Hélcio Machado Pimentel

Protocolo de Comunicação Segura para Plataforma de Distribuição de Vídeo em Redes Sobrepostas

São Paulo
2011

Hélcio Machado Pimentel

Protocolo de Comunicação Segura para Plataforma de Distribuição de Vídeo em Redes Sobrepostas

Dissertação apresentada à Escola Politécnica de Universidade de São Paulo para a obtenção do título de Mestre em Engenharia.

São Paulo
2011

Hélcio Machado Pimentel

Protocolo de Comunicação Segura para Plataforma de Distribuição de Vídeo em Redes Sobrepostas

Dissertação apresentada à Escola Politécnica de Universidade de São Paulo para a obtenção do título de Mestre em Engenharia.
Orientadora: Profa. Dra. Graça Bressan

São Paulo
2011

Dedico esta monografia primeiramente a Deus, que está sobre tudo e todos, a meus pais, meu irmão, minha irmã e amigos, que me apoiaram continuamente por essa jornada.

Agradecimentos

A Deus, acima de tudo e todos, que me guia sempre

A meus pais, Jacira L. M. Pimentel e Hugo Silva Pimentel

Às professoras Dr. Graça Bressan e Dr. Regina Melo Silveira, orientadora e coorientadora respectivamente, por sempre me motivarem, ensinarem, e abrirem portas a novas oportunidades pelas quais pude crescer na vida profissional e pessoal.

A todos os professores, pois me instruíram no caminho da riqueza intocável, o conhecimento.

A equipe de coordenação do curso, Sílvia e Selma, que diariamente dão condições e muita simpatia no desenrolar das atividades cotidianas.

Aos amigos e colegas de projeto Samuel K., Lucas C., Ricardo M., Rosiane C., Marcos S. e Reinaldo M., por partilhar a amizade e o muito trabalho que contribuiram imensamente para o meu crescimento.

Aos amigos Marcos S. e Samuel K., que, além de grandes camaradas, foram de grande importância na etapa final deste trabalho.

A todos pesquisadores e colegas do LARC pela troca de experiência e por tornar este laboratório em um ótimo ambiente de trabalho.

A todos aqueles que estiveram presentes, e que, de maneira direta ou indireta, contribuíram para a realização deste trabalho, meus sinceros agradecimentos.

“É impossível alcançar a perfeição de caráter
sem sacrifício de nós mesmos”
E. W. - 9 Te, 53

“Perfection of character cannot possibly be attained
without self-sacrifice”
Ellen G. White - Testimony for the Church Volume Nine (1909), page 53

Resumo

As redes de distribuição de vídeo têm sido amplamente utilizadas na atualidade pela Internet. O sucesso de Portais de Vídeo evidencia tal uso. Por poderem ser redes de grande porte, há uma grande preocupação com as vulnerabilidades existentes nessas redes. A comunicação de seus elementos deve ser segura o bastante para garantir a disponibilidade, o sigilo e integridade de suas mensagens e a autenticidade dos seus elementos.

Este trabalho apresenta um protocolo de comunicação segura que busca atender a tais necessidades de uma maneira eficiente - pois consegue atender aos requisitos de desempenho na entrega do conteúdo aos usuários - e genérica - pois pode ser utilizado em outras plataformas de distribuição. A validação do trabalho é feita de maneira a mostrar que a proposta consegue atender aos requisitos de um sistema de distribuição de vídeo seguro.

Abstract

Video delivery network has been widely used across the Internet nowadays. The success of Video Portals is an evidence of this use. Due to its potential to turn into large infrastructures, there is a concern about its vulnerabilities. The communication among its elements must be secure enough to guarantee the availability, the secrecy and integrity of messages and the authenticity of its elements.

We present in this work a secure communication protocol to meet such requisites in an efficient - since it meets the performance requisites for delivering the content to the users - and generic way - because it can be used by other distribution systems. The validity of this work is done in order to show that this proposal can meet the requisites of a secure video delivery system.

Lista de Figuras

Lista de Tabelas

Lista de Abreviações e Siglas

AES (Advanced Encryption Standard) - Padrão de Criptografia Avançada

AS (Autonomous System) - Sistema Autônomo

BGP (Border Gateway Protocol) - Protocolo de Gateway de Borda

BOS (Basic Overlay Services) - Serviços Básicos Sobrepostos

C-I-A (Confidentiality-Integrity-Availability) - Confidencialidade, Integridade e Disponibilidade

CBC (Cipher Block Chaining) - Criptografia de Blocos Encadeados

CDN (Content Delivery (Distribution) Network) - Rede de Distribuição de Conteúdo

CFB (Cipher Feedback) - Realimentação de Cifra

CMAC (Cipher-Based Message Authentication Code) - Código de Autenticação de Mensagem Baseado em Cifras

CTR (Counter) - Contador

DDoS (Distributed Denial of Service) - Negação de Serviço Distribuído

DES (Data Encryption Standard) - Padrão de Encriptação de Dados

DHT (Distributed Hash Table) - Tabelas Hash Distribuídas

DNS (Domain Name System) - Sistema de Nomes de Domínio

DoS (Denial of Service) - Negação de Serviço

DRM (Digital Rights Management) - Gestão de Direitos Digitais

ECB (Electronic CodeBook) - Livro-código Eletrônico

HTTP (Hypertext Transfer Protocol) - Protocolo de Transferência de Hipertexto

IP (Internet Protocol) - Protocolo de Internet

ISP (Internet Service Providers) - Provedores de Acesso à Internet

MAC (Message Authentication Code) - Código de Autenticação de Mensagem

NIST (National Institute of Standards and Technology) - Instituto Nacional de Padrões e Tecnologia

OAS (Overlay Autonomous System) - Sistema Autônomo Sobreposto

OCP (Overlay Communication Protocol) - Protocolo de Comunicação (de Rede) Sobreposta

OFB (Output Feedback) - Realimentação de Saída

OVERSEA (Scalable and Effective Architecture for Overlay Networks) - Arquitetura Escalável e Eficaz para Redes Sobrepostas

P2P (Peer-to-Peer) - Par-a-Par

PSTN (Public Switched Telecommunications Network) - Rede de Telecomunicações Pública Comutada

QoS (Quality of Service) - Qualidade de Serviço

RNP (Rede Nacional de Ensino e Pesquisa)

RSA (Rivest, Shamir and Adleman) - Rivest, Shamir e Adleman

SBRC (Simpósio Brasileiro de Redes de Computadores)

SMNP (Simple Network Management Protocol) - Protocolo Simples de Gerenciamento de Redes

SON (Service Overlay Networks) - Rede de Serviços Sobrepostos

SSL (Secure Socket Layer) - (Protocolo de) Camada de Sockets Segura

TCP (Transmission Control Protocol) - Protocolo de Controle de Transmissão

VoD (Video on Demand) - Vídeo Sob Demanda

VoIP (Voice over Internet Protocol) - Voz sobre Protocolo de Internet

VPN (Virtual Private Network) - Rede Privada Virtual

WAF (Web Application Firewall) - Firewall de Aplicação Web

WRNP (Workshop da Rede Nacional de Ensino e Pesquisa)

Sumário

1 Introdução

Com a crescente demanda por aplicações multimídia através da Internet, torna-se evidente que os sistemas de distribuição de conteúdo multimídia precisam ter maior eficiência, e ainda garantir alta disponibilidade de seus serviços. Em outras palavras, o consumo de objetos digitais pelos usuários necessita ser garantido mesmo em face da imensa quantidade de usuários acessando conteúdos de maneira concorrente (BUYYA; PATHAN; VAKALI, 2008).

Nos últimos anos, tal eficiência tem sido alcançada com a implementação de **Plataformas de Distribuição de Conteúdos**, que podem utilizar as **Redes de Distribuição de Conteúdo** (CDN - do inglês, *Content Distribution (Delivery) Networks*) para otimizar tal distribuição aos diversos usuários dispersos nessas redes (CLARK et al., 2006).

De maneira prática, as redes de distribuição são implementadas com o uso da técnica de sobreposição - também conhecida como *Overlay Network* (Rede Sobreposta), que se trata da criação de rede virtual, que, por sua vez, mapeia a rede física da arquitetura TCP/IP (UCHôA et al., 2007), (CLARK et al., 2006) e (DOVAL, 2003).

De fato, as redes de distribuição de conteúdo são redes sobrepostas que efetuam armazenamento (*caching*) dinâmico de conteúdo e oferecem serviços de forma distribuida por toda a Internet, e podem representar uma boa parcela do tráfego na Internet atual. Podemos citar como exemplo de utilização desta estratégia as ofertas comerciais da *Akamai*[1] e aplicações de armazenamento *P2P (Peer-to-Peer)*, como o *BitTorrent*[2] (CLARK et al., 2006).

O benefício do uso da técnica de sobreposição é a possibilidade da implementação de funcionalidades em camada de aplicação, sem a necessidade de se configurar *switches* e roteadores (FAHMY; KWON, 2007). Essas funcionalidades abrangem o armazenamento dinâmico de conteúdos, controle de QoS (*Quality of Service* - Qualidade de Serviço), difusão por *Multicast*, segurança, monitoramento, contabilização, entre outras (CLARK et al., 2006) e (BUYYA; PATHAN; VAKALI, 2008).

[1] **Akamai**: http://www.akamai.com
[2] **BitTorrent**: http://www.bittorrent.com/

Sucintamente, essa infraestrutura de rede pode ser utilizada para a distribuição de diversas mídias, para diferentes propósitos, como áudio, vídeo, texto, serviços de diretórios, transferência de arquivos, etc., (CLARK et al., 2006) e (BUYYA; PATHAN; VAKALI, 2008). Salientando que os conteúdos de vídeo são os que mais geram carga sobre a rede e que implicam em requisitos mais restritos de qualidade.

De forma a atender os usuários, o processo de distribuição de conteúdo por rede sobreposta envolve o uso de serviços, como, por exemplo, reconhecimento da topologia e seus elementos, definição da melhor rota (como no caso de transmissão de vídeo), que geram necessidade de comunicação entre os elementos da rede de distribuição.

Esta comunicação se caracteriza por mensagens de sinalização de controle que precisam ser seguras (fim-a-fim) de forma a se evitar a concretização de ameaças devido às vulnerabilidades da rede.

Em relação às vulnerabilidades presentes na rede sobreposta, existem diversas delas que não são resolvidas com as técnicas tradicionais implementadas em outras camadas, devendo-se à implantação na camada de aplicação. Essas vulnerabilidades podem ser relacionadas a ameaças à própria rede ou a terceiros.

No primeiro caso das vulnerabilidades de rede sobreposta, a própria rede pode ter sua estrutura e serviços providos comprometidos. Por exemplo, algum invasor pode inserir conteúdo impróprio na rede para que esse seja disseminado, ou então, permitir que usuários acessem conteúdo ao qual não tenham permissões de acesso. Além disso, esse conteúdo impróprio pode originar problemas jurídicos por poderem ferir a reputação e o bom nome de pessoas e instituições.

No segundo caso, devido ao potencial das redes sobrepostas em evoluir para grandes infraestruturas, existe a possibilidade de torná-las candidatas a serem empregadas para promover ataques a outros hospedeiros ou mesmo redes inteiras (DEFRAWY; GJOKA; MARKOPOULOU, 2007).

Como consequência, essas redes são convertidas nas chamadas *botnets*, que são redes comprometidas e controladas por algum invasor que, posteriormente, utiliza os elementos da rede para potencializar o ataque. Isso é possivel quando um atacante invade os

elementos da rede e os transforma em *zumbis*, ou seja, por meio de execução de programas que abrem conexões para, por fim, direcionar seus fluxos de dados a qualquer vítima - servidores ou redes, sobrecarregando-os até derrubá-los (DEFRAWY; GJOKA; MARKOPOULOU, 2007).

Esse tipo de ataque, conhecido como **Negação de Serviço** (*DoS - Denial of Service*) não se caracteriza pela invasão do alvo, mas em tornar indisponíveis os seus serviços ou impossibilitar seu acesso, como manobra para prática de extorsão a indivíduos ou instituições, que sejam responsáveis pelo fornecimento desses serviços aos seus usuários (MIRKOVIC et al., 2004).

Quando empregado em redes de grande porte, o ataque de negação causa consequências muito mais drásticas, pois facilita seu emprego de maneira distribuída sobre a vítima em questão, o que caracteriza um *Ataque de Negação Distribuído* (*DDoS - Distributed Denial of Service*).

Portanto, faz-se necessário o sigilo das informações da rede de distribuição para que não sejam facilitados os procedimentos para invasão de elementos da rede e, consequentemente, o ataque de negação de serviço.

Neste contexto, este trabalho tem como foco propor um protocolo a ser usado como mecanismos de segurança para as **sinalizações de controle** (comunicação de controle seguro) de um sistema de distribuição de conteúdo, particularmente no cenário de conteúdo de vídeo.

Temos como principais requisitos a robustez e qualidade no serviço de transmissão de vídeo aos usuários do sistema. Além disso, tomamos em consideração evitar que haja sobrecarga excessiva devido ao uso da solução proposta.

Os mecanismos de segurança devem abranger a distribuição de chaves, autenticação de elementos da topologia e sigilo de mensagens de controle do sistema, que se beneficiam do protocolo proposto para permitir que a comunicação, além de segura, possa atender aos diversos serviços disponibilizados pelo sistema de distribuição de vídeos.

Outro importante requisito está relacionado à integração de serviços de maneira segura.

Essa integração é realizada em redes vizinhas, por meio dos serviços de interdomínio, que podem dividir entre si a carga das solicitações de vídeo de usuários, o que otimiza o uso de suas infraestruturas de rede.

Em tal cenário interdomínio, a comunicação segura também deve estar disponível, permitindo que redes vizinhas compartilhem informações de suas topologias e de seus recursos, aplicando políticas de compartilhamento e de modo seguro.

A validação deste protocolo é feita no contexto de uma rede de distribuição de vídeo, viabilizando a comunicação dos diversos elementos que a compõe. Entretanto, o enfoque é na segurança do fluxo de sinalização de controle do sistema e não inclui a segurança do fluxo de vídeo em si. Por fim, é integrado à arquitetura Oversea, que é uma plataforma de distribuição de vídeo de propósitos acadêmicos (KOPP et al., 2010).

1.1 Motivação

Pesquisas na área de redes de distribuição de conteúdo têm sido feitas, nos últimos anos, objeto de estudo no LARC (Laboratório de Arquitetura e Redes de Computadores). Juntamente a projetos como GTGV (Grupos de Trabalho de Gerência de Vídeo) e GT Overlay (Grupo de Trabalho em Rede de Serviços Sobrepostos), subsidiados pela RNP (Rede Nacional de Ensino e Pesquisa), desenvolveu-se uma plataforma de gerência de vídeo, que já está em operação na RNP - Video@RNP e na USP (IPTV-USP). Tal plataforma explora o conceito de redes sobrepostas em redes de distribuição de vídeo.

Dos benefícios resultantes desses projetos mencionados, podemos destacar: (i) a construção de uma infraestrutura para distribuição de vídeo que emprega o *multicast* implementado com redes sobrepostas e (ii) o desenvolvimento de uma arquitetura para redes de serviços sobrepostos, a arquitetura Oversea (UCHôA et al., 2007).

Por meio dessa arquitetura é possível maximizar o uso de diversas redes sobrepostas, com um conjunto de serviços comuns, reduzindo a sobrecarga de tráfego, já que evita redundâncias de medições na rede para a descoberta da topologia e sua monitoração (UCHôA et al., 2007).

Em tal contexto, este trabalho elabora mecanismos de segurança de sinalização de controle que podem ser integrados a esta plataforma, permitindo que usuários possam acessar os conteúdos de forma mais adequada, com aumento da segurança e robustez do sistema.

Os desafios apresentados são fatores que permitirão dar contribuições às áreas de segurança e distribuição de conteúdo em redes de computadores.

1.2 Objetivo

Este trabalho tem como objetivo principal a proposição de um protocolo para comunicação segura para as mensagens de sinalização de controle do sistema, implantado em uma rede de serviços sobreposta, para distribuição de vídeo com mecanismos que garantam requisitos de segurança.

Usando uma abordagem preventiva busca minimizar a sobrecarga à rede, visto que o tipo de aplicação, ou seja, distribuição de vídeo, requer bom desempenho (AMUTHARAJ; RADHAKRISHNAN, 2007), em conjunto à alta disponibilidade, que pode ser afetada sob ataques.

O uso do protocolo e mecanismos propostos aumentam a disponibilidade deste sistema, pois, além de baixa sobrecarga, permite prevenir que sejam expostas as informações da rede a algum atacante que queira atingir tanto à rede em si quanto a outro alvo utilizando-se da rede comprometida.

Também é objetivo desse trabalho que tal protocolo seja utilizado na sinalização que controla transmissões, ao vivo ou sob-demanda, utilizando *multicast* em camada de aplicação e através de domínios sobrepostos diferentes, isto é, *multicast* interdomínio, atendendo as garantias aqui apresentadas.

1.3 Justificativa

Apesar dos benefícios de se obter segurança durante as comunicações dos diversos elementos da rede sobreposta, há de se considerar a existência de sobrecarga à própria rede, além, é claro, do custo computacional no processamento das mensagens seguras.

A abordagem reativa é uma das que causam maior efeito à rede, pois necessitam de mecanismos que constantemente monitorem a rede em busca de quaisquer ameaças potenciais, o que poderia ser, até certo ponto, inviável em cenários em que o desempenho é vital, tal como ocorre em redes de distribuição de conteúdo de vídeo.

De maneira a tratar essas questões, a proposta utiliza um protocolo com mecanismos de garantias dos requisitos de segurança apresentados, que são satisfeitos por uma abordagem preventiva, com menor impacto, concernente à sobrecarga sobre a rede, o que implica em maior robustez do sistema.

É relevante destacar o aspecto da coexistência e integração de várias das técnicas de otimização em sistemas de distribuição de conteúdo, além das questões de segurança, ou seja, *multicast*, armazenamento, monitoramento, que apresentam bastante complexidade em seu gerenciamento, inclusive são geradores de grande aumento de sobrecarga de tráfego para o controle da rede.

Com essas considerações, neste trabalho, a utilização de redes sobrepostas de serviços, por meio da arquitetura Oversea, que tem como motivação a redução desse tráfego excessivo, com a abordagem de serviços comuns.

Podemos identificar as seguintes contribuições deste trabalho:

- Elaboração de um protocolo e de mecanismos associados para garantias de requisitos de segurança em redes de distribuição de vídeo;
- Elaboração de um protótipo que utilize o protocolo proposto;
- Integração do protótipo do sistema de segurança a um serviço de *multicast* em sistemas autônomos;

- Identificação dos impactos que o uso do protocolo e dos mecanismos de segurança podem causar em um sistema de distribuição de vídeo;
- Validação do protocolo proposto por meio de análise de dados obtidos em testes utilizando o protótipo.

1.4 Organização

Este trabalho está organizado da seguinte maneira. No Capítulo 2 são demonstrados os conceitos de redes de distribuição de conteúdo e de redes sobrepostas, *multicast* sobreposto, bem como a definição e descrição da arquitetura Oversea utilizada. Também é apresentado o cenário interdomínio utilizado. No Capítulo 3 são apresentados os conceitos de segurança em redes, como criptografia, assinaturas digitais e outros princípios básicos da área de segurança, seguindo com as vulnerabilidades identificadas das redes sobrepostas no contexto de distribuição de vídeo. Neste capítulo ainda, apresentamos os trabalhos correlatos e o estado da arte, assim como uma análise comparativa dessas soluções à proposta deste trabalho. Todo esse levantamento tem o intuito de embasar o protocolo descrito no Capítulo 4. No Capítulo 5, realiza-se a validação da proposta, com análise de desempenho e comparações analíticas. E por fim, são apresentadas as considerações finais e conclusões no Capítulo 6.

2 Distribuição de Conteúdo Multimídia

Há diversas definições para **Redes de Distribuição de Conteúdo**. Em uma delas, tais redes são infraestruturas compartilhadas implantadas para entrega ou distribuição dos conteúdos da **Web** de terceiros aos usuários na Internet (TRIUKOSE; AL-QUDAH; RABINOVICH, 2010).

Ao compartilhar seus vastos recursos entre um grande e diversificado número de sítios *Web*, as redes de distribuição tiram como consequência a economia de escala, ou seja, conseguem melhor escalabilidade (TRIUKOSE; AL-QUDAH; RABINOVICH, 2010).

Essa economia se realiza porque diferentes sítios experimentam picos de demanda por conteúdos, denominados *flash crowds*, em tempos distintos, e assim, a mesma capacidade em períodos menos ativos pode ser usada para absorver a demanda inesperada para sítios múltiplos (TRIUKOSE; AL-QUDAH; RABINOVICH, 2010).

Logisticamente, nessas redes de distribuição, o conteúdo é distribuído para os servidores de armazenamento - servidores de réplicas ou replicação - localizados próximos aos usuários, o que resulta em aplicações e serviços *Web* rápidos e confiáveis aos usuários (PALLIS; VAKALI, 2006).

De modo mais específico, as redes de distribuição mantêm múltiplos PoP's (*Points of Presence* - Pontos de Presença) com *clusters* de servidores de réplicas que armazenam cópias de conteúdo idêntico, tal que as solicitações dos usuários são atendidas pelo sítio mais apropriado (PALLIS; VAKALI, 2006).

Deve-se ao uso do armazenamento nesses servidores de réplicas que as redes de distribuição protegem os sítios *Web* detentores do conteúdo do excesso de carga devido às solicitações dos muitos clientes (TRIUKOSE; AL-QUDAH; RABINOVICH, 2010).

É nesse intuito que as arquiteturas CDN objetivam minimizar o impacto em rede durante a entrega de conteúdo, bem como superar o problema de sobrecarga, que é uma séria ameaça a sítios ou sistemas que ficam ocupados em atender a demanda por conteúdos populares (AMUTHARAJ; RADHAKRISHNAN, 2007).

Em uma rede de distribuição de conteúdo, a comunicação cliente-servidor é substituída por dois fluxos de comunicação (PALLIS; VAKALI, 2006):

- Entre o cliente e o servidor de réplicas;
- Entre o servidor de réplicas e o servidor de origem ou fonte.

Esta distinção em dois fluxos de comunicação reduz o congestionamento - em especial para servidores populares - e aumenta a distribuição e disponibilidade dos conteúdos (PALLIS; VAKALI, 2006).

Das vantagens do uso de redes de distribuição, podemos indicar (PALLIS; VAKALI, 2006):

- Redução dos custos operacionais de gerência de infraestrutura;
- Desvio de congestionamentos de tráfego na rede, visto que os dados estão mais próximos ao usuário e não há necessidade de transmitir o fluxo desde o servidor principal, evitando a geração de pontos de congestionamento;
- Melhora na qualidade e velocidade de entrega, e disponibilidade do conteúdo;
- Redução da carga sobre os servidores fonte do conteúdo.

