Universidad de las Ciencias Informáticas

Facultad 2

"Seguridad del Protocolo SIP en la Voz sobre IP"

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores: *Mayrelis Cruz Acosta*

Orlando Sánchez Castro

Tutor: *Ing. Rodney Del Valle Torres*

Consultante: *David Pérez de la Llera*

Ciudad de La Habana, Junio de 2008

"El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres."

Steve Job

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo, por lo que autorizamos a la Facultad No. 2 de la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los __ días del mes de _____ del _____.

_____ _____

Firma de la Autora

(Mayrelis Cruz Acosta)

Firma del Autor

(Orlando Sánchez Castro)

_____

Firma del Tutor

(Ing. Rodney Del Valle Torres)

DATOS DE CONTACTO

oscastro@estudiantes.uci.cu

mcacosta@estudiantes.uci.cu

rodneyvt@uci.cu

AGRADECIMIENTOS

Mayrelis y Orlando

Queremos agradecer en primer lugar a nuestro invencible Comandante en Jefe Fidel Castro Ruz, por ser nuestro guía.

A nuestra Revolución por darnos la posibilidad de graduarnos como Ingenieros de las Ciencias Informáticas.

A nuestro tutor Rodney por su dedicación y por guiarnos de manera excepcional en la realización de este trabajo de diploma.

A los compañeros David y Erkins de Miramar Try Center por toda la ayuda que nos brindaron en la investigación y desarrollo de esta tesis.

A nuestros compañeros del grupo 2504, y a los que no son del grupo que de una forma u otra nos apoyaron o nos aclararon dudas importantes para lograr el objetivo de este trabajo.

Mayrelis

Al llegar a esta etapa de mi vida siento la satisfacción de haber logrado uno de mis más anhelados sueños, verme graduada universitaria. Incontables han sido los sacrificios y las alegrías que he experimentado en el decursar de estos cinco años. También han sido muchas las personas que han contribuido con la realización de este sueño, a ellas quiero agradecer a través de estas pequeñas palabras:

Mi primer y nunca suficiente agradecimiento es para mis padres Mayda y Carlos quienes han dedicado su vida y esfuerzo a mi formación personal y profesional, a quienes debo todo lo que soy. Gracias a los dos por enseñarme el camino a recorrer, por su amor incondicional, su cariño y por vivir para mi. Los quiero mucho. Agradezco a mi hermanita, por su amor y cariño, por todo lo que hemos pasado juntas, buenos y malos momentos y porque ahora nuestras peleas de niños se convirtieron en nuestra amistad.

A toda mi familia, por sus contantes preocupaciones, por contribuir en mi crecimiento personal y por ser parte importante en mi vida. Mis abuelos, los que están y los que no, por ser tan comprensivos y buenos conmigo, en especial a mi abuelita Paula por su amor infinito. Mis tíos, especialmente Elsa, Gloria, Amarilys por todo el cariño que me han demostrado siempre, por su ayuda tanto en lo personal como en lo profesional y por permitirme contar con ellas siempre. A mis primos, a los grandes y chicos

A mis amigos, a los que quiero mucho y forman parte de muchos recuerdos lindos, los que siempre me han brindado su mano y han compartido conmigo momentos de tristeza y alegría, a Mariosky y a Laury mis amigas de la infancia, por brindarme su amistad sincera y pura que sobrepasa las fronteras de la distancia, a Sulanis mi compañera de cuarto durante los dos últimos años, a Noisy por ser tan especial conmigo y porque siempre que la necesito está ahí para brindarme su ayuda. Al resto de mis amigos también les agradezco hacer de cada día junto a ellos una evocación inolvidable.

A todos, incluso aquellos que no mencione porque están de alguna manera reflejados en las líneas anteriores, simplemente gracias

Orlando

Como hijo, como familiar, como amigo, siempre tendré presente a personas con las cuales he compartido y vivido momentos de los cuales nunca me arrepentiré. Momentos que de una forma u otra influyeron en mi vida tanto laboral, estudiantil como personal, por esto y mucho más los quiero y les agradezco todo lo que me han brindado. Gracias una y mil veces más pues realizaré una de mis mayores metas: la de graduarme y ser un profesional en el país donde nací y crecí. Por este logró y por estar siempre presentes en mi vida le agradezco:

A mi madre Lolita por todas las preocupaciones que le he dado desde el primer día que vine al mundo, además de ayuda, apoyo, consejos y la dedicación que siempre tuve de su parte, no solo para la realización de este trabajo sino en toda mi vida, te agradezco esto y mucho más madre mía, te quiero mucho sin dudarlo, sin pensarlo, solo te quiero.

A mi padre Orlando por toda su ayuda y preocