Renata Coelho Borges

Um estudo para o provimento de segurança nas transmissões de Voz sobre redes IP

São José – SC

março / 2009

Renata Coelho Borges

Um estudo para o provimento de segurança nas transmissões de Voz sobre redes IP

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:
Prof. Emerson Ribeiro de Mello, Dr.

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC

março / 2009

Monografia sob o título *"Um estudo para o provimento de segurança nas transmissões de Voz sobre redes IP"*, defendida por Renata Coelho Borges e aprovada em 09 de março de 2009, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Emerson Ribeiro de Mello, Dr.
Orientador

Prof. Evandro Cantú, Dr.
IFSC

Prof. Odilson Tadeu Valle, M. Eng.
IFSC

A auto-satisfação é inimiga do estudo.
Se queremos realmente aprender alguma coisa, devemos começar por libertar-nos disso.
Em relação a nós próprios devemos ser 'insaciáveis na aprendizagem'
e em relação aos outros, 'insaciáveis no ensino'."
Mao Tse Tung

Agradecimentos

Agradeço primeiramente a Deus, pois sem Ele nada seria possível.

A toda minha família, em especial meus pais, Solange e Adair, e minha irmã, Amanda, pelo esforço, dedicação e compreensão em todos os momentos dessa e de outras caminhadas.

Ao meu namorado, Leandro, por sua paciência e todo o seu carinho, e pelo apoio em todos os momentos dessa jornada.

Aos amigos Cesar, Deise e Cleiber pela amizade, companheirismo e apoio.

Ao Prof. Dr. Emerson, pela orientação e confiança depositadas para a realização deste trabalho.

A Instituição IFSC e seus docentes, e à estrutura proporcionada onde fora possível a conquista desta vitória.

A todas as pessoas que de direta e indiretamente contribuíram com este trabalho.

Resumo

O uso da tecnologia Voz sobre IP (VoIP) vem se tornando cada vez mais comum ao longo do tempo. No entanto, a implantação dessa tecnologia associada à necessidade de garantia de segurança na comunicação ainda é um desafio. Apesar de haver algumas discussões de como prover segurança em uma conferência VoIP ainda não há um consenso de como prover um canal seguro para a sinalização e para o transporte da mídia. Este trabalho tem como objetivo o estudo dos métodos existentes para o provimento de segurança em um sistema baseado na transmissão de voz sobre redes IP. O foco do trabalho se manterá sobre os protocolos *Transport Layer Security* (TLS), *Secure Real Time Protocol* (SRTP) e *Inter Asterisk eXchange* (IAX). Para a realização dos experimentos, optou-se pelo PABX IP Asterisk. Com este estudo pretende-se apresentar formas para prover um serviço de voz sobre IP seguro.

Abstract

Voice over IP (VoIP) has recently been receiving more attention. However, the implementation of VoIP environment that cover all security properties is still a challenge. There are some discussions about how to provide a security VoIP communication (signaling and voice), however there is not a consensus yet. This work presents a study over security mechanisms that can be used with VoIP communications. The focus of this work consist to verify how difficult is and which security properties could be cover when using Transport Layer Security (TLS), Secure Real Time Protocol (SRTP) or Inter Asterisk eXchange (IAX). To achieve this goal, some experiments were performed with a PBX IP, Asterisk. This study aims to present ways to provide a security Voice over IP communication.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

A consolidação da rede mundial de computadores, a Internet, marcou na década de 90 o início da convergência de tecnologias tradicionais existentes com as redes IP, tornando o acesso às informações cada vez mais rápido. A transmissão da Voz sobre redes IP (*Voice over Internet Protocol* - VoIP) é uma das representantes dessa transformação e pode ser definida como uma aplicação telefônica que utiliza o protocolo de rede IP para transmissão de voz. VoIP é considerado como uma inovação importante e cada vez mais se torna uma alternativa à rede de telefonia convencional.

Os motivos para que haja a migração para a tecnologia de Voz sobre IP são vários, como redução de custos na implantação e manutenção, isso porque utiliza apenas a rede de dados IP que já se encontra disponível, a redução de custos em ligações de longa distância (DDD e DDI) e a possibilidade de introdução de novas funcionalidades, como conferências, chamada em espera.

Diferentemente da rede convencional de telefonia, VoIP é baseado na comutação de pacotes através da rede IP, uma rede descentralizada e que apresenta diversas vulnerabilidades. Esse fato acarreta na inexistência de um controle adequado sobre o canal que irá trafegar o fluxo de mídia e sinalização. Isso faz que, quando comparada à sua antecessora, VoIP seja visto com restrições em cenários onde há a necessidade de segurança.

Apesar da existência de alguns padrões que definem como fazer a proteção de uma chamada VoIP, eles ainda são muito recentes e apresentam algumas dificuldades. Como esta tecnologia faz uso de diversas recomendações e protocolos, qualquer vulnerabilidade em um deles pode tornar frágil todo o sistema. Por isso, ao se fazer a adoção do VoIP é necessário que se considere as várias implicações na área de segurança.

É importante salientar que apesar de um sistema baseado em Voz sobre IP estar sujeito a uma grande quantidade de riscos, os quais serão vistos no decorrer deste trabalho, estes não devem ser considerados como fatores limitantes para o uso desta tecnologia. O desafio é utilizar mecanismos já existentes para prover segurança e atingir um nível aceitável de risco, no entanto,

sem comprometer sua usabilidade.

O presente trabalho tem como objetivo analisar a eficácia da implementação de protocolos de segurança em um ambiente VoIP. Para isso serão feitos testes em alguns cenários utilizando protocolos como *Secure Real Time Protocol* (SRTP), *Transport Layer Security* (TLS) e *Inter Asterisk eXchange* (IAX), e comparando-os com chamadas sem nenhum tipo de mecanismo de proteção, através da captura de pacotes na rede. Os experimentos serão realizados com o PBX IP Asterisk, que possui *software* aberto, flexível, tem tido grande destaque na área de tecnologia IP e oferece todas as funcionalidades de um PBX convencional.

1.1 Motivação

A tecnologia VoIP oferece diversas vantagens se comparada à rede de telefonia convencional, como economia nas ligações e infra-estrutura barata quando se possui a existência da rede IP em funcionamento. Apesar de todos os benefícios, VoIP ainda apresenta vulnerabilidades referentes à segurança, o que possibilita, por exemplo, a escuta indevida de uma chamada.

Esses foram os fatores que motivaram a realização desse trabalho, que consiste em um estudo sobre os métodos para se prover segurança à infra-estrutura utilizada na trasmissão de Voz sobre IP, fazendo uso do PBX IP Asterisk.

1.2 Organização do texto

- **Capítulo 2 - Fundamentação Teórica:** apresenta uma breve introdução de Voz sobre redes IP, descrevendo os protocolos utilizados nesse sistema, e o PBX IP Asterisk;
- **Capítulo 3 - Segurança no Ambiente VoIP:** nesse capítulo serão abordadas as questões referentes à segurança no ambiente VoIP, definindo os conceitos básicos de segurança e criptografia. Também serão descritos os mecanismos de segurança existentes, como os protocolos SRTP e TLS;
- **Capítulo 4 - Ambiente de Testes:** descreve as ferramentas utilizadas, o processo de implementação de segurança no ambiente VoIP e análise dos resultados obtidos;
- **Capítulo 5 - Conclusão:** apresenta uma conclusão sobre o estudo realizado e apresenta algumas recomendações para o bom desenvolvimente de trabalhos futuros.

2 *Fundamentação Teórica*

Neste capítulo serão apresentados os conceitos de algumas tecnologias e serviços necessários para a compreensão deste trabalho, como a tecnologia VoIP, que permite a transmissão de voz através de redes IP, os protocolos envolvidos, e o Asterisk, um software de PBX que utiliza a tecnologia voz sobre IP.

2.1 Voz sobre IP

Voz sobre IP (*Voice over Internet Protocol* - VoIP) é uma tecnologia que permite o tráfego de voz em redes que utilizam o protocolo IP (*Internet Protocol*). A idéia por trás do VoIP tem sido discutida desde o início dos anos 1970, mas a tecnologia só se tornou popular na década de 90, quando a empresa *Vocaltec Communication* lançou um *software* chamado *Internet Phone*. No entanto a qualidade da comunicação através do VoIP era precária e não podia ser comparada à telefonia convencional (COLCHER, 2005).

Na sequência do rápido crescimento do mercado de Internet, especialmente da Web, durante o início dos anos 1990 e acompanhando o investimento em infra-estrutura de rede IP pelas empresas, VoIP finalmente torna-se uma alternativa viável para o envio da voz sobre redes IP.

Voice over Internet Protocol (COLCHER, 2005) foi desenvolvido com o intuito de poder realizar chamadas telefônicas através da internet com alta qualidade e baixo custo. Para que a transmissão de voz pela rede IP aconteça, a voz é submetida a um processo de codificação e decodificação, que são responsáveis pela conversão do sinal de voz analógico em sinal digital. Já digitalizada, a voz é distribuída em pacotes IP que possuem informações como o endereço de origem e destino em seu cabeçalho. Os pacotes são transmitidos pela rede IP através de protocolos de transporte, tais como TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) e UDP (*User Datagram Protocol*), passando pelos nós da rede. Ou seja, em cada nó ele é recebido e então, com base no endereço contido no pacote, é determinado o caminho da rota a ser seguido. Ao chegarem ao destinatário esses pacotes são reordenados e convertidos

para a forma analógica.

A transmissão de voz em redes baseadas na comutação de pacotes exige algumas propriedades básicas como baixo atraso origem-destino, baixa variação de atraso (*jitter*) e taxas de perdas de pacotes e de erros de bits pequenas. Mas, como o desenvolvimento inicial da Internet e do protocolo IP não visava a transmissão de mídia em tempo real, a rede caracteriza-se pela perda e atrasos de pacotes o que pode interromper o fluxo de dados por alguns instantes, causando dificuldade na comunicação (BALBINOT et al., 2003).

Uma chamada telefônica realizada através da rede IP pode se tornar inviável caso haja, por exemplo, atrasos na transmissão de pacotes (WALSH; KUHM, 2005). Além disso, a voz é dividida em pequenos pacotes para poder ser enviada pela rede, o que pode causar atrasos não uniformes desses pacotes (*jitter*) fazendo com que eles cheguem fora da seqüência correta. A reordenação dos pacotes é feita pela aplicação, já que o transporte dos dados é geralmente baseado em UDP (*User Datagram Protocol*). Problemas com o *jitter* também podem tornar a comunicação inviável pois adiciona atraso na comunicação. Alguns protocolos implementam *buffers* específicos para o tratamento do *jitter*. Esses *buffers* armazenam uma certa quantidade de pacotes para tentar solucionar o problema da variação no atraso. Caso alguns pacotes cheguem muito atrasados, eles serão descartados. Outro fator que pode inviabilizar a chamada é a perda de pacotes. Apenas 5% de perda de pacotes já pode comprometer uma chamada VoIP.

VoIP pode ser implementado de várias formas, como mostra a figura 2.1. O primeiro cenário representa uma chamada de voz feita entre telefones ligados à Rede de Telefonia Pública Comutada (RTPC). Esta ligação pode ser transmitida tradicionalmente, sobre linhas analógicas, ou pode ser convertida em IP, e depois volta para a rede convencional. O cenário 2 retrata uma chamada de voz feita a partir de um telefone da RTPC para uma aplicação de voz em um computador pessoal. Finalmente, o terceiro cenário ilustra uma chamada de voz iniciada a partir do computador através do seu servidor VoIP, atuando em uma RTPC, que é encaminhado através da Internet para um telefone ligado a uma organização, normalmente um *Private Branch Exchange* (PBX).

O que realmente impulsiona e faz crescer cada vez mais o número de pessoas adeptas ao uso da Voz sobre IP são aspectos como relação custo/benefício: reduzir significativamente o custo com telefonia, efetuar chamadas de longas distâncias a custos locais, possibilidade de integração da rede fixa com a rede IP, baixo custo com infra-estrutura considerando que não é necessário criar uma outra estrutura apenas para o tráfego de voz uma vez que o cliente já possua uma infra-estrutura IP existente. Mobilidade, que permite qualquer usuário com acesso à Internet se conectar e fazer ligações com custo local, arquitetura aberta, possibilitando aos usuários

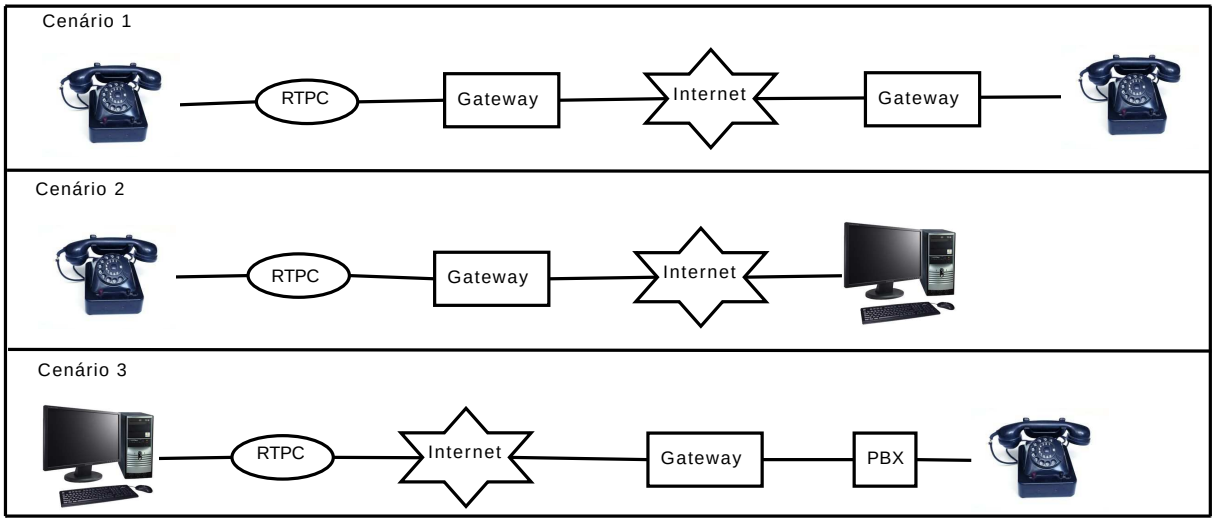

Figura 2.1: Aplicações VoIP.

escolherem seus provedores e não estarem presos a operadoras de telefonia, e principalmente as aplicações, pois o VoIP provê recursos, assim como na telefonia convencional, como chamada em espera, conferência, etc.

Dificuldades para a consolidação do uso da tecnologia VoIP

Pesquisas recentes divulgaram altos índices de crescimento no uso da tecnologia VoIP (TELEGEOGRAPHY, 2005), no entanto, por não contemplar amplamente as propriedades básicas de segurança (capítulo 3) como, por exemplo, a confidencialidade dos dados, a tecnologia para provimento de voz sobre IP ainda não é totalmente aceita (VARSHNEY et al., 2002).

Alguns dos problemas encontrados para a adoção do VoIP serão resolvidos com o tempo, mas outros ainda retardarão a adoção desse novo sistema por muito tempo. As principais deficiências encontradas nos dias de hoje são:

Confiabilidade. ainda falta confiança quanto a disponibilidade do serviço VoIP, quando se precisar dele como, por exemplo, para ligações de emergência. O VoIP depende do fornecimento de energia elétrica diferentemente da telefonia convencional. Utilizando VoIP, a falta de energia elétrica significa que não podemos fazer nem receber ligações;

Qualidade de Serviço. as redes IP atuais, atuam oferecendo um serviço do tipo "melhor esforço"[1] o que reduz a qualidade do serviço oferecido;

[1] O IP oferece um serviço de datagramas não confiável (também chamado de melhor esforço); ou seja, o pacote vem quase sem garantias. O pacote pode chegar desordenado (comparado com outros pacotes enviados entre os mesmos hosts), também podem chegar duplicados, ou podem ser perdidos por inteiro. (WIKIPEDIA, 2009)

Custo. apesar da notável redução de custos das ligações, os produtos VoIP ainda não conseguem apresentar um custo compatível com os oferecidos pelos produtos de telefonia tradicionais;

Investimento. enquanto muitos dos potenciais usuários de VoIP já tem uma conexão razoável de internet, a necessidade de estabelecer serviços de VoIP confiáveis e de qualidade faz com que seja necessária a substuição dos equipamentos (roteadores) já utilizados por modelos que tenham garantia de qualidade integrados. Também pode ser necessária a compra de *no-breaks* para garantir o fornecimento de energia no caso de uma queda.

Protocolos

Os protocolos são necessários e fundamentais para a troca de informações de gerenciamento e controle dos serviços de rede nas transmissões de Voz sobre IP. Esses componentes são oferecidos pelos protocolos de sinalização, como por exemplo, o SIP (*Session Initiation Protocol* - Protocolo de Inicialização de Sessão) e o IAX (*Inter Asterisk eXchange*), que serão abordados no decorrer deste capítulo. Para o transporte serão apresentados os protocolos RTP (*Real Time Transport Protocol*) e o RTCP (*Real Time Transport Control Protocol)*.

2.2 Protocolos para Sinalização e Transporte

2.2.1 H.323

O H.323 é um padrão (ITU-T, 2006) e pertence à série H da família de recomendações *International Telecommunication Union Telecommunication Standardization sector* (ITU-T) H.32x, e trata de "Sistemas Audiovisuais e Multimídia". A recomendação H.323 possui os objetivos de fazer a especificação de sistemas de comunicações multimídia cujas redes são baseadas em pacotes e não fornecem um garantia de Qualidade de Serviço (QoS). Além disso, define que os produtos baseados no padrão H.323 de um fabricante possam interoperar com produtos H.323 de outros fabricantes, através de padrões estabelecidos pela recomendação para codificação e decodificação de fluxos de áudio e vídeo.

O padrão H.323 especifica o uso de áudio, vídeo e dados em comunicações multimídia, no entanto, é obrigatório apenas o suporte à mídia de áudio.

Componentes do H.323

O padrão H.323 especifica quatro tipos de componentes que em conjunto possibilitam a comunicação multimídia (CONSORTIUM, 2001). São eles:

- Terminais - computadores utilizados em uma rede, que provêem comunicação em tempo real. Como especificado na recomendação, todos os terminais devem prover suporte a voz, já o suporte à video e dados é opcional;
- *Gateways* - componentes que têm como função prover a comunicação de terminais H.323 com outros terminais de padrões diferentes (H.310, H.321, H.322). São elementos opcionais em conferências H.323;
- *Gatekeepers* - componentes que atuam como ponto central para todas as chamadas dentro de sua zona[2] e provêem serviços de controle de chamada para registrar participantes. São também responsáveis pelo gerenciamento da largura de banda em conferências H.323;
- *Multipoint Control Units* (MCUs) - suportam conferências entre três ou mais participantes. Sob H.323, um MCU consiste de um *Multipoint Controller* (MC) e zero ou mais *Multipoint Processors* (MP);
  - o MC é responsável por determinar capacidades comuns para processamento de áudio e vídeo da rede, fazendo-o através da manipulação das negociações entre todos os terminais da rede;
  - o MP é responsável por chavear, mesclar e processar os bits de áudio, vídeo e/ou dados.