Como meio de diminuir a carga sobre os servidores populares e melhorar a experiência do usuário final, cópias de conteúdos são armazenados em locais diferentes (AMUTHARAJ; RADHAKRISHNAN, 2007).

Havendo cópias de um mesmo conteúdo em múltiplos servidores, é necessário que se utilizem mecanismos que forneçam o melhor tempo de resposta, para a escolha do servidor que deve atender a uma determinada requisição, e com desempenho resultante determinístico, o que depende do servidor selecionado (AMUTHARAJ; RADHAKRISHNAN, 2007).

Tacitamente, as redes de distribuição de conteúdo, na realidade, são **Redes Sobrepostas** que fazem armazenamento de conteúdos e disponibilização de serviços dinami-

camente distribuídos por toda a Internet. De maneira geral, o conteúdo pode ser arquivos de documentos, áudio, vídeos e outros (CLARK et al., 2006).

Este trabalho tem seu foco especificamente na transmissão de vídeo, em vista que tal conteúdo é o que apresenta maiores desafios quanto à sua distribuição na Internet atual, pois normalmente apresentam requisitos de largura de banda e tempo de transmissão significativos devido ao grande volume de dados (AMUTHARAJ; RADHAKRISHNAN, 2007).

2.1 Redes Sobrepostas

A Internet teve sua origem em uma rede de pesquisas criada pelo governo dos E.U.A. com fins militares utilizando como infraetrutura a rede PSTN (*Public Switched Telecommunications Network* - Rede de Telecomunicações Pública Comutada), que é a rede telefônica de utilização pública. Dessa forma, a Internet é uma rede sobreposta - *overlay* - que complementa a infraestrutura básica subjacente de telefonia, que, então, adiciona uma nova funcionalidade: rede de dados de comutação de pacotes (CLARK et al., 2006).

Partindo de sua comercialização nos anos 80, para então emergir como uma plataforma de mercado em massa para comunicações de banda larga nos anos 90, de forma crescente, a infraestrutura TCP/IP da Internet tem se tornado não mais uma aplicação sobreposta, mas sim em infraestrutura básica (CLARK et al., 2006).

O sucesso da Internet se deve à interoperabilidade e conectividade da arquitetura TCP/IP e de seu princípio de simplicidade em que a complexidade é tratada pelos sistemas finais. Apesar disso, a Internet enfrenta grandes desafios, como o crescimento de serviços heterogêneos, já que as necessidades e capacidades desses serviços são diferenciadas, bem como novas necessidades e requisitos gerados por serviços de tempo-real e de segurança ampliada (CLARK et al., 2006).

Foi então que as redes sobrepostas surgiram como um modo eficaz de se dar suporte a novas aplicações, bem como protocolos, sem que isso obrigue a mudanças na camada de TCP/IP (LI; MOHAPATRA, 2004).

Recentemente, tais redes têm ganhado destaque como infraestrutura de suporte a mecanismos para superar barreiras de implantação à solução no nível de roteadores de diversos problemas de rede (FAHMY; KWON, 2007).

Há diversos exemplos de redes sobrepostas que vão desde redes P2P para compartilhamento de arquivos - que estão associadas a aplicações como *BitTorrent*[1] e serviços de voz sobre IP oferecidos via aplicações como o *Skype*[2], até redes de cache e distribuição de conteúdo - implementadas por empresas como a *Akamai*[3] - e várias redes de *testbed*, tais como o *PlanetLab*[4] (CLARK et al., 2006).

A tabela 1 mostra os tipos mais comuns de redes sobrepostas, com suas características e exemplos de implementação.

Tabela 1: Exemplos de Redes Sobrepostas (CLARK et al., 2006)

Tipo	**Função/Propósito**	**Exemplo**
Peer-to-Peer (P2P)	Compartilhamento de Arquivos (ex. MP3)	Napster, Gnutella
Rede de Distribuição de Conteúdo	*Caching* de Conteúdo para reduzir atraso de acesso e custos de transporte	Akamai, Digital Island
Roteamento	Reduz atraso de roteamento, redes sobrepostas de roteamento resilientes	Resilient Overlay Network (RON)
Segurança	Aumenta a segurança e privacidade no usuário final	Virtual Private Network (VPNs), roteamento onion (Tor, I2P), armazenagem de conteúdo anônimo (Freenet, Entropy), redes sobrepostas resistentes à censura (Publius, Infranet, Tangler)
Experimental	Facilita inovação, implementação de novas tecnologias, experimentação	Próposito Geral (PlanetLab, I3)
Outras	Várias	Email, VoIP (Skype), Multicast (MBone, 6Bone, TRIAD, IP-NL), Redes tolerantes a atraso, etc.

Uma rede sobreposta é composta de um conjunto de servidores, implantados dentro de nós *peers* ou pares, que fornecem uma infraestrutura para uma ou mais aplicações, responsabilizando-se, de alguma forma, pelo repasse e tratamento dos dados dessas aplicações, trazendo como benefício o desacoplamento de outras camadas de rede (CLARK et al., 2006).

Na realidade, as redes sobrepostas criam uma topologia virtual estruturada acima do nível de protocolo de transporte básico, o que facilita buscas determinísticas e garante convergência das informações de topologia (DOVAL, 2003).

[1] **BitTorrent**: http://www.bittorrent.com/
[2] **Skype**: http://www.skype.com
[3] **Akamai**: http://www.akamai.com/
[4] **PlanetLab**: http://www.planet-lab.org/

As redes sobrepostas organizam os nós de uma forma que possam estar posicionados independentes da topologia física inferior. Os elementos da rede sobreposta podem ter outros elementos vizinhos dentro de uma mesma subrede ou ao longo da Internet (DOVAL, 2003).

Embora algumas redes se adaptem à topologia física inferior, tal otimização não é requerida para que o algoritmo de roteamento - que é responsável por gerenciar as tabelas e tomar as decisões de roteamento (TANENBAUM, 2003) - opere adequadamente, mas é encorajada para evitar maiores problemas devidos a gargalos e baixa latência (DOVAL, 2003).

Rede de Serviços Sobrepostos

Diversos tipos de redes sobrepostas têm sido projetadas para dar suporte a soluções de provisionamento de serviços que dificilmente são suportados nas camadas TCP/IP. Estão incluídas a essas redes sobrepostas as redes de distribuição de conteúdo, (DILLEY et al., 2002), e de compartilhamento de arquivos (P2P - *Peer-To-Peer*), (RIPEANU, 2001) e (LEE; KIM, 2009).

Como uma inovação às rede sobrepostas, um novo tipo de rede sobreposta denominada Rede de Serviços Sobrepostos (SON - *Service Overlay Network*) foi proposta para facilitar o uso de serviços como VoIP (Voz sobre IP - *Voice over IP*) e VoD (Vídeo sob Demanda - *Video on Demand*) que requerem QoS (Qualidade de Serviço - *Quality of Service*) (LEE; KIM, 2009). Desse modo, a rede de serviços sobrepostos é um meio efetivo de tratar problemas de fornecimento de serviços fim-a-fim na atual Internet (LIU; LUI, 2004).

De fato, a rede de serviços sobrepostos consiste em uma rede sobreposta que pode abarcar muitos sistemas autônomos, o que pode facilitar a criação e implantação de serviço com valor agregado na Internet (POMPILI; SCOGLIO; LOPEZ, 2008) e (LIU; LUI, 2004).

2.1.1 Multicast Sobreposto

A funcionalidade de se transmitir fluxos de vídeo e de áudio de alta qualidade ao vivo pela Internet, a baixo custo, tem se mostrado de difícil implementação (ANDREEV et al., 2003).

A popularidade desse tipo de transmissão, como, por exemplo, *broadcast* televisivo ou radiofônico de eventos esportivos ou chamadas comerciais, tem aumentado conforme cada vez mais empresas e organizações mostram-se interessadas em disponibilizar por esses meios o seu conteúdo ao público mundial da Internet (ANDREEV et al., 2003).

Em vista dessa grande demanda, tem sido de grande preocupação a implantação de mecanismos que permitam um uso otimizado da rede, reduzindo o excesso de carga conforme aumenta o número de usuários do sistema distribuído sobre a infraestrutura de rede.

Por isso, uma arquitetura de redes sobrepostas deve proporcionar alto desempenho, alta escalabilidade e custos reduzidos ao dar suporte a serviços que exijam grande carga da rede e do sistema (CIDON; UNGER, 2003).

Para tal fim, metodologias de *multicast* podem ser empregadas para entregar o conteúdo utilizando servidores regionais ou *proxies* para os usuários finais próximos a eles e ainda dar suporte à sincronização de conteúdo de maneira econômica e temporal entre os servidores distribuídos (CIDON; UNGER, 2003).

O *multicast* é extremamente útil para um número crescente de aplicações de redes, tais como a transmissão contínua de mídia em CDNs, que requerem que a entrega de pacotes sejam feitas de um ou mais remetentes a um grupo de receptores (KUROSE; ROSS, 2005).

Aliás, uma definição apropriada para *multicast* é o envio de um pacote de um único remetente para múltiplos receptores com apenas uma operação simples de “transmissão” (KUROSE; ROSS, 2005).

Contudo, com relação aos grupos de usuários receptores, o desafio está em enviar mensagens a grupos bem definidos que têm um tamanho numericamente grande, porém que são pequenos quando comparados à rede como um todo (TANENBAUM, 2003).

Sendo assim, o *multicast* exige a existência de um gerenciamento de grupos de usuários. Havendo necessidade de se utilizar algum método de criação e destruição de grupos, além de permitir que usuários entrem e saiam dos grupos (TANENBAUM, 2003).

Os hospedeiros devem informar seus roteadores - que são responsáveis pelo gerenciamento dos grupos - sobre alterações na associação a grupos, ou então, os próprios roteadores devem se encarregar de consultar seus hospedeiros (TANENBAUM, 2003). Os roteadores executam o roteamento por *multicast* calculando a árvore de amplitude ou escoamento (*Spanning Tree*) que engloba todos os outros roteadores da sub-rede que contenham hospedeiros que pertençam a um mesmo determinado grupo ou conjunto de grupos (TANENBAUM, 2003).

Quando um processo envia um pacote de *multicast* a um grupo, o primeiro dos roteadores examina sua árvore de amplitude e a poda (*pruning*), retirando todas as linhas que não conduzam a hospedeiros que pertençam ao mesmo grupo. Assim os pacotes são somente encaminhados ao longo da árvore de amplitude apropriada (TANENBAUM, 2003).

Nessa abordagem um único datagrama é transmitido do hospedeiro remetente. Esse datagrama - ou uma cópia dele - é, portanto, replicado num roteador da rede sempre que for necessário reencaminhá-lo em múltiplos enlaces de saída, a fim de alcançar os receptores (KUROSE; ROSS, 2005).

A razão de ser considerada ideal da abordagem se deve em fazer *multicast* com uso mais eficiente da banda de rede na qual apenas uma simples cópia de um datagrama atravessa um enlace.

Em comunicação *multicast*, encaramos dois problemas comuns - que são sensivelmente mais complicados que um simples caso de *unicast*. Esses problemas são em como identificar os receptores de um datagrama *multicast* e como endereçar um datagrama enviado a esses receptores. Dessa forma, o endereçamento de datagramas *multicast* é feito usando “endereçamento indireto” - tradução livre de *address indirection*. Isso consiste em

que um simples “identificador” seja usado para um grupo de receptores e uma cópia do datagrama endereçado ao grupo com tal identificador é entregue a todos os receptores associados a esse grupo. Na Internet um endereço *multicast* Classe D é usado como tal identificador (KUROSE; ROSS, 2005).

Há outras questões que precisam ser tratadas para a solução de *multicast* IP nativo, como quando se inicia um grupo e como finalizá-lo, como escolher o endereço do grupo, como adicionar novos hospedeiros, restrições de ingresso e outras que não serão tratadas nesse trabalho.

Por outro lado, a solução de *multicast* nativo da camada de rede precisa de considerável suporte da camada de rede para implementá-lo (KUROSE; ROSS, 2005).

Esse suporte, de maneira global, enfrenta ainda muitos desafios técnicos e de implantação atualmente, uma vez que há necessidade de cooperação de diversos ISPs (*Internet Service Providers* - Provedores de Acesso à Internet) ao configurarem seus roteadores para suporte à *multicast* nativo (YIU; CHAN, 2008).

Em muitas aplicações que utilizam *multicast* para distribuição de conteúdos, há o desejo de que o transmissor possa transmitir seus dados de maneira encriptada para que apenas usuários assinantes e autorizados sejam capazes de decriptá-los. Portanto, suporte a canais de comunicação segura entre os *peers* ou nós é essencial para tais aplicações (YIU; CHAN, 2008).

Agora de maneira prática e do ponto de vista de redes, a ideia abstrata de *multicast*, ou seja, uma única operação de envio que resulta em cópias dos dados enviados sendo entregues a muitos receptores, pode ser implementado de diversas maneiras (KUROSE; ROSS, 2005).

Uma possibilidade é que o remetente use uma conexão de transporte *unicast* - um para um - para cada um dos receptores. Essa abordagem implementa uma abstração de *multicast* “um-remetente-para-muitos-receptores” usando *unicast* da camada de rede subjacente, sem exigir suporte a *multicast* da camada de rede explicitamente (KUROSE; ROSS, 2005).

Essa é uma solução parecida com a usada em *multicast* sobreposto. Entretanto, o *multicast* sobreposto gera menos sobrecarga à rede, já que consegue transmitir um fluxo *unicast* único da fonte, e esse fluxo só será dividido no nó da árvores em que os usuários estão, de acordo com o número de usuários nesse nó.

Por meio da abordagem de **Multicast Sobreposto**, pode-se obter soluções para diversos problemas desafiantes, entre eles: o próprio *multicast*, roteamento, distribuição de conteúdo e serviços de servidores pares. Em *multicast* sobreposto, os servidores participantes em uma sessão multicast formam uma rede sobreposta. Tais servidores utilizam *unicast* apenas entre pares de servidores, que sejam vizinhos na árvore de distribuição de rede sobreposta, para disseminarem os dados (FAHMY; KWON, 2007).

O conceito de *Rendezvous Point*, que é responsável por fazer com que os elementos participantes do *multicast* sejam localizados e a rota de entrega de distribuição do conteúdo aos participantes seja criada, é possível por meio de algum elemento gerenciador que pode ser centralizado ou distribuído.

Ademais, esses servidores lidam exclusivamente com a gerência de grupo, roteamento e construção de árvores, sem a necessidade de suporte à tais serviços sejam configurados nos roteadores na Internet, já que a extensão de suporte a esses serviços é feito em camada de aplicação nos agentes de *multicast* nos nós de serviços sobrepostos (FAHMY; KWON, 2007).

Entretanto há um custo de desempenho imposto nessa solução comparada à alternativa no nível de roteadores. Porque, enquanto que a solução de multicast IP é considerada a mais otimizada, a solução multicast sobreposto, e qualquer serviço em sobreposição, ou seja, em rede sobreposta, claramente consome mais banda de rede e aumenta a latência, porém com a vantagem ainda de maior flexibilidade, adaptatividade e facilidade de implantação (FAHMY; KWON, 2007).

2.2 Plataforma OVERSEA

A arquitetura OVERSEA (*Scalable and Effective Architecture for Overlay Networks*) foi proposta e desenvolvida a fim de fornecer serviços de rede que não podem ser implantados diretamente nas camadas inferiores TCP/IP (UCHôA et al., 2007).

Essa arquitetura se compõe basicamente de quatro subcamadas (UCHôA et al., 2007):

- **Rede**: considerada como suporte da camada IP, ou seja, serviço de datagramas de melhor-esforço;
- **BOS (Serviços Básicos Sobrepostos - *Basic Overlay Services*)**: extensão direta da camada de rede que fornece suporte à camada superior SON;
- **SON (Redes de Serviços Sobrepostos - *Service Overlay Networks*)**: subcamada em que os serviços extensíveis específicos estão implantados;
- **Aplicações**: são as aplicações que se beneficiam do bom uso da infraestrutura completa de maneira transparente, escalável e homogênea.

O melhor esforço da camada IP significa que essa camada faz o melhor que pode para entregar unidades de dados, contudo não garante a entrega do pacote ou a sua sequência de transmissão. Com isso a arquitetura OVERSEA pode ser vista como contendo roteadores de rede implementando o serviço de datagramas *unicast*, e nós de serviços - que de fato são *proxies* - implementando quaisquer outras funcionalidades de camadas superiores (UCHôA et al., 2007).

A principal utilidade da camada BOS é reduzir a sobrecarga por redundância de monitoramento de informações e de controle da plataforma, evitando a implementação de redes sobrepostas específicas para cada serviço. O suporte fornecido pela camada BOS compreende prover funcionalidades comuns aos serviços sobrepostos, tais como (UCHôA et al., 2007):

- Busca de elementos da rede sobreposta (*lookup*);

- Designação de nomes para identificação dos elementos da rede;
- Instanciação de topologia virtual;
- Roteamento em caso de solicitações de conteúdo de vídeo;
- Monitoramento dos elementos da rede;
- Segurança na comunicação dos elementos da rede.

Esse suporte provê tais funcionalidades na forma de nós de serviços que são implantados ao longo da rede virtual.

A camada SON segue o conceito apresentado nesse capítulo de uma **rede de serviços sobrepostos**, na qual todos esses serviços podem ser gerenciados ou orquestrados de maneira otimizada e não concorrente, porém até mesmo complementares em alguns casos. Nessa camada os serviços específicos são implantados, utilizando-se dos serviços sobrepostos mais básicos fornecidos pela camada BOS (UCHôA et al., 2007).

Na subcamada Aplicações, as aplicações desenvolvidas utilizam a infraestrutura, beneficiando-se pelo ganho em suporte a serviços de *multicast*, QoS, segurança e gestão baseada em políticas de modo escalável e efetiva (UCHôA et al., 2007). A figura 1 ilustra a arquitetura Oversea, indicando as suas subcamadas e serviços que podem ser implantados, além da camada de comunicação segura com o protocolo da proposta.

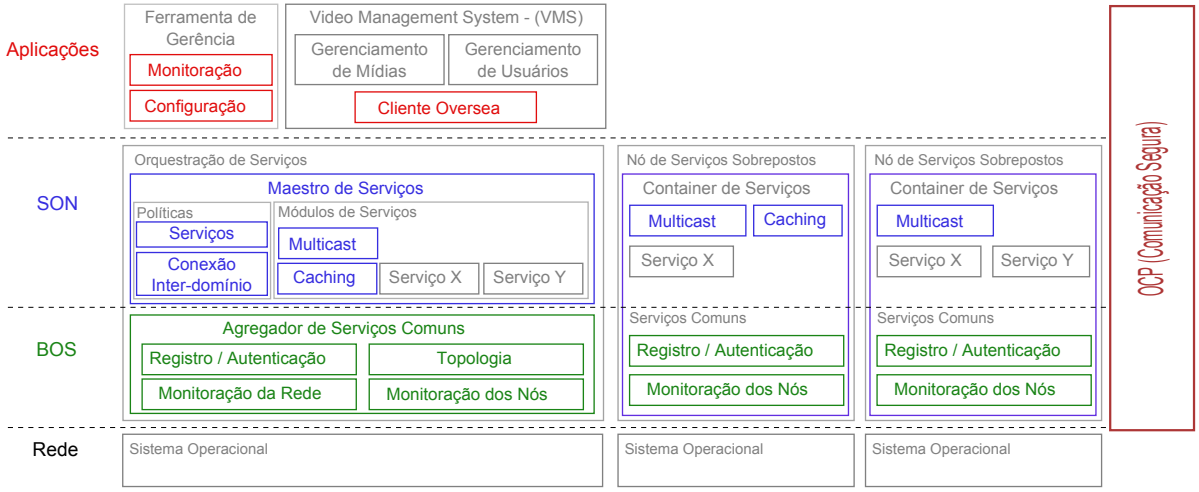

Figura 1: Arquitetura Oversea

Dentro da camada BOS, que pode ser vista como agregadora de serviços comuns, temos os seguintes componentes:

- **Componente de Registro e Autenticação**: esse componente é responsável por fazer o registro de todos os elementos que fazem parte da rede sobreposta, juntamente à autenticação desses componentes. Quando um nó de serviço ingressa a rede sobreposta, é necessário que ele seja registrado. Entretanto, ele precisa obter permissões de ingresso previamente configuradas pelo gerente ou administrador da rede por meio de autenticação. Isso é relevante para evitar que algum intruso possa ser considerado como parte da rede sobreposta.
- **Componentes de Monitoramento da Rede e dos Nós**: esses componentes buscam informações da rede, coletadas por meio do BGP (*Border Gateway Protocol*) na camada de rede, e informações dos nós de serviço utilizando o protocolo SMNP (*Simple Network Management Protocol*). A razão dessas coletas se dá em se saber o *status* dos elementos da rede, mas também para se obter informações para construção e manutenção da topologia virtual da rede.
- **Componente de Topologia**: componente que provê a descoberta e a criação da topologia virtual a partir do monitoramento dos elementos tanto da rede subjacente (TCP/IP) como da rede sobreposta. Esse componente, por exemplo, é utilizado na montagem de rotas em caso de solicitações de vídeo dentro da rede de distribuição de vídeo.

Na camada SON, em que se realizam as “orquestrações” dos serviços, temos o seguinte componente:

Componente Maestro (Gerente) de Serviços: componente que gerencia todos os elementos da rede. O Maestro realiza o registro a autenticação, e centraliza as informações de monitoramento dos elementos da rede. É com ele também que são configuradas, impostas e distribuídas as políticas de uso da infraestrutura de distribuição e dos serviços, como, por exemplo, as políticas de compartilhamento de informações topológicas em interdomínios. É prerrogativa do Maestro gerenciar os serviços de *multicast*, armazenamento e quaisquer outros serviços que possam ser implantados na rede.

Por fim, na camada de aplicação, temos as aplicações que são cliente da infraestrutura

OVERSEA, como o **Cliente Oversea**, que no cenário de distribuição de vídeo é, na realidade, o *Portal de Vídeos*. O Portal de Vídeos faz a gerência das mídias, quanto a sua localização física e os seus metadados, e dos usuários que utilizam o conteúdo, como suas informações cadastrais e *login* do usuário no sistema.

Outras ferramentas de gerência também podem ser acessadas pelo Portal, como as ferramentas de monitoramento e configuração, acessíveis e utilizadas pelo gerente ou administrador da rede de distribuição.

2.2.1 Interdomínio de Redes Sobrepostas

Uma das mais promissoras características da plataforma de distribuição do OVERSEA é o suporte ao uso interdomínio de redes sobrepostas, ou seja, integração de serviços sobrepostos, implementados em diferentes sistemas autônomos.

Os domínios do OVERSEA são denominados OAS (*Overlay Autonomous Systems* - Sistemas Autônomos Sobrepostos) e cada domínio corresponde a uma rede sobreposta ou uma rede de distribuição de conteúdo. Cada domínio tem autonomia de gerência de seus recursos. Contudo, é possível compartilhá-los com quais outros domínios de acordo com políticas.