Padrões que estendem as funcionalidades do padrão H.323

O H.323 pode ser estendido adicionando novos serviços através da utilização de outros padrões definidos pelo ITU-T. São eles (CONSORTIUM, 2001):

- *H.235 Security and Encryption for H-Series (H-323 and other H.245-based) Multimedia Terminals* - essa recomendação introduz um ambiente de segurança para o H.323, provendo serviços de autenticação, integridade e confidencialidade. Usa os mecanismos de criptografia dos protocolos de suporte ao *Internet Protocol Security* (IPSEC) e ao *Transport Layer Security* (TLS);

[2]Zona é o conjunto de todos terminais, gateways e MCUs gerenciados por um único gatekeeper, e deve incluir, pelo menos, um terminal

- Série H.450.x adapta o H.323 à Telefonia IP. Serve para introduzir alguns serviços suplementares ao H.323 que são comuns aos sistemas telefônicos;

A série H.450.x constitui-se de recomendações para: protocolos genéricos (H.450.1), espera de chamadas (H.450.6), transferência de chamadas (H.450.2), desvio de chamadas (H.450.3), retenção de chamadas (H.450.4), retenção e retomada de chamadas (H.450.5), indicação de mensagens em espera (H.450.7), identificação de nomes (H.450.8) e procedimentos de completar chamadas se ocupado (H.450.9), oferta de chamada (H.450.10), intrusão em chamadas (H.450.11) e características de informações comuns adicionais de rede (H.450.12).

2.2.2 Session Initiation Protocol - SIP

O *Session Initiation Protocol* (SIP) é um protocolo localizado na camada de aplicação do modelo TCP/IP responsável por estabelecer, modificar e terminar comunicações multimídia entre dois ou mais usuários (ROSENBERG et al., 2002). Ou seja, o SIP não é um protocolo para transmissão de dados via rede IP, mas para a sinalização necessária ao estabelecimento do canal de comunicação entre as partes envolvidas.

A estrutura das mensagens SIP são em modo texto, baseadas nos protocolos *HyperText Transfer Protocol* (HTTP) e *Simple Mail Transport Protocol* (SMTP). Do HTTP o SIP herda o funcionamento baseado em requisições e respostas, sendo que o estabelecimento da comunicação se dá através do envio de requisições a um servidor SIP responsável por enviar uma ou mais respostas. Do SMTP herda o esquema de endereçamento. O SIP usa o *Uniform Resource Identifier* (URI) que define os endereços de cada usuário na rede IP. A URI é formada por três campos: nome do protocolo a ser usado, nome do recurso e por fim o domínio em que ele está localizado. Um exemplo de SIP URI pode ser visto na figura 2.2, onde o protocolo é sip, o nome do usuário é renata e o domínio sj.cefetsc.edu.br.

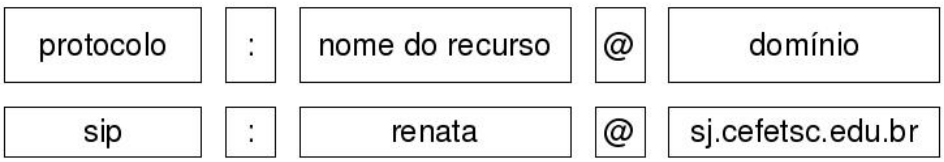

Figura 2.2: Exemplo de URI SIP

O protocolo SIP utilizado em conjunto com outros protocolos do IETF formam uma arquitetura completa de comunicação multimídia. Entre esses protocolos podem ser citados o

Real Time Transport Protocol (RTP) (SCHULZRINNE et al., 2003) para o transporte de dados em tempo real, o *Real-Time Transport Control Protocol* (RTCP) (SCHULZRINNE et al., 2003) que faz o controle dos pacotes na rede, o *Session Description Protocol* (SDP) (ANDREASEN; BAUGHER; WING, 2006) que faz a descrição das características da sessão multimídia, o *Media Gateway Control Protocol* (MEGACO) (BLATHERWICK; BELL; HOLLAND, 2001) responsável, entre outros, por controlar os *gateways* que se comunicam com a rede pública de telefonia.

Como dito anteriormente o SIP é um protocolo para sinalização de sessão, feitas através de requisições enviadas de um cliente para um servidor e as respostas à essas requisições. O estabelecimento de uma comunicação multimídia é feita a partir da troca inicial de mensagens, ou seja, o processo de sinalização necessário para o estabelecimento de uma sessão. Após o início da sessão a sincronização dos pacotes fica sob responsabilidade do protocolo RTP e a definição do conteúdo e das características da sessão é responsabilidade do protocolo SDP, e assim por diante. O SIP somente inicializa, modifica, gerencia e finaliza uma sessão.

O sistema SIP é composto por duas entidades: *User Agents* (Terminais SIP) e *SIP Network Servers* (Servidores SIP).

O *User Agent* (UA) pode ser entendido como qualquer entidade capaz de enviar ou receber requisições em uma rede IP. Os *user agents* podem ser divididos em dois tipos. Quando estão responsáveis por fazer a requisição inicial de uma comunicação eles serão os *User Agents Clients* (UAC). Ao terminal que recebe a requisição dá-se o nome de *User Agent Server* (UAS). Um UA pode atuar tanto como UAC como UAS, porém, apenas de um modo a cada transação, dependendo de quem iniciou o pedido. Qualquer dispositivo capaz de iniciar ou receber requisições para estabelecimento de uma sessão através da rede IP pode ser considerado um UA. Alguns exemplos de *User Agents* são: computadores de mesa, computadores portáteis, telefone celular, etc.

Os servidores SIP (*SIP Network Servers*) têm como principais funções efetuar a resolução de nomes e a localização dos usuários. Quando um UA deseja iniciar uma sessão seu UAC envia uma mensagem de requisição de início de sessão para os UAS de outro UA. Geralmente o solicitante não conhece o endereço IP do destinatário, ele possui apenas o endereço URI (figura 2.2), endereço o qual permite que usuários possam ser localizados em um ambiente SIP. O UAC poderá então, determinar qual servidor SIP será capaz de resolver o SIP URI para um endereço IP. Este servidor pode redirecionar a requisição para outros servidores até que se encontre o endereço IP do destinatário. Os servidores SIP são divididos em três categorias:

- *SIP Registrar Servers* - são servidores responsáveis por processar o pedido dos UAC, e registrar suas informações, figura 2.3. Quando um UA envia uma requisição para registro, estes servidores armazenam algumas informações como o endereço IP, por exemplo, e informações adicionais que permitem determinar sua localização (passo 1). Estas informações armazenadas são enviadas ao servidor solicitante (passo 2). A figura 2.3 mostra Bob registrando sua localização em um *Registrar Server*;

- *SIP Proxy Servers* - são servidores que atuam como intermediários em uma comunicação multimídia, fazendo o redirecionamento de requisições e respostas. O *SIP Proxy Server* recebe uma requisição e determina para qual servidor ela será encaminhada, redirecionando-a como se fosse o requisitante. O *SIP Proxy* pode encaminhar a requisição para qualquer outro servidor, inclusive para um UA. A mensagem SIP, então, pode trafegar por diversos servidores no seu caminho entre um UAC e um UAS. Assim que recebe a resposta, encaminha ao destinatário. O *SIP Proxy Server* também disponibiliza serviços como: autenticação, autorização, roteamento, retransmissões de pedidos e segurança.

- *SIP Redirect Server* - são servidores responsáveis por receber as requisições e informar ao requisitante o endereço do destinatário, ou seja, informa qual caminho a chamada deve tomar, possibilitando assim que o cliente entre diretamente em contato com os *User Agents*. Para que sejam determinadas as rotas, tanto o *SIP Proxy Server* quanto o *SIP Redirect Server*, podem utilizar meios como consulta a banco de dados. Além disto, um servidor proxy também pode duplicar a requisição, enviando cópias destas para os próximos servidores. Isto proporciona que uma requisição de início de chamada tente diferentes rotas e a primeira localização que responder é conectada com o cliente chamador. A figura 2.4 ilustra o funcionamento do Redirect Server. Alice pretende iniciar uma chamada com Bob, utilizando uma aplicação SIP em seu computador. Ao enviar o convite, o UA de Alice tenta localizar Bob através do endereço público que ele possui (passo 1). No domínio empresa.com, existe um SIP *Redirect Server* controlando os convites para início de sessão, que se destinam a este domínio. O *Redirect Server* sabe que Bob pode ser encontrado tanto no endereço: SIP:bob@empresa.com, quanto no endereço: SIP: bob@universidade.com, então o *Redirect Server* informa Alice para que tente localizar Bob, nos endereços conhecidos (passos 2 e 4). Resumidamente, o *Redirect Server*, faz somente o roteamento das requisições e respostas enviando uma mensagem aos clientes com o endereço SIP procurado (passos 3 e 5).

Esses três tipos de servidores SIP possuem distinção exclusivamente lógicas, podendo todos eles estarem instalados em um mesmo hardware.

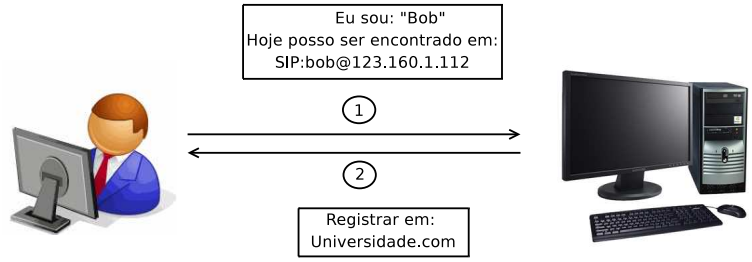

Figura 2.3: Registro da localização em um *Registrar Server*

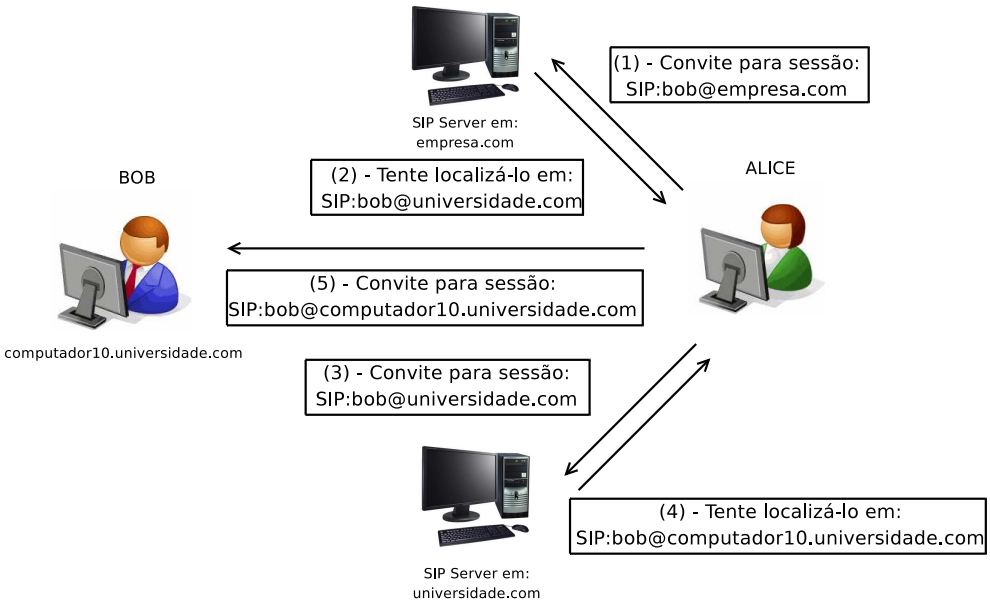

Figura 2.4: Funcionamento do *Redirect Server*

Sinalização SIP

Mensagens SIP podem ser requisições efetuadas pelo cliente ao servidor e respostas enviadas pelo servidor ao cliente (ROSENBERG et al., 2002). Existem seis tipos de mensagens de requisição definidas pela RFC 3261 e seis classes de mensagens de respostas, sendo que as cinco primeiras foram herdadas do protocolo HTTP. Outros tipos de mensagens são definidas em RFC diferentes.

As mensagens de requisição devem ser realizadas especificamente pelo UAS. *INVITE, REGISTER, ACK, BYE, CANCEL* e *OPTIONS*, mostradas na tabela 2.1, são mensagens originais do protocolo SIP. Já as mensagens *REFER, SUBSCRIBE, NOTIFY, MESSAGE, UPDATE* e *PRACK*, mostradas na tabela 2.2, são as mensagens de requisição estendidas definidas por outras RFC.

Mensagem	**Funcionalidade**
INVITE	Convida pessoas para participar de uma sessão
ACK	Confirma o recebimento de uma resposta final para um INVITE
BYE	Solicita o término de uma sessão
CANCEL	Solicita que uma sessão prévia seja cancelada, diferente do BYE
REGISTER	Registra a informação de contato de um indivíduo
OPTIONS	Consulta servidores com respeito a suas capacidades

Tabela 2.1: Mensagens de Requisições do SIP

Mensagem	**RFC**	**Funcionalidade**
INFO	2976	Carrega informações de controle geradas durante a sessão
MESSAGE	3428	Permite a transferência de mensagens instantâneas
NOTIFY	3265	Permite a notificação de eventos específicos
PRACK	3262	Confirma a recepção de uma mensagem de resposta informativa
PUBLISH	3903	Publica o estado de um evento
REFER	3515	Solicita que o receptor faça contato com um terceiro participante
SUBSCRIBE	3265	Permite se subscrever para um estado particular de um recurso
UPDATE	3311	Permite a atualização dos parâmetros de uma sessão

Tabela 2.2: Mensagens de Requisições estendidas do SIP

Um servidor *proxy* consiste em um UA responsável por intermediar as transações SIP entre os pares. Um *proxy* fundamentalmente recebe uma requisição de um terminal iniciador (UAS), localiza o seu destino e direciona para o terminal destino essa mensagem (UAC). Assim que o destino responde a requisição inicial o *proxy* redireciona essa resposta para o terminal de origem, fazendo com que durante todo o processo de sinalização, os terminais nunca interajam diretamente.

Classe	Descrição	Ação
1xx	Informativo	Indica o estado da mensagem antes que seja completada
2xx	Sucesso	A requisição foi recebida com sucesso
3xx	Redirecionamento	O servidor retornou possíveis localidades. O cliente deve reenviar a requisição para outros servidores
4xx	Erro no cliente	A requisição falhou devido a um erro no cliente
5xx	Falha no servidor	A requisição falhou devido a um erro no servidor
6xx	Falha global	A requisição falhou e não deve ser enviada a este ou outros servidores

Tabela 2.3: Classes de respostas do SIP

Na figura 2.5, vê-se um exemplo do estabelecimento de uma sessão SIP simples utilizando um *Proxy Server* entre dois dispositivos SIP. No exemplo ilustrado, assume-se que ambos dispositivos estão conectados em uma rede IP e cada um deles está autenticado no mesmo *Proxy Server*.

Figura 2.5: Estabelecimento de uma sessão SIP simples com um Proxy Server

Neste exemplo, o chamador inicia a troca de mensagens enviando uma mensagem SIP *INVITE* para o usuário 7000. A mensagem de *INVITE*, contém os detalhes do tipo de sessão que será estabelecida, que pode ser tanto uma chamada com voz como uma conferência com vídeo. A estrutura da mensagem *INVITE* pode ser vista na figura 2.6.

Ao receber a requisição *INVITE* do terminal iniciador, o *proxy* irá verificar as credenciais do seu usuário, checando se ele possui ou não permissão para realizar a chamada desejada. Após isso, o *proxy* irá analisar o endereço do terminal destino, e então direcionar a chamada para o

endereço chamado.

```
1  INVITE sip:7000@172.18.21.28 SIP/2.0
2  Via: SIP/2.0/UDP 172.18.21.161:5060;branch=
       z9hG4bK00ccba33c0c3dd118153000c6e0ff7ea;rport
3  From: "PhonerLite" <sip:6000@172.18.21.28>;tag=3720367345
4  To: <sip:7000@172.18.21.28>
5  Call-ID: 00CCBA33-C0C3-DD11-8152-000C6E0FF7EA@172.18.21.161
6  CSeq: 24 INVITE
7  Contact: <sip:6000@172.18.21.161:5060>
8  Content-Type: application/sdp
9  Allow: INVITE, OPTIONS, ACK, BYE, CANCEL, INFO, NOTIFY, MESSAGE, UPDATE
10 Max-Forwards: 70
11 Supported: 100rel, replaces
12 User-Agent: SIPPER for PhonerLite
13 Content-Length: 504
```

Figura 2.6: Mensagem *INVITE*

A linha de início do *INVITE* SIP contém o tipo de pedido enviado pelo cliente SIP, o endereço SIP do usuário destino e a versão SIP utilizada. De acordo com (HERSENT; GURLE; PETIT, 2001) a mensagem INVITE contém os seguintes campos:

- *Via* - possui a versão do SIP, o protocolo da camada rede, o endereço IP do usuário que faz a chamada e a porta utilizada;
- *From* - contém o endereço do originador da chamada e um nome que pode ser mostrado opcionalmente. Deve estar presente tanto nas requisições quanto nas respostas SIP. Nas respostas, o campo 'From' simplesmente é copiado a partir do pedido;
- *To* - indica o destino da chamada, sendo obrigatório em todas as requisições e respostas SIP;
- *Call-ID* - a primeira parte deste campo deve ser um padrão único dentro de cada agente, e a última parte, o nome de domínio ou endereço IP. Um novo "Call-ID" deve ser gerado para cada chamada;
- *Cseq* - este campo guarda um número escolhido aleatoriamente sem sinal que identifica o tipo de pedido que está sendo enviado;
- *Content-Type* - define o protocolo a ser usado pelo SIP para descrever a sessão. Entende-se como descrição da sessão qualquer informação que define os parâmetros fim a fim necessários para a construção do canal multimídia;
- *Content-Length* - contém o número de octetos do corpo da mensagem.

O terminal destino no momento que recebe a requição *INVITE*, deverá responder ao *proxy* o seu estado, informando se está disponível ou não para realizar a conferência solicitada. Essa resposta é feita através da mensagem *180 Ringing*, e indica que a outra parte, no caso o usuário 7000, recebeu o *INVITE*, e está sendo alertado de que alguém quer contatá-lo. O *180 Ringing* é um tipo de mensagem SIP de resposta. Conforme visto na tabela 2.3 as mensagens de resposta são numéricas e classificadas pelo primeiro dígito da sequência. O *180 Ringing*, por exemplo, é classificado como classe de informação, identificado pelo primeiro digito, o número 1. A estrutura da mensagem *Ringing* pode ser vista na figura 2.7

```
1  SIP/2.0 180 Ringing
2  Via: SIP/2.0/UDP 172.18.21.161:5060;branch=
       z9hG4bK00ccba33c0c3dd118154000c6e0ff7ea;received=172.18.21.161;rport=5060
3  From: "PhonerLite" <sip:7000@172.18.21.28>;tag=3720367345
4  To: <sip:6000@172.18.21.28>;tag=as673e3580
5  Call-ID: 00CCBA33-C0C3-DD11-8152-000C6E0FF7EA@172.18.21.161
6  CSeq: 25 INVITE
7  User-Agent: Asterisk PBX SVN-trunk-r81432M
8  Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
9  Supported: replaces
10 Contact: <sip:6000@172.18.21.28>
11 Content-Length: 0
```

Figura 2.7: Mensagem *180 Ringing*

Esta mensagem é composta pela cópia de alguns campos da mensagem *INVITE*, como: Via, To, From, Call-ID e CSeq. O parâmetro *received* é adicionado ao campo Via, que contém o endereço IP do agente chamador, que tipicamente é o mesmo endereço URI no campo Via.