Esse compartilhamento deve estar em concordância a políticas de roteamento e compartilhamento de topologia preajustadas de tal forma que seu administrador possa selecionar os serviços e os blocos de endereços que queira anunciar em compartilhamento de seus recursos.

Para facilitar o uso de políticas, as redes ou domínios sobrepostos receberam dois tipos de classificações:

- **Redes *stub***: são redes que não permitem repassar informações de outras redes sobrepostas;
- **Redes *transit***: redes que permitem repassar informações de outras redes sobre-

postas, inclusive, ao importar informações dessas outras redes, podem permitir anexar as suas informações de topologia de rede de modo arbitrário.

Nesse tipo de integração, surge um novo componente à arquitetura OVERSEA: o **Agente de Borda**. Esse Agente de Borda na realidade é um nó de serviço que fica na fronteira entre as redes sobrepostas vizinhas, e de fato cria a interconexão dessas redes. De fato, eles são posicionados aos pares, um dentro de cada domínio ou rede.

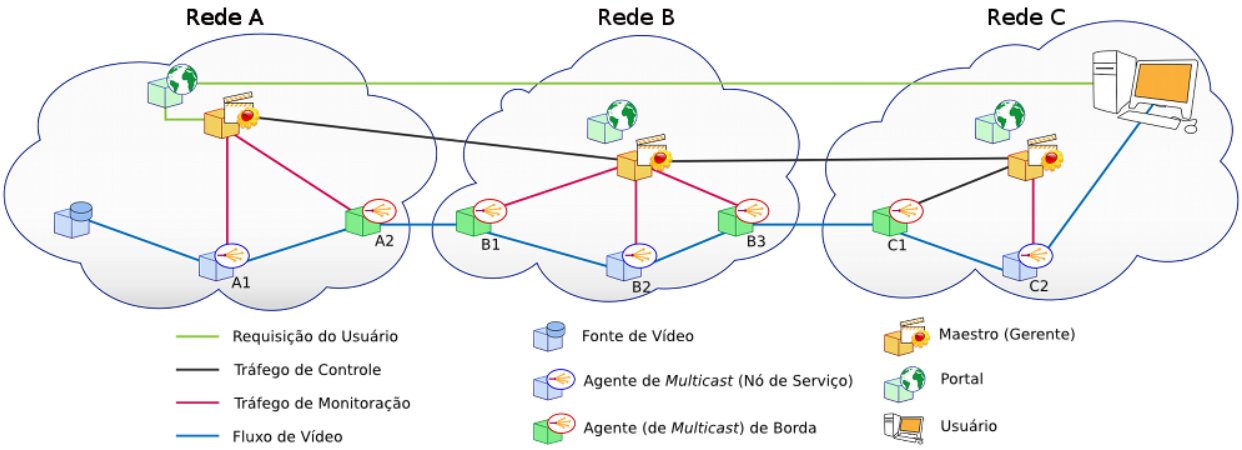

Figura 2: Cenário de Utilização Interdomínio

A figura 2 mostra um cenário de três sistemas autônomos sobrepostos. Esse cenário é minimamente suficiente para exemplificar os perfis de domínios de redes sobrepostas (*transit* ou *stub*) e suas interrelações ao serem integrados.

Há dois tipos de perfis para tal cenário: duas redes (virtuais), são do tipo *stub* - as redes A e C - e uma outra rede é do tipo *transit* - a rede B.

Com a aplicação das políticas de compartilhamento de informações de topologia, tais informações são anunciadas e compartilhadas por um mecanismo de compartilhamento de informações de topologia. É possível para cada rede sobreposta também compartilhar o seu conteúdo de distribuição e equilibrar a carga de trabalho gerada, quando clientes são pertencentes a outra rede sobreposta.

No cenário em questão, figura 2, um usuário pertencente ao OAS C (rede C) solicitará uma transmissão (ao vivo ou sob demanda) ao portal de vídeos do OAS A (rede A). A

transmissão deverá atravessar o OAS B (rede B), que é uma rede *transit* e está entre A e C.

A transmissão deve se beneficiar do serviço de *multicast* sobreposto para que apenas uma cópia do conteúdo trafegue entre as redes até o nó de serviço mais próximo ao primeiro usuário, se houver outros usuários no mesmo nó de serviço.

Dessa forma, o nó de serviço mais próximo desses usuários realiza cópias do conteúdo para cada usuário, diminuindo um tráfego que antes haveria se todas as cópias de seus respectivos usuários passassem por todo os *backbones* das três redes.

Na realidade, sem esse compartilhamento de recursos pelo interdomínio, o usuário da rede ou OAS C acabaria sendo designado a um nó *default* no OAS A, visto que ele não pertence a esse OAS. Solicitando todos os outros usuários do OAS C o mesmo conteúdo no OAS A, eles também seriam encaminhados ao nó *default* do OAS A. Assim, em certo ponto, os nós *default* do OAS A seriam sobrecarregados e várias cópias do conteúdo trafegariam pela rede física, sobrecarregando-a.

Com isso em mente, podemos concluir que o principal benefício do uso integrado dos serviços de diferentes domínios é passar o “peso” que os dados de transmissão causam ao *backbone* da rede física para as áreas periféricas, ou seja, as mais próximas dos usuários. Isso, de fato, permite que mais e mais usuários possam acessar os diversos conteúdos em todas as redes de modo mais otimizado, devido ao uso de interdomínio em redes sobrepostas.

É relevante mencionar que a comunicação entre esses sistemas precisam ter garantias de sigilo e integridade das mensagens de sinalização, bem como dos elementos participantes, já que sem tais garantias essas redes se tornariam suscetíveis a ataques ou então induzidas a participar em ataques como os de Negação de Serviço.

Para entender as questões de segurança dessas redes no capítulo 3 serão aprofundadas as questões de segurança de sistemas de informação.

3 Segurança

Pra resolver a vulnerabilidade de serviços de transmissão em redes sobrepostas, é necessário prover mecanismos de comunicação segura entre os elementos da rede sobreposta.

Este capítulo apresenta os conceitos e técnicas utilizadas para prover segurança que permitirão a comunicação segura das aplicações e sistemas de distribuição de conteúdos.

3.1 Princípios de Segurança em Redes

Quando se fala sobre segurança em redes, três princípios chave devem ser considerados: confidencialidade, integridade e disponibilidade; conhecidos como (C-I-A - *Confidentiality-Integrity-Availability*) (COLE; KRUTZ; CONLEY, 2005). Dependendo do tipo de aplicação e do contexto, um desses princípios será o de maior relevância sobre os demais.

Confidencialidade - diz respeito a evitar a divulgação não autorizada de informação sensível. Essa divulgação pode ser intencional, como, por exemplo, a quebra de uma cifra e a leitura da informação, ou, então, pode ser sem intenção, por descuido ou incompetência dos que estejam tratando a informação. Conforme mencionado no capítulo 1, se esse princípio não for atendido as informações expostas podem ser utilizadas para auxiliar em algum eventual ataque promovido para outros alvos que, não necessariamente, a própria rede. Ademais, o vazamento de informações confidenciais podem gerar problemas jurídicos afetando a honra e o bom nome de pessoas ou instituições.

Integridade - quanto a esse princípio, há três objetivos: (i) prevenção a modificações da informação por usuário não autorizados, (ii) prevenção a modificações não autorizadas ou não intencionais da informação por usuários autorizados e (iii) conservação de consistência interna e externa.

- Consistência Interna deve assegurar que os dados internos estejam consistentes. Para compreender melhor, considere que, em uma base de dados organizacional, o número total de itens que uma organização possui deva ser igual à soma dos

mesmos itens mostrados na base de dados conforme estejam mantidos por cada elemento da organização. Sendo assim, tais dados não devem ser alterados, intencionalmente ou não.

- Consistência Externa assegura que os dados armazenados na base de dados sejam consistentes com o mundo real. Com relação ao exemplo de consistência interna, o número total de itens fisicamente sobre a prateleira devem, por exemplo, igualar ao número total de itens indicados pela base de dados.

Disponibilidade - esse princípio assegura que usuários autorizados de um sistema tenham acesso adequado e ininterrupto às informações do sistema e de rede.

Há alguns outros princípios a serem considerados, definidos a seguir (COLE; KRUTZ; CONLEY, 2005):

Identificação - é o ato de o usuário declarar uma dada identidade ao sistema, um identificador de *logon* por exemplo.

Autenticação - verifica se a identidade alegada pelo usuário é válida, por senhas por exemplo.

Contabilização - é a capacidade de determinar quais ações e comportamentos foram realizadas por um indivíduo de modo único no sistema, além de o tornar responsável por seus atos.

Autorização - ocorre quando um indivíduo ou processo é alocado privilégios que lhe permitam acessar algum dos recursos do sistema.

3.2 Criptografia

A maneira mais utilizada para garantir a segurança à informação, garantindo, portanto, esses princípios, é por meio da **criptografia**, como será discutido a seguir.

O termo **Criptografia** provém do grego ($\kappa\rho\upsilon\pi\tau o\varsigma$, “escondido, secreto”, $\gamma\rho\alpha\phi\omega$, “escrevo” - verbo escrever) para denominar a arte de ocultar informações, é um ramo da matemática que se baseia em transformações de dados. Ela fornece uma importante ferramenta para a proteção da informação e tem sido usada em muitos aspectos de segurança de dados de computadores (GUTTMAN, 1995) e em redes (COLE; KRUTZ; CONLEY, 2005).

Enquanto que, em geral, segurança é o processo de construir muros para prevenir um ataque ou gerenciar riscos na ocorrência de um, a Criptografia desempenha um papel importante no esquema de segurança como um todo. A criptografia é uma ferramenta que por meio de provas matemáticas, portanto determinísticas, dá suporte ao nível de segurança. No entanto, como é comum em matemática, essas provas se aplicam somente a situações específicas, e com frequência, existe o caso em que as pessoas tentam direcionar protocolos ou usar primitivas criptográficas de maneira para que nunca foram intencionados, o que resulta em um sistema inseguro (COLE; KRUTZ; CONLEY, 2005).

Não obstante, criptografia é, de longe, a mais importante ferramenta automatizada para segurança de comunicação e rede, e há dois tipos comuns: a convencional ou simétrica e a de chave pública ou assimétrica. A chamada criptoanálise, ou “quebra de código”, é a área em que compreendem-se as técnicas usadas para decifrar uma mensagem sem qualquer conhecimento dos detalhes de encriptação. A criptologia corresponde à união das áreas de criptografia e criptoanálise (STALLINGS, 2005).

3.2.1 Criptografia Simétrica

Criptografia simétrica é uma forma de sistema criptográfico em que tanto a encriptação quanto a decriptação são executadas usando a mesma chave. Também chamada de criptografia de chave única, era o único tipo de criptografia em uso antes do desenvolvimento da criptografia de chave pública nos anos de 1970, e se mantém a mais utilizada entre as duas. Na encriptação o **texto claro**, ou *plain text*, é transformado em **texto cifrado**, utilizando-se uma **chave secreta** e um **algoritmo de encriptação**. Ao passo que, a decriptação, usando a mesma chave secreta e um **algoritmo de decriptação**, faz o processo inverso, transformando texto cifrado em texto claro. Há dois tipos de ataques que podem ser aplicados em algum algoritmo de encriptação, que são a criptoanálise, baseando-se em propriedades de encriptação, e a força bruta, que consiste em tentar

todas as chaves possíveis (STALLINGS, 2005).

Para o esquema de criptografia simétrica, tem-se os seguintes ingredientes: (i) o texto claro, que é a mensagem ou dados originais inteligíveis que alimentam o algoritmo como entrada; (ii) algoritmos de encriptação e decriptação, que executa diversas substituições e transformações no texto claro e cifrado respectivamente; (iii) chave secreta, que também serve como entrada dos algoritmos criptográficos, sendo independente do texto claro e do algoritmo - o algoritmo produzirá saídas diferenciadas dependendo da chave específica usada no instante, e as substituições e transformações executadas pelo algoritmo são dependentes da chave; (iv) texto cifrado, mensagem misturada produzida como saída e que depende tanto do texto claro como da chave secreta. Para uma dada mensagem, duas chaves diferentes resultarão em dois textos cifrados distintos. É um fluxo de dados aparentemente aleatório que se torna ininteligível (STALLINGS, 2005).

Faz-se necessário um algoritmo criptográfico forte, em que, no mínimo, se faça com que algum oponente que conheça o algoritmo e tenha acesso a um ou mais textos cifrados seja incapaz de decifrar o texto cifrado ou descobrir a chave, sendo esta, necessariamente, tranferida entre o remetente e o destinatário de maneira segura, caso contrário, toda a comunicação usando-a será legível (STALLINGS, 2005).

Cifras de Bloco

Cifra de blocos é um esquema criptográfico em que um bloco de texto claro é tratado como um inteiro e usado para produzir um bloco cifrado de igual tamanho. Tipicamente, são empregados blocos de 64 ou 128 *bits*. A grande maioria de aplicações criptográficas simétricas baseadas em redes fazem uso de cifras de blocos (STALLINGS, 2005).

Em relação ao tamanho do bloco, quanto maior esse, maior a segurança obtida, porém a velocidade de um dado algoritmo criptográfico acaba sendo reduzida. Blocos de tamanho de 64 bits têm sido tradicionalmente considerados um *tradeoff* razoável, o que o tornara quase universal em projeto de cifras de bloco. Contudo, o algoritmo AES (*Advanced Encryption Standard*) utiliza blocos de 128 bits. O mesmo *tradeoff* ocorre com relação ao tamanho das chaves, sendo 128 bits um tamanho comum (STALLINGS, 2005).

Vetor de inicialização (*IV - Initialization Vector*) é um vetor escolhido cujas células contêm

valores aleatórios e com o qual será efetuada uma operação de ou exclusivo (XOR) com o primeiro bloco da mensagem em texto não-cifrado a ser enviado.

AES

O AES (*Advanced Encryption Standard - Padrão de Encriptação Avançado*) é uma cifra de bloco simétrica que surgiu para substituir o padrão antigo chamado DES (*Data Encryption Standard - Padrão de Encriptação de Dados*) como o padrão aprovado para uma gama ampla de aplicações. A sua publicação foi feita pelo NIST (*National Institute of Standards and Technology - Instituto Nacional (estadunidense) de Padrões e Tecnologia*) em 2001. O AES corresponde ao algoritmo criptográfico chamado **Rijndael** desenvolvido pelos criptógrafos belgas Dr. Daemen e Dr. Vincent Rijmen. A proposta do Rijndael para o AES definiu uma cifra na qual os comprimentos tanto do bloco como da chave podem ser independentemente especificados em 128, 192 ou 256 bits. Todavia, a especificação do AES usa três alternativas de tamanho de chave e limita o tamanho do bloco a 128 bits, que vem a ser o mais usualmente implementado. Rijndael foi projetado para ter como características resistência contra todos os ataques conhecidos, velocidade e solidez de código em uma ampla gama de aplicações, e simplicidade de projeto. Como toda cifra de blocos, são feitas iterações em que a cada rotação são executadas 4 funções separadas: substituição, permutação, operações aritméticas em corpo finito, e XOR - “Ou Exclusivo” - com a chave.

Modos de Operação

Um algoritmo de cifra de bloco é um bloco de construção básico com a finalidade de fornecer segurança de dados. Com o intuito de utilizá-la em diversas aplicações, a NIST definiu cinco modos de operação, que são técnicas para se aumentar o efeito do algoritmo criptográfico ou, então, adaptá-lo a uma dada aplicação, como por exemplo, aplicando a cifra de blocos a uma sequência de blocos de dados ou um fluxo de dados.

Os cinco modos são os seguintes:

1. *Electronic Codebook (ECB)* - cada bloco de texto claro de 64 bits é codificado de modo independente usando a mesma chave. Aplicações típicas são transmissões seguras de valores únicos (ex., uma chave criptográfica);

2. *Cipher Block Chaining (CBC)* - a entrada ao algoritmo criptográfico é o resultado de um ou-exclusivo dos próximos 64 bits do texto claro com os 64 bits precedentes do texto cifrado. Aplicações típicas são as de transmissão orientadas a blocos de propósito geral, e as de autenticação;

3. *Cipher Feedback (CFB)* - a entrada é processada em *j* bit por vez. O texto cifrado precedente é usado como entrada para o algoritmo de criptografia produzir saída pseudoaleatória, em que se aplica um ou-exclusivo com o texto claro para produzir a próxima unidade de texto cifrado. As aplicações típicas são as de transmissão orientadas a fluxos de próposito geral, e as de autenticação;

4. *Output Feedback (OFB)* - semelhante ao CFB, com a exceção de que a entrada ao algoritmo criptográfico é a saída DES precedente. As aplicações típicas são as de transmissão orientada a fluxo em canais com ruído, como por exemplo, comunicação via satélite;

5. *Contador (CTR)* - cada bloco de texto claro é operado com ou-exclusivo junto a um contador encriptado. O contador é incrementado para cada bloco subsequente.

Modo Contador

Alguns tipos de mensagens ou dados em geral necessitam ser acessados de maneira não-sequencial, como por exemplo, arquivos em uma base de dados, cujos dados estejam encriptados. Uma proposição custosa seria a de que ao se precisar acessar um bloco aleatório, haja necessidade de se decriptar desde o primeiro bloco até o bloco pretendido, como ocorreria em *cipher block chaining*. Por essa razão, que surgiu o modo contador (*counter mode*), no qual o texto claro não é cifrado diretamente, mas um vetor de inicialização somado a uma constante e a soma resultante é operada com ou exclusivo junto ao texto claro. Esse vetor é incrementado em 1 a cada novo bloco do texto, o que torna fácil decriptar um bloco em qualquer posição no texto cifrado, sem precisar desse modo decriptar os blocos predecessores (TANENBAUM, 2003).

Entretanto, esse modo de operação tem um requisito importante de que tanto as chaves como os vetores de inicialização devam ser escolhidos independentemente e aleatoriamente. Isso significa que se uma chave for utilizada duas vezes ao menos, o texto claro ainda é mantido em sigilo se o vetor de inicialização for diferente a cada vez que for usado (TANENBAUM, 2003).

3.2.2 Criptografia Assimétrica

A criptografia assimétrica, também conhecida como criptografia de chave pública, é uma forma de sistema criptográfico em que as operações de encriptação e decriptação são executadas com o uso de diferentes chaves, a saber, uma chave pública e uma chave privada. Seu desenvolvimento é considerado a maior, e talvez única, revolução da história inteira da criptografia.

Todavia, devido à sobrecarga computacional dos esquemas de encriptação de chave pública atuais, não se pode prever aparentemente o provável abandono da criptografia simétrica. A sua evolução conceitual teve início em um tentativa de se resolver dois problemas associados à criptografia simétrica: a distribuição de chaves e assinaturas digitais (STALLINGS, 2005).

De maneira geral, os algoritmos de criptografia se fiam em uma chave para encriptação e outra diferente, mas relacionada, para decriptação. A encriptação consiste em transformar o texto claro em cifrado usando uma dessas chaves e um algoritmo de encriptação. Enquanto que a decriptação consiste em uma operação análoga, com uso do par de chaves para transformar o texto cifrado em claro.

Entre as finalidades desse tipo de criptografia têm-se a de garantir a confidencialidade e realizar a autenticação. É importante notar que a criptografia assimétrica tem a propriedade de que descobrir uma chave a partir da outra deve ser tão difícil quanto decriptar a mensagem sem chave alguma. Em outras palavras, o poder computacional requerido para mensagem criptografada assimetricamente é aproximadamente o mesmo de se deduzir uma chave assimétrica a partir da outra (COLE; KRUTZ; CONLEY, 2005).

Como exemplo, considere que Alice[1] crie as duas chaves requeridas para a criptografia assimétrica e disponibilize publicamente uma delas. Então, todo mundo, incluindo Bob, tem acesso à chaves pública de Alice. E também significa que Bob, e qualquer outro no mundo, pode encriptar dados e os enviar à Alice de tal forma que somente ela possa lê-los, já que somente ela, ou seja, a pessoa com a segunda chave que não foi enviada - a chave privada de Alice neste caso.

[1] **Alice, Bob**: esses são nomes tradicionalmente utilizados em exemplificações para conceitos de segurança.

Assim, a solução do problema de distribuição de chaves simétricas se torna simples da seguinte forma (COLE; KRUTZ; CONLEY, 2005):

1. Bob cria uma chave simétrica;
2. Ele utiliza a chave pública de Alice para encriptar a chave simétrica de maneira a que ninguém mais possa lê-la;
3. Ele envia a chave simétrica à Alice;
4. Alice a recebe, a decripta com sua chave privada e inicia comunicações com Bob usando a chave simétrica por ele criada.

O algoritmo mais comumente utilizado para o criptosistema de chave-pública é o RSA (segundo os nomes de seus inventores: Rivest, Shamir e Adleman em 1977), que possui a complexidade ao atacá-lo semelhante à dificuldade de se encotrar fatores primos de um número composto e é o mais importante algoritmo de encriptação e decriptação que tem se mostrado praticável para a criptografia assimétrica (STALLINGS, 2005), (MENEZES; Van Oorschot; VANSTONE, 1997).

Além das implicações já mencionadas, a velocidade é o que impossibilita o abandono de criptografia simétrica em favor da assimétrica. Ao se utilizar o RSA, pode-se criptografar 35633 mensagens de 1024 bits em dez segundos em um computador mediano. Em contrapartida, utilizando-se o AES no modo CBC, pode-se criptografar 69893 mensagens de 1024 bits em apenas três segundos. Concluindo que a criptografia simétrica nesse caso é 6,5 vezes mais rápida que a assimétrica(COLE; KRUTZ; CONLEY, 2005).

Essa lentidão se deve basicamente ao uso de propriedades da teoria dos números para derivar sua força. A adição e multiplicação de números muito grandes, na ordem de 1024 bits, leva muitíssimo tempo em computadores comparadas às operações binárias executadas na criptografia simétrica. (COLE; KRUTZ; CONLEY, 2005).

Assinaturas Digitais e *Tag* de Autenticação

Uma das ferramentas fundamentais usadas em segurança de informações é a assinatura, que é um bloco de construção para muitos outros serviços tais como os que, por exemplo,

provêm *irretratabilidade*, autenticação de origem de dados, identificação (MENEZES; Van Oorschot; VANSTONE, 1997).

As transações da Internet, por sua natureza de anonimidade, têm permitido que, em âmbitos comerciais, possam ser rapidamente direcionadas a fraudes. Esse tipo de natureza abre questões de como provar a identidade de alguém em um meio que permite total manipulação de informação (ALBANESE; SONNENREICH, 2004).