Quando a parte chamada aceita a sessão, atendendo ao telefone, por exemplo, uma resposta de *200 OK* é enviada ao agente chamador, e também significa que o tipo de mídia proposta para a sessão foi aceito. Na mensagem *200 OK*, enviada pelo usuário 7000 no campo *Content-Type* é indicado que será usado o protocolo SDP (seção 2.2.3) utilizado para a troca de informações necessárias ao estabelecimento correto da sessão multimídia.

A mensagem *200 OK* é semelhante a mensagem *180 Ringing*. Possui o mesmo valor no campo *To*, e o *URI* do contato. A capacidade da mídia deve ser mostrada junto à mensagem SDP adicionada na resposta.

A finalização para confirmação e início da sessão, se dá por uma requisição de reconhecimento (*acknowledgment*). Este reconhecimento indica que o originador da chamada recebeu uma resposta. Após esse procedimento de troca de mensagens a sessão é estabelecida usando outro protocolo, como o RTP (seção 2.2.4). A figura 2.8 mostra a estrutura do *acknowledgment* (*ACK*).

```
ACK sip:6000@172.18.21.28 SIP/2.0
Via: SIP/2.0/UDP 172.18.21.161:5060;branch=
    z9hG4bK80e9e637c0c3dd118154000c6e0ff7ea;rport
From: "PhonerLite" <sip:7000@172.18.21.28>;tag=3720367345
To: <sip:6000@172.18.21.28>;tag=as673e3580
Call-ID: 00CCBA33-C0C3-DD11-8152-000C6E0FF7EA@172.18.21.161
CSeq: 25 ACK
Contact: <sip:7000@172.18.21.161:5060>
Max-Forwards: 70
Content-Length: 0
```

Figura 2.8: Mensagem *ACK*

Depois de estabelecida a conexão, o usuário 7000 envia uma requisição de BYE, agindo então como um cliente, e o usuário 6000 agora age como o servidor, quando responde à requisição. A confirmação de resposta do *BYE* é feita através da mensagem *200 OK*.

2.2.3 Session Description Protocol - SDP

O SIP utiliza o *Session Description Protocol* (SDP) (ANDREASEN; BAUGHER; WING, 2006) para descrever as sessões multimídia. O SDP é responsável por fazer a troca de informações sobre a capacidade e as configurações que os terminais SIP desejam usar durante a conferência a ser criada.

O SDP provê ao SIP os meios necessários para troca de parâmetros pelos quais os terminais envolvidos na comunicação desejam estabelecer o canal multimídia. Desta forma, o SIP concentra-se apenas no correto encaminhamento das mensagens de sinalização, onde dentre outras informações, estarão os dados SDP, fazendo com que essas mensagens cheguem ao seu destino final.

Uma mensagem SDP é simples de ser entendida. Consiste de um conjunto de informações do tipo *atributo=valor*, onde o atributo define a propriedade da conferência a ser descrita seguido pelo seu respectivo valor. As tabelas 2.4 e 2.5 apresentam os principais atributos que podem ser definidos no SDP. Apesar da importância de todos os atributos, apenas alguns são obrigatórios e serão representados através do uso do símbolo (*).

O parâmetro *version (v)* define a versão do protocolo usado. O parâmetro *owner (o)* contém parâmetros como o proprietário e o identificador global da sessão e o endereço IP onde esta foi criada a sessão. O parâmetro *session (s)* descreve a sessão pretendida. O parâmetro *time (t)* permite definir o tempo durante o qual a sessão está ativa. O parâmetro *media (m)* serve para identificar para o par remoto a porta de dados e o protocolo no qual o terminal local deseja

Atributo	Descrição
*v**	versão do protocolo
*o**	criador e identificador da sessão
*s**	nome da sessão
i	informação da sessão
u	descrição do URI
p	número do telefone
c	informação da conexão
b	informação da largura de banda
z	ajuste do fuso horário
k	chave para a cifra
a	zero, uma ou mais linhas de atributo da sessão

Tabela 2.4: Descrição temporal dos atributos do SDP

realizar a conferência multimídia.

Atributo	Descrição
*t**	tempo em que a sessão está ativa
r	zero ou mais vezes de repetição
*m**	nome do meio e endereço de transporte
i	título do meio
c	informação da conexão - opcional se incluida no nível de sessão
b	informação da largura de banda
k	chave para a cifra
a	zero, uma ou mais linhas de atributo da sessão

Tabela 2.5: Descrição do meio dos atributos do SDP

Um dos cabeçalhos mais poderosos do SDP é o atributo (*a*), pois é ele que permite a extensão do SDP sem a necessidade de alterar o núcleo do protocolo. O formato geral desse atributo é: *a=<attribute>:valor*, onde *attribute* é o nome do atributo SDP estendido.

O atributo *a* pode ser usado para várias finalidades. O *rtpmap* é o que possibilita ao par em comunição a escolha dinâmica do formato de mídia que deseja usar em uma conferência. Essa escolha é considerada dinâmica, ao receber a mensagem SDP o terminal escolhe, dentre as opções ofertadas, a que melhor lhe convier.

Os atributos do SDP são mostrados na figura 2.9, que apresenta o formato de uma mensagem SDP que informa a outro terminal remoto que deseja realizar uma conferência de áudio na porta 23337, usando o protocolo RTP/AVP (2.2.4).

```
v=0
o=6000 29743 233 IN IP4 172.18.23.169
s=Mizu
c=IN IP4 172.18.23.169
t=0 0
m=audio 23337 RTP/AVP 106 105 18 4 97 104 0 8 101
a=rtpmap:106 speex/32000
a=fmtp:106 mode=8;mode=any
a=rtpmap:105 speex/16000
a=fmtp:105 mode=8;mode=any
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:4 G7231/8000
a=rtpmap:97 iLBC/8000
a=fmtp:97 mode=30
a=rtpmap:104 speex/8000
a=fmtp:104 mode=3;mode=any
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=sendrecv
```

Figura 2.9: Mensagem SDP

2.2.4 Real Time Transport Protocol / Real Time Transport Control Protocol - RTP/RTCP

Anteriormente foram descritos os protocolos SIP (seção 2.2.2) e SDP (seção 2.2.3) que possibilitam à terminais multimídias estabelecerem um canal de comunicação entre si. No entanto, esses protocolos não fornecem um canal de comunicação de dados fim a fim entre os envolvidos na comunicação, função que é de responsabilidade do *Real Time Transport Protocol* (RTP) e do *Real Time Transport Control Protocol* (RTCP) (SCHULZRINNE et al., 2003).

Inicialmente, poderia-se pensar em usar os protocolos TCP ou UDP para a realização do transporte da mídia, uma vez que eles são especificados para essa tarefa. No entanto, ambos possuem problemas que os tornam inadequados para realizar a tarefa de transportar dados em tempo real (PERKINS, 2003). O TCP é confiável pois garante a entrega do pacote através da confirmação do seu recebimento (AGENCY, 1981). Porém, os problemas com atraso e a sobrecarga (*overhead*) causada pelo processo de garantia, nem sempre garante a disponibilidade dos recursos necessários para o perfeito funcionamento da aplicação em um ambiente onde a banda é pequena. Outra opção existente na especificação do protocolo IP seria a utilização do UDP. No entanto, é necessário manter algum controle sobre a entrega dos pacotes enviados, funcionalidade que não está presente no UDP (POSTEL, 1980).

O RTP foi desenvolvido para prover serviços de envio e recebimento de dados em tempo real como, por exemplo, em uma videoconferência. O RTP cuida do transporte da mídia com o mínimo de sobrecarga possível, muito semelhante ao UDP, enquanto o RTCP cria um canal de controle do tráfego RTP, gerando estatísticas de qualidade e garantia parcial de entrega, possibilitando assim algum controle sobre a mídia, tal como no TCP.

Uma das características principais do RTP/RTCP é que ele procura interferir o mínimo na rede, evitando a necessidade do aumento da banda necessária para que o pacote multimídia possa trafegar.

O protocolo RTP atua entre as camadas de aplicação e os protocolos da camada de transporte. Por ser um protocolo que independe das camadas de rede e de transporte, pode ser implementado sobre qualquer protocolo. Com a implementação do RTP sobre o UDP, seu uso mais comum, além da simplicidade, mais dois serviços são disponibilizados: a multiplexação e a correção de erros.

Funcionalidades RTP

O protocolo RTP possui inúmeras funcionalidades, dentre elas:

- Seqüência: é atribuido um número a cada pacote, que é utilizado para verificação de perdas e/ou reordenamento de pacotes;
- Sincronismo: a variação de atraso que os pacotes podem sofrer pode interferir na reprodução da mídia com qualidade, por isso o RTP provê informações sobre o tempo de cada pacote para o caso da necessidade de reordenação dos pacotes;
- Identificação de origem: indica quem enviou o pacote. Este campo é importante em conferência *multicast*, onde os dados podem ter diversas origens;
- Criptografia: alguns fluxos de RTP podem ser criptografados;
- Controle da sessão: permite aos participantes trocarem informações pessoais;
- Qualidade de Serviço: o destinatário tem a possibilidade de fornecer informação sobre a qualidade da recepção.

Pacote RTP

Um pacote RTP, conforme visto na figura 2.10 é bastante simples de ser entendido. Ele consiste de um conjunto de cabeçalhos totalizando aproximadamente 40 bytes e um campo de

carga (*payload*) onde irão trafegar os dados multimídia propriamente ditos (SCHULZRINNE et al., 2003). Na primeira linha e coluna do pacote é apresentado o número de bytes que cada cabeçalho ocupa. A linha organiza o pacote em conjunto de 32 bits e a coluna diz qual a posição do primeiro bit onde a informação será armazenada.

bit offset	0-1	2	3	4-7	8	9-15	16-31
0	Ver.	P	X	CC	M	PT	Sequence Number
32	Timestamp						
64	SSRC identifier						
96	CSRC identifiers (optional) ...						
96+(CC×32)	Extension header (optional).						
96+(CC×32) + (X×((EHL+1)×32))	Data						

Figura 2.10: Pacote RTP.

O campo *sequence number* provê funcionalidades importantes de controle. Através desse valor, o terminal pode perceber com clareza se os pacotes estão chegando na ordem sequencial correta e se existe alguma descontinuidade no seu recebimento.

No entanto, essa informação sozinha não é capaz de definir a situação em que se encontra a rede. Para obter essa informação, o RTP utiliza o campo *timestamp*, que tem por finalidade definir o horário no qual o pacote RTP foi gerado em sua origem, importante para resolver problemas com relação ao *jitter*.

Outro campo existente em um pacote RTP é o *Synchronization Source Identifier* (SSRC). O SSRC possui o objetivo de identificar de forma única um determinado terminal em uma conferência, possibilitando-lhe conversar com uma infinidade de outros terminais de forma simultânea.

Nos casos onde a mídia é criada por vários terminais (áudio-conferência) é usado o campo *Contributed Source Identifiers* (CSRC) para identificar cada SSRC que contribui na formação da mídia. Para ser possível que a aplicação saiba quantos CSRC existem no pacote é utilizado o campo CC, responsável por definir esse número de usuários na áudio-conferência.

O RTP possui outros campos que possuem um caráter informativo e auxiliar, são eles:

- *V (version)*: indica a versão do RTP para construir a mensagem;
- *P (padding)*: informa à aplicação se o pacote sofreu algum tipo de preenchimento para

que atingisse um tamanho múltiplo de 32;

- *M (marker)*: utilizado para otimização do canal.

O campo *payload type* (PT), tem por finalidade descrever o tipo de mídia contido no pacote bem como auxiliar a aplicação a decodificar adequadamente o pacote recebido, já que ele é o responsável por informar à aplicação o *codec* utilizado para digitalizar o áudio. O tipo de áudio a ser utilizado é negociado através da transações efetuadas pelo protocolo SIP e SDP, conforme visto na seção 2.2.3.

Diferentemente do RTP, os pacotes RTCP não possuem uma arquitetura comum de formação, pois em um unico pacote podem haver vários relatórios de tipos diferentes, fazendo com que cada pacote tenha um formato diferente (PERKINS, 2003).

Sessão RTP

Uma sessão RTP é uma associação de usuários que se comunicam através do protocolo RTP. Cada participante utiliza dois endereços de transporte para cada sessão: um para o fluxo RTP e um para as mensagens RTCP. Quando uma transmissão é feita através da difusão seletiva (*multicast*) todos os participantes usam o mesmo par de endereços de transporte multicast. Os fluxos de dados que estão em uma mesma sessão devem compartilhar um canal RTCP comum (SOUZA, 2003).

Se em uma conferência estiver sendo transmitido áudio e vídeo, estes serão transmitidos em sessões RTP distintas. Os pacotes RTCP vindos da origem terão o mesmo identificador, e as sessões RTP podem associar-se. Essa divisão ocorre para que os participantes da sessão possam escolher as mídias a receber de acordo com os recursos e processamento da rede.

2.2.5 Inter Asterisk eXchange - IAX

O *Inter Asterisk eXchange* (IAX) é um protocolo para trasmissão e controle de mídia utilizado em redes IP, e otimizado para chamadas VoIP (SPENCER et al., 2009). É utilizado no PABX IP Asterisk e está em sua segunda versão (GONÇALVES, 2005).

O IAX foi desenvolvido como uma alternativa aos protocolos SIP e H.323. Pode ser utilizado, assim como o SIP, para qualquer tipo de sessão multimídia (voz, dados). O seu desenvolvimento se deu principalmente pelos seguintes objetivos: a redução do consumo de banda; suporte a NAT transparente; transmissão de informações de plano de discagem e controle de chamadas usando voz sobre IP.

O IAX multiplexa a sinalização e o transporte de dados na mesma porta UDP (4569) entre dois nós. Essa característica o faz distinto de outros protocolos, que utilizam separadamente o controle e a transmissão de dados.

O *Inter Asterisk eXchange* é mais eficiente em relação ao consumo de banda por ter sido desenvolvido como protocolo binário, otimizado especificamente para a redução do consumo de banda em chamadas de voz feitas através da rede IP. O IAX provê suporte a autenticação através de chaves assimétricas (públicas e privadas) e *trunking*.

O *trunking* permite que múltiplos fluxos de voz compartilhem o mesmo canal, reduzindo assim o *overhead* causado pelos pacotes IP. Ou seja, ele remove a redundância do pacote IP para cada fluxo, no entanto, isso só será possível quando as chamadas estiverem ocorrendo entre os mesmos nós. Com a utilização do *trunking* o primeiro canal utilizará um certo valor de largura de banda, dependendo do *codec* utilizado e a partir do segundo canal, devido a não existência do mesmo *overhead* nos pacotes IPs, a largura de banda necessária é menor, já que todo o tráfego é feito pela mesma porta UDP.

A unidade de comunicação básica no protocolo IAX é um quadro (*frame*), existem basicamente 3 tipos (SPENCER et al., 2009):

- *FULL FRAME* - pode ser usado para para enviar dados de mídia ou de sinalização. São enviados de forma confiável, ou seja, sempre que um destinatário receba um *full frame* é exigido que ele retorne um *ACK*. Entretanto, não é recomendado usá-lo para enviar dados de mídia devido ao *overhead*, causado pelo *ACK* e pelo tamanho do *frame*, que é de 12 octetos;

- *MINI FRAME* - assim como o *full frame*, pode ser usado para enviar dados de sinalização ou de mídia. Não é exigida confirmação para este tipo de *frame*, pois como o protocolo IAX é otimizado para transmissão de voz e o *overhead* para garantir entrega é grande, acaba por limitar a banda, além disto a perda de um *frame* em uma comunicação de voz não impossibilita a conversa. O cabeçalho deste *frame* é de apenas 4 octetos;

- *META FRAME* - são usados com dois propósitos: transmissão de vídeos com um cabeçalho otimizado para este fim (*Meta Video Frames*) e Agrupamento de mútiplos *streams* entre dois pontos usando um único cabeçalho para minimizar o consumo de banda (*Meta Trunk Frames*)

Sinalização IAX

O processo de efetuar uma ligação através do protocolo IAX é similar ao processo efetuado pelo SIP, porém muito mais simples. As mensagens IAX podem ser requisições efetuadas pelo cliente ao servidor e respostas enviadas pelo servidor ao cliente. As mais utilizadas podem ser vistas na tabela 2.6.

Método	Funcionalidade
NEW	Realiza uma nova chamada
PING	Efetua o comando *ping* para verificar a presença de um usuário/servidor na rede
PONG	Responde ao ping
ACK	Usado para confirmar uma requisição
ACCEPT	Faz a aceitação de um pedido
REJECT	Rejeita um pedido que foi feito. Pode ser utilizado para indicar que uma chamada não pode ser realizada, por exemplo
AUTHREQ	Faz um pedido de autenticação. Por exemplo: Um servidor pode, durante uma chamada que ainda esta para ser estabelecida, utilizar o *AUTHREQ* para pedir ao cliente que o mesmo se autentique. O usuário, então, responde esse pedido com o comando *AUTHREP*
AUTHREP	Responde a um pedido de autenticação
HANGUP	Utilizado para finalizar uma chamada

Tabela 2.6: Mensagens de Requisições do IAX

Na figura 2.11 uma pessoa, quando deseja efetuar uma ligação, envia um *NEW* (passo 1) para a pessoa desejada. Caso a pessoa chamada esteja disponível e aceite a ligação, ela envia de volta uma mensagem *ACCEPT* (passo 2). Em resposta ao *ACCEPT*, o chamador envia uma mensagem de reconhecimento, *ACK* (passo 3). A partir desse momento, a pessoa chamada envia duas mensagens: uma *RINGING* (passo 4) e a outra *ANSWER* (passo 6), que são respondidas, pela pessoa que efetuou a ligação, com um *ACK* para cada mensagem (passos 3 e 5) . Depois do segundo *ACK*, a chamada é estabelecida e os dados começam a trafegar.

A finalização da chamda acontece da seguinte maneira: o assinante chamado envia uma mensagem *HANGUP*, a qual é respondida com um *ACK* pelo assinante chamador. Assim que a mensagem *ACK* é recebida, a ligação é finalizada.