Fazendo-se uma analogia ao mundo real, há a necessidade de se provar a identidade de alguém por meio de uma identidade com foto confiável, como um passaporte ou identificação emitida pelo governo. Considerando, agora, uma identidade sem fotografia. Isso é o que se equivale a um **certificado digital**. Nesse certificado contêm dados pessoais com os quais pode-se indentificar o possuidor ou proprietário. A sua emissão é feita por uma organização de confiança, o que torna o próprio certificado fidedigno, e presume-se que parte dele é mantido em segredo e pode apenas ser apresentado pelo seu legítimo dono (ALBANESE; SONNENREICH, 2004).

Com uma assinatura convencional, uma assinatura é fisicamente parte do documento sendo assinado. Já a assinatura digital não é anexada fisicamente à mensagem que é assinada, de forma que o algoritmo a ser usado deve atar a assinatura, de alguma maneira, à mensagem (STINSON, 1995).

O sistema todo de certificação digital compreende entidades ou pessoas que buscam provar a identidade, outras que buscam verificar essa identidade, além de um ou mais terceiros, que sejam confiáveis, que são capazes de executar a verificação. A esse sistema todo se dá o nome de **PKI** (*Public Key Infrastructure* - Infraestrutura de Chaves Públicas) (ALBANESE; SONNENREICH, 2004).

No envio de informações importantes, dois objetivos devem ser assegurados (KUROSE; ROSS, 2005):

- O remetente dos dados é quem diz ser, ou seja, esse remetente assinou os dados e essa assinatura pode ser conferida;
- Os dados transmitidos não foram modificados, uma vez que o remetente os criou e

os assinou.

Uma maneira eficiente de se alcançar esses dois objetivos é por meio de **resumos criptográficos** (*message digest*) (KUROSE; ROSS, 2005).

Resumo criptográfico funciona, muitas vezes, como um *checksum*. Fazendo-se uso de um algoritmo de resumo criptográfico, esse toma uma mensagem de tamanho arbitrário e processa uma "impressão digital" de tamanho fixo dos dados, que resulta no resumo criptográfico. Dessa forma, a mensagem se torna protegida, pois se houver modificações, os resumos criptográficos da mensagem original e da mensagem modificada não serão compatíveis (KUROSE; ROSS, 2005).

É importante dizer que os resumos criptográficos são, da mesma forma que *checksum*, ou códigos de detecção de erros mais sofisticadas como as verificações de redundância cíclica, todos exemplos de funções de *hash* (KUROSE; ROSS, 2005).

Por outro lado, as denominadas *tags* de autenticação, ou simplesmente *MAC* (*Message Authentication Codes - Código de Autenticação de Mensagem*) são semelhantes às funções de *hash*, visto que são funções irreversíveis usadas para criar resultados de tamanho fixo a partir de uma mensagem de entrada de tamanho arbitrário. É por essa razão que são chamadas também de **Funções de *Hash* com Chave** (*Keyed Hash Functions*). A distinção entre as funções existe porque os algoritmos de *MACs* tomam uma chave secreta compartilhada entre o transmissor e o receptor como argumento, juntamente à mensagem a ser autenticada, tendo como resultado o *MAC* (MENEZES; Van Oorschot; VANSTONE, 1997).

O uso do *MAC* para autenticação de mensagem é simples (STALLINGS, 2005):

- O transmissor calcula o *MAC* da mensagem (considerada correta ou conhecida no instante do envio), e envia juntamente à mensagem o *MAC* gerado;
- O receptor autentica a mensagem comparando o *MAC* recebido a um *MAC* regerado da mensagem enviada.

O tamanho do *MAC* é um parâmetro importante de segurança, uma vez que implica na

proporcionalidade da resistência a ataques de tentativa de colisão do *MAC* (DWORKIN, 2005).

Se o *MAC* for validado para autenticação, pode ser que um atacante sem acesso a chave ou ao processo de geração do *MAC* tenha conseguido adivinhar o *MAC* correto para a mensagem, e assim forjar a mensagem. Especificamente, se o atacante selecionar uma *MAC* aleatoriamente de um conjunto dados de tamanho *t bits*, então a probabilidade de que tenha obtido um *MAC* válido é de 1 em 2^t. Como consequência, valores maiores de 2^t fornecem maior proteção contra tal evento. Mas para a maior parte das aplicações, um valor de tamanho de pelo menos 64 *bits* são suficientes para prover proteção à ataques tentando adivinhar o *MAC* que seja dado como correto e usado para forjar a mensagem (DWORKIN, 2005).

CMAC

Assim como o algoritmo de criptografia simétrica AES, o CMAC (*Cipher-Based Message Authentication Code* - Código de Autenticação de Mensagem Baseado em Cifra) é uma função de *hash* com chave que é baseada em uma cifra de blocos de chaves simétricas. O CMAC fornece maior segurança de integridade de dados que um *checksum* ou um código de detecção de erros, já que o CMAC é projetado para detectar modificações de dados não autorizadas e intencionais, assim como modificações acidentais (SONG et al., 2005).

Duração ou Extensão da Chave para Mensagens

A duração ou extensão de uma chave é número total de mensagens para as quais *MACs* são gerados através de todas implementações de CMAC com essa chave. Essa duração da chave pode afetar a segurança dos sistema contra ataques que se baseiam na detecção de um par distinto de mensagens que resultam em um mesmo *MAC* antes que esse seja truncado. Denominamos esse par de mensagens como sendo colisões. Assim como com outros algoritmos *MAC* baseados em cifras de bloco, o atacante pode explorar um evento de colisão a fim de obter um *MAC* válida para uma nova mensagem, cujo conteúdo convenha ao atacante. Isso resultaria em uma falha fundamental de segurança com autenticação (DWORKIN, 2005).

Apesar de ser comum existirem colisões, uma vez que há muito mais mensagens possíveis que *MACs* gerados, elas não devem ocorrer entre mensagens, cujos *MACs* são de fato gerados durante a duração de uma chave. Consequentemente, a probabilidade de que haja ao menos uma colisão depende, na maior parte das vezes, da extensibilidade da mensagem da chave é relativa ao tamanho do bloco, *b*, da cifra de blocos subjacente. Por exemplo, uma colisão é esperada ocorrer entre um conjunto de $2^{b/2}$ mensagens arbitrárias. Isso quer dizer que seriam 2^{64} mensagens para o algoritmo AES com 128 *bits* (DWORKIN, 2005).

Em qualquer sistema que implemente o CMAC, há necessidade de se limitar apropriadamente o risco de atacantes consigam detectar e explorar uma colisão. Um modo de se alcançar isso é adequadamente limitar a vigência de uma chave, o que acaba limitando, por sua vez, a ocorrência de colisões. Assim, de forma geral, a recomendação é limitar o uso da chave a não mais que 2^{48} para blocos de 128 *bits*, e 2^{21} para blocos de 64 *bits* (DWORKIN, 2005).

É importante que haja uma infraestrutura de gerência e distribuição de chaves, para, aliada à duração de chaves limitada, haja um controle para restringir apropriadamente o período de validade de uma chave no sistema.

3.3 Segurança em Redes Sobrepostas

Atualmente vários serviços que estão em produção para a distribuição de midias utilizam as arquiteturas CDN e técnicas de sobreposição. Podemos citar, como exemplo, a *Akamai*[1] e a *Limelight*[2], que oferecem serviços de distribuição para clientes, como *YouTube*[3], *Electronic Arts*[4] e *Dreamworks*[5], para garantir que seus usuários tenham acesso rápido e eficiente a seus conteúdos.

Antes de se introduzir o estado da arte das redes atuais, algumas considerações devem ser apresentadas. Primeiramente, incluindo-se as questões citadas na seção anterior, as redes de distribuição de conteúdo podem lidar com conteúdos de terceiros, que precisam ser protegidos, mantendo sua confiabilidade e validade, e a estabilidade de seus serviços.

Segundo Buyya (BUYYA; PATHAN; VAKALI, 2008), uma das maiores preocupações de uma CDN é fornecer soluções de segurança potenciais para conteúdo confidencial e de alto valor. Basicamente, segurança diz respeito à proteção contra acesso e modificação não autorizados. Caso não haja controle de segurança adequados, uma plataforma de CDN pode se tornar sujeita a fraude eletrônica, ataques de negação de serviço distribuídos, vírus, e outras intrusões que podem prejudicar o negócio.

De maneira geral, uma rede de distribuição de conteúdo precisa prover medidas combativas contra quaisquer riscos que incluam ataques DDoS ou outras atividades maliciosas que interrompam o negócio.

Atualmente, existem dois tipos de CDNs: as comerciais (por exemplo, Akamai, EdgeStream[6], Limelight Networks e Mirror Image[7]) e as acadêmicas (por exemplo, CoDeeN[8], Coral[9] e Globule[10]).

[1] **Akamai**: http://www.akamai.com/
[2] **Limelight**: http://www.limelightnetworks.com/
[3] **YouTube**: http://www.youtube.com/
[4] **Electronic Arts**: http://www.ea.com/
[5] **Dreamworks**: http://www.dreamworksanimation.com/
[6] **EdgeStream**: http://www2.edgestream.com/
[7] **Mirror Image**: http://www.mirror-image.com/site/
[8] **CoDeeN**: http://codeen.cs.princeton.edu/
[9] **Coral**: http://www.coralcdn.org/
[10] **Globule**: http://www.globule.org/

A rede Akamai cobre 85% do mercado, com 25.000 servidores em 900 redes em 69 países (BUYYA; PATHAN; VAKALI, 2008), e dentre seus clientes fazem parte *Adobe*[11], *Amazon*[12], *Apple*[13], *Fox Broadcasting*[14], *Globo.com*[15] e muitos outros (AKAMAI, 2007).

Em relação à segurança, a rede Akamai utiliza a abordagem de segurança em camadas, sendo que na camada de aplicação são utilizadas ferramentas como firewalls de aplicações web (*WAF - Web Application Firewall*), controles de autorização HTTP e priorização de usuário (relacionado à alta disponibilidade) (AKAMAI, 2009). São tratados, também, segurança em DNS (*Domain Name System*) e uso de SSL (*Secure Socket Layer*), certificados digitais e mecanismos de mitigação de ataques de negação de serviço.

Por outro lado, entre as CDNs acadêmicas, destaca-se a rede *open-source* colaborativa **Globule**. Essa CDN foi desenvolvida pela *Vrije Universiteit* (*"Universidade Livre"*) de Amsterdam, Holanda, e usa um modelo colaborativo baseado em P2P (PIERRE; STEEN, 2006).

Em sua gestão de segurança, a rede Globule tem entre suas maiores preocupações garantir que qualquer conteúdo potencialmente malicioso não danifique os servidores, por meio de uso excessivo de recursos, acesso à informação confidencial, ou qualquer outro tipo de má conduta. Isso vem da premissa de que os servidores hospedam conteúdos que não são, muitas vezes, pertencentes aos administradores da rede, e tal ameaça se torna presente, em especial, quando hospedando conteúdo dinâmico, no qual código arbitrário pode ser executado (PIERRE; STEEN, 2006).

Outra questão relacionada à segurança na rede Globule, segundo (PIERRE; STEEN, 2006), tem a ver com o fato de que o proprietário do conteúdo espera que haja garantias de que os servidores de réplicas do conteúdo atuarão de maneira confiável as suas tarefas designadas. Isso, de fato, é relevante, pois um servidor de réplicas malicioso poderia, por exemplo, rejeitar solicitações de conexões, aparentando que o serviço está indisponível aos clientes, ou distribuir conteúdo impróprio em lugar do original.

[11] **Adobe**: http://www.adobe.com/
[12] **Amazon**: http://www.amazon.com/
[13] **Apple**: http://www.apple.com/
[14] **Fox Broadcasting**: http://www.fox.com/
[15] **Globo.com**: http://www.globo.com/

Globule trata dessas questões com uma solução que consiste em instrumentar certos clientes com a capacidade de informar ao servidor de origem do caminho como suas solicitações às réplicas do conteúdo foram tratadas. O servidor de origem pode, assim, detectar comportamento inesperado das réplicas e, então, finalizar a cooperação com os clientes se forem considerados não confiáveis (PIERRE; STEEN, 2006).

Essa técnica, conforme o próprio Pierre afirma, contradiz o objetivo do projeto original de evitar a necessidade de uma aplicação cliente junto ao usuário do sistema. Por isso, atualmente, tem havido uma busca por explorar abordagens que permitam aos servidores de origem obter **provas criptográficas** do comportamento das réplicas, sem o envolvimento do cliente no processo (PIERRE; STEEN, 2006).

Enquanto que as CDNs correspondem à arquitetura do sistema, a técnica para sua implementação, redes sobrepostas, precisam constituir defesas contra suas principais vulnerabilidades. A maioria das soluções se baseiam, em geral, em duas principais abordagens, isto é, a preventiva e a reativa, ou alguma combinação das duas.

A reativa é principalmente usada quando a preventiva se mostra muito complexa de ser desenvolvida, cuja complexidade depende da aplicação (GUERBER; FONSECA, 2009). Entretanto, essa abordagem depende não somente em quão bem elas reduzam a eficácia dos ataques, mas também na acurácia do sistema em determinar quais defesas são requeridas para tratarem de um ataque específico, quando se invocar o sistema defensivo e onde implantá-lo (MIRKOVIC et al., 2004). Dependendo do modo em que se é implementado, poderá causar grande sobrecarga à rede, devida às constantes monitorações no sistema. Ademais, essa abordagem pode deixar aberta a porta para outros ataques que utilizarem métodos mais sofisticados para mascarar seu tráfego (KEROMYTIS; MISRA; RUBENSTEIN, 2002).

Já a abordagem preventiva é a considerada ideal (MIRKOVIC et al., 2004), pois, quando possível de ser feita, pode tornar alguns ataques mais difíceis de serem bem-sucedidos.

Além do tipo de abordagens existentes para tratar de segurança em CDNs, Perez cita algumas questões que devem ser consideradas em redes sobrepostas para que lhes seja creditada segurança (PEREZ et al., 2008). Dentre elas, têm-se as seguintes:

Criptografia e distribuição de chaves - onde cada domínio administrativo precisa ter um conjunto de serviços para gerenciar a informação criptografada (certificados digitais, por exemplo) usadas pelos processos (processos de gerência de rede sobreposta segura), dispositivos (*gateways* seguros) e usuários (administradores de rede). Todas essas entidades-fim estão envolvidas na criação e gestão da rede sobreposta segura e na provisão de seus serviços sobrepostos. O conjunto de serviços de gerência de certificados compõem uma infraestrutura de chave pública, que se responsabiliza pela distribuição das chaves nos domínios e entre domínios conectados.

Provisão segura e dinâmica de serviços - os cenários multidomínio requerem que haja um conjunto de relações de confiança entre os domínios administrativos. A certificação cruzada em P2P - que são redes sobrepostas (CLARK et al., 2006) - seria o mecanismo mais conveniente como parte de pesquisa para se estabelecer confiança entre infraestruturas de chaves públicas existentes em diferentes domínios administrativos.

Localização dos Nós - um dos maiores desafios para a abordagem de redes sobrepostas corresponde a desvantagem que os nós sobrepostos possuem em relação aos sistemas baseados em roteadores, devendo-se ao fato de que os nós não estão localizados às bordas da rede. Esse retrocesso impõe um sério problema, visto que os nós sobrepostos não estão sempre em uma posição de poder observar o tráfego na rede que não esteja explicitamente direcionado a eles. Portanto, para lidar com esse problema que se define um elemento gerenciador sobreposto de segurança, atuando como ponto de decisão de política sobre os roteadores de acesso de um domínio administrativo.

Provisão de enlaces seguros - entre diferentes domínios administrativos é necessária a provisão de enlaces seguros que forneça mecanismos para sobrepor uma rede segura privada acima das redes IPs públicas.

3.3.1 Vulnerabilidades em Redes de Distribuição de Vídeo

As principais vulnerabilidades de redes sobrepostas dizem respeito ao sigilo das informações transmitidas entre os elementos da rede, isto é, mensagem de sinalização, e da autenticação desses elementos.

De maneira geral, Redes Sobrepostas são mais suscetíveis a ataques promovidos por atacantes internos infiltrados na rede sobreposta ou que deixam comprometidos alguns de seus nós (WALTERS; ZAGE; ROTARU, 2008).

A seguir são citadas as vulnerabilidades que são intrínsecas às redes sobrepostas e as potenciais ameaças correspondentes, algumas baseadas no trabalho de Kwon (KWON; KIM; NAH, 2009):

Mensagem Não Autenticada

Essa vulnerabilidade permite que o conteúdo das mensagens trafegadas possam ser modificadas indevidamente. Dentre as ameaças e seus respectivos alvos estão:

- Ameaça ao conteúdo - caso o ataque possa modificar as mensagens, seria possível realizar solicitações a conteúdos não autorizados. Além disso, poderia inserir conteúdo indevido, ou seja, realizar *cache poisoning*;
- Ameaça à CDN - nesse caso o atacante poderia modificar a mensagem de forma a aceitar fluxo de algum transmissor não autorizado. Também seria possível executar um ataque de negação de serviço tendo como alvo a própria rede.
- Ameaça a outras redes ou hospedeiros - o atacante poderia modificar a mensagem para induzir os elementos da CDN a direcionarem os seus fluxos de vídeo a algum hospedeiro ou a alguma outra rede.

Cliente Não Autenticado

A não autenticação do cliente, de maneira semelhante à vulnerabilidade de não autenticação da mensagem, torna possível ao atacante solicitar conteúdo que não lhe é permitido, além de condicionar a realização de ataques do tipo *request replay*.

Nó de Serviço Não Autenticado

Corresponde ao caso em que os elementos da própria rede não são confiáveis. Dentre as ameaças e seus respectivos alvos têm-se:

- Ameaça ao conteúdo - semelhantemente à vulnerabilidade de não autenticação da mensagem, é possível ao atacante inserir conteúdo indevido no cache dos nós de serviço da CDN, o já citado *cache poisoning*, bem como acessar conteúdos não autorizados;
- Ameaça à CDN - com tal ameaça, o atacante pode extrair informações sigilosas da CDN e dos seus usuários, assim como, induzir ataques de negação de serviço à própria CDN;
- Ameaça a outras redes ou hospedeiros - ataques de negação de serviço podem ser induzidos a atacarem outras redes ou hospedeiros por meio da infiltração de algum ou vários nós não confiáveis na rede

Mensagem Expondo Infraestrutura

Caso a mensagem contendo informações da infraestrutura da rede não seja trafegada dentro da rede de maneira sigilosa, qualquer atacante pode potencializar o seu ataque, pois tendo acesso a informações restritas, toda a infraestrutura se torna comprometida a ser alvo ou a promover os ataques.

3.4 Trabalhos Correlatos

Dos trabalhos existentes, que abordam a segurança em redes sobrepostas, destaca-se (ZHU et al., 2007), que cria mecanismos que lidam com dois problemas de segurança para multicast sobreposto: controle de acesso à rede e gerenciamento de chave de grupo. Além desses, apresenta um esquema de distribuição de chaves resiliente a DoS. De maneira geral, busca prover uma solução de roteamento multicast sobreposto seguro e confiável que consiste de uma transmissão fim-a-fim (em camada de aplicação) com o próprio uso de multicast para a distribuição de chaves juntamente ao controle de acesso. Portanto, trata-se de uma solução preventiva, que no caso de alguns de seus nós sejam comprometidos, devido à atualização das chaves, é capaz de manter a sua disponibilidade em face de ataques.

Por sua vez, Keromytis propõe uma interessante solução pró-ativa a ataques DoS, cuja ar-

quitetura é construída com o uso de um combinação de *tunelamento* sobreposto seguro, roteamento via *hashing* consistente e filtragens. De certa forma, utiliza-se mecanismos que visam, em face de ataques, após detectados, desviar o fluxo de ataques para os roteadores centrais de maior velocidade, de modo a "escoá-lo" e assim mitigá-lo. Inclusive, introduz-se aleatoriedade e anonimato na arquitetura, para dificultar que o atacante identifique alvos ao longo do curso para algum destino específico que esteja protegido no sistema, (KEROMYTIS; MISRA; RUBENSTEIN, 2002).

Além desses, Kurian propõe basicamente um esquema de controle de acesso escalável, de forma que cada nó em uma rede de serviços sobrepostos possa decidir se aceita ou não algum tráfego de usuário para ingressar à rede para processamento e repasse em direção ao destino desejado. Essa solução se baseia em um *token* de acesso que contém um novo conjunto de credenciais para o usuário que o possuir, sem que haja necessidade, para a verificação, de manutenção do estado do usuário no nó de serviço sobreposto, fazendo uso de pares de chaves pública e privada (KURIAN; SARAC, 2008).

Ainda tratando-se de redes sobrepostas, dois trabalhos abordam a questão de segurança em redes sobrepostas: a proposta MOSES (SIDIROGLOU; STAVROU; KEROMYTIS, 2007) e uma proposta que busca analisar as vulnerabilidades de mecanismos de *multicast* sobreposto para IPTV baseado em P2P (KWON; KIM; NAH, 2009).

A arquitetura MOSES (*Mediated Overlay Services*) tem como objetivo compor serviços de segurança de redes, tais como anti-*spam*, antivírus, detecção e mitigação de vulnerabilidades automatizadas, e filtragens, com a promessa de escalabilidade e redução de custos de propriedade (SIDIROGLOU; STAVROU; KEROMYTIS, 2007).

Essa arquitetura é uma espécie de solução terceirizada que disponibilizaria serviços diversos, entre esses serviços de segurança em vários aspectos em composição de um grande conjunto de serviços de segurança que podem ser expandidos (SIDIROGLOU; STAVROU; KEROMYTIS, 2007).

Já a proposta de uma infraestrutura de *multicast* sobreposto seguro para serviço de IPTV baseado em P2P se baseia na arquitetura chamada *Scribe*. A ideia dessa proposta é fornecer segurança à arquitetura de *multicast* da Scribe, que é uma infraestrutura descentralizada de *multicast* em camada de aplicação construída sobre a *Pastry*, uma rede

sobreposta P2P baseada em tabelas DHT (*Distributed Hash Table* - Tabelas *Hash* Distribuídas). A rede *Pastry* é um substrato de roteamento e localização de objetos P2P auto-organizáveis e escalável sobreposto à camada física da arquitetura TCP/IP (KWON; KIM; NAH, 2009).

Todas essas soluções abordam, em partes, alguns dos diversos aspectos da área de segurança de redes sobrepostas e, por conseguinte, para redes de distribuição de conteúdo. Entretanto, há de se considerar que a segurança em redes é uma área de atuação vasta, que utiliza diversos conceitos e técnicas para assegurar a proteção do sistema.

3.4.1 Análise

Em comparação a essas soluções, a solução proposta nesse trabalho estende alguns dos benefícios encontrados nessas soluções citadas, como o controle de acesso, distribuição de chaves simétricas e uma abordagem preventiva que não sobrecarrega a rede com mensagens de monitoramento, além de evitar os ataques prevenindo a exposição das informações trafegadas, com a diferença de não haver necessidade constante de tunelamento, o que permite melhor desempenho, bem como a não necessidade de mitigação de efeitos dos ataques.