2.3 Asterisk

Os primeiros *Private Branch eXchange* (PBX) foram desenvolvidos como um equipamentos analógicos que permitem realizar várias ligações entre os telefones sem a necessidade de ter

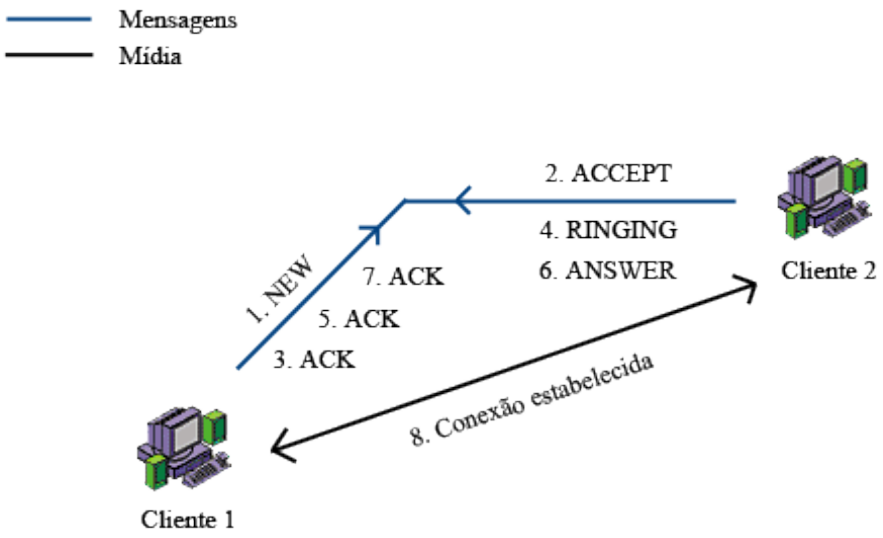

Figura 2.11: Estabelecimento de sessão IAX. Fonte (GPWM, 2008)

um operador. É um equipamento que executa serviços privados de telefonia, ou seja, é uma central telefônica privada onde chegam as linhas da rede pública e saem os ramais para os usuários (3CX, 2008).

Por ser um equipamento desenvolvido por empresas, possui *hardware* e *software* proprietário. Devido a este fato, qualquer funcionalidade adicional que um usuário queira agregar ao seu PBX está limitada somente ao fabricante e qualquer manutenção ou modificação de programação depende de algum profissional especializado.

O PBX IP surgiu como uma alternativa aos problemas apresentados pelo PBX tradicional. O PBX IP possui uma arquitetura diferente, maior flexibilidade e alguns com soluções livres, acabando assim com a dependência existente nos PBXs tradicionais. Além disso, agrega inúmeras vantagens em relação a um PBX convencional, entre elas a possibilidade de se agregar novas funcionalidades facilmente, personalização e controle total sobre o seu equipamento.

O Asterisk (ASTERISK, 2008) é um *software* que implementa um PBX. Foi desenvolvido pela empresa Digium e está sob uma licença de *software* livre (*General Public Licence* – GPL). O Asterisk possui diversos desenvolvedores e colaboradores por todas as partes do mundo, tendo como principal colaborador e mantenedor a própria empresa Digium. Trata-se de uma solução livre e que foi desenvolvido para ser flexível e facilmente extensível, ao contrário dos PBXs proprietários que são caros e limitados quanto à manutenção e personalização.

O Asterisk oferece todas funcionalidades e soluções de um PBX tradicional: possui funci-

onalidades básicas e outras extremamente avançadas, encontradas em grandes sistemas de PBX proprietários. Algumas das principais funcionalidades do Asterisk são: gravação de chamadas, *Call Detail Records* (CDR), integração com bases de dados, transferência de chamadas, música de espera, texto para fala ou *Text-to-Speech* (TTS), resposta de voz interativa ou *Interactive Voice Response* (IVR), segurança e suporte a diversos protocolos de VoIP (ASTERISK, 2008), incluindo o SIP (seção 2.2.2) e IAX (seção 2.2.5).

O PBX IP Asterisk possui algumas opções para agregar segurança ao serviço de telefonia IP. Dentre elas destacam-se os protocolos (YORK, 2008):

- *TLS-Encrypted SIP*;
- *Secure RTP (SRTP)*;
- *IAX*;

Esse protocolos referentes à segurança estão presentes nas versões 1.6 (TLS e IAX) e 1.4 (SRTP e IAX) do Asterisk, e serão vistas no próximo capítulo.

O controle das operações do Asterisk é realizado através de arquivos de configuração em formato de texto puro. Os principais arquivos de configuração são (MAHLER, 2004):

- *asterisk.conf* - localização das pastas de componentes do Asterisk;
- *extensions.conf* - configuração do plano de discagem;
- *iax.conf* - configuração referente ao protocolo IAX;
- *manager.conf* - configuração da API do Asterisk;
- *modules.conf* - configuração dos módulos a serem carregados;
- *sip.conf* - configuração referente ao protocolo SIP;
- *zapata.conf* - configuração dos hardwares de telefonia;

3 *Segurança VoIP*

Neste capítulo serão apresentados os conceitos de segurança e criptografia, os tipos de ataques a que estão sujeitos os sistemas de comunicação VoIP e os métodos utilizados para prover segurança no meio IP.

Um dos pontos discutidos será a proteção da sinalização existente em uma conferência VoIP utilizando-se o *Transport Layer Security* (TLS). Aplicações multimídias também estão vulneráveis à interceptação e manipulação dos dados transmitidos. O protocolo RTP, geralmente utilizado para transporte da mídia, voz e/ou dados, pode ser substituído pelo *Secure Real Time Protocol* (SRTP) responsável por fazer a segurança da mídia transmitida. E, o *Inter Asterisk eXchange* (IAX) capaz de proteger sinalização e mídia simultaneamente.

3.1 Conceitos Básicos de Segurança

Inicialmente, e durante algum tempo, as redes de computadores eram usadas por poucas pessoas e principalmente para envio de correio eletrônico. Por tal motivo a seguraça não representava tanta importância. Com o decorrer dos anos a Internet se desenvolveu e com a introdução do VoIP a necessidade de segurança se torna imprescindível.

Segundo a (ISO/IEC, 2005) a proteção da informação pode ser definida como a proteção da informação à qualquer tipo de ameaça que gere a descontinuidade do negócio, minimize o retorno sobre seus investimentos e as oportunidades desse negócio. Para que seja possível essa proteção, três propriedades principais da informação devem ser mantidos (LANDWEHR, 2001): a integridade, a confidencialidade e a disponibilidade.

A **integridade** (ISO/IEC, 2004) pode ser definida como a propriedade que garante que determinada informação mantenha as características originais estabelecidas pelo proprietário da informação, ou seja, garantir que ela está completa e exata. Essa propriedade é responsável por garantir ao usuário que determinado conteúdo acessado seja realmente o original, sem qualquer tipo de adulteração. A **confidencialidade** (STALLINGS, 2000) tem a função de proteger a

informação de acessos indevidos, ou seja, garante que somente os usuários autorizados pelo proprietário da informação possam acessá-la. A **disponibilidade** (ISO/IEC, 2004) garante que a informação esteja sempre acessível e utilizável para os usuários autorizados.

Além dessas, outras propriedades podem ainda estar associadas ao conceito de segurança, tais como a **autenticidade** e **não repúdio** (ISO/IEC, 2005). A **autenticidade** possui o objetivo de garantir que o remetente e o destinatário de uma determinada informação sejam devidamente identificados, evitando que um terceiro usuário (intruso) assuma o papel de um deles e realize operações ilegais. O **não repúdio** garante que somente o originador da informação a tenha criado, não podendo negar sua autoria futuramente.

3.1.1 Principais Ameaças na comunicação Voz sobre IP

Existem diversos tipos de ataques que podem ser realizados com o objetivo de tentar comprometer as propriedades básicas de segurança. São estes:

Ataque por negação de serviço (*Denial of service* - DoS). Ataques do tipo *DoS* têm o objetivo de congestionar e indisponibilizar, temporariamente, os recursos disponíveis em uma rede. Geralmente os alvos desse tipo de ataque são servidores conhecidos, tais como, servidores DNS, servidores Web e roteadores (CISCO, 2009).

Escuta do meio (*Eavesdropping*). *Eavesdropping* é uma técnica que se baseia na violação da confidencialidade, ou seja, em uma rede onde se usa VoIP, o *Eavesdropping* pode ser comparado à uma escuta telefônica não autorizada. Em redes onde não há meio para se proteger os dados que trafegam, é possível que um atacante capture os dados transmitidos e tenha acesso à mídia de transmissão (FOCUS, 2009).

Homem no meio (*Man in the Middle ou MITM*). Nesse tipo de ataque é possível que o atacante fique entre o emissor e o destinatário da mensagem, sendo capaz de ler, inserir e modificar as mensagens. Em sistemas VoIP é possível escutar perfeitamente a mídia transmitida (TELECO, 2009).

Ataque de mensagens antigas (*Replay-Packets Attacks*). Nos ataques do tipo *Replay* o atacante consegue, através de um analisador de tráfego de rede (*sniffer*), capturar os pacotes de uma conferência em andamento. Após essa captura o atacante reinsere os pacotes modificados na rede como se eles fossem legítimos (DENG; LI; AGRAWAL, 2002).

Mascaramento (*Spoofing*). Consiste em mascarar (*spoof*) pacotes IP com endereços remetentes falsificados. Existem diversos tipos de ataques que utilizam tal técnica, como *Man in*

the Middle e *Denial of Service.* (FERGUSON; SENIE, 1998).

Personificação. essa técnica permite que o atacante falsifique a sua identidade para se passar por algum ou todos os usuários envolvido na chamada, de forma a induzir os usuários acreditem estar conversando com uma outra pessoa, ou seja, o atacante finge ser alguém que não é para obter informações sobre a conversa (STALLINGS, 2000).

Sequestro de chamadas (*Call Hijack*). Os protocolos de sinalização que não utilizam criptografia estão sujeitos a interceptação de pacotes no momento do registro. Isso proporciona ao atacante obter informações privilegiadas, podendo ainda fazer com que as chamadas, destinadas a um terceiro, passem a ser desviadas para ele próprio (THERMOS, 2007).

3.2 Conceitos Gerais de Criptografia

A criptografia é a ciência utilizada para alterar a aparência dos dados com o objetivo de mantê-los seguros. Ou seja, é a transformação de uma mensagem clara em uma forma ilegível a fim de esconder o seu significado. O fator chave para uma criptografia forte é a dificuldade de engenharia reversa (TRAPPE; WASHINGTON, 2001).

A função matemática utilizada para a cifragem e decifragem é chamada algoritmo criptográfico ou cifra. A segurança de uma cifra pode ser baseada exclusivamente em manter em segredo a sua funcionalidade e, neste caso, é uma cifra restrita. Existem muitos inconvenientes para cifras restritas. É muito difícil manter um algoritmo em segredo quando é utilizado por muitas pessoas. Se ele está incorporado em um produto comercial, por exemplo, é apenas uma questão de tempo e dinheiro antes de que se faça uso da engenharia reversa. Por estas razões, os algoritmos utilizados atualmente são protegidos, ou seja, a cifragem e decifragem faz uso de um parâmetro, conhecido como a chave. A chave pode ser escolhida a partir de um conjunto de valores possíveis, chamado de tamanho da chave.

Quanto maior o tamanho da chave melhor, uma vez que diariamente surgem computadores com mais capacidade de processamento. Somente o tamanho da chave não é suficiente para garantir a segurança do canal, a robustez do algoritmo utilizado também deve ser levado em consideração (TRAPPE; WASHINGTON, 2001).

As chaves podem ser consideradas *assimétricas* quando há uma chave específica para cifrar e outra para decifrar, esse sistema também é conhecido como *criptografia de chave pública*. Quando uma única chave é utilizada para ambos os processos, as chaves são chamadas de simétricas, ou *criptografia de chave privada* (STINSON, 2006).

Para que duas pessoas possam se comunicar de forma segura utilizando criptografia simétrica, eles devem acordar uma chave secreta e mantê-la entre si. Se eles estão em diferentes locais físicos, eles devem confiar em algum meio, como o telefone, por exemplo, ou algum outro meio de comunicação seguro para impedir a divulgação da chave secreta durante a transmissão. Quem intercepta a chave no trânsito pode ler mais tarde, modificar e forjar todas as informações cifradas com essa chave.

O conceito apresentado pode ser melhor entendido observando-se a figura 3.1. Na figura pode-se perceber a existência de um canal inseguro entre dois usuário e chaves para a realização da cifragem e decifragem da mensagem transmistida. Os usuários, Alice e Bob, desejam realizar uma comunicação segura. O objetivo do sistema apresentado na figura é garantir a confidencialidade dos dados transmitidos.

Para proteger o conteúdo da comunicação de ações ilícitas, é inserido no canal transmissor um sistema cifrador e no canal receptor um sistema decifrador. O cifrador é o responsável por tornar a mensagem ilegível à pessoas sem autorização. Já o decifrador tem a finalidade de recompor a mensagem à forma original para que o devido receptor possa ter acesso ao conteúdo da informação transmitida.

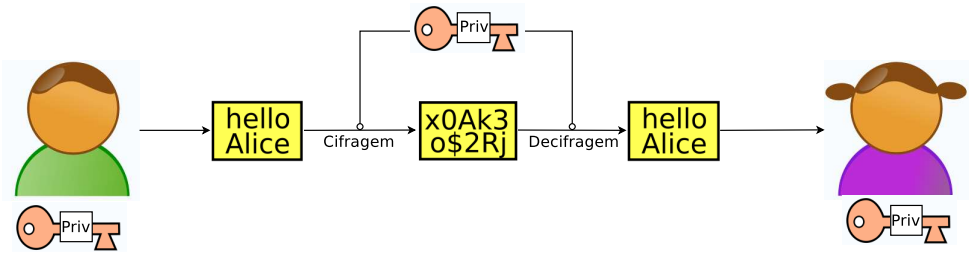

Figura 3.1: Cenário básico de uso de criptografia. Fonte (PARZIALE et al., 2006)

A negociação dos parâmetros de segurança (chaves, tamanho das chaves, algoritmos, etc) do canal de comunicação que estarão estabelecidos durante a conferência é feita durante a troca de sinalização, essa tarefa é conhecida como *Key Agreement*.

3.2.1 Criptografia de Chave Privada

Na criptografia de chave privada são utilizados algoritmos simétricos, onde a chave de cifragem é a mesma utilizada pra a decifragem. Estes são algoritmos criptográficos convencionais onde o remetente e o receptor devem concordar com a chave antes do início da troca de mensagens entre eles. A figura 3.1 ilustra um esquema utilizando algoritmo simétrico. A criptografia de chave privada tem uma vantagem sobre a criptografia de chave pública que consome mais

tempo e processamento para decodificação dos dados (STINSON, 2006).

Os algoritmos que utilizam chave privada são:

- **DES** - *Data Encryption Standard* (1977 - IBM);
- **3DES** - uma evolução do DES;
- **Diffie-Hellman** - criado por Whitfield Diffie e Martin Hellman em 1976.

3.2.2 Criptografia de Chave Pública

A criptografia assimétrica, ou criptografia de chave pública se dá quando um usuário deseja receber e enviar informações criptografas através de um canal seguro utilizando pares de chaves. Uma dessas chaves, a utilizada para cifrar a mensagem, é chamada de chave pública, pois pode ser distribuída abertamente e entregue sem proteção para qualquer usuário ou entidade que deseja lhe enviar uma mensagem segura. A outra chave, utilizada para decifrar, é chamada de chave privada e deve ser mantida em total segurança pois, se for capturada por um atacante, ele será capaz de decifrar todas as mensagens endereçadas ao usuário (TRAPPE; WASHINGTON, 2001).

Na figura 3.2 é mostrado um exemplo de comunicação utilizando criptografia de chave pública. Alice deseja se comunicar com Bob e para isso envia-o sua chave pública. Bob ao enviar a mensagem para Alice utiliza a chave pública enviada por ela para cifrá-la. Alice por sua vez decifra a mensagem enviada por Bob utilizando sua chave privada.

Os algoritmos que utilizam chave pública são:

- **RSA** - *Rivest Shamir Adleman* ;
- **DSA** - *Digital Signature Algorithm*.

3.2.3 Certificado Digital

A identificação dos usuários em uma rede pode ser realizada de forma simétrica ou assimétrica. Na forma simétrica, a autenticidade das chaves é garantida mediante a instituição de uma *Autoridade Certificadora* (AC). A Autoridade Certificadora é um órgão central no qual todos os envolvidos confiam durante um processo de comunicação existente. Essa AC, através da sua chave privada, cifra as chaves públicas dos usuários da sua rede de segurança, gerando assim um novo documento conhecido como *Certificado Digital* (PARZIALE et al., 2006).

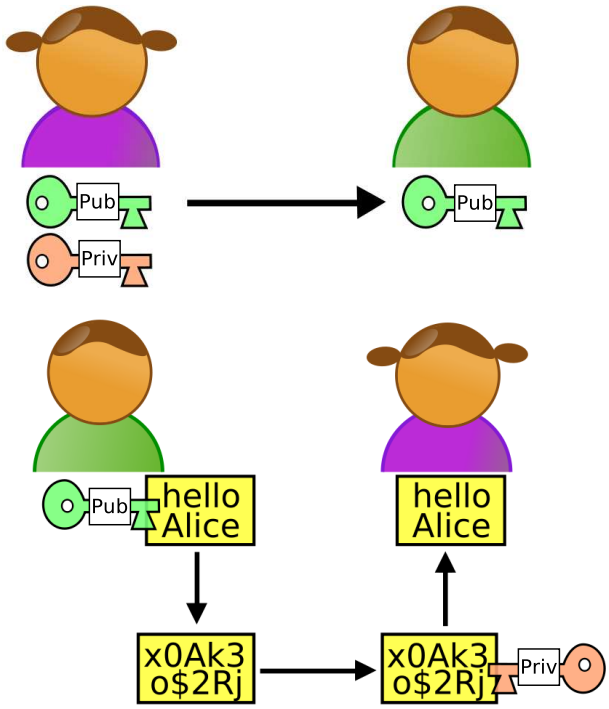

Figura 3.2: Cenário básico do uso de criptografia de chaves pública.

Os certificados podem ser emitidos para diversos fins como, por exemplo, autenticação de usuário na *Web*, segurança da camada de transporte (TLS) e assinaturas de códigos.

Um *certificado digital* é normalmente composto pelas seguintes informações (GOMES; RIBEIRO, 2004):

- Da chave pública cifrada do usuário;
- Das informações para identificação do usuário, como endereço de *e-mail* ou um *alias* (apelido);
- Do período de validade (período em que o certificado será considerado válido);
- Da identificação da AC responsável pela autenticação da chave do usuário.

Para que um usuário possa checar a autenticidade da chave pública existente no certificado digital, é necessário somente que ele o decodifique usando a chave pública da AC que a assinou. Se funcionar corretamente é comprovado que o certificado existe e foi devidamento assinado pela AC.

A necessidade de criação de uma rede de relacionamento confiável pode fazer com que haja obrigatoriamente a existência de uma infra-estrutura de chaves públicas. Como alternativa

(ROSENBERG et al., 2002) cita o uso de *certificados auto-assinados*. Um certificado auto-assinado é criado localmente e não possui referência a nenhuma Autoridade Certificadora, o que impossibilita a garantia de validade e autenticidade do mesmo. Esse tipo de certificado está vulnerável a ataques do tipo MITM.

3.3 O Protocolo TLS

O *Transport Layer Security* (TLS) (DIERKS, 2006) é um protocolo que atua na camada de transporte da pilha TCP/IP e é responsável pelo serviço de segurança da sinalização SIP, utilizando o protocolo TCP como base para realização desta tarefa. O protocolo TLS protege apenas as informações da camada de transporte da pilha TCP/IP, sendo assim, o endereço IP dos envolvidos na processo de comunicação irá trafegar em claro.