Ademais, inerentemente, essas soluções não visam atender aos requisitos em cenários como o de distribuição de vídeo, cuja aplicação precisa obter bom desempenho, devido à característica de gerar grande carga à rede. Na presente proposta, tais questões são consideradas na elaboração da solução.

4 Protocolo de Comunicação Segura em Rede Sobreposta

Um mecanismo de comunicação segura necessita que seu protocolo de comunicação possa garantir confidencialidade, integridade e confiabilidade de suas mensagens. Com esse intuito que propomos aqui o protocolo **OCP** (Overlay Communication Protocol - Protocolo de Comunicação (de Rede) Sobreposta). Com esse protocolo é possível que todas as mensagens de sinalizações e controle do sistema se tornem seguras, tais como o registro de elementos, distribuição de atualização de chaves simétricas e outras solicitações de serviços, como as solicitações de vídeo, por exemplo.

Para entendermos a importância do OCP para a comunicação entre os elementos da rede, a seguir explicamos a necessidade e ocorrência de cada um dessas mensagens de sinalizações e controle dentro do sistema:

- **Registro de Elementos**: o registro de elementos da rede é importante para que somente elementos autorizados e autenticados possam ingressar na rede de distribuição. Quando cada nó de serviço é inicializado, faz uma solicitação de registro ao Maestro (Gerente) da rede, que contém um Gerente de Registro, que é o componente de registro e autenticação citado na seção 2. O Gerente de Registro verifica, junto ao Gerente de Topologia, se o elemento pode ser integrado ao sistema. Caso seja aprovada o seu ingresso, é verificado se esse nó de serviço é apenas interno ou se ele pode realizar o interdomínio de acordo com políticas de integração de sistemas acordadas pelos Administradores de Domínios interligados. Caso ele possa realizar o interdomínio, então esse nó de serviço se torna um Agente (de *Multicast*) de Borda e precisa receber o conjunto de chaves de interdomínio, juntamente ao conjunto de chaves internas de seu próprio domínio. Os conjuntos de chaves recebidos pelos nós de serviços são de chaves de criptografia simétrica e cada conjunto de chaves possui cinco chaves, para que caso alguma dessas chaves tenha expirado, as outras quatro possam ser utilizadas ainda. É importante salientar que no serviço de autenticação, os nós de serviço para contactarem o Maestro precisam previamente possuir as chaves públicas e privadas de ambos os lados, pois a autenticação deve ser mútua. Assim, criptografia assimétrica é utilizada para a distribuição dos conjunto de chaves simétricas;
- **Distribuição de Atualização de Chaves Simétricas**: para garantir a segurança

proporcionada pelos mecanismos de autenticidade e integridade da mensagem, é preciso que haja uma atualização periódica das chaves simétricas. O Gerente de Registro é responsável pela redistribuição das chaves renovadas a cada intervalo de tempo. Esse mecanismo utiliza também a criptografia assimétrica;

- **Solicitação de Rota**: toda vez que um usuário solicita um vídeo pelo Portal de Vídeos, esse precisa repassar a solicitação do usuário ao Maestro, para que o Gerente de Topologia dentro do Maestro faça a montagem da rota desde a fonte do vídeo até o usuário do Portal. Caso haja o interdomínio, cada Maestro das redes que estejam interconectadas precisa fazer a montagem do trecho da rota dentro de seu respectivo domínio. Faz parte da montagem da rota a aplicação de políticas de compartilhamento de informações de topologia configurada por cada um dos Administradores de seus respectivos domínios. Após a montagem da rota, o Maestro retorna ao Portal a mensagem contendo a rota gerada. O Portal envia a mensagem de rota ao usuário para que haja o consumo dessa mensagem, que é feita pelos refletores contidos nos nós de serviço, que realizam o *multicast* sobreposto de acordo com o que está designado nessa rota. Nessa ocasião, os nós de serviços são tratados como **Agentes de *Multicast***, levando a solicitação de vídeo contida na mensagem de rota, desde o usuário até a fonte do vídeo solicitado, para, por fim, começar a transmissão no caminho inverso, ou seja, da fonte, passando por todos os agentes de *multicast* contidos na rota, até o usuário;
- **Monitoramento**: com as informações de monitoramento o sistema pode saber como estão o *status* e quais as capacidades de banda de cada um dos elementos que fazem parte da rede. Além disso, essas informações são importantes para que o Gerente de Topologia possa fazer o roteamento entre esses elementos de maneira eficiente e otimizada.

Um ponto importante no serviço de registro é que ele ocorre entre os Maestros de redes interconectadas para realizar integração de serviços. Para isso, a cada par de Maestros, um deles deve ser eleito pelos administradores dos respectivos domínios como o gerador de chaves. O Maestro gerador de chaves é denominado *master*, já o outro é denominado *slave*, e o *slave* deve solicitar registro ao *master* para que haja a troca de chaves simétricas para o uso em interdomínio.

De todos esses serviços, o de maior complexidade é o de solicitação de rota, pois implica no uso da técnica de *multicast* sobreposto, e num cenário interdomínio a complexidade

é ainda maior. Por essa razão, a idealização do OCP buscou garantir as vantagens do *multicast* sobreposto, aliado com os mecanismos de segurança, mas também obter uma abordagem genérica, para que os demais serviços pudessem também ser atendidos por este protocolo.

4.1 Uso do Protocolo

Primariamente, esse protocolo foi desenvolvido para atender a uma necessidade de confidencialidade e sigilo de mensagens em diferentes OAS's em situações nas quais uma rota interdomínio é solicitada, ao passo que em tais circunstâncias cada domínio constroi uma sub-rota que será parte de uma rota completa do cliente em um dado domínio até a fonte de vídeo em um outro domínio. É relevante que tais sub-rotas sejam conhecidas somente pelos seus respectivos domínios, e políticas de compartilhamento de informação de topologia devam ser associadas durante os anúncios dessas informações.

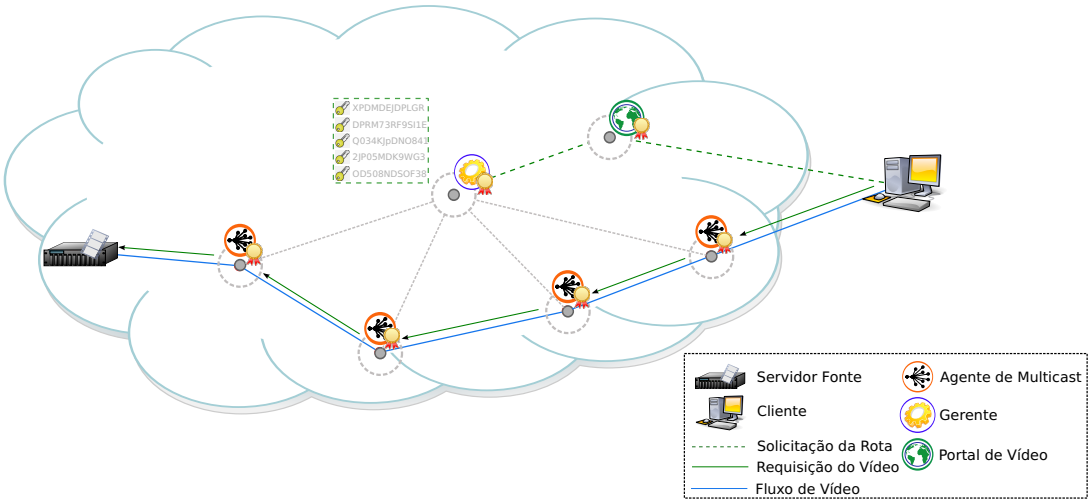

Figura 3: Cenário Básico Ilustrativo do Funcionamento do Protocolo OCP

Para melhor compreender o seu funcionamento, a figura 3 ilustra o funcionamento básico do protocolo. Nessa situação, um usuário faz uma requisição de vídeo ao Portal, que, por sua vez, faz uma solicitação de rota ao Maestro. Assim que a rota é montada, ela é retornada ao Portal que a repassa ao Usuário para que esse solicite ao Agente de *Multicast* mais próximo - informado na própria rota. Esse Agente repassa as solicitações até chegar ao Agente mais próximo da fonte de vídeo. Utilizando o protocolo OCP é possível

garantir que por todo esse caminho, a mensagem foi trafegada de maneira segura e tendo garantias de integridade e confidencialidade.

4.1.1 Descrição do Uso em Cenário Interdomínio

A figura 4, reinserida aqui para facilitar o entendimento, mostra um cenário de três domínios, conforme explicado na seção 2.2.1. Esse cenário é útil para exemplificar um caso em que se demonstra os benefícios resultantes do uso do protocolo, que é a questão de manutenção de sigilo e autenticidade da comunicação e dos elementos da rede de distribuição de vídeo. Também foi o cenário utilizado durante o *10º WRNP*[1] (Workshop da Rede Nacional de Ensino e Pesquisa), que ocorreu em 2009 no Simpósio Brasileiro de Redes de Computadores (*SBRC*[2]), onde foi apresentada a plataforma Oversea utilizando o protocolo de comunicação segura proposto.

Conforme será visto, esse será o cenário para a validação do próprio protocolo.

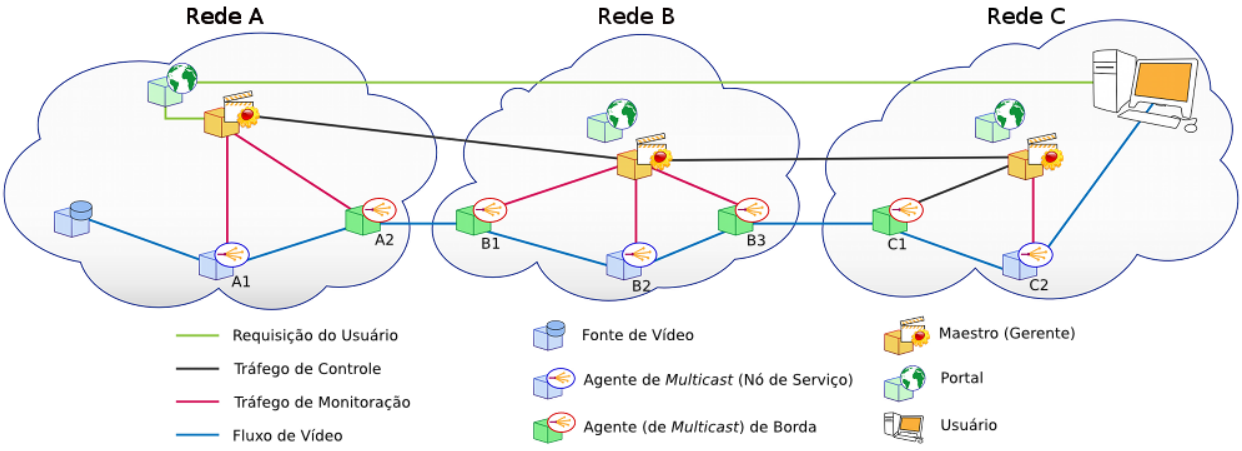

Figura 4: Cenário de Utilização Interdomínio

A seguir serão mostrados resumidamente os passos em um caso de solicitação de um vídeo armazenado em um servidor na primeira rede - nesse caso, de exemplo, a rede A - e de um usuário (que quer acessar os conteúdos disponíveis no Portal da rede A, que é conhecido por esse usuário, acessando-o via *browser*) na terceira rede - aqui exempli-

[1] **SBRC 2009**: http://www.sbrc2009.ufpe.br/
[2] **10º WRNP**: http://www.rnp.br/wrnp/2009/

ficado como a rede C, cuja conexão com a rede A se dá por intermédio da segunda rede - exemplificada pela rede B:

1. A princípio, o cliente ou usuário da rede C faz uma solicitação de visualização de um vídeo da rede A, por meio da aplicação Web, que é o Portal de Vídeos da rede A;

2. O Portal repassa a solicitação ao Maestro, que possui um Gerente de Topologia que se responsabiliza por montar a rota interdomínio;

3. O Maestro ao consultar a sua topologia, verifica que o usuário não pertencente à sua rede e toma a seguinte providência: (i) averigua se a(s) rede(s) vizinhas têm a informação de onde o usuário está de acordo com as informações de topologia enviadas periodicamente pelas demais redes; (ii) caso não encontre o cliente, o procedimento é eleger algum dos nós chamados *default* como sendo o mais próximo do usuário desconhecido. Nesse cenário, entretanto, o Maestro descobriu que a rede B sabe onde se localiza o usuário;

4. Nesse passo, o Maestro da rede A cria uma sub-rota até o agente de *multicast*, e que nesse caso é chamado de **Agente de Borda**, que é um dos agentes dos pares de agentes que ficam na fronteira entre os dois domínios - eles são o elo de ligação interdomínio das redes;

5. Após montar essa sub-rota, o Maestro da rede A contacta o Maestro da rede B e lhe solicita a montagem do restante da rota até o usuário e os procedimentos 3 e 4 são repetidos. Nesse cenário, o Maestro da rede B descobre que o usuário está na rede C, consultando o Gerente de Topologia;

6. Após o Maestro da rede B montar uma sub-rota desde o seu Agente de Borda até o outro Agente de Borda na fronteira, ou no interdomínio, com a rede C, o Maestro da rede B novamente repete os passos 3 e 4, contactando o Maestro da rede C e lhe solicitando o restante da rota;

7. O Maestro da rede C descobre que o usuário é pertencente à sua rede, assim ele monta uma sub-rota desde o seu Agente de Borda da fronteira com a rede B até o nó mais próximo do usuário e responde à solicitação do gerente dessa rede, lhe enviando essa sub-rota;

8. Agora, cascateadamente, o Maestro da rede B ao receber a resposta do Maestro da rede C, repassa a sub-rota, juntamente a sua sub-rota interna, ao Maestro da rede A;

9. Assim, o Maestro da rede A recebendo a resposta do Maestro da rede B, possui toda a rota desde a fonte do vídeo presente em sua rede até o usuário solicitante e pertencente à terceira rede, que é a rede C;

10. O Maestro da rede A repassa a rota como resposta ao Portal de Vídeos da rede A que a põe em formato de URL e envia ao usuário solicitante para que o seu *player* possa fazer a solicitação do fluxo de vídeo ao agente de *multicast* mais próximo do cliente, que é o primeiro na rota;

11. A solicitação é repassada a todos os agentes até que se chegue ao agente mais próximo da fonte para que em sentido contrário, ou seja, da fonte ao usuário o fluxo de vídeo seja enviado.

Algumas considerações devem ser levadas em conta:

- Toda a comunicação é feita com mensagens no formato do protocolo OCP;
- Anteriormente à comunicação exemplificada, deverá ter sido feita a distribuição de chaves, que ocorre durante o registro dos elementos da rede, ou seja, o Portal de Vídeos, os Agentes de *Multicast* e até mesmo os Maestros e os Agentes de *Multicast* de Borda devem possuir as chaves dos domínios em que atuam;
- As chaves simétricas usadas na encriptação e autenticação da mensagem são distribuídas por meio de criptografia de chave pública, e algumas dessas conexões, como por exemplo entre Maestros de diferentes OASs, precisam ser negociadas entre os administradores dessas redes, para que o serviço de compartilhamento de topologia possa fazer a troca de informações de topologia entre elas, obedecendo às políticas de compartilhamento também preajustadas;
- Esse cenário apenas mostra uma situação de uso do protocolo, solicitação de vídeo e montagem de rota, que também pode ser utilizado em outros serviços da rede, como o registro dos elementos, compartilhamento de informações de topologia, adaptação de vídeo durante transmissão e monitoramento;

- Somente foi demonstrado o serviço de solicitação de rota, pois esse mostra o uso mais pleno do protocolo e seu potencial em cenários de interdomínio;
- A mensagem trafegada é mutável a cada passo, pois ela aumenta durante a montagem da rota e diminui em seu consumo, que é quando o usuário, com a rota montada, solicita o fluxo de vídeo.

Especificações da Implementação

O protótipo implementado utilizou as seguintes especificações para a segurança da comunicação do sistema:

- Criptografia Assimétrica: para a distribuição das chaves simétricas, a criptografia assimétrica utilizou o algoritmo RSA, com certificados digitais e chaves produzidas para cada um dos elementos da rede;
- Criptografia Simétrica: para obter sigilo das informações na mensagem foi utilizado o algoritmo AES para blocos de 128 *bits* e o modo de operação escolhido é o modo contador, já que a mensagem pode variar de tamanho pelo trajeto ponto a ponto;
- Integridade: para garantir a integridade da mensagem o algoritmo escolhido foi o CMAC.

4.2 Modelo de Mensagens

Para a elaboração deste protocolo, foi determinado como requisito a definição de um único modelo de mensagem que pudesse satisfazer todas as necessidades de troca de sinalização segura entre os elementos da rede sobreposta.

Portanto, a seguir será feita a descrição do modelo de mensagem do protocolo, indicado na figura 5, e o detalhamento de cada um de seus campos.

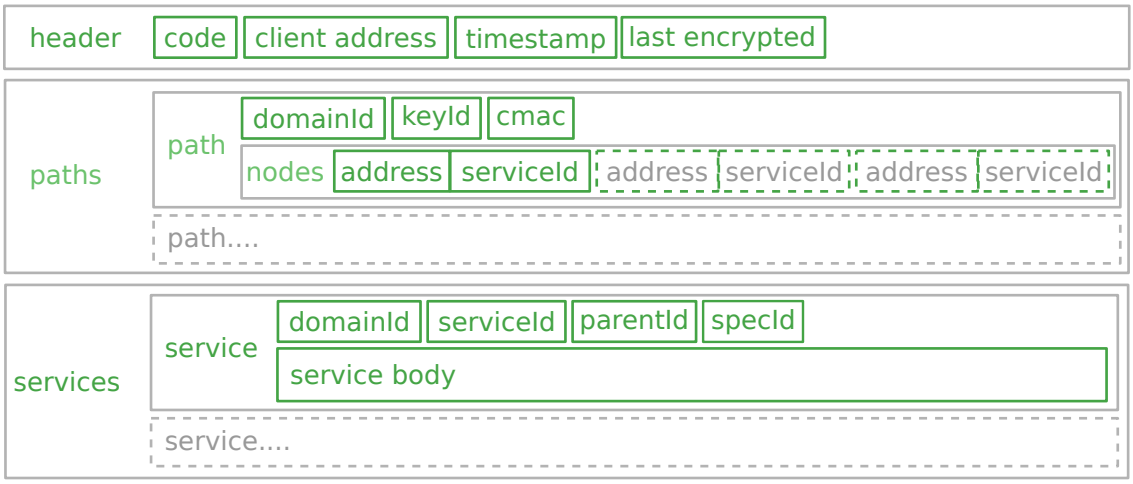

Figura 5: Modelo de Mensagem do Protocolo OCP

Detalhamento dos Campos

A mensagem OCP é composta basicamente de três partes, a saber, cabeçalho (*header*), *paths* e serviços (*services*).

A tabela 2 mostra os campos contidos no cabeçalho, bem como os tamanhos de cada campo com suas descrições.

Tabela 2: Campos do Cabeçalho

Campo	**Número de Bytes**	**Descrição**	**Valores Possíveis**
Code	4	refere-se ao código da mensagem.	códigos OCP
Timestamp	8	indica o instante em que a mensagem foi gerada. Cada mensagem tem um prazo de validade a ser com parado no receptor com esse valor.	numérico
Client Address	8	contém o endereço IP e a porta do cliente, ou seja, usuário do Portal de Vídeos.	endereço IP
Last Encrypted	1/8(1 *bit*)	indica que já chegou ao último elemento da rota que precisa vir em aberto.	0 ou 1
Paths	vide tab. 3	contém os nós de serviço que fazem parte da rota.	lista de *paths*.
Services	vide tab. 4	contém os serviços que serão atendidos durante o consumo da mensagem de rota.	lista de serviços.

O campo *code* pode conter os seguintes códigos:

- “010”: solicitação de rota;
- “020”: solicitação de registro;
- “021”: solicitação de registro com troca de chaves;
- “100”: solicitação de monitoramento;
- “200”: 200 ok, resposta à solicitação bem sucedida;
- “300”: redirecionar solicitação;
- “400”: falha no cliente;
- “401”: falha de chave inválida no cliente;
- “500”: falha no servidor;
- “600”: falha global.

O campo *client address* contém o endereço IP e porta do cliente ou usuário[1], usado também para compor o vetor de inicialização e autenticar se o cliente é válido numa dada topologia e se a solicitação é válida, se houve *spoofing* por exemplo.

O campo *timestamp* corresponde ao instante em que a solicitação foi gerada, considerando que na resposta, o seu valor não é alterado, sendo usado tanto para compor o vetor de inicialização quanto se verificar o tempo válido da mensagem, ou seja, o *timeout*. O vetor de inicialização é utilizado pelo algoritmo AES no modo contador, e o seu valor deve permitir ao contador obter valores únicos conforme é incrementado ou decrementado a cada bloco da mensagem a ser encriptada. No OCP o vetor de inicialização é composto da concatenação dos valores dos campos *timestamp* e *client address*.

O campo *paths* possui subcampos conforme mostrados na tabela 3.

[1] **Cliente ou Usuário**: neste trabalho convencionamos dizer que para a solicitação de rota o termo usado é “usuário”, em outros tipos de solicitação o termo é “cliente”, porém ambos valores são postos no campo *client address*.

Tabela 3: Campos do *Path*

Campo	**Número de Bytes**	**Descrição**	**Valores Possíveis**
Domain Id	2	refere-se ao identificador do domínio do *path*.	numérico.
Key Id	2	refere-se ao identificador da chave a ser usada na criptografia.	numérico.
CMAC	8	contém a *tag* da mensagem gerada dentro do domínio do *path*.	*bytes*.
Nodes	(4 + 2)xN	contém os nós de serviço que fazem parte da rota dentro do *path* junto ao identificador do serviço que está disponível em cada nó.	lista de *bytes*.

O campo *Domain Id* é o identificador do domínio corrente, usado na identificação de qual conjunto (ou *pool*) de chaves deverá ser utilizado no *path*.

O campo *Key Id* é o identificador da chave dentro do conjunto de chaves para criptografia dos nós dentro do *path*, assim como para a geração do resumo criptográfico dentro do *path*.

O campo *CMAC* contém *MAC* da mensagem, que é usado para garantir a autenticação e a integridade da mensagem. Recebeu esse nome porque o algoritmo escolhido para geração do *MAC* da mensagem é o algoritmo CMAC.

A geração do *MAC* será repetida em cada elemento por onde a mensagem passar. Em particular, toda vez que há uma inserção de um novo *path* ou um novo nó, é feito um *commit* da mensagem, que é basicamente a criação do *MAC*.

O campo *Nodes* identifica um conjunto de nós e possui duas informações em cada um: o endereço do nó e o identificador do serviço a que ele está associado, ou melhor dizendo, qual o serviço que ele fornece ao usuário. O “N” na tabela 3 indica o número de nós no *path*.

Em relação aos serviços, esses podem ser para o registro de elementos e de domínio (para o interdomínio), monitoramento, solicitação de rota para vídeo (serviço de transmissão de vídeo ao vivo ou sob-demanda) e compartilhamento de informações de topologia.

Tabela 4: Campos do Serviço

Campo	**Número de Bytes**	**Descrição**	**Valores Possíveis**
Domain Id	2	refere-se ao identificador de domínio para o serviço.	numérico
Service Id	2	refere-se ao identificador ou índice do serviço.	numérico (1, 2, . . .)
Parent Id	2	refere-se ao identificador do serviço "pai" deste serviço.	numérico.
Spec Id	2	refere-se ao identificador de informações específicas do serviço.	numérico.
Service Body	-	contém informações não serializáveis do serviço.	characteres.

O campo *Domain Id* da Tabela 4, caso não seja nulo, indica que tal serviço é encriptado apenas para um dado domínio. A razão de seu uso se deve à necessidade de alguns serviços serem reencriptados a cada domínio. Um exemplo seria o serviço de mídia, que, no cenário interdomínio, precisa ser encriptado para cada domínio desde o usuário e que precisa ser decriptado por todos os nós de serviço da rota interdomínio até a fonte.

O campo *Service Id* recebe seu valor pela ordem de instanciação do serviço, começando com o valor numérico 1, sendo incrementado conforme são acrescentados serviços.

Já o campo *Parent Id* é uma maneira de associar serviços que estão implantados em cada nó de serviço, pois indica quais outros serviços, anteriormente instanciados, estão associados a um dado serviço. Um exemplo seria o "serviço de mídia", que contém as informações para o serviço de transmissão de vídeo, que pode ser ao vivo ou sob-demanda. Esse serviço está associado a um outro serviço, aqui chamado de "serviço de fonte", que contém informações sobre a fonte de vídeo. O identificador do "serviço de fonte" é o valor que é posto no campo *Parent Id* do "serviço de mídia", para que ocorra a associação. Dessa forma, é possível realizar a associação de diversos serviços, cuja associação tem seu critério livre e a cargo de cada implementação.

O campo *Spec Id*, referencia o identificador de informações específicas do serviço, como por exemplo, se esse serviço é de registro, roteamento, fonte, mídia e adaptação. Com esse identificador o Maestro e os nós de serviço podem saber por exemplo, qual o protocolo, portas e outras informações estão associadas ao serviço da mensagem. Apenas o

identificador é inserido na mensagem para que tais informações não precisem ser transportadas.

Por fim, o campo *Service Body* contém as informações específicas e não serializáveis de cada serviço, cujo tamanho depende do serviço instanciado.

É importante salientar que os endereços dos nós e os identificadores de serviços associados são criptografados, com exceção do último nó de serviço mais próximo do usuário. Também são criptografados os conteúdos no *body* de cada serviço. Todas as demais informações são mantidas em texto claro.

De agora em diante, faz-se necessário o detalhamento do funcionamento do protocolo OCP para o seu completo entendimento, apresentando as modificações ocorridas na mensagem ao trafegar pelos três domínios citados.

Mensagem OCP para Montagem da Rota

A seguir será exemplificada como a montagem de rota é feita em um cenário interdomínio.

Para melhor compreensão, a figura 6 mostra o diagrama de sequência para a montagem da rota.

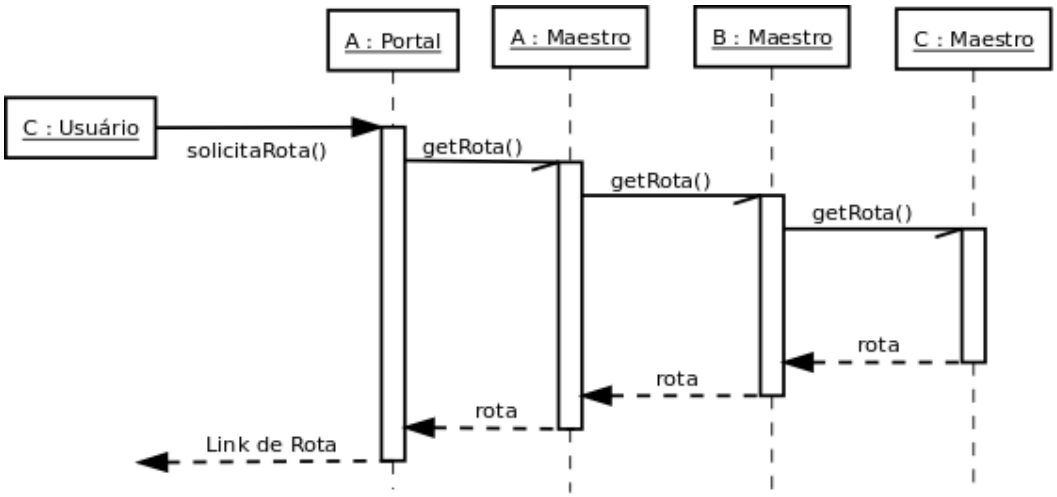

Figura 6: Diagrama de Sequência para Montagem da Rota

Em todas as figuras a seguir, os campos, cujos valores também estão com fundo acinzen-

tado e letras sublinhadas, são os que foram modificados a cada passo correspondente.

Portal rede A ao Maestro rede A

A figura 7 mostra a mensagem enviada do Portal pertencente à rede A ao Maestro da mesma rede. Isso se dá após o usuário da rede C ter feito a solicitação a um vídeo pertencente à rede A no Portal, e esse último a repassa ao Maestro da rede A.

Nessa etapa, o Portal indica ao Maestro A que a solicitação é de montagem de rota - pelo código “010” da mensagem - e que o usuário é de endereço IP igual a C3.

header	**code**	**client address**	**timestamp**	**last encrypted**	
	010	C3	t1	true	
paths	***path***	**domainId**	**keyId**	**cmac**	
		d1	k1	#1	
		node	**address**	**serviceId**	
			-	-	
services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d1	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 7: Mensagem enviada do Portal na rede A ao Maestro da rede A

Dois serviços foram instanciados: o “serviço de fonte” e o “serviço de mídia”.

No serviço de fonte há a *url* para acesso à fonte de vídeo - contida no *body* do serviço.

Já o serviço de mídia contém informações da mídia em si, como o idenficador da mídia, a sua instância, qual o multiplexador e a sessão do vídeo, que indica qual o tipo de transmissão, se é ao vivo ou sob-demanda.

Para a associação dos dois serviços, nesse exemplo, o serviço de fonte foi primeiramente instanciado e seu identificador é referenciado no campo *Parent Id* do serviço de mídia, que é instanciado em seguida. Esse tipo de associação é arbitrário, sendo necessário garantir que todos os elementos consigam conhecer essas associação ao receber e trans-

mitir a mensagem.

Previamente ao envio da mensagem ao Maestro A, o Portal deve gerar o *timestamp* para verificação de *timeout* da mensagem, assim como gerar um *MAC* da mensagem - campo *cmac* - para que a verificação da integridade da mensagem durante o envio até o Maestro A seja constatada.

Por fim, devem ser inseridos na mensagem os identificadores de domínio e chaves da rede A para que a mensagem possa ser decriptada.

Para que o *cmac* possa ser inserido na mensagem é preciso que haja um novo *path* sem nós de serviço, apenas com as informações de domínio, chaves e o *cmac* propriamente. Portanto, esse *path* serve apenas para verificação de integridade da mensagem.

Maestro rede A ao Maestro rede B

Agora, a mensagem é enviada do Maestro A ao Maestro B, conforme a mensagem na figura 8. Isso ocorre logo após o Maestro da rede A verificar que o usuário não é pertencente à sua rede e que o Maestro B tem como montar o restante do caminho até o usuário.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	***#2***
		node	**address**	**serviceId**
			A1	***2***
	path	**domainId**	**keyId**	**cmac**
		d12	***k12***	***#3***
		node	**address**	**serviceId**
			A2	***2***

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d12	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 8: Mensagem enviada do Maestro na rede A ao Maestro da rede B

Os Maestros utilizando o Gerente de serviço de topologia podem obter a rota com o menor caminho entre a fonte e o usuário.

Assim, antes de enviar a mensagem de solicitação para o restante da rota ao Maestro B, o Maestro A adiciona a sub-rota interna à mensagem. Isso pode ser visto na mensagem no primeiro nó do primeiro *path* da mensagem, que contém no campo *addresss* o valor “A1”, que na realidade seria o valor do IP desse nó em *bytes*, porém posto assim para fins didáticos.

O identificador de serviço desse nó aponta para o valor “2”, o que significa que esse nó está associado ao serviço de mídia, mais especificamente, vídeo ao vivo.

Ambos valores são criptografados com a chave de identificador “k1”.

O Maestro gera um outro *MAC* e o insere no primeiro *path* com o valor representativo de “#2”. O valor do *MAC* é na realidade um vetor de 8 *bytes*, conforme visto no detalhamento da mensagem.

Em seguida, o Maestro A adiciona uma novo *path* e lhe adiciona o nó de serviço do tipo agente de borda para que esse nó possa atender ao fluxo interdomínio.

Dessa maneira, é preciso que a informação desse nó seja criptografada com uma chave interdomínio, ou seja, uma chave que pertença a um conjunto de chaves compartilhado entre os dois domínios. Nesse exemplo, o identificador de domínio “d12” e o identificador de chave “k12” é utilizado no *path* contendo o agente de borda “A2”.

Novamente é feito outro *MAC*, porém para integridade da mensagem no interdomínio.

Uma coisa importante a ser reiterada é que as informações do serviço de mídia precisam ser decriptadas para a montagem da rota ao passar por um novo domínio, pois elas são necessárias na seleção dos nós a serem adicionados à rota. Essas informações precisam ser novamente encriptadas com a chave compartilhada para que o Maestro B possa também ter acesso a essas informações para montar seu trecho de rota.

Maestro B a Maestro C

Ao chegar no domínio da rede B, novamente, o Maestro da rede B verifica que o usuário não está em sua rede e é necessário solicitar ao Maestro da rede C o restante da rota ao usuário, pois já possui a informação de que o usuário está na rede C. Modificações são feitas na mensagem, indicadas na figura 9, com a criação de um novo *path*, que conterá a sub-rota do domínio da rede B, ou seja, os nós de serviço “B1” e “B2”. Ambos nós atendem ao serviço de mídia, indicado com o valor “2”.

Mais um *MAC* é feito, a saber, o valor “#4”, utilizando-se o conjunto de chaves do identificador de domínio “d2” e a chave do identificador de chaves “k2”.

Como o Maestro B precisa solicitar o restante da rota, ele inclui na mensagem o seu Agente de Borda - “B3”. Para isso, um novo *path* é criado, o identificador de domínio é o “d23” e o identificador de chave é “k23”, que são identificadores compartilhados entre as

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	***k2***	***#4***
		node	**address**	**serviceId**
			B1	***2***
			B2	***2***
	path	**domainId**	**keyId**	**cmac**
		d23	***k23***	***#5***
		node	**address**	**serviceId**
			B3	***2***

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d23	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 9: Maestro B a Maestro C

rede B e C.

Para garantir a integridade da mensagem nesse interdomínio - rede B e rede C - novamente é feito um *MAC*, com o valor “#5”.

Antes de enviar a mensagem, o serviço de mídia é recriptografado usando o idenficador de domínio “d23”, comum às duas redes.

A volta: Maestro C a Maestro B

A mensagem agora é chegada ao Maestro C. O Maestro C se encarrega de montar a sub-rota em seu domínio para chegar ao usuário, que pertence a essa rede. Novas

informações são acrescentadas à mensagem, como mostrado na figura 10.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	false
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	#4
		node	**address**	**serviceId**
			B1	2
			B2	2
	path	**domainId**	**keyId**	**cmac**
		d23	k23	#5
		node	**address**	**serviceId**
			B3	2
	path	**domainId**	**keyId**	**cmac**
		d3	***k3***	***#6***
		node	**address**	**serviceId**
			C1	2
			C2	2
	path	**domainId**	**keyId**	**cmac**
		d23	***k23***	***#7***
		node	**address**	**serviceId**
			-	-

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d3	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 10: Maestro C a Maestro B

Tais novas informações são o acréscimo da sub-rota no domínio da rede C, que são os nós de serviço “C1” e “C2”. Esses nós são inseridos em um novo *path*.

É importante salientar que o nó de serviço mais próximo do usuário, isto é, o nó “C2”, não é criptografado ao ser inserido no *path*. Isso se deve ao fato de que o usuário não tem

como ter tal informação se essa for encriptada, o que o obrigaria a possuir as chaves.

Portanto, para indicar que esse nó não foi encriptado, o campo *last encrypted* recebe o valor “*false*”.

O serviço de mídia é reencriptado com a identificação de domínio “d3”, pois esse serviço será visualizado, a princípio, apenas pelos nós de serviço da rede C durante o consumo da mensagem.

Mais um *path* é criado para o *MAC* de valor “#7”, a fim de garantir integridade dentro do interdomínio de redes A e B ao enviar a resposta de solicitação de rota vindo da rede C à rede B.

A partir da resposta à solicitação de rota vindo da rede C à rede B, a mensagem contendo a rota completa está finalizada, necessitando apenas trafegar o caminho de volta até a rede A.

A volta: Maestro B a Maestro A

Então, não há muitas mudanças na mensagem - figura 11 - que o Maestro B repassa ao Maestro A.

A exceção se dá em que o último *path* é alterado para ter o identificador de domínio de valor “d12” e identificador de chaves de valor “k12” e o *MAC* da mensagem com o valor “#8", utilizando esses identificadores para garantir a integridade da mensagem no interdomínio da rede A e da rede B.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	false
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	#4
		node	**address**	**serviceId**
			B1	2
			B2	2
	path	**domainId**	**keyId**	**cmac**
		d23	k23	#5
		node	**address**	**serviceId**
			B3	2
	path	**domainId**	**keyId**	**cmac**
		d3	k3	#6
		node	**address**	**serviceId**
			C1	2
			C2	2
	path	**domainId**	**keyId**	**cmac**
		d12	***k12***	***#8***
		node	**address**	**serviceId**
			-	-

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d3	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 11: Maestro B a Maestro A

Rota montada

Para finalizar a montagem da rota, a figura 12 mostra a resposta à solicitação de rota a vídeo que o Portal deve entregar ao usuário para que esse possa visualizá-lo por algum *player* ou navegador *Web*.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	false
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	#4
		node	**address**	**serviceId**
			B1	2
			B2	2
	path	**domainId**	**keyId**	**cmac**
		d23	k23	#5
		node	**address**	**serviceId**
			B3	2
	path	**domainId**	**keyId**	**cmac**
		d3	k3	#6
		node	**address**	**serviceId**
			C1	2
			C2	2

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d3	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 12: Resposta do Portal ao Usuário de Vídeo

Podemos destacar que a única modificação efetuada pelo Portal é retirar o *path* usado para inserção do *MAC* que garante a integridade da mensagem dentro do sistema.

Mensagens OCP durante Consumo da Rota

Após a montagem de mensagem com a rota interdomínio, *player* ou navegador do usuário poderá solicitar ao refletor no nó de serviço mais próximo a visualização do vídeo. Esse processo é chamado de **consumo da mensagem de rota**.

O passos a seguir mostram as modificações que a mensagem de solicitação de vídeo passa ao longo do caminho pelos nós de serviço da rede de distribuição de conteúdo:

Usuário solicita vídeo ao nó de serviço mais próximo

A mensagem que o Portal manda para o usuário, figura 12, é a mesma que ele envia ao nó de serviço mais próximo. Com essa mensagem OCP contendo a rota completa para a fonte de vídeo atravessando os três domínios, o *player* do usuário passa a solicitação ao nó de serviço “C2”, que é o mais próximo do usuário da rede C.

Nó C2 repassa solicitação ao C1 - Domínio da rede C

Então, o nó de serviço “C2” verificou que o conteúdo buscado precisa ser acessado por um outro nó de serviço, ou seja, o nó de serviço “C1”. A figura 13 mostra a mensagem enviada.

Portanto, para lhe repassar a solicitação de vídeo, o nó de serviço “C2” modifica a mensagem, retirando dela o nó com a sua denominação e, por ter de modificar o conteúdo do último *path*, será necessário gerar um novo *MAC*, tendo esse o valor de “#10”.

Outra importante modificação é no campo *Last Encrypted*, que acaba sendo atribuído com o valor de “true”, pois de agora em diante não há mais nós em aberto, ou seja, precisam ser decriptados.

Nó C1 repassa solicitação ao B3 - Interdomínio

O nó de serviço “C1” precisa repassar a solicitação de vídeo ao próximo nó de serviço, ou seja, nó de serviço “B3” e a mensagem chega, conforme a figura 14.

header	code	client address	timestamp	last encrypted
	010	C3	t1	***true***
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	#4
		node	**address**	**serviceId**
			B1	2
			B2	2
	path	**domainId**	**keyId**	**cmac**
		d23	k23	#5
		node	**address**	**serviceId**
			B3	2
	path	**domainId**	**keyId**	**cmac**
		d3	k3	***#10***
		node	**address**	**serviceId**
			C1	2
			-	-

services	*service*	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d3	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 13: Usuário da rede C a nó de serviço “C2” também da rede C

A mensagem está passando no interdomínio entre a rede C e a rede B, e assim, necessita modificar o campo de identificador de domínio do serviço de mídia para valor “d23”, isto é, o serviço de identificador de serviço com valor “2”.

Com esse identificador, o conteúdo do serviço de mídia é reencriptado para que os nós de serviço entre os dois domínios, ou redes, possam ter acesso a tais informações de serviço.

A última modificação antes do envio, é a retirada do último *path* que continha o endereço

header	code	client address	timestamp	last encrypted
	010	C3	t1	true
paths	*path*	domainId	keyId	cmac
		d1	k1	#2
		node	address	serviceId
			A1	2
	path	domainId	keyId	cmac
		d12	k12	#3
		node	address	serviceId
			A2	2
	path	domainId	keyId	cmac
		d2	k2	#4
		node	address	serviceId
			B1	2
			B2	2
	path	domainId	keyId	cmac
		d23	k23	#5
		node	address	serviceId
			B3	2

services	*service*	domainId	serviceId	parentId	specId
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	domainId	serviceId	parentId	specId
		d23	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 14: Mensagem de C1 da rede C a B3 da rede B

e serviço do nó de serviço “C1”.

Nó B3 repassa solicitação ao B2 - Domínio da rede B

A mensagem agora passa pelo domínio da rede B, partindo do nó de serviço “B3” ao nó de serviço “B2”, ambos da rede B. A figura 15 demonstra a mensagem enviada.

O nó de serviço faz apenas duas mudanças: retira o *path* que possuía identificadores de chave e domínio do interdomínio entre as redes 2 e 3; e reencripta o serviço de mídia novamente com o identificador de domínio de valor “d2”.

Nó B2 repassa solicitação ao B1 - Domínio da rede B

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	#6
		node	**address**	**serviceId**
			B1	2
			B2	2

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d2	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 15: Mensagem de B3 a B2, ambos da rede B

Ainda na rede B, a mensagem trafega do nó de serviço “B2” ao nó de serviço “B1”, e chega a esse último nó conforme a figura 16.

O nó de serviço “B2” faz as seguintes mudanças: retira o nó de serviço “B2” e por isso precisa gerar um novo *MAC* de valor “#11” a ser inserido no último *path*.

Nó B1 repassa solicitação ao A2 - Interdomínio

Novamente a mensagem trafega em interdomínio - entre a rede B e rede A - partindo do nó de serviço “B1” até o nó de serviço “A2” conforme a figura 17.

O nó de serviço “B1”, basicamente, retira o *path* do domínio da rede B e reencripta o serviço de mídia com o identificador de domínio “d12” de uso para o interdomínio entre a rede B e a rede A.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2
	path	**domainId**	**keyId**	**cmac**
		d2	k2	***#11***
		node	**address**	**serviceId**
			B1	2
			-	-

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d2	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 16: Mensagem de B2 a B1, ambos da rede B

Nó A2 repassa solicitação ao A1 - Domínio da rede A

A mensagem agora está na rede A, partindo do nó de serviço “A2” ao nó de serviço “A1”, como mostra a figura 18.

O nó de serviço “A2” retira o *path* do interdomínio da rede B e rede A e reencripta o serviço de mídia com o identificador de domínio “d1” de uso para o domínio da rede A.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2
	path	**domainId**	**keyId**	**cmac**
		d12	k12	#3
		node	**address**	**serviceId**
			A2	2

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d12	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 17: Mensagem de B1 da rede B a A2 da rede A

Nó A1 repassa solicitação à fonte de vídeo - Domínio da rede A

Esse é o último passe da solicitação de vídeo, já que o nó de serviço “A1” é o mais próximo da fonte e somente necessita de uma única informação, que é a *url* de acesso à fonte: “mmsh://ip:port/mediaName”.

Nessa *url*, há o protocolo de comunicação (mmsh), IP e porta da fonte e o nome da mídia.

header	**code**	**client address**	**timestamp**	**last encrypted**
	010	C3	t1	true
paths	***path***	**domainId**	**keyId**	**cmac**
		d1	k1	#2
		node	**address**	**serviceId**
			A1	2

services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		d1	1	-	2
		service body			
		mmsh://ip:port/mediaName			
	service	**domainId**	**serviceId**	**parentId**	**specId**
		d1	2	1	3
		service body			
		mediaId, mediaInstance, muxer, session			

Figura 18: Mensagem de A2 a A1, ambos da rede B

Transmissão ao Usuário

Por fim, é finalizada a comunicação de sinalização e controle do OCP por meio da solicitação de vídeo, pois o fluxo de vídeo, que faz o caminho inverso, isto é, da fonte ao usuário, para que esse possa visualizá-lo pelo Portal de Vídeos.

Para melhor compreensão, a figura 19 mostra o diagrama de sequência para o consumo da rota.

Esse exemplo, aqui explicado, foi utilizado para as medições apresentadas no capítulo 5.