O protocolo TLS é indicado por (ROSENBERG et al., 2002) para prover a negociação dos parâmetros criptográficos necessário ao estabelecimento de um canal de comunicação SRTP de forma protegida, como será visto na seção 3.4.

O funcionamento do TLS consiste de uma fase inicial conhecida como *handshake*. Nessa fase são negociados os parâmetros de segurança envolvidos na sessão (chave, algoritmo criptográfico, etc) assim como a autenticação de cada elemento necessário para o estabelecimento do canal. Essa autenticação pode ser feita de duas maneiras: unidirecional ou bidirecional. Na forma unidirecional somente um dos usuários é autenticado. Na forma bidirecional tanto os servidores quanto os clientes se autenticam mutuamente.

Na figura 3.3 a mensagem `ClientHello` é enviada no início da comunicação TLS. Essa mensagem é utilizada pelo cliente para solicitar ao servidor que inicie a negociação dos parâmetros de segurança TLS. Os atributos da mensagem `ClientHello` podem ser vistos na tabela 3.1.

O servidor ao receber a mensagem `ClientHello` responde com a mensagem `ServerHello`. Na mensagem `ClientHello` o cliente oferece algumas opções para negociação dos parâmetros de segurança. Na mensagem `ServerHello` o servidor informa se aceita ou recusa esses parâmetros, pois é dele o papel para indicar se a sessão será ou não estabelecida. Os atributos da mensagem `ServerHello` podem ser vistos na tabela 3.2.

A mensagem `ServerKeyExchange` complementa o campo `CipherSuite` da mensagem `Serverhello`, enviando as informações referentes à chave pública. A chave vai depender do algoritmo utilizado e pode ser transmistida sem nenhum tipo de segurança sem que haja

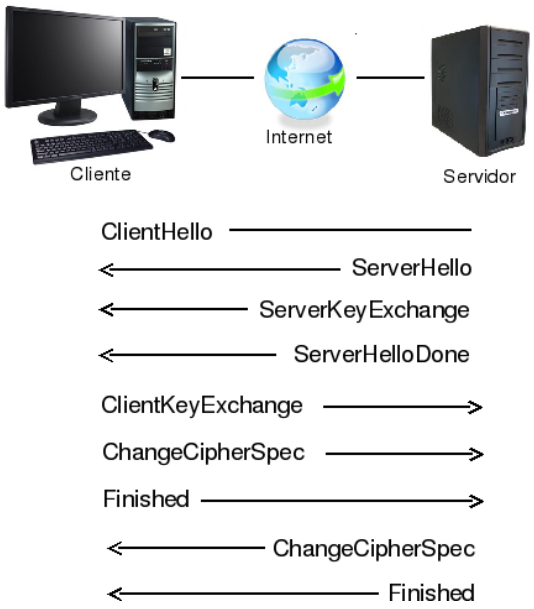

Figura 3.3: Funcionamento do TLS

prejuízos. O cliente depende dessa chave para cifrar uma chave de sessão e permitir que somente o servidor possa decifrá-la. Logo após a mensagem `ServerKeyExchange`, o servidor envia a mensagem `ServerHelloDone` para indicar ao cliente o término das mensagens iniciais para o estabelecimento da sessão. Tal mensagem não possui nenhum campo, mas é importante para que o cliente possa identificar o fim das negociações e dê início aos próximos passos para o estabelecimento da sessão.

Campo	**Descrição**
Version	Indica a versão mais recente do protocolo TLS suportada pelo cliente
RandomNumber	Número randômico de 32 bytes utilizado para os cálculos criptográficos
SessionID	Identifica uma sessão TLS
CipherSuites	Lista os parâmetros criptográficos que o cliente suporta
CompressionMethods	Lista os padrões de compressão que o cliente suporta

Tabela 3.1: Atributos da mensagem ClientHello

O cliente após receber a mensagem `ServerHelloDone`, cria uma chave de sessão e a cifra com a chave pública do servidor. Tal chave é então encaminhada ao servidor através da mensagem `ClientKeyExchange`. Como a chave de sessão foi cifrada pela chave pública do servidor, o cliente pode certificar-se de que somente o servidor em questão poderá obtê-la. Isso garante que nenhuma outra parte conseguirá decifrar as mensagens trocadas entre o cliente e o servidor.

Campo	Descrição
Version	Indica a versão do protocolo TLS que será utilizada
RandomNumber	Número randômico de 32 bytes utilizado para os cálculos criptográficos
SessionID	Identifica uma sessão TLS
CipherSuites	Indica os parâmetros criptográficos que serão utilizado na comunicação
CompressionMethods	Indica o padrão de compressão que vai ser utilizados na comunicação

Tabela 3.2: Atributos da mensagem ServerHello

Após a troca da chave de sessão, cliente e servidor enviam a mensagem `ChangeCipherSpec` que é responsável por indicar o início do uso dos parâmetros de segurança definidos durante a etapa de negociação. Após tal mensagem ambos enviam a mensagem `Finished` a qual indica que a negociação ocorreu com sucesso.

3.3.1 Negociação de chaves utilizando TLS

Para fazer uso do TLS no SIP, o usuário necessita da utilização de uma URI (seção 2.2.2) especial chamada *SIPs*. O SIPs é responsável por informar às entidades envolvidas na conferência que o protocolo TLS deve ser preferencialmente utilizado para realizar o processo de sinalização SIP.

O uso do TLS juntamento com a sinalização SIP afeta a escalabilidade de soluções VoIP, pois para que a chamada seja encaminhada corretamente o ambiente necessita que todos os servidores *proxy* tenham estabelecido algum tipo de relacionamento de confiança (ROSENBERG et al., 2002).

Essa limitação não torna o uso do TLS inviável, pelo contrário, o suporte ao TLS não é obrigatório, no entanto, um *proxy* ao identificar que o próximo elemento intermediário não pertence a sua rede de confiança ou não provê suporte ao TLS, encaminhará a mensagem em claro, usando uma URI comum.

Além dessa limitação, o uso de mensagens SIPs oferece um problema de desempenho. Para que os *proxys* pertençam à mesma rede de confiança é estabelecida uma relação de confiança antes que qualquer mensagem SIP seja enviada (JENNINGS, 2004). Esse processo ocasiona uma sobrecarga que aumenta o tempo para estabelecimento dos canais de comunicação. Em enlaces cuja banda disponível seja pequena poderá ocorrer momentos em que a chamada demore muito pra completar fazendo com que o usuário desista da chamada (OHTA, 2006).

3.4 O protocolo SRTP/SRTCP

Os protocolos *Secure Real Time Protocol* (SRTP) responsável pela proteção da mídia e *Secure Real Time Control Protocol* (SRTCP) que visa proteger o conteúdo das mensagens de controle, são perfis específicos do RTP/RTCP (SCHULZRINNE et al., 2003) e foram desenvolvidos com o intuito de oferecer confidencialidade e integridade para todo o pacote RTP/RTCP (AL., 2004). Além de possuírem funções diferentes, no SRTCP é obrigatório que se faça a cifragem e autenticação dos pacotes, no SRTP, somente na carga multimídia é obrigatório que se faça a encriptação, porém a autenticação é uma forte aliada contra ataques do tipo *replay-packets*.

O SRTP/SRTCP trabalha com um conceito muito importante chamado *contexto criptográfico* (AL., 2004), o qual se refere ao conjunto das informações necessárias ao estabelecimento de um canal seguro. Essa informações (chaves, protocolos envolvidos, etc.) devem ser mantidas em segurança durante o tempo em que o canal de comunicação estiver em uso.

Para prover segurança contra ataques do tipo *replay-packets* o contexto criptográfico utiliza um vetor que armazena o valor dos últimos pacotes autenticados (*replay-list*). Quando o pacote chega ao destino, é feita uma verificação na lista e mensagens sobre o índice do pacote de voz. Caso seu índice não conste no vetor o pacote será processado. No entanto, para que essa funcionalidade esteja presente é necessário que todos os pacotes sejam autenticados.

A *taxa de derivação da chave* é outra variável utilizada no contexto criptográfico e é utilizada pelo protocolo para calcular o número de pacotes que utilizarão uma única chave.

Para maior eficiência o SRTP trabalha com o processo de derivação de chaves, onde mediante uma única chave de sessão negociada entre os pares e transportadas por um protocolo de sessão, se gera conjuntos de chaves individuais responsáveis pela autenticação e cifragem oferecidas pelo protocolo.

O processo de derivação de chaves fornece ao processo de comunicação uma segurança adicional, pois se apenas uma das chaves individuais é quebrada, somente as mensagens associadas à ela estarão disponíveis. Se a chave de autenticação for decifrada, por exemplo, as mensagens de áudio ou vídeo continuarão seguras.

Outra informação existente no contexto criptográfico e que provê o aumento da proteção das chaves existentes pode ser oferecida pelo *master salt*. Esse processo consiste no preenchimento de valores aleatórios (*padding*) em uma chave calculada, dificultando o cálculo do seu valor pelo atacante (TRAPPE; WASHINGTON, 2001).

Além dessas informações, o contexto criptográfico ainda trabalha com um atributo chamado

rollover counter (ROC). ROC é o índice que define quantas vezes foi zerado o valor do número de sequência do pacote SRTP. O ROC também determinará o índice final do pacote, usado no processo de derivação da chave de criptografia e descriptografia.

O processo de negociação e estabelecimento do contexto criptográfico ocorre antes do funcionamento dos protocolos SRTP/SRTCP. A formação desse contexto deve ocorrer através de um protocolo de negociação de cheves (*Key Agreement*) podendo estar associado ao SIP ou por mensagens iniciais no canal de comunicação (ZIMMERMANN; JOHNSTON; CALLAS, 2006).

A negociação do contexto criptográfico no SRTP pode ser feita de duas formas: através da arquitetura *Mikey* e *SDES*. Essas funcionalidades serão abordadas na sequência.

3.4.1 Descrição de Parâmetros Criptográficos usando o SDES

O SDP (seção 2.2.3) possui apenas um atributo para descrever o contexto criptográfico, o *encryption key* (*k*), responsável pelo transporte da chave mestra que será utilizada na sessão. Para contextos criptográficos que necessitam de mais informações o esquema oferecido pelo SDP não é adequado, como é o caso do SRTP/SRTCP. O *Session Description Protocol Security Description for Media Streams* (SDES) é uma extensão do SDP desenvolvida em 2006 para ser utilizado como mecanismo de negociação do contexto criptográfico do SRTP (ANDREASEN; BAUGHER; WING, 2006).

O SDES utiliza o mesmo modelo pedido/resposta do SDP, em que o remetente propõe os parâmetros que deseja para estabelecer o canal de comunicação e o destinatário responde aceitando os parâmetros, que podem ser os mesmos ofertados ou novos escolhidos pelo próprio destinatário.

A descrição do contexto criptográfico (figura 3.4) do SDES é realizada através do atributo *crypto* apresentado na equação a=crypto:<crypto-suite><key-params>[<session-params>].

O campo *tag* possui a função de identificar unicamente o atributo criptográfico dentro do escopo do SDP. Isso porque no processo de oferta o remetente pode enviar no atributo *crypto* mais de uma opção para possibilitar que o destinatário escolha o mecanismo de construção do contexto criptográfico que mais se adeque ao seu objetivo. Sendo assim, o destinatário deve informar com quais atributos será realizada a comunicação, sinalizando a *tag* para isso.

O atributo *crypto-suite* tem a finalidade de identificar qual o algoritmo será utilizado para cifrar a mensagem e outros parâmemtros utilizados como tamanho da chave, *master salt*, etc.

Outro atributo chamado *key-params* é responsável por definir a negociação de chaves.

Mesmo não sendo de uso obrigatório o *session-params* é importante pois possibilita que sejam definidas as demais informações necessárias para a criação do contexto criptográfico. Quando esse atributo não é usado, é necessário que os envolvidos no processo compartilhem previamente as informações ou algum tipo de padrão a ser estabelecido. Para inicialização do SRTP, são utilizadas ainda as informações (seção 2.2.4): *SSRC*, *ROC* e *SEQ* (número de sequência do pacote).

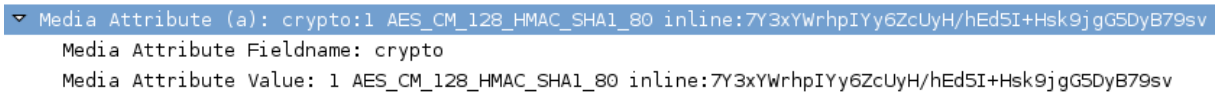

Figura 3.4: Descrição dos Parâmetros Criptográficos - SDES

Essas informações são necessárias porém disponíveis apenas após a inicialização do protocolo (ANDREASEN; BAUGHER; WING, 2006). Elas propõem uma função chamada *late binding of one or more SSRV's to a crypto context*, cuja idéia é compor temporariamente o contexto criptográfico através da recuperação da porta e endereço IP do usuário através dos atributos *c* e *m* do SDP. O contexto criptográfico é atualizado, no momento em que estas informações estiverem disponíveis, no recebimento dos primeiros pacotes SRTP/SRTCP.

O SDES possui uma forma simples e completa de fazer a negociação do contexto criptográfico, no entanto possui a necessidade de ser transportada através de um canal seguro, sua maior limitação.

3.4.2 Multimedia Internet KEying - MIKEY

O *Multimedia Internet KEying* (MIKEY) (ARKKO et al., 2004) tem a finalidade de descrever os parâmetros necessários para a composição do contexto criptográfico além de implementar proteção a esses atributos, tornando possível sua utilização em canais desprotegidos.

Uma das grandes vantagens do MIKEY é a negociação dos parâmetros criptográficos em uma única troca de mensagens de oferta e aceitação/recusa, possibilitando seu uso com o protocolo SDP, sem grandes modificações nos protocolos de sinalização existentes (ARKKO et al., 2006).

O MIKEY foi desenvolvido para possuir algumas características fundamentais:

- Simples;

- Eficiência (baixo consumo de largura de banda carga computacional, código pequeno, possuir número mínimo de ida e volta dos pacotes);
- Possibilidade de integração com os protocolos de estabelecimento de sessão, como o SDP;
- Independente de funcionalidades de segurança específicas da camada de transporte;
- Segurança fim a fim, garantindo que somente os participantes da sessão terão acesso às chaves geradas.

Em sua especificação original o MIKEY (ARKKO et al., 2004) apresenta três modos para negociação e transporte do contexto criptográfico: MIKEY-PS, MIKEY-PK e MIKEY-DH. Pelo fato de existirem algumas limitações nesses modos foram desenvolvidos o MIKEY-RSA-R e o MIKEY-DHHMAC.

Modo de Transporte Baseado em Chave Compartilhada (MIKEY-PS). A estrutura MIKEY-PS é a mais simples, baseada no transporte de chave compartilhada. Sua principal característica é o acordo de uma única chave simétrica entre os envolvidos na comunicação antes do início da troca de sinalização.

Modo de Transporte Baseado em Chave Pública (MIKEY-PK). O modo de transporte baseado em chave pública tem como principal vantagem proporcionar grande escalabilidade ao sistema VoIP. O MIKEY-PK realiza a negociação dos parâmetros criptográficos através de chaves públicas e privadas.

No modo MIKEY-PK o remetente necessita possuir o certificado do destinatário para poder construir o pacote cifrado para transportar a chave de sessão que será compartilhada, fato que o torna bastante limitado.

Modo de Transporte Baseado em Diffie-Hellman (MIKEY-DH). O MIKEY-DH se baseia no modelo proposto por Diffie-Hellman (DH) para negociação de chaves. O DH é um algoritmo que permite que os envolvidos na comunicação consigam compartilhar, sem a necessidade de mecanismos de proteção, um segredo em um canal desprotegido.

Modo de Transporte Reverso de Chaves Públicas (MIKEY-RSA-R). O modo MIKEY-RSA-R é bastante similar ao MIKEY-PK, a diferença é que no modo MIKEY-RSA-R a chave de sessão é inserida na resposta e não na requisição inicial, ou seja, quem a cria é o destinatário. Isso faz com que não seja necessária a existência de informações compartilhadas antes do início da negociação.

Essa estrutura é dependende de uma arquitetura de chaves públicas, fazendo com que todos os usuários possuam um certificado e uma chave privada assinados por uma CA.

Modo de Transporte Baseado em Diffie-Hellman com autenticação HMAC (MIKEY-DHHMAC). O uso do algoritmo Diffie-Hellman permite que os envolvidos em uma sessão possam realizar uma chamada sem requisitos de infra-estrutura. É necessário o uso de certificados, fazendo assim o uso de chave pública indispensável. Para solucionar esse problema foi desenvolvido em 2006 um novo modo, o MIKEY-DHHMAC (figura 3.5). Esse novo modo consiste também, no uso do algoritmo Diffie-Hellman para o processo de negociação dos parâmetros criptográficos. A mudança se faz na autenticação, onde é usado um algoritmo simétrico e o segredo usado como chave é compartilhado anteriormente entre as partes.

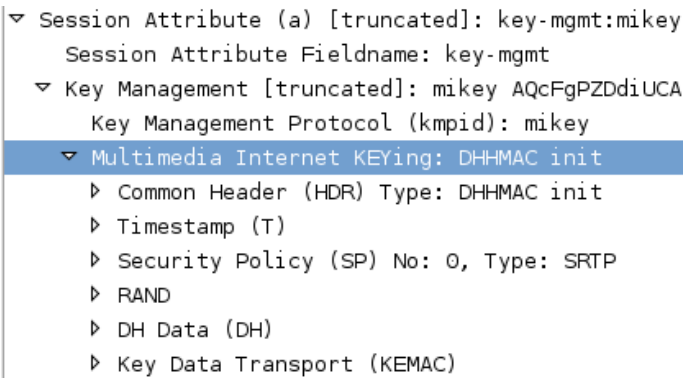

Figura 3.5: Descrição dos Parâmetros Criptográficos - MIKEY

3.5 Criptografia com o Protocolo IAX

O Protocolo *Inter Asterisk eXchange* (IAX) (SPENCER et al., 2009) provê suporte a chamadas usando criptografia de chave simétrica, bloco cifrado *Rijndael* [AES] (também chamado AES - *Advanced Encryption Standard*). *Rijndael* é um bloco cifrado de 128-bits utilizando um segredo compartilhado.

O processo inicia com a mensagem de texto *NEW* indicando, além dos outros parâmetros existentes na mensagem, que a chamada deve ser cifrada. Esta indicação de criptografia é enviada através do *Information Element* (IE) em conjunto com a solicitação da mensagem NEW. O IE é usado para especificar informações de usuários ou das chamadas. Essas mensagens são anexadas a um cabeçalho e podem conter zero, um ou vários elementos de informações.

Caso o usuário chamado possua suporte a criptografia ele responderá com a mensagem AUTHREQ, que vem acompanhada do ENCRYPTION IE. A partir desse momento todas as chamadas subsequentes serão cifradas. Se o usuário chamado não possuir suporte a criptografia, a mensagem AUTHREQ enviada em resposta a mensagem NEW não incluirá o IE ENCRYPTION. Sendo assim o assinante chamado pode encerrar o pedido com a mensagem HANGUP, ou então continuar a chamada sem criptografia.