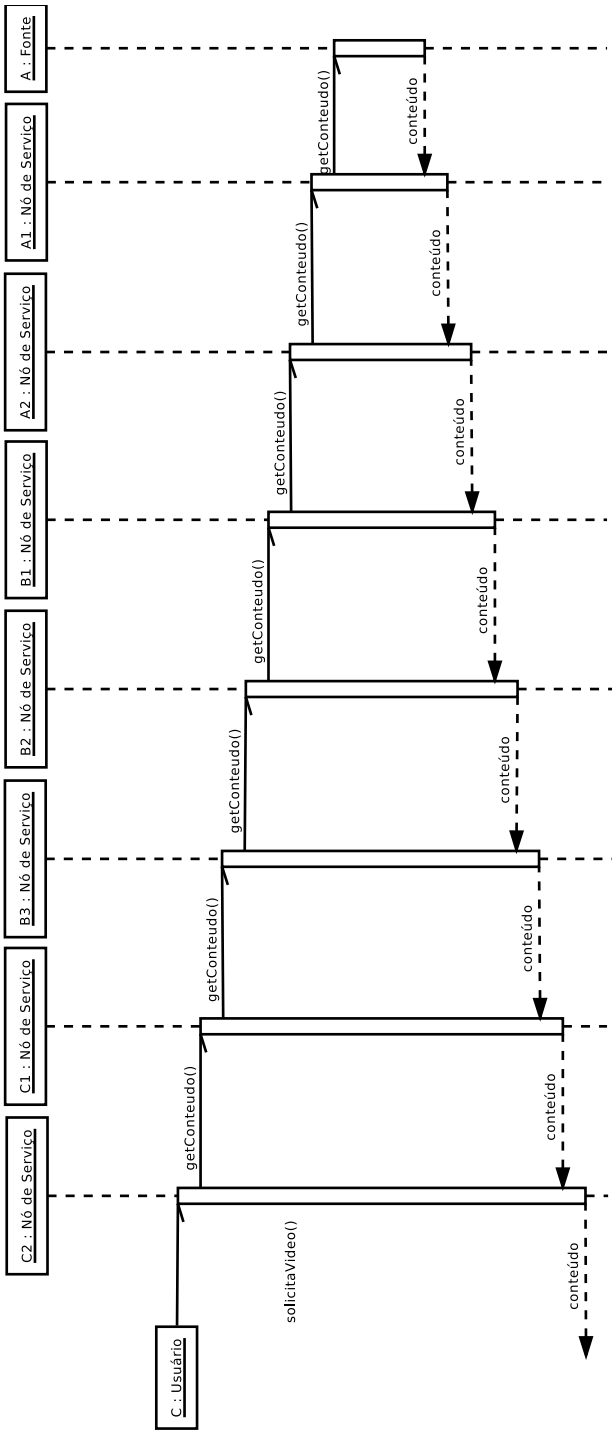

Figura 19: Diagrama de Sequência para Consumo da Rota

5 Análise e Validação

Para se validar o protocolo e os mecanismos de segurança propostos, dois critérios foram utilizados, a saber, o desempenho durante as comunicações e a robustez da segurança empregada.

5.1 Metodologia

A seção a seguir descreve os critérios adotados para se validar a solução proposta neste trabalho, bem como qual a metodologia utilizada para esse fim.

5.1.1 Montagem de Cenário de Medições

O cenário utilizado para avaliação é o descrito na seção 4.1.1, numerando os nós “A1”, “A2”, “B1”, “B2”, “B3”, “C1” e “C2”, como nós 1, 2, 3, 4, 5, 6 e 7 respectivamente.

E para realização das medições para validação, foram utilizados:

- Oito computadores com o sistema operacional Linux da Distribuição Ubuntu[1] 10.04 LTS (*Lucid Lynx*) dentro do LARC;
- Sete computadores foram utilizados para cada um dos nós de serviço;
- Um computador foi utilizado para captura das mensagens, contendo o Portal de Vídeos, com o qual se capturou a montagem da rota, e também contendo o usuário, com o qual se capturou o consumo da mensagem de rota;
- Dos sete computadores, três também continham cada um dos três Maestros, representando cada um dos três domínios respectivamente;
- Foi configurada estaticamente três topologias correspondendo a cada domínio, com as quais seus respectivos Gerentes de Topologia montaram os trechos de rota em seus respectivos domínios;

- Toda a implementação foi feita na linguagem de programação Java.

Tal configuração de ambiente de medições foi suficiente para a realização das medições para validação da proposta.

5.1.2 Desempenho

Para se medir o desempenho da solução de comunicação segura com o protocolo OCP, duas métricas foram adotadas:

1. **Latência**: com o mesmo objetivo de verificar o impacto da solução em relação ao desempenho da comunicação dentro da CDN e por conseguinte dos sistemas que estejam implantandos sobre ela, faz-se necessária a utilização da métrica de latência, que corresponde ao tempo desde a solicitação de algum tipo de serviço até o envio do primeiro pacote de resposta à tal solicitação. A maneira como tal métrica será medida, se dará por meio de um *sniffer*, tal como o **Wireshark**[2], posicionado junto ao cliente, o que permitiria uma **comparação experimental** da solução em dois cenários relevantes:

   - Sem a solução de comunicação segura;
   - Com a solução de comunicação segura (OCP).

   Dessa forma, poder-se-ia quantizar a sensação do cliente relativa à resposta do sistema em solicitações como, por exemplo, o pedido de execução de algum vídeo no Portal de Vídeos, sendo a latência, do ponto de vista do cliente, o tempo desde o clique no *thumbnail* do vídeo até à sua execução no *player*.

2. **Número de Mensagens trocadas**: A partir da medição do número de mensagens trocadas, é possível determinar o quanto a solução proposta impacta sobre a rede de distribuição de conteúdo, ou seja, qual o valor de seu *overhead*. A ideia é que o protocolo de comunicação segura evite que esse número de mensagens seja excedente, buscando ser sucinto e de fluxo direto, o que evita sobrecarga a rede com tráfego de controle. Essa solução contemplará dois cenários:

[1] **Ubuntu**: http://www.ubuntu.com/
[2] **Wireshark**: http://www.wireshark.org/

- Sem a solução de comunicação segura;
- Com a solução de comunicação segura (OCP);

Com esses dois cenários, busca-se uma **comparação analítica** que permita mostrar que a solução não produz grande sobrecarga à rede para a troca de mensagens durante a comunicação entre os diversos elementos e para os diversos serviços disponibilizados pela CDN, além de mostrar que o fluxo se mantém direto, ou seja, não há necessidade de desvios para autenticação, algo que também impacta no desempenho das comunicações do sistema.

É importante salientar que, com ambas as métricas, o cenário de multidomínio será empregado, visto que esse foi um dos motivadores de se projetar uma solução de comunicação segura em redes de distribuição de conteúdo. Na comparação com latência, o tipo de solicitação será a de exibição de vídeo, e as medições serão feitas incrementando-se o número de nós participantes da rota, que é possível ao se deslocar o cliente desde a posição mais próxima da fonte, e nesse caso no mesmo domínio, até a posição mais distante da fonte, já em outro domínio, conforme o cenário ilustrado pela figura 4 na seção 4.1.1.

5.1.3 Nível de Segurança

Para a obtenção do nível de segurança que a solução garante, é importante considerar o número de mensagens que podem ser trocadas de maneira segura.

Isso está diretamente relacionado à entropia que o vetor de inicialização possui, uma vez que o contador desse vetor não pode repetir seu valor em cada outro bloco de texto claro, ou seja, o vetor precisa ter um valor único, para ser utilizado corretamente no modo contador.

Além desse critério, um modo de validação é o tratamento dado às vulnerabilidades presentes nas redes de distribuição de vídeo. Será feita uma análise comparativa entre a solução baseada na rede Scribe para uma infraestrutura de *multicast* sobreposto seguro para serviço de IPTV baseado em P2P (KWON; KIM; NAH, 2009), uma vez que além

de ser relacionada à distribuição de vídeo, apresentou uma lista de vulnerabilidades das redes de distribuição e como foram tratados tais problemas.

5.2 Resultados e Análises

Esta seção apresenta os resultados obtidos pela medições realizadas em um cenário multidomínio. Também apresentamos as análises desses resultados nessa seção.

5.2.1 Desempenho

De acordo com a metodologia apresentada o desempenho da solução foi medido e analisado conforme a seguir:

Latência: Para se mensurar o desempenho dos dois cenários foram considerados, ou seja, utilizando-se uma solução multidomínio sem mecanismos de segurança e outra solução utilizando protocolo de comunicação segura do OCP.

As medições ocorreram em duas ocasiões:

- Montagem da Rota: corresponde ao instante após o Portal de vídeos receber a solicitação de vídeo e a repassar ao Maestro - para que esse monte a rota da fonte do conteúdo requerido até o usuário solicitante - até a chegada da resposta com a rota montada ao Portal;
- Consumo da Rota: O Portal retorna a URL contendo a rota para que o usuário possa acessar o conteúdo. As medições corresponderam ao instante em que o *browser* ou o *player* do usuário, utilizando essa URL, solicita o conteúdo ao nó de serviço mais próximo do usuário, conforme definido na URL, até a resposta contendo os cabeçalhos que indicam o início da transmissão.

Montagem da Rota

Na montagem da rota, o usuário foi deslocado do primeiro nó até o sétimo nó e foram feitas 10 iterações para medir o atraso médio.

A solução sem segurança foi comparada graficamente a solução com OCP e obtivemos o seguinte gráfico na figura 20. Esse gráfico mostra os desvios padrão dos valores medidos em barras verticais para cada abordagem.

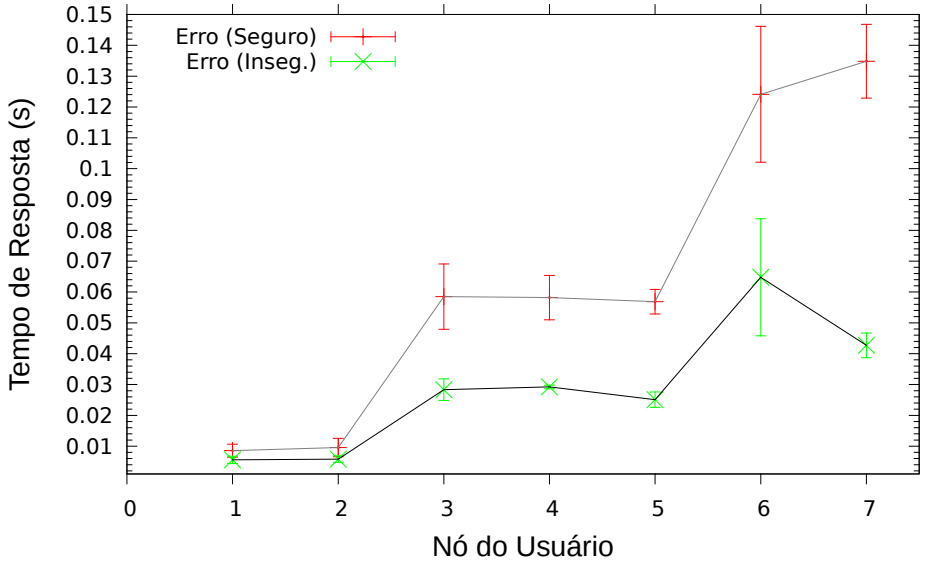

Figura 20: Montagem da rota sem segurança e com o OCP

Desse gráfico, na solução sem segurança, podemos verificar que há alguns instantes em que o atraso se mantém praticamente constante (entre os nós 1 e 2, e os nós 3, 4 e 5), mas ocorrem subidas praticamente lineares entres os nós 2 e 3, e os nós 5 e 6, além de uma queda brusca entre os nós 6 e 7.

Contudo o interessante nesse caso é que entre os nós que fazem parte de um mesmo domínio, há uma constância no atraso - os nós 1 e 2 são do domínio da rede 1, os nós 3, 4 e 5 são do domínio da rede 2. A única exceção corresponde aos nós 6 e 7 com uma queda no atraso.

Por outro lado, quando ocorre a troca de domínio, o atraso é aumentado, e isso se deve ao fato de que nesses instantes os Maestros precisam se comunicar para que cada um contribua com o seu trecho da rota, o que gera tal atraso acentuado.

Na solução com o protocolo OCP, verificamos que o comportamento, conforme esperado,

é análogo ao da solução sem segurança, ou seja, com atrasos constantes dentro dos domínios e com subidas acentuadas entre os domínios.

Entretanto os valores de atraso não são os mesmos, obviamente. Sendo esse tempo um pouco maior na solução com segurança, já que faz-se uso das técnicas que garatem sigilo e autenticidade da mensagem transmitida.

Para entendermos melhor essas diferenças, a tabela 5 mostra a diferença entre os atrasos quando o usuário se localizava em cada um dos sete nós durante a montagem de rota.

Tabela 5: Diferença dos Atrasos medidos - Montagem de Rota

Nó	**Atraso (s)**
1	0,002973
2	0,003755
3	0,030155
4	0,028965
5	0,031769
6	0,059336
7	0,092120

Podemos observar que as diferenças entre os tempos não são tão acentuadas, sendo que há uma constância em nós de mesmo domínio.

Quando há troca de domínio, é necessário que não somente as informações dos nós sejam encriptadas, mas também algumas informações de serviços precisam ser reencriptadas, como é o caso dos serviço de mídia, que precisa trafegar de modo seguro, e ao fim da montagem da rota precisa estar encriptado com a chave do domínio em que o usuário se encontra. Isso explica em parte esse aumento de atraso, juntamente com toda a criptografia de sigilo e autenticidade que ocorre na comunicação com o OCP.

Não obstante, o pior caso, ocorre com 7 nós, portanto com os três domínios, em que essa diferença é em torno de 92 milisegundos, e no melhor caso, ou seja, com apenas 1 ou 2 nós, essa diferença é de apenas 2 a 3 milisegundos.

Consumo da Rota

Da mesma maneira da montagem da rota, o usuário foi deslocado do primeiro nó até o

sétimo nó e foram feitas 10 iterações para medir o atraso médio.

O gráfico da figura 21 mostra o consumo da rota para a solução sem segurança.

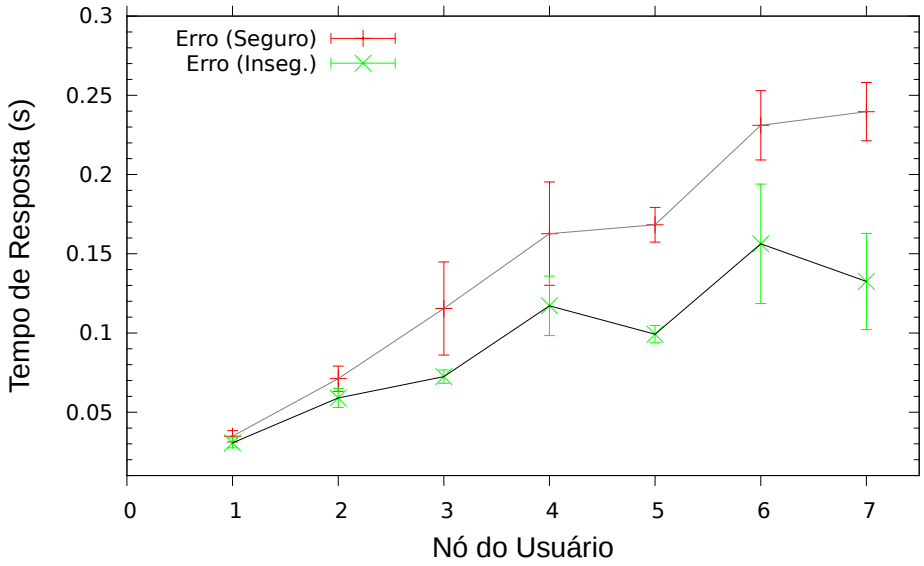

Figura 21: Consumo da rota sem segurança e com o OCP

Nesse caso, o gráfico explica o comportamento dos refletores nos nós de serviço, conforme é aumentado o número de nós na rota.

Os valores obtidos não são afetados pela questão de interdomínio. Assim se esses refletores em cada nó de serviço estivessem todos num mesmo domínio, o gráfico seria equivalente.

Podemos verificar que há uma subida e posteriormente uma queda no atraso, entretanto o atraso resultante é de aumento.

Interessantemente, para a comunicação com o OCP, há uma ligeira modificação no comportamento dos refletores nos nós de serviço.

A oscilação que havia na solução sem segurança foi atenuada, se tornando um pouco mais linear a princípio para no fim oscilar com uma certa amplitude.

Ademais, na solução do OCP, há a influência da mudança de domínio, pois ao trocar de

domínio, é preciso que as informações de mídia sejam reencriptadas para que o domínio seguinte tenha acesso a essas informações.

Isso poderia explicar porque há alguma constância nos nós em que não há troca de chaves, como ocorre no nó 4 e possivelmente, por projeção, ocorreria no nó 7.

Por ora, é interessante saber as diferenças nos valores de atraso das duas abordagens, conforme é mostrado na tabela 6.

Tabela 6: Diferença dos Atrasos medidos - Consumo de Rota

Nó	**Atraso (s)**
1	0,004141
2	0,012180
3	0,042970
4	0,045593
5	0,069034
6	0,074866
7	0,107163

Novamente, vemos que o pior caso, com 7 nós e em interdomínio, há um acréscimo de 107 milisegundos, mas, no melhor caso, esse acréscimo é de apenas 4 a 12 milisegundos.

Portanto, podemos dizer que em um cenário de três redes, em que 7 nós de serviço espalhados por essas redes, há um custo de menos de 200 milisegundos para se obter sigilo, confiabilidade e integridade da mensagem trafegada por entre esses três domínios, o que consideramos factível e razoável para garantir a segurança do sistema.

Número de Mensagens trocadas: o número de mensagens enviadas implicam no desempenho da solução.

Durante as medições nos cenários de comunicação segura e insegura e de acordo com os pacotes capturados, constatou-se que ambos cenários tiveram o mesmo número de pacotes transmitidos no ponto de captura dos pacotes.

A figura 22 e a figura 23 mostram exemplos de captura tanto com comunicação insegura como com comunicação segura para a montagem da rota.

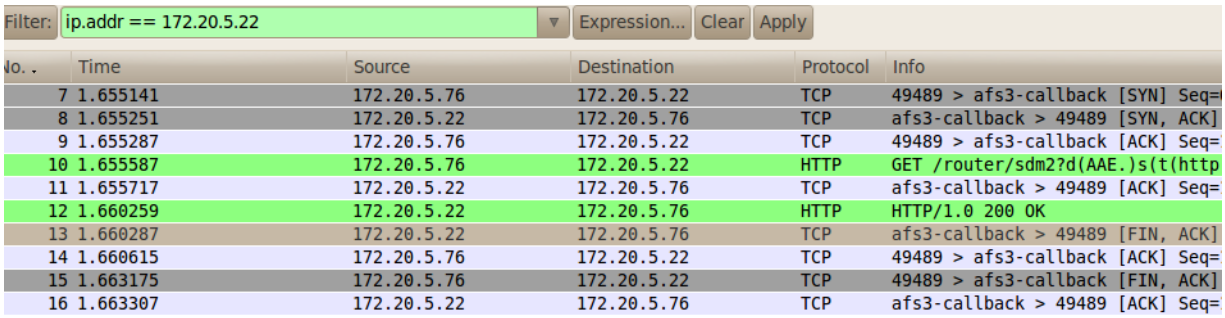

Figura 22: Amostra de Captura da Montagem da Rota Sem Segurança

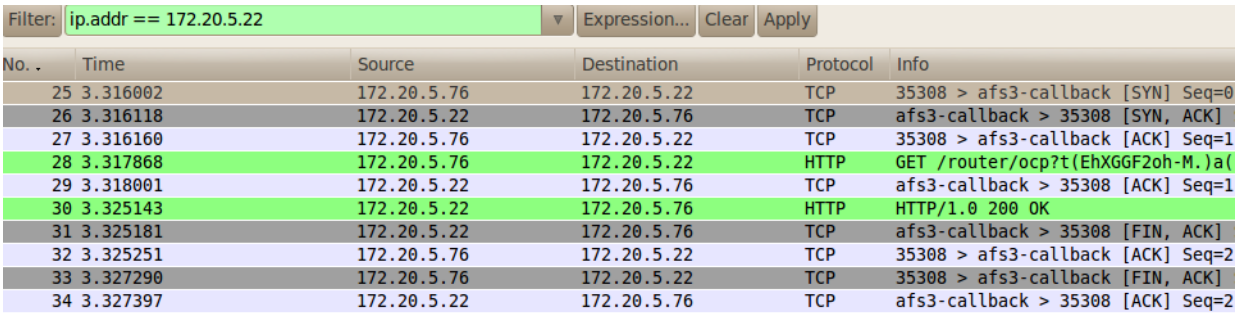

Figura 23: Amostra de Captura da Montagem da Rota Com Segurança pelo OCP

Essas figuras mostram que o número de pacotes transmitidos entre o Portal de Vídeos e o Maestro da rede 1 se manteve igual.

Já as figuras 24 e 25 mostram exemplos de captura tanto com comunicação insegura como com comunicação segura para o consumo da rota.

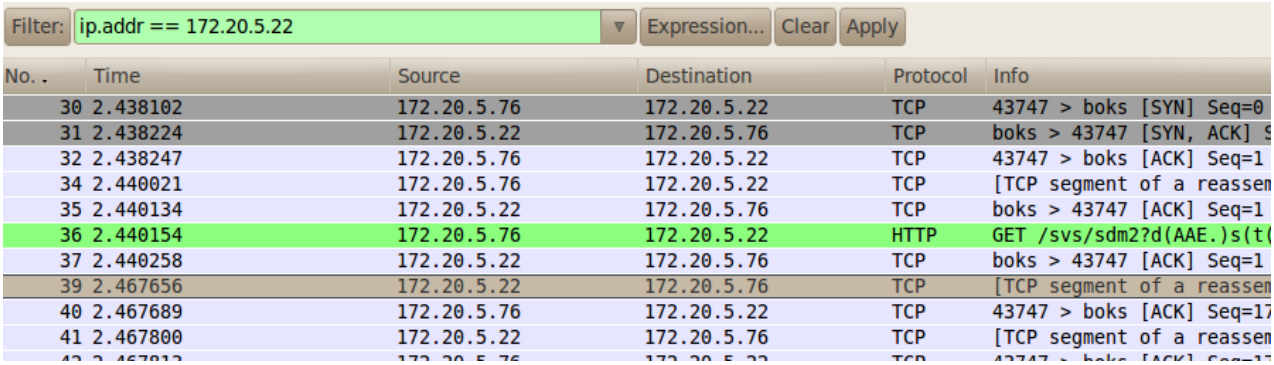

Figura 24: Amostra de Captura da Consumo de Rota Sem Segurança

Novamente, podemos observar que o número de pacotes transmitidos entre o usuário de vídeo e o nó de serviço mais próximo a ele se manteve igual nos dois cenários

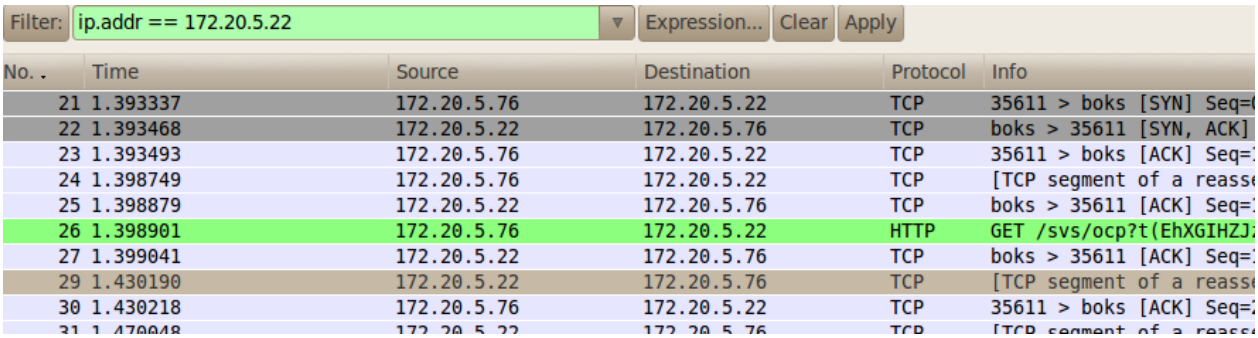

Figura 25: Amostra de Captura da Consumo de Rota Com Segurança pelo OCP

Como nesse ponto, entre o usuário e o nó de serviço, podemos ver o tamanho total que a mensagem de consumo da rota tem, pode-se inferir que o mesmo ocorra por todas as redes desses dois casos, ou seja, com segurança e sem segurança. O mesmo raciocínio é válido analogamente para a montagem da rota.