A chave de cifragem utilizada nas mensagens é calculada tomando o conhecimento do IE do AUTHREQ e concatenando qualquer uma das senhas compartilhadas, em seguida, computando MD5 a 128-bit dessa combinação. Na decifragem, caso haja mais de uma senha para os envolvidos na comunicação, cada chave deve ser julgada até que a mensagem seja decifrada com sucesso. A chave permanece constante durante todo o decorrer da chamada. Apenas os dados das mensagens são cifrados.

4 Ambiente de Testes

Este capítulo consiste na descrição da implementação de mecanismos segurança para chamadas VoIP em ambientes de testes, avaliar o impacto causado e garantir o seu funcionamento.

4.1 Descrição dos Aplicativos Utilizados

A tabela 4.1 mostra os aplicativos utilizados para realização dos cenários de testes e suas respectivas funções.

Aplicativo	**Funcionalidade**
Minisip	*Softphone* para chamadas TLS e SIP
MizuPhone	*Softphone* para chamadas SRTP
Wireshark	*Software* para captura de pacotes
Asterisk	*Software* PBX IP

Tabela 4.1: Aplicativos utilizados nos testes

Softphone é um *software* que possui as funcionalidades de um telefone comum, e é utilizado para realização de chamadas VoIP através da *Internet*. Os seguintes *softphones* foram usados nos testes:

MizuPhone. É um aplicativo desenvolvido para o sistema operacional *Windows* e foi usado com o objetivo de testar as funcionalidades do protocolo SRTP, que provê segurança à mídia.

Minisip. Minisip é um *softphone* desenvolvido para os sistemas operacionais *Linux* e *Windows*. O minisip possui suporte ao TLS, responsável pela segurança da sinalização SIP, e foi usada com o intuito de comprovar a eficácia desse protocolo. Também foi utilizado para testes com chamadas SIP.

O *Wireshark* é um *software* bastante utilizado para análise de tráfego de rede. Ele captura os pacotes e possui a opção de organizá-los por protocolo, facilitando sua análise.

Para complementar os testes e estruturar um cenário completo foi utilizado o sofware Asterisk. Como descrito na seção 2.3, o Asterisk é PBX IP baseado em *software* livre, simples e flexível. Possui suporte aos protocolos de segurança TLS, SRTP e IAX. Para os testes foram utilizadas duas versões do *software* Asterisk: 1.4 e 1.6, conforme apêndice A.

4.2 Descrição dos Cenários

Nesta seção serão descritos com detalhes os cenários utilizados como base de testes deste trabalho. Os cenários serão testados tanto com os protocolos TLS, SRTP e quanto o IAX que não é um protocolo específico para prover segurança, porém provê suporte a tal funcionalidade.

Cenário 1. A figura 4.1 ilustra simplificadamente um dos ambientes utilizados para os testes. Neste cenário é utilizado um computador com o *software* PBX IP Asterisk, e dois *softphones* que estão autenticados nesse PBX. Os *softphones* serão escolhidos conforme o teste a ser realizado.

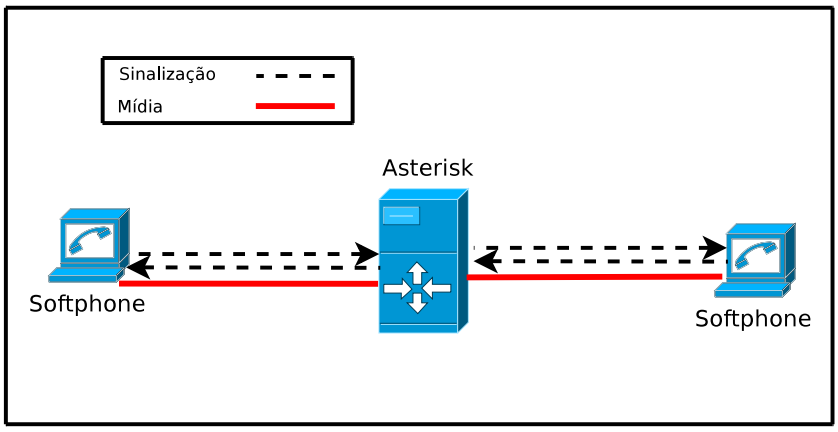

Figura 4.1: Cenário 1

Cenário 2. O cenário 2 é ilustrado na figura 4.2, onde pode-se notar a presença de dois PBXs IP Asterisk. São dois Asterisk interligados via rede IP, e dois *softphones* que estão autenticados um em cada PBX. Os *softphones* serão escolhidos conforme o teste a ser realizado.

4.3 Testes utilizando o Protocolo TLS

Como visto em (seção 3.3) o TLS é um protocolo para proteção da sinalização em uma comunicação de Voz sobre IP. Abaixo serão descritos todos os procedimentos utilizados para a realização dos teste utilizando o protocolo TLS.

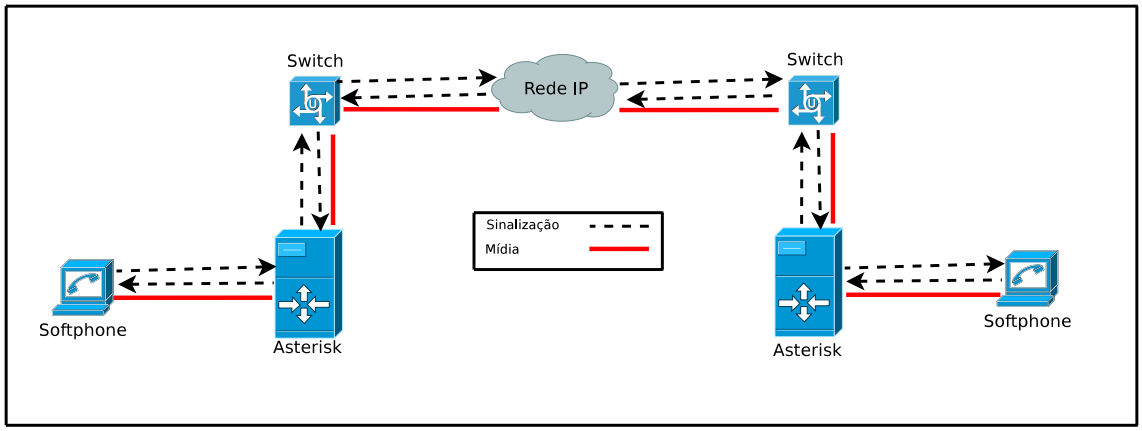

Figura 4.2: Cenário 2

Certificados e Chaves para o TLS

Para assegurar a autenticidade dos participantes de uma sessão SIP, é necessário que se gere os certificados de todos os servidores e usuários do sistema. Se a autenticidade de um deles não for comprovada pelos demais participantes, a conexão não é estabelecida. Há diversas razões para o não estabelecimento da sessão: falta de autenticação do cliente, erro nos certificados, certificados e algoritmos não confiáveis, entre outros. Os certificados gerados serão auto-assinados. Os certificados e chaves foram gerados conforme mostra a figura 4.3.

Para o cenário 1 foram gerados os certificados para os dois *softphones* utilizados e um para o Asterisk. Para o cenário 2 foi necessária a geração dos certificados dos *softphones* e um para cada PBX IP Asterisk, sendo que no Asterisk cliente a opção *Common Name* será preenchida com o endereço IP do Asterisk servidor.

```
1  Gerando certificados para o sip.conf:
2
3  Asterisk:
4  openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout asterisk.pem -out
       asterisk.pem
5
6  Gerando certificados para o Minisip:
7
8  Minisip1:
9  openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout minisip1.key -out
       minisip1.crt
10
11 Minisip2:
12 openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout minisip2.key -out
       minisip2.crt
```

Figura 4.3: Certificados e Chaves TLS

Os parâmetros utilizados para geração de um certificado auto assinado e suas respectivas chaves serão detalhados:

req -x509. Serve para gerar uma requisição de assinatura do tipo X.509, que é um padrão especificado pela ITU-T para infra-estrutura de Chave Pública (*Public Key Infrastructure* - PKI). A X.509 define, entre outras coisas, padrões para Certificados de Chave Pública (*Public Key Certificates*) e o algoritmo que verifica um certificado emitido (Certification Path Validation Algorithm).

-nodes. Utilizado quando não é necessária a geração de senhas para o uso dos certificados.

-days. Define o tempo de validade do certificado.

-newkey rsa:1024. Gera uma nova chave utilizando o algoritmo RSA com um tamanho de 1024 bits.

-keyout. Indica o arquivo em que será armazenada a chave.

-out. Indica o arquivo em que será armazenado o certificado.

As extensões utilizadas são padrão:

- Certificados: *.crt*
- Chaves: *.key*
- Certificados e Chaves combinados: *.pem*

O arquivo *.pem* consiste na combinação do certificado público e da chave privada. O Asterisk provê suporte somente a este tipo de extensão.

Nas linhas 9 e 12 da figura 4.3 podemos notar o uso das extensões *.key* para as chaves e *.crt* para os certificados. Essas extensões foram utilizadas desta maneira pois o *softphone* usado nos testes funcionam somente com arquivos *.crt* e *.key*.

Configuração dos Softphones

Para que o softphone possa se comunicar com o servidor SIP de forma segura, é necessário importar o certificado *asterisk.pem*, gerado conforme mostrado anteriormente na figura 4.3.

Primeiramente devemos habilitar o suporte à TLS no *softphone* escolhido, nesse caso utilizamos o **Minisip**. Clicando no menu *File* e em seguida *Preferences* aparecerá a tela *Minisip - Settings*, como podemos ver na figura 4.4. Na aba *Network* deverá estar habilitada somente a opção TLS.

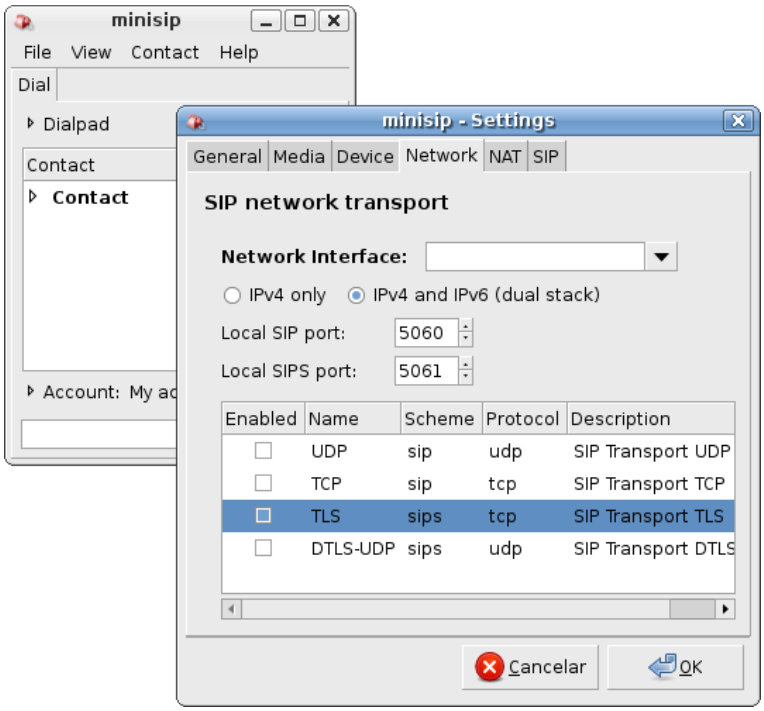

Figura 4.4: Configuração do *Softphone* - Protocolo TLS

Em seguida, conforme mostra a figura 4.5, voltando à tela *Minisip - Settings* na aba *General* (Passo 1) clicamos em *Edit*. Serão habilitadas as opções *Enable certificate based key exchange* e *Enable certificate verification* e clicamos em *Certificate Settings* (Passo 2). Finalmente serão importados os certificados e a chaves criados anteriormente (Passo 3).

Configuração do Asterisk

Após configurar os clientes SIP (*softphone*) foi necessário configurar o Asterisk para que fosse o Servidor SIP e também foi necessário configurar o plano de discagem.

O diretório que contém os arquivos de configuração do Asterisk é o */etc/asterisk*. O arquivo de configuração do protocolo SIP é o */etc/asterisk/sip.conf* e o do plano de discagem é o */etc/asterisk/extensions.conf*.

No arquivo *sip.conf* serão habilitados os parâmetros mostrados na figura 4.6 para que se possa fazer uso do protocolo TLS.

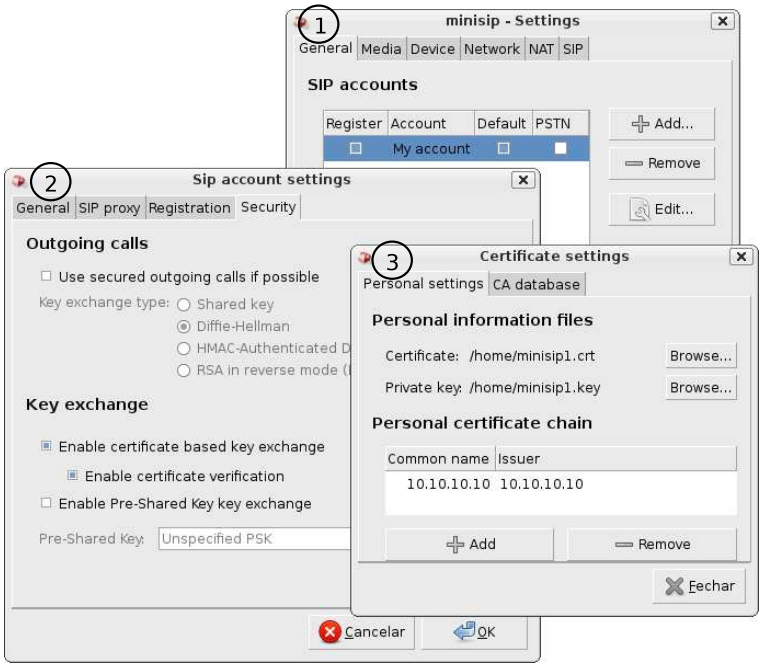

Figura 4.5: Configuração do *Softphone* - Certificados TLS

```
1 tlsenable=[yes|no]
2    Habilita/Desabilita servidor TLS.
3
4 tlscertfile=</path/to/certificate>
5    Indica o caminho para o arquivo do certificado.
```

Figura 4.6: Sip.conf

4.3.1 Resultado dos Testes

Os testes realizados utilizando-se o protocolo TLS para proteção da mídia funcionaram corretamente tanto no cenário 1 como no cenário 2. Na figura 4.7 podemos ver a troca de sinalização para o estabelecimento de uma sessão multimídia baseada na transmissão de voz sobre IP utilizando SIP. Já na figura 4.8 é possível a troca da sinalização de forma segura utilizando o TLS, ou seja, os dados não trafegam em claro pela rede.

Figura 4.7: Troca de Sinalização SIP

Nota-se que as mensagens referentes à sinalização não são explícitas quando feitas através do TLS, pois elas estão cifradas e são transmitidas através do campo *Encrypted Application Data* (figura 4.9), fato que não ocorre quando utilizamos o SIP, em que as mensagens podem ser identificadas facilmente analisando-se o campo *Cseq* (figura 4.10).

Para que o sistema opere funcionando perfeitamente com o uso do TLS é necessário que se utilize *softphones*, telefones IP ou ATA (Adaptador de Telefone Analógico) com suporte a este protocolo. Caso contrário, os resultados não serão obtidos satisfatoriamente. Também é preciso que os certificados sejam gerados corretamente e uma das maneira de fazê-lo é como foi mostrado na figura 3.3.

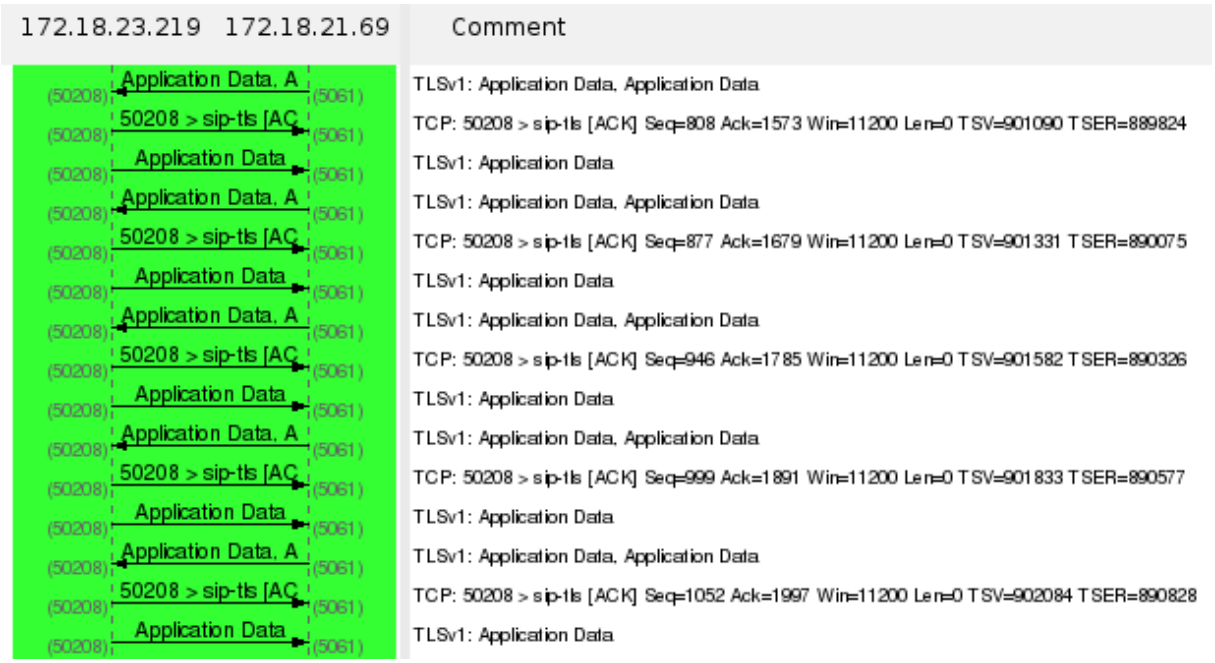

Figura 4.8: Troca de Sinalização TLS

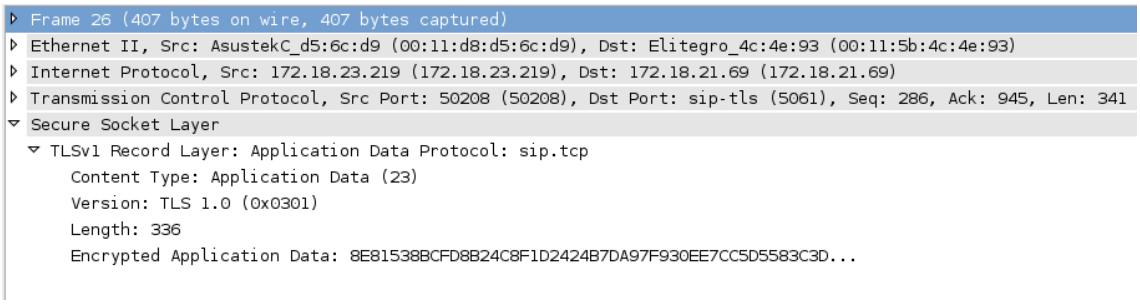

Figura 4.9: Mensagens TLS

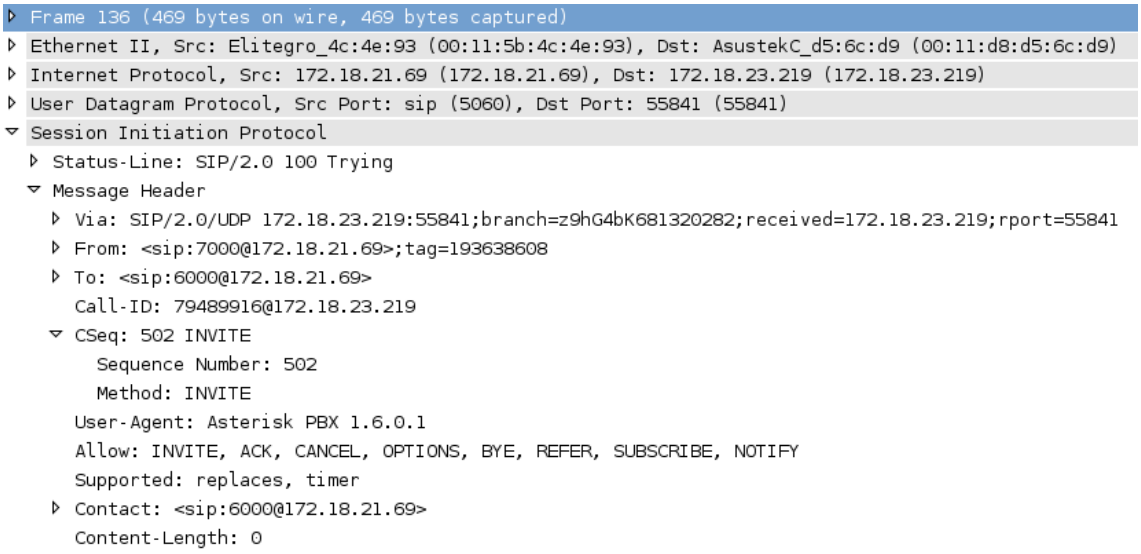

Figura 4.10: Mensagens SIP

4.4 Testes utilizando o Protocolo SRTP

Como visto em (seção 3.4) o SRTP é um protocolo para proteção da mídia em uma comunicação de Voz sobre IP. Abaixo serão descritos todos os procedimentos utilizados para a realização dos teste utilizando o protocolo SRTP.