Para comprovar isso, a figura 26 mostra os campos de informações comuns, ou seja, *payload* (carga útil), que necessitam ser transmitidos por toda a infraestrutura das redes em ambos cenários. Os campos com fundo cinza são exclusivos do OCP, já os demais correspondem aos que contêm a carga útil, os valores de tamanho são apresentados em *bytes*.

header	**code**	**client address**	**timestamp**	**last encrypted**	
	4	8	8	0,125	
paths	***path***	**domainId**	**keyId**	**cmac**	
		10	10	40	
		node	**address**	**serviceId**	
			28	14	
services	***service***	**domainId**	**serviceId**	**parentId**	**specId**
		4	4	4	4
		service body			
		64			

Total de Overhead	78,125	*bytes*
Total de Mensagem	202,125	*bytes*
Diferença Percentual	38,65%	

Figura 26: Informações acrescentadas pelo OCP e o percentual de acréscimo

A quantidade de dados excedentes (sobrecarga) para a comunicação segura do OCP tem seu tamanho total dependente do número de *paths* e de serviços.

A figura 12 da seção 4.1.1 contém cinco *paths* e dois serviços. E novamente na figura 26 é possível ver o percentual acrescido à mensagem dos campos exclusivos do OCP.

Assim, podemos constatar que o OCP acrescentou 38,65 % de sobrecarga de uma mensagem que contém por volta de 202 *bytes*, isto é, desse total, em torno de apenas 78 *bytes* representam garantias de sigilo e autenticidade da mensagem transmitida no cenário de três domínios apresentado para validação.

5.2.2 Nível de Segurança

Como meio de garantir um bom nível de segurança funcional o vetor de inicialização precisa ser único.

Para isso, o protocolo OCP faz com que o valor do contador para cada mensagem seja inicializado com um *timestamp* de precisão de nanosegundos. Assim, para um intervalo de *t* milisegundos entre as chegadas de quaisquer duas mensagens, o número de valores do contador disponíveis para cada uma delas é de *t* nanosegundos.

Contudo o fator mais limitador é o tempo de validade de chave para mensagens devido ao uso do algoritmo CMAC na geração do *MAC* da mensagem, conforme visto no capítulo 3.

Com o tamanho de 8 *bytes* para o *MAC* e a criptografia sendo feita com o AES-128, o *MAC* resultante tem descartado metade do *MAC* gerado.

Dessa forma, o número de mensagens possíveis sem que haja colisão é de 2^21, o que torna possível ainda manter uma margem segura na manutenção da integridade da mensagem, já que, dentro desse limite, a probabilidade de colisão é de que ocorra colisão em menos de 1 em um milhão. Além disso, no OCP há mecanismos de atualização periódica das chaves o que dificulta mais ainda a ocorrência desse evento.

Outros pontos a serem considerados são as abordagens para lidar com as vulnerabilidades das redes de distribuição. A seguir mostramos as vulnerabilidades em redes P2P para *multicast* sobreposto para distribuição de vídeo apresentadas no trabalho de Hyeokchan Kwon (KWON; KIM; NAH, 2009) e as soluções com que o protocolo OCP trata a cada problema.

Os problemas apresentados são:

1. **Rendezvous-Point Malicioso**: caso um *peer* malicioso faça o papel de um *Rendezvous Point*, tornará o serviço de *multicast* indisponível;
2. **Partições da árvore sobreposta**: um conjunto de nós maliciosos podem tornar malicioso o *Rendezvous Point*. E eles podem reencaminhar a mensagem *"join"* de um novo nó para um *Rendezvous Point* malicioso;
3. **Desorientar o roteamento *multicast***: um membro de grupo malicioso poderia reencaminhar dados de *multicast* para um nó incorreto ou descartá-lo;
4. **Vazamento de Informação**: alguns membros da árvore de *multicast* não têm direito de visualizar a árvore de *multicast*. Estes membros poderiam utilizar *sniffers*, armazenar ou redistribuir dados de *multicast* ilegalmente;
5. **Ataque DoS**: um conjunto de nós maliciosos encaminham dados de *multicast* para um nó alvo específico cooperativamente;
6. **Falsificação de mensagem de roteamento**: um nó intermediário pode forjar e entregar mensagens de roteamento. Especificamente, o sistema de roteamento pode entrar em colapso em caso de falsificação de mensagens de controle.

Muitas dessas vulnerabilidades já foram apresentadas e complementadas na seção 3.3.1. Agora veremos como a proposta apresentada nesse trabalho trata cada uma dessas vulnerabilides, bem como as apresentadas na seção 3.3.1:

- Na arquitetura OVERSEA, o *Rendezvous Point* corresponde ao Gerente, pois é responsável por fazer com que os nós de serviço sejam localizados e a rota de entrega do vídeo aos usuários seja criada (UCHôA et al., 2007). Na comunicação

OCP, durante o registro dos elementos da rede, é feita uma certificação mútua, o que dessa forma, inviabilizaria a inserção de um Gerente falso à infraestrutura de rede. Assim, é possível tratar das vulnerabilidades 1 e 2;

- Para tratar a vulnerabilidade 3, a comunicação via OCP impede o ingresso de um nó não autenticado, e sem tal autenticação os repasses feitos a outros nós são descartados. Se houver alguma alteração na mensagem dentro do sistema, ela não consegue ser repassada sem a autenticação;
- Para a vulnerabilidade 4, com a encriptação da mensagem, não há possibilidade de algum atacante acessá-las, a não ser que consiga quebra a segurança proporcionada pela criptografia ou então entrar no sistema e possuir as chaves. Entretanto, isso pode ser evitado com segurança no próprio hospedeiro, independentemente do sistema;
- Os ataques de negação de serviço, mostrados na vulnerabilidade 5, quando específicos ao sistema, são bloqueados, já que para que as mensagens sejam aceitas pelos nós de serviço, é preciso que essas mensagens provenham de nós autenticados. Seria preciso ao atacante quebrar a segurança de certificados digitais;
- A vulnerabilidade 6 é tratada por meio da autenticação da mensagens de roteamento pelo campo "*cmac*", o que novamente implica na quebra da segurança obtida do resumo criptográfico e das chaves utilizadas na autenticação da mensagem. Ademais, há o campo "*timestamp*" e a validação do campo com a informação de endereço do usuário que são usados para verificar o tempo de expiração e a procedência da mensagem como válidos ou não.

Portanto, como as vulnerabilidades apresentadas na seção 3.3.1, o protocolo OCP consegue garantir a segurança na comunicação dos elementos da rede basicamente pelas garantias de sigilo da mensagem e as autenticações dos nós de serviço, do cliente ou usuário e da própria mensagem.

6 Considerações Finais e Conclusões

As aplicações de distribuição de vídeo estão em evidência atualmente, o que é comprovado pela crescente demanda na Internet. Entretanto, é preciso garantir que os usuários dessas aplicações terão alta disponibilidade em conjunto à eficiência dos serviços disponibilizados pelas redes de distribuição. Para obter essas garantias, a comunicação dos elementos das redes de distribuição precisa ser segura, mesmo em cenários complexos como os de interdomínio.

Por meio desse trabalho foi possível comprovar que a segurança e eficiência da comunicação dos elementos de rede é alcançável, fazendo uso de diversas técnicas de segurança em uma infraestrutura de distribuição de conteúdo que permite o uso de diversos serviços, como o *multicast* sobreposto, auxiliando na distribuição otimizada dos conteúdos.

Com esse contexto, desenvolvemos o protocolo OCP (*Overlay Communication Protocol*), que é um protocolo de comunicação genérico, pois atende diversos serviços da rede de distribuição, e seguro, pois toda a comunicação é sigilosa e a integridade das mensagens é garantida.

Para validar o protocolo, realizamos um conjunto de testes com medições que foram utilizadas para comparações de desempenho da solução com outro cenário sem segurança, além de análises comparativas que permitiram averiguar o potencial de segurança que o trabalho abordado propõe.

Os resultados da validação mostraram que o uso do OCP é pertinente, tanto em termos de latência, quanto em termos de número de mensagens geradas, além de garantir um fluxo único de comunicação entre os elementos, tanto na montagem como no consumo da mensagem de rota, bem como atendendo aos requisitos de segurança apresentados.

6.1 Trabalhos Futuros

Por meio de pesquisas realizadas para o desenvolvimento deste trabalho, verificamos algumas evoluções possíveis, sendo essas:

- Estender o trabalho para integração interdomínio de rede P2P às redes sobrepostas, cujo desafio é a gerência de chaves e domínios em redes P2P;
- Criptografia do fluxo de vídeo, com DRM (*(Digital Rights Management - Gestão de Direitos Digitais)*), compartilhamento de chaves criptográficas para cada vídeo, que no trato em interdomínio, haveria chaves compartilhadas entre os domínios, evitando a reencriptação pela mudança de domínio.

Referências

AKAMAI. **Akamai - Customer List**. 2007. Acessado em: Agosto de 2010. Disponível em: <http://www.akamai.com/html/customers/customer_list.html>.

AKAMAI. **[Whitepaper] Akamai Security Capabilities: Protecting Your Online Channels and Web Applications Table of Contents**. 2009.

ALBANESE, J.; SONNENREICH, W. **Network Security Illustrated**. [S.l.]: McGraw-Hill, 2004. ISSN 1618-727X.

AMUTHARAJ, J.; RADHAKRISHNAN, S. Dominating Set Theory Based Semantic Overlay Networks for Efficient Content Distribution. **2007 International Conference on Signal Processing, Communications and Networking**, Ieee, p. 228–232, February 2007. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper-.htm?arnumber=4156618>.

ANDREEV, K. et al. Designing overlay multicast networks for streaming. **Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures - SPAA '03**, ACM Press, New York, New York, USA, p. 149, 2003. Disponível em: <http://portal.acm.org/citation.cfm?doid=777412.777437>.

BUYYA, R.; PATHAN, M.; VAKALI, A. **Content Delivery Networks**. [S.l.]: Springer, 2008. ISBN 978-3-540-77886-8.

CIDON, I.; UNGER, O. Optimal content location in ip multicast based overlay networks. **23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings.**, Ieee, p. 916–921, 2003. Disponível em: <http://ieeexplore.ieee.org-/lpdocs/epic03/wrapper.htm?arnumber=1203668>.

CLARK, D. et al. Overlay Networks and the Future of the Internet. **Communications and Strategies**, v. 63, n. 3, p. 1–21, 2006. Disponível em: <http://www.citeulike.org/group-/1598/article/1058133>.

COLE, E.; KRUTZ, R.; CONLEY, J. **Network security bible**. [S.l.]: Wiley-India, 2005.

DEFRAWY, K. E.; GJOKA, M.; MARKOPOULOU, A. Bottorrent: Misusing bittorrent to launch ddos attacks. **In SRUTI07 Proceedings of the 3rd USENIX workshop on Steps to reducing unwanted traffic on the internet**, 2007.

DILLEY, J. et al. Globally distributed content delivery. **IEEE Internet Computing**, v. 6, p. 50–58, 2002.

DOVAL, D. Overlay networks a scalable alternative for p2p. **IEEE Internet Computing**, v. 7, n. 4, p. 73–82, julho 2003.

DWORKIN, M. C o m p u t e r S e c u r i t y. **Nist Special Publication**, 2005.

FAHMY, S.; KWON, M. Characterizing Overlay Multicast Networks and Their Costs. **IEEE/ACM Transactions on Networking**, v. 15, n. 2, p. 373–386, April 2007. ISSN 1063-6692. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper-.htm?arnumber=4154753>.

GUERBER, C. R.; FONSECA, M. Mars : Um Mecanismo de Defesa DoS Reativo para Redes Virtuais. **SBRC09**, p. 71–83, 2009.

GUTTMAN, B. **An introduction to computer security: The NIST Handbook**. [S.l.]: DIANE Publishing, 1995. ISBN 0788128302.

KEROMYTIS, A.; MISRA, V.; RUBENSTEIN, D. SOS: Secure overlay services. **ACM SIGCOMM Computer Communication Review**, ACM, v. 32, n. 4, p. 72, 2002. Disponível em: <http://portal.acm.org/citation.cfm?id=633032>.

KOPP, S. et al. A New Approach for Video Adaptation through Overlay Services Networks. **LatinCom**, 2010.

KURIAN, J.; SARAC, K. **A security framework for service overlay networks: Access control**. IEEE, 2008. 412–419 p. ISBN 978-1-4244-2391-0. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4769117>.

KUROSE, J.; ROSS, K. **Computer networks: A top down approach featuring the internet**. [S.l.]: Pearson Addison Wesley, 2005. ISBN 8177588788.

KWON, H.; KIM, S.; NAH, J. Secure overlay multicast infrastructure for P2P-based IPTV service. **11th International Conference on Advanced**, v. 3, p. 2041–2043, 2009. Disponível em: <http://ieeexplore.ieee.org/xpls/absine all-.jsp?arnumber=4809482&tag=1>.

LEE, H.; KIM, J. A service availability-aware construction of profitable service overlay network. In: **Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on**. IEEE, 2009. v. 2, p. 1252–1256.

ISSN 1738-9445. Disponível em: <http://ieeexplore.ieee.org/xpls/absackslash_all-.jsp?arnumber=4809641>.

LI, Z.; MOHAPATRA, P. QRON: QoS-Aware Routing in Overlay Networks. **IEEE Journal on Selected Areas in Communications**, v. 22, n. 1, p. 29–40, janeiro 2004. ISSN 0733-8716. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper-.htm?arnumber=1258113>.

LIU, K.; LUI, J. On service replication strategy for service overlay networks. **2004 IEEE/IFIP Network Operations and Management Symposium (IEEE Cat. No.04CH37507)**, Ieee, p. 643–656, 2004. Disponível em: <http://ieeexplore.ieee.org-/lpdocs/epic03/wrapper.htm?arnumber=1317752>.

MENEZES, A.; Van Oorschot, P.; VANSTONE, S. **Handbook of applied cryptography**. [S.l.]: CRC, 1997. ISBN 0849385237.

MIRKOVIC, J. et al. **Internet Denial of Service: Attack and Defense Mechanisms (Radia Perlman Computer Networking and Security)**. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004. ISBN 0131475738.

PALLIS, G.; VAKALI, A. Insight and perspectives for content delivery networks. In: **Communications of the ACM**. Citeseer, 2006. v. 49, n. 1, p. 418. ISBN 3540778861. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.74.4021>.

PEREZ, G. M. et al. Secure overlay networks for federated service provision and management. **Computers & Electrical Engineering**, v. 34, n. 3, p. 173–191, maio 2008. ISSN 00457906. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii-/S0045790607000420>.

PIERRE, G.; STEEN, M. van. Globule: a collaborative content delivery network. **IEEE Communications Magazine**, v. 44, n. 8, p. 127–133, ago. 2006. `http://www.globule.org/publi/GCCDN_commag2006.html`.

POMPILI, D.; SCOGLIO, C.; LOPEZ, L. Multicast algorithms in service overlay networks. **Computer Communications**, v. 31, n. 3, p. 489–505, February 2008. ISSN 01403664. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0140366407003040>.

RIPEANU, M. **Peer-to-Peer Architecture Case Study: Gnutella Network**. 2001. Disponível em: <http://citeseer.ist.psu.edu/ripeanu01peertopeer.html>.

SIDIROGLOU, S.; STAVROU, A.; KEROMYTIS, A. D. Mediated overlay services (MOSES): Network security as a composable service. **2007 IEEE Sarnoff Symposium**, Ieee, p. 1–7, abr. 2007. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03-/wrapper.htm?arnumber=4567338>.

SONG, J. et al. **The AES-CMAC algorithm**. 2005. 1–21 p. Disponível em: <http://www-.ietf.org/rfc/rfc4493.txt>.

STALLINGS, W. **Cryptography and network security: principles and practice**. 4th editio. ed. [S.l.]: Prentice Hall, 2005. ISBN 0136097049.

STINSON, D. **Cryptography: theory and practice**. First. [S.l.]: CRC press, 1995. ISBN 1584885084.

TANENBAUM, A. s. **Redes de Computadores**. 4. ed. [S.l.]: Campus, 2003.

TRIUKOSE, S.; AL-QUDAH, Z.; RABINOVICH, M. Content delivery networks: protection or threat? **Computer Security–ESORICS 2009**, Springer, p. 371–389, 2010. Disponível em: <http://www.springerlink.com/index/r052153707478k22.pdf>.

UCHôA, D. C. et al. OVERSEA: towards a scalable and effective architecture for overlay networks. **XIII Webmedia**, 2007.

WALTERS, A.; ZAGE, D.; ROTARU, C. N. A Framework for Mitigating Attacks Against Measurement-Based Adaptation Mechanisms in Unstructured Multicast Overlay Networks. **IEEE/ACM Transactions on Networking**, v. 16, n. 6, p. 1434–1446, dez. 2008. ISSN 1063-6692. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03-/wrapper.htm?arnumber=4460573>.

YIU, W.-P. K.; CHAN, S.-H. G. Offering data confidentiality for multimedia overlay multicast. **ACM Transactions on Multimedia Computing, Communications, and Applications**, v. 5, n. 2, p. 1–23, November 2008. ISSN 15516857. Disponível em: <http://portal.acm.org/citation.cfm?doid=1413862.1413866>.

ZHU, S. et al. Efficient security mechanisms for overlay multicast based content delivery. **Computer Communications**, v. 30, n. 4, p. 793–806, 2007. ISSN 01403664. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0140366406003793>.

A Detalhamento dos Resultados dos Testes

Os valores utilizados para a comparação das estratégias estão descritos nas tabelas a seguir. Estes valores foram calculados a partir dos valores obtidos com os testes utilizando o protótipo implementado.

Tabela 7: Valores de Atraso Medidos (em seg.) Para Montagem da Rota (Sem Segurança)

Nó	Iteração 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5	Iter. 6	Iter. 7	Iter. 8	Iter. 9	Iter. 10
1	0,005118	0,008346	0,004935	0,005728	0,005090	0,004776	0,004597	0,005152	0,006910	0,005090
2	0,005181	0,004924	0,005259	0,006135	0,007252	0,005292	0,004735	0,004946	0,007491	0,006950
3	0,027409	0,026493	0,026150	0,025779	0,026582	0,029510	0,030535	0,037318	0,026730	0,026917
4	0,029975	0,029384	0,029543	0,029584	0,029632	0,029473	0,029330	0,028372	0,028808	0,028103
5	0,024631	0,031801	0,024551	0,024538	0,025074	0,024073	0,024127	0,023547	0,022763	0,025858
6	0,116002	0,057630	0,055714	0,055746	0,061664	0,055654	0,056543	0,066388	0,071458	0,050970
7	0,044723	0,040113	0,039376	0,040204	0,040226	0,042865	0,046743	0,051678	0,042164	0,039142

Tabela 8: Valores de Atraso Medidos (em seg.) Para Montagem da Rota (Com Segurança OCP)

Nó	Iteração 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5	Iter. 6	Iter. 7	Iter. 8	Iter. 9	Iter. 10
1	0,009141	0,007853	0,007793	0,008573	0,007645	0,007744	0,007405	0,007721	0,014314	0,007280
2	0,008591	0,008959	0,008385	0,008868	0,008540	0,018030	0,008578	0,008162	0,008547	0,009059
3	0,049973	0,056844	0,056614	0,056806	0,058825	0,086279	0,056995	0,062324	0,050096	0,050218
4	0,061629	0,070862	0,064633	0,053941	0,060476	0,055206	0,055859	0,053999	0,060823	0,044429
5	0,058649	0,061512	0,057455	0,057497	0,064981	0,054511	0,053913	0,054103	0,053645	0,052385
6	0,151566	0,122627	0,131163	0,119870	0,113170	0,100729	0,150225	0,154107	0,101616	0,096051
7	0,146457	0,122967	0,155230	0,142995	0,128154	0,114877	0,135671	0,129066	0,131833	0,141184

Tabela 9: Valores de Atraso Medidos (em seg.) Para Consumo da Rota (Sem Segurança)

Nó	Iteração 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5	Iter. 6	Iter. 7	Iter. 8	Iter. 9	Iter. 10
1	0,029554	0,028731	0,027194	0,031175	0,038147	0,027344	0,031566	0,029311	0,031207	0,032337
2	0,060147	0,050673	0,059468	0,054921	0,058653	0,061849	0,059404	0,051400	0,071700	0,061381
3	0,078301	0,073209	0,077202	0,069177	0,072275	0,071557	0,063169	0,074512	0,072746	0,072690
4	0,160483	0,124720	0,122304	0,127119	0,105319	0,106521	0,099608	0,093765	0,116089	0,115139
5	0,103110	0,108605	0,092998	0,101481	0,095642	0,102085	0,099466	0,090906	0,102803	0,095206
6	0,203996	0,225360	0,163933	0,180075	0,133741	0,128861	0,133021	0,161752	0,114381	0,117128
7	0,214753	0,110561	0,145333	0,118823	0,124733	0,124090	0,119166	0,120654	0,117249	0,129678

Tabela 10: Valores de Atraso Medidos (em seg.) Para Consumo da Rota (Com Segurança OCP)

Nó	Iteração 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5	Iter. 6	Iter. 7	Iter. 8	Iter. 9	Iter. 10
1	0,032157	0,036853	0,032812	0,037359	0,035585	0,039394	0,032579	0,027795	0,034463	0,038981
2	0,064060	0,075690	0,072528	0,087632	0,074777	0,060170	0,065731	0,073527	0,063744	0,073539
3	0,175069	0,136443	0,139187	0,094059	0,135523	0,106273	0,090272	0,098353	0,087064	0,092293
4	0,165265	0,223188	0,162839	0,162427	0,140761	0,131073	0,131644	0,133891	0,161851	0,214061
5	0,158802	0,175664	0,165154	0,170858	0,153598	0,156818	0,166596	0,174322	0,191291	0,169539
6	0,267552	0,250077	0,207502	0,239965	0,222909	0,225402	0,233020	0,204093	0,206265	0,254122
7	0,240934	0,240901	0,263667	0,226706	0,237310	0,244497	0,267513	0,222174	0,246978	0,205992