Configuração dos Softphones

Uma conferência VoIP com a mídia protegida se faz através do uso do protocolo SRTP. Para tal, é necessário que o *softphone* e o PBX IP Asterisk estejam habilitados para realizar essa segurança.

O *softphone* utilizado foi o **Mizuphone**. Ao iniciar o *softphone* devemos clicar na aba *Settings - SIP Settings* e na seção *Security* habilitar a opção *Encrypt Media (SRTP)*.

Configuração do Asterisk

Depois de configurados os *softphones* é necessário habilitar o uso do SRTP no PBX Asterisk. O arquivo a ser editado para que a mídia transmitida possa ser protegida é o *extensions.conf*. Como pode ser visto na figura 4.12, o SRTP pode ser habilitado utilizando dois métodos de negociação MIKEY ou SDES.

O uso do SRTP possui alguns problemas ainda não solucionados, como:

- o método Mikey suporta apenas cifragem obrigatória;
- O usuário chamado não pode forçar o método de criptografia a ser usado;
- o usuário chamador precisa terminar a chamada se o requerimento de cifragem não estiver disponível.

4.4.1 Resultados dos Testes

Cenário 1. este cenário foi testado utilizando-se as duas formas de negociação do SRTP: MIKEY (figura 4.13) e SDES (figura 4.14). Tanto com Mikey quanto com SDES o cenário apresentou problemas.

Quando utilizado o padrão MIKEY somente o usuário chamador consegue enviar o áudio corretamente, já que o usuário chamado apenas ouve e somente os dados transmitidos por

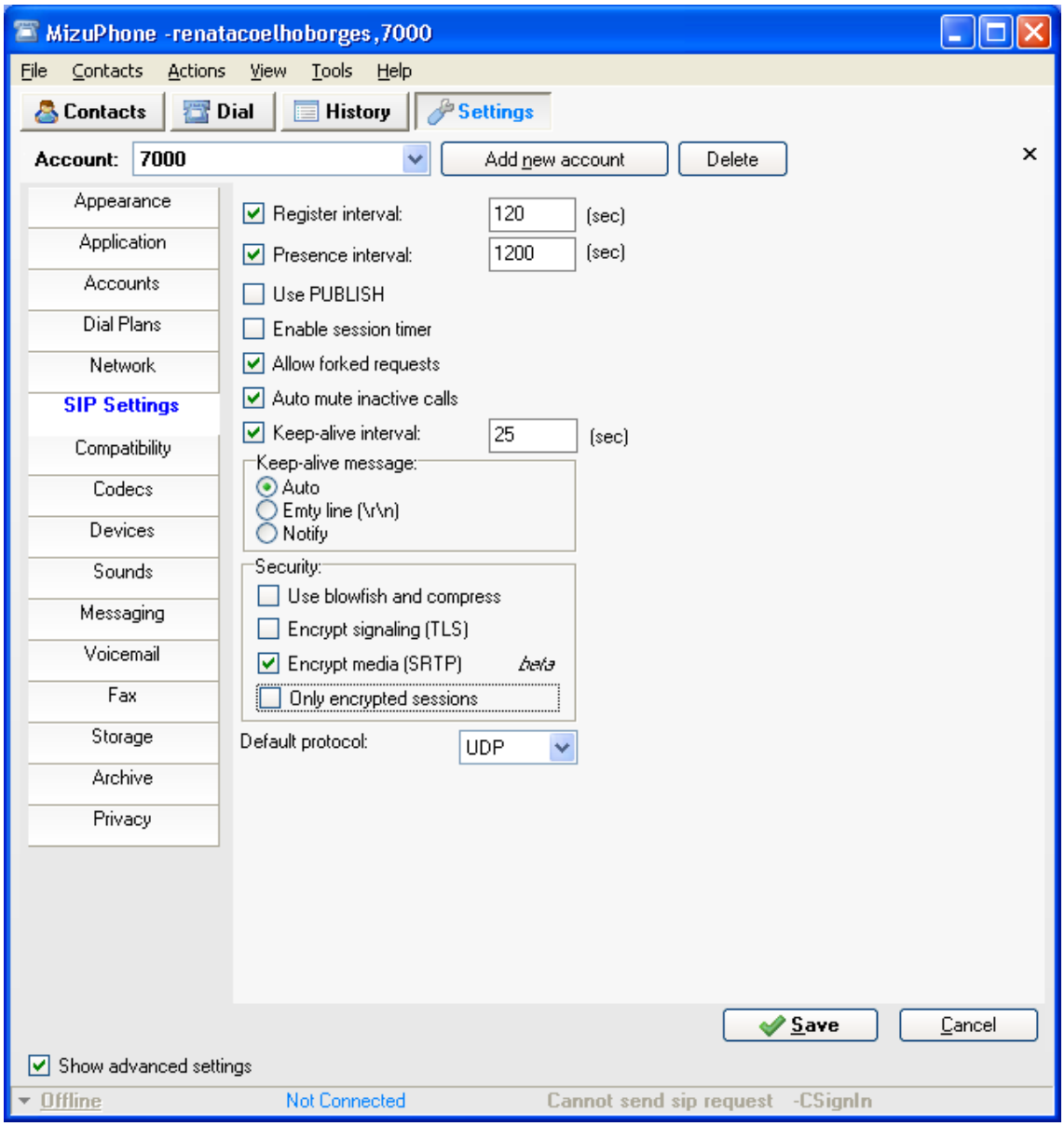

Figura 4.11: Configuração do *Softphone* - SRTP

```
Configurando SRTP no arquivo extensions.conf

SIPSRTP=<any>
Outgoing methods:
SIPSRTP_CRYPTO=disable - Enable/disable sdescriptions
SIPSRTP_MIKEY=disable - Enable/disable MIKEY DH-HMAC

Exemplo

Utilizando SDES
exten => xxx,1,Set(_SIPSRTP=optional)
exten => xxx,n,Set(_SIPSRTP_CRYPTO=enable)
exten => xxx,n,hangup

Utilizando MIKEY
exten => xxx,1,Set(_SIPSRTP=require)
exten => xxx,n,Set(_SIPSRTP_MIKEY=enable)
exten => xxx,n,hangup
```

Figura 4.12: SRTP

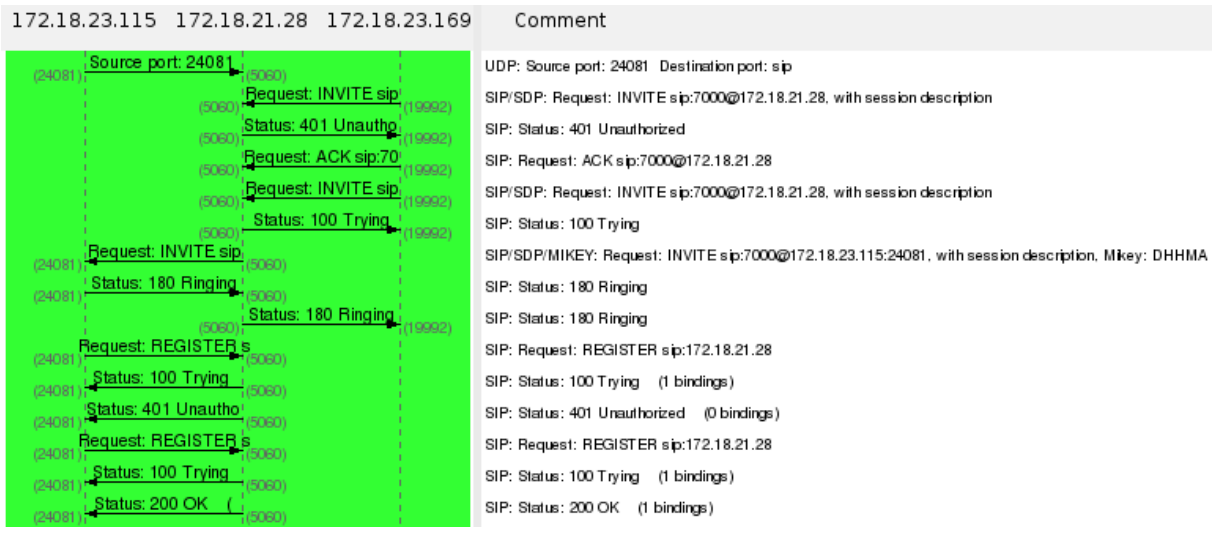

Figura 4.13: SRTP - MIKEY

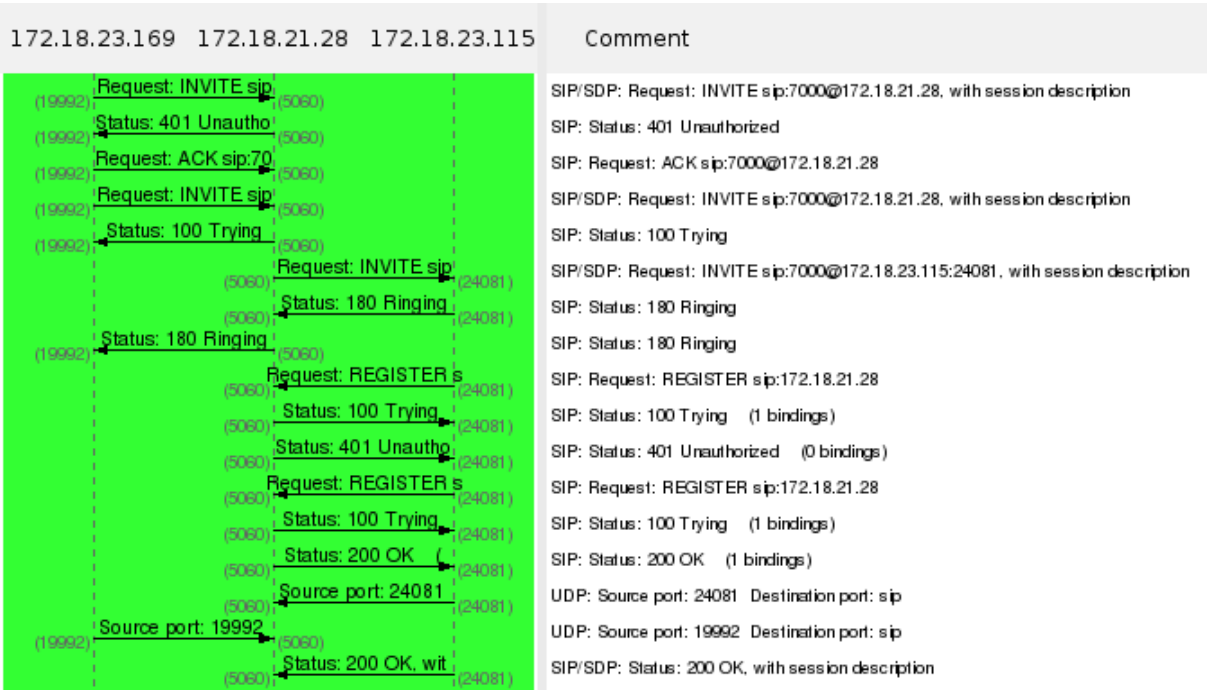

Figura 4.14: SRTP - SDES

ele são cifrados, mas vão somente até o servidor Asterisk e não chegam até o seu destino, conforme pode ser visto na figura 4.15.

Figura 4.15: Fluxo de dados SRTP - MIKEY (Cenário 1)

Já quando utilizamos o padrão SDES, podemos notar na figura 4.16 que os dados enviados pelo usuário chamado são cifrados até chegarem no servidor Asterisk, a partir de então

passam a trafegar em claro. Os dados enviados pelo usuário chamador trafegam todo o tempo em claro sem nenhum tipo de proteção.

Figura 4.16: Fluxo de dados SRTP - SDES (Cenário 1)

Cenário 2. este cenário também apresentou problemas, tanto com o padrão MIKEY quanto com o SDES.

O padrão MIKEY não fez a negociação dos parâmetros criptográficos, e todo o fluxo de mídia trafegou sem proteção. No padrão SDES a negociação foi feita corretamente através do atributo *crypto*, como mostra a figura 4.17.

Figura 4.17: Negociação dos parâmetros criptográficos - SDES (Cenário 2)

Apesar de ter estabelecido os parâmetros criptográficos corretamente o fluxo da mídia

não funcionou perfeitamente. Os dados transmitidos foram protegidos pelo SRTP quando estavam entre os servidores Asterisk, caso contrário trafegam em claro.

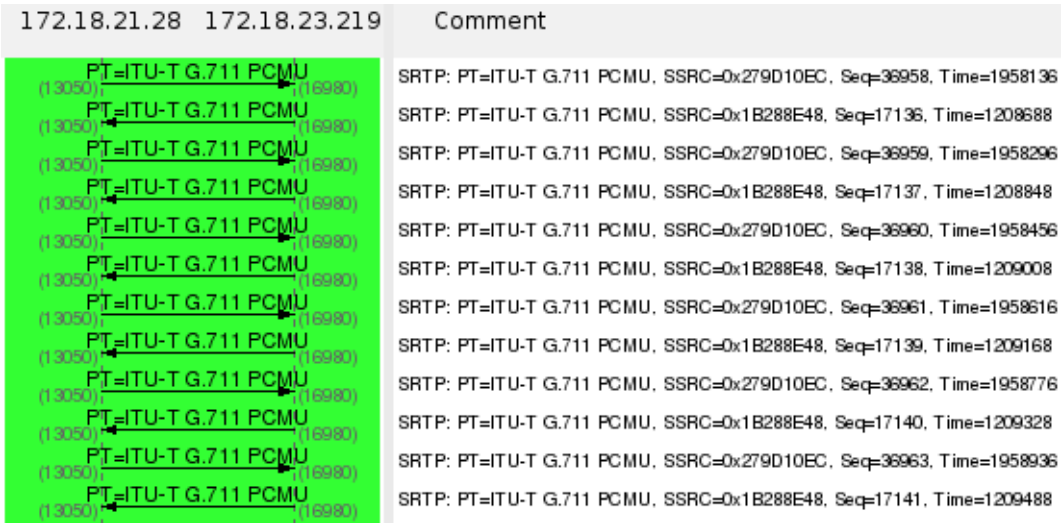

Figura 4.18: Fluxo de dados SRTP - SDES (Cenário 2)

O protocolo SRTP é muito eficaz, mas por estar ainda em fase de desenvolvimento no Asterisk apresenta alguns problemas, os quais deverão ser solucionados com próximas versões desse PBX IP. No entanto, no cenário 2, é possível notar que grande parte do caminho a ser percorrido pela mídia estará protegido, dificultando a ação de atacantes na rede.

Os problemas ocorridos com esse protocolo devem-se principalmente à esse tipo de mecanismo não estar totalmente integrado ao Asterisk. Para a implementação do SRTP é necessário a utilização de um *patch* e compilação de diversas bibliotecas extras. Como esse PBX está em constante desenvolvimento é possível que esses problemas logo sejam resolvidos.

4.5 Testes utilizando o Protocolo IAX

Como visto em (seção 2.2.5) o IAX é um protocolo para trasmissão e controle de mídia utilizado em redes IP. Abaixo serão descritos todos os procedimentos utilizados para a realização dos teste utilizando este protocolo.

Configuração dos Softphones

Para a realização dos testes com o protocolo IAX foi utilizado o *softphone* Minisip. Como nenhum *softphone* possui suporte à segurança opcional do protocolo IAX os testes foram realizados apenas com o cenário 2, utilizando o IAX entre os dois servidores Asterisk.

Primeiramente devemos habilitar o suporte à chamads SIP no *softphone* escolhido, nesse caso utilizamos o **Minisip**. Clicando no menu *File* e em seguida *Preferences* aparecerá a tela *Minisip - Settings*, como podemos ver na figura 4.19. Na aba *Network* deverão estar habilitadas as opção UDP e TCP.

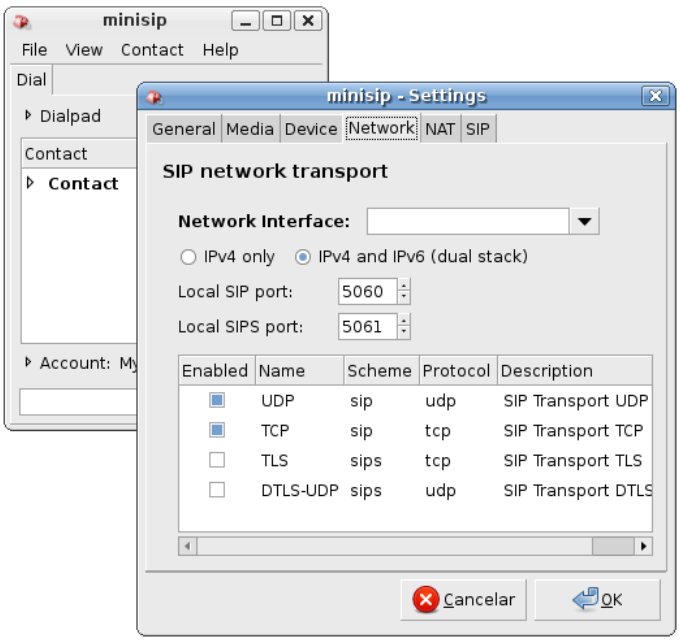

Figura 4.19: Configuração do *softphone* - Protocolo IAX

Configuração do Asterisk

Depois de configurados os *softphones* é necessário habilitar o uso do protocolo IAX nos dois PBX Asterisk. O arquivo a ser editado para que haja segurança na transmissão dos dados transmitidos é o *iax.conf*, como pode ser visto em 4.20.

4.5.1 Resultados dos Testes

A partir dos testes realizados foi comprovada o funcionamento correto da opção de segurança existente no protocolo IAX. Como pode ser visto na figura 4.21 a troca de sinalização entre os dois servidores Asterisk acontece de forma clara e facilmente identificada.

Após terem sido configurados para operarem de forma segura a sinalização entre os servidores aparece cifrada, conforme mostra a figura 4.22. Como dito na seção 3.5, a indicação de criptografia é enviada através do *Information Element* (IE) em conjunto com a solicitação da mensagem NEW (figura 4.23). O usuário chamada responde com a mensagem AUTHREQ

```
Configurando opcao de seguranca no iax.conf

auth=md5
encryption=aes128

Exemplo:

[iax-encryption]
type=friend
host=x.x.x.x
auth=md5
secret=1234
trunk=no
notransfer=no
encryption=aes128
```

Figura 4.20: IAX

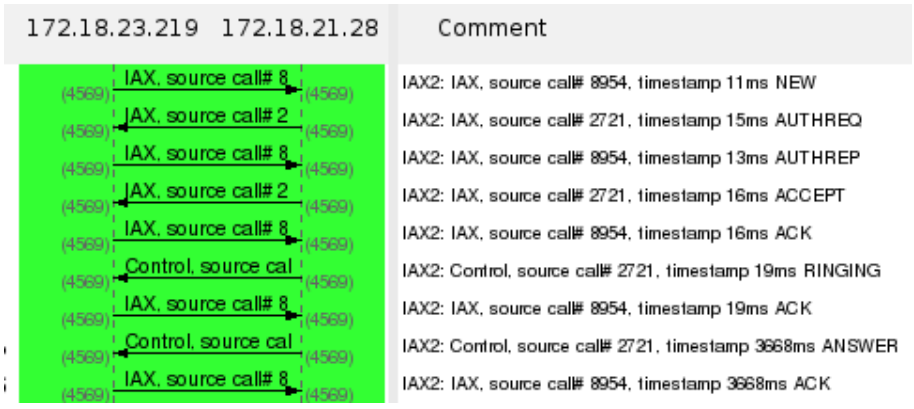

Figura 4.21: Sinalização IAX sem Segurança

acompanhada do ENCRYPTION IE (figura 4.24), a partir de então todas as mensagens serão cifradas.

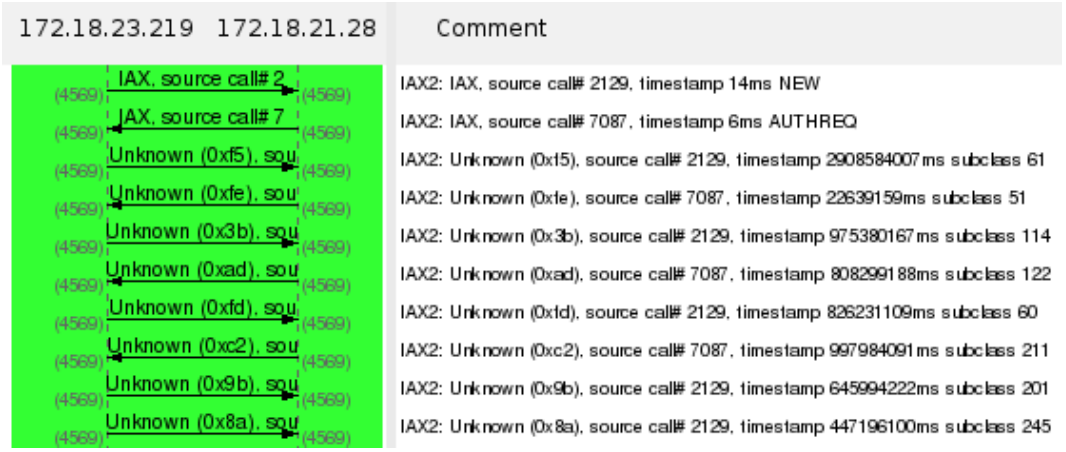

Figura 4.22: Sinalização IAX com Segurança

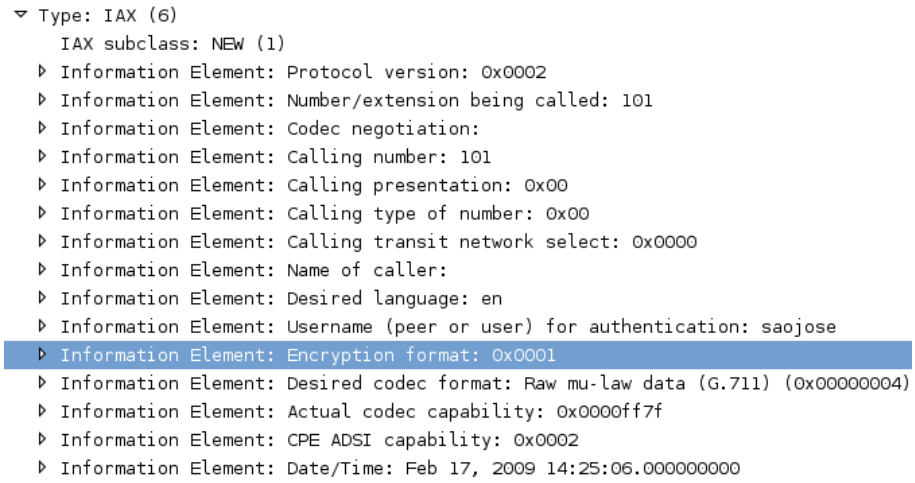

Figura 4.23: Mensagem NEW

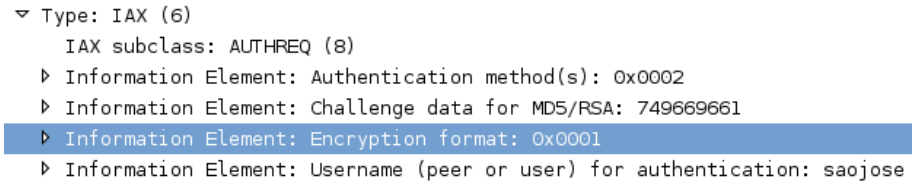

Figura 4.24: Mensagem AUTHREQ

4.6 Conclusões do Capítulo

A partir dos resultados obtidos com os testes realizados observa-se que apenas os protocolos TLS e IAX funcionam conforme o esperado. O protocolo TLS é usado principalmente

para cifragem da negociação do contexto criptográfico necessário ao estabelecimento de uma chamada segura com o uso do SRTP e garantir a segurança da mídia transmitida. No entanto, o protocolo SRTP não funcionou corretamente. Por estar disponível apenas em uma versão espefícida do *software* PBX IP Asterisk e ainda estar em desenvolvimento apresenta alguns erros, fato que dificulta a garantia de segurança. Com o uso do protocolo IAX entre os servidores Asterisk é possível que se tenha resultados satisfatórios, sendo que a sinalização e a mídia foram cifradas, dificultando o acesso de atacantes às mensagens transmitidas.

5 *Conclusões*

O importante fato de não haver uma solução mundialmente aceita para prover segurança em um ambiente VoIP é um fator limitante para o seu crescimento e sua aceitação em cenários em que os requisitos de segurança sejam indispensáveis.

Esse fator foi o maior motivador para o desenvolvido do presente estudo pois, enquanto essa abordagem não for resolvida será muito difícil as indústrias incorporarem essa funcionalidade em larga escala em seus produtos.

Este trabalho foi iniciado introduzindo-se os conceitos de VoIP, as motivações para sua adoção e alguns problemas existentes. Seguido de uma descrição dos protocolos envolvidos no processo de comunicação de Voz sobre IP, com grande ênfase para os protocolos SIP, IAX e RTP, e para o PBX IP Asterisk, utilizado como plataforma principal dos estudos.

Os conceitos de segurança e criptografia possibilitaram uma visão geral de como são tratados esses assuntos e como é possível implementá-los, apresentando na sequência os protocolos de segurança TLS e SRTP, responsáveis pela proteção da sinalização e da mídia, respectivamente. A partir dos estudos feitos partimos para o desenvolvimento de cenários que agregam o PBX IP Asterisk à segurança aplicada no ambiente VoIP, passos que foram descritos no capítulo 4.

Foram propostos dois cenários para testes implementando-se mecanismos de segurança tanto na mídia quanto na sinalização. O processo de instalação e configuração do ambiente foi dificultado pela falta de uma documentação mais detalhada das ferramentas utilizadas, exigindo tempo e esforço para solucionar alguns problemas encontrado ao longo dos testes.

Os testes feitos com o protocolo IAX apresentaram um bom desempenho com relação à cifragem das mensagens. Todo as mensagens de sinalização e mídia trafegam de forma segura quando a opção de segurança do IAX é habilitada.

O *software* Asterisk, em constante desenvolvimento, provê suporte aos protocolos de segurança TLS e SRTP em diferentes versões, o que impossibilitou o uso desses protocolos em

um mesmo cenário. No entanto, os testes foram realizados separadamente e perante os estudos realizados pudemos perceber que é possível o uso desse dois protocolos em conjunto.

O Asterisk, que está na versão 1.6, e ainda não tem previsão para lançar aos usuários uma versão com suporte aos dois protocolos de segurança mais conhecidos (SRTP e TLS), possibilitando estruturar um sistema totalmente seguro. No entanto, o protocolo IAX já provê segurança entre dois servidores, podendo neste caso substituir o SRTP e o TLS.

Apesar de não estar funcionando perfeitamente, o protocolo SRTP, utilizado na versão 1.4 do Asterisk, já melhora consideravelmente em alguns aspectos a segurança nos cenários testados, uma vez que o atacante conseguirá capturar os pacotes em apenas um trecho do caminho percorrido.

Dessa forma, pode-se concluir que a utilização de protocolos de segurança (TLS e SRTP) em ambientes VoIP, aumenta consideravelmente o nível segurança do sistema, evitando ataques, como escuta do tráfego e personificação.

A partir dos conhecimentos obtidos foram analisados alguns pontos que podem ser continuados em outros trabalhos futuros, como:

- Analisar a qualidade de voz em ambientes VoIP que utilizam os mecanismos de segurança estudados nesse trabalho;
- Contribuir para o desenvolvimento do Asterisk possibilitando o uso de TLS e SRTP em uma mesma versão do *software*.

APÊNDICE A – Instalação e configuração do Asterisk

As versões do *software* Asterisk utilizadas foram a 1.6.0.5 (para os testes com TLS e IAX) e a versão 1.4 revisão 81432 (para os teste com SRTP).

Para a instalação do Asterisk 1.6 foi necessário fazer o *download* do código fonte e compilá-lo. Como essa versão do Asterisk será utilizada para os testes com TLS será necessário também fazer a compilação da biblioteca OpenSSL. Seguindo os passos da figura A.1

```
1  Opensssl
2  apt-get install openssl
3
4  Asterisk
5  wget http://www.digium.com/elqNow/elqRedir.htm?ref=http://downloads.digium.com/
       pub/asterisk/releases/asterisk-1.6.0.3.tar.gz
6  tar -vzxf asterisk-1.6.0.3.tar.gz
7  cd asterisk-1.6.0.3
8  ./configure
9  make
10 make install
11 make samples
```

Figura A.1: Asterisk 1.6

Para o Asterisk 1.4, versão utilizada para os testes com SRTP, é necessário fazer o *download* do código fonte e compilá-lo. Além do Asterisk serão necessárias outras bibliotecas. Como será visto na figura A.2.

```
LIBSRTP -- download http://srtp.sourceforge.net/download.html
tar -xzf srtp-tarball
./configure --prefix=/usr
make
make runtest
make install

Libraries
svn co svn://svn.minisip.org/minisip/trunk
cd minisip-trunk

Compiling and installing libmutil
First we run the bootstrap script to generate the configure script.
cd libmutil
libmutil$ ./bootstrap

Now we are ready to compile the source code of libmutil and install it:
libmutil$ ./configure --prefix=/usr
libmutil$ make && make install

Compiling and installing libmnetutil
Configuring, compiling and installing libmnetutil is done the same way as with
    libmutil:
cd ../libmnetutil
libmnetutil$ ./bootstrap
libmnetutil$ ./configure --prefix=/usr
libmnetutil$ make && make install

Compiling and installing libmcrypto
Configuring, compiling and installing libmcrypto is done the same way as with
    libmutil:
cd ../libmcrypto
libmcrypto$ ./bootstrap
libmcrypto$ ./configure --prefix=/usr
libmcrypto$ make && make install

Compiling and installing libmikey
Configuring, compiling and installing libmikey is done the same way as with
    libmutil:
cd ../libmikey
libmikey$ ./bootstrap
libmikey$ ./configure --prefix=/usr
libmikey$ make && make install

Asterisk
svn co -r81432 http://svn.digium.com/svn/asterisk/trunk asterisk-trunk
cd asterisk-trunk
wget "http://bugs.digium.com/file_download.php?file_id=15384&type=bug"
patch -p1 < ast_srtp_r81432_mikey_r3412.patch
./bootstrap
./configure
make && make install
make samples
```

Figura A.2: Asterisk 1.4

Referências Bibliográficas

3CX, S. B. P. for W. Sistema de telefonia - pbx. Http://www.3cx.com.br/voip-sip/sistema-telefonia-pbx.php. 2008.

AGENCY, D. . D. A. R. P. *Transmission Control Protocol (TCP)*. [S.l.], 1981.

AL., M. Baugher et. *The Secure Real-time Transport Protocol (SRTP)*. [S.l.], 2004.

ANDREASEN, F.; BAUGHER, M.; WING, D. *Session Description Protocol (SDP)*. [S.l.], 2006.

ARKKO, J. et al. *MIKEY: Multimedia Internet KEYing*. [S.l.], 2004.

ARKKO, J. et al. *Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)*. [S.l.], 2006.

ASTERISK. The open source pbx. Http://www.asterisk.org/. 2008.

BALBINOT, R. et al. Voz sobre ip - tecnologia e tendências. In: *Simpósio Brasileiro de Redes de Computadores, 2003, Natal. Anais do XXIII Simpósio Brasileiro de Redes de Computadores*. [S.l.: s.n.], 2003.

BLATHERWICK, P.; BELL, R.; HOLLAND, P. *Megaco IP Phone Media Gateway Application Profile*. [S.l.], 2001.

CISCO, S. Call manager denial of service. Http://www.cisco.com/en/US/products/products_security_advisory09186a00805e8a55.shtml. 2009.

COLCHER, S. *VoIP: Voz sobre IP*. [S.l.]: Elsevier, 2005.

CONSORTIUM, I. E. H.323. Http://www.iec.org/online/tutorials/h323/topic01.html. 2001.

DENG, H.; LI, W.; AGRAWAL, D. Routing security in wireless ad hoc networks. *Communications Magazine, IEEE*, v. 40, n. 10, p. 70–75, 2002.

DIERKS, T. *The TLS Protocol*. [S.l.], 2006.

FERGUSON, P.; SENIE, D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. [S.l.]: BCP 38, RFC 2827, May 2000, 1998.

FOCUS, S. Eavesdropping in voip. Http://www.securityfocus.com/print/infocus/1862. 2009.

GOMES, C. F. S.; RIBEIRO, P. C. C. *Gestão da cadeia de suprimentos*. [S.l.]: Cengage Learning, 2004.

GONÇALVES, F. E. A. *Asterisk PBX: Guia de configuração.* [S.l.]: Voffice, 2005.

GPWM, G. de Pesquisa Web e M. Inter asterisk exchange. Http://gpwm.devin.com.br/index.php/Inter-Asterisk_Exchange_(IAX). 2008.

HERSENT, O.; GURLE, D.; PETIT, J.-P. *Telefonia IP: Comunicação multimídia baseada em pacotes*. [S.l.]: Makron Books, 2001.

ISO/IEC. *Iso/iec 13335-1: Management of information and communications technology security*. [S.l.], 2004.

ISO/IEC. *Iso/iec 17799: Código de prática para gestão da segurança da informação*. [S.l.], 2005.

ITU-T. *Packet Based Multimedia Communications Systems - H.323*. [S.l.], 2006.

JENNINGS, C. *Example Call Flows Using SIP Security Mechanisms*. [S.l.], 2004.

LANDWEHR, C. E. Computer security. *International Journal of Information Security*, v. 1, p. 3–13, 2001.

MAHLER, P. *Voip Telephone with Asterisk: a technical overview of the open source PBX*. [S.l.]: Signate, 2004.

OHTA, M. Overload control in a SIP Signaling Network. *Transactions on Engineering, Computing and Technology V*, 2006.

PARZIALE lydia et al. *TCP/IP Tutorial and Technical Overview*. [S.l.]: ibm.com/redbooks, 2006.

PERKINS, C. *RTP: Audio and Video for the Internet*. [S.l.]: Addison Wesley, 2003.

POSTEL, J. *User Datagram Protocol (UDP)*. [S.l.], 1980.

ROSENBERG, J. et al. *Session Initiation Protocol (SIP)*. [S.l.], 2002.

SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. [S.l.], 2003.

SOUZA, J. P. P. d. *sIPtel - Um sistema de IPtel com suporte para vídeo utilizando o protocolo SIP*. Dissertação (Mestrado) — Utad: Universidade de Trás-os-Montes e Alto Douro, 2003.

SPENCER, M. et al. *Inter Asterisk eXchange (IAX) Version 2*. [S.l.], 2009.

STALLINGS, W. *Network Security Essentials. Applications and Standards*. [S.l.]: Prentice-Hall, 2000.

STINSON, D. R. *Cryptography: Theory and Practice*. [S.l.]: CRC Press, 2006.

TELECO. Segurança voip - conceitos e nomenclatura. Http://www.teleco.com.br/tutoriais/tutorialsegvoip/pagina_2.asp. 2009.

TELEGEOGRAPHY. Brazil and nigeria fastest-growing voip destinations. Http://www.telegeography.com/press/releases/2005-12-15.php. 2005.

THERMOS, P. *Securing VoIP Networks: Threats, Vulnerabilities, and Countermeasures*. [S.l.]: Pearson Education, 2007.

TRAPPE, W.; WASHINGTON, L. *Introduction to Cryptography with Coding Theory*. [S.l.]: Prentice Hall, 2001.

VARSHNEY, U. et al. Voice over ip. *Communications of the ACM*, 2002.

WALSH, T.; KUHM, D. Challenges in securing voice over ip. *Security and Privacy Magazine*, v. 3, p. 44–49, 2005.

WIKIPEDIA. Protocolos de internet. Http://pt.wikipedia.org/wiki/Protocolo_de_Internet. 2009.

YORK, D. Suggestions for a 'security roadmap' for asterisk. Http://voipsa.org/blog/2007/10/09/suggestionsforasecurityroadmapforasterisk/. 2008.

ZIMMERMANN, P.; JOHNSTON, A.; CALLAS, J. *ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP*. [S.l.]: Internet Engineering Task Force, 2006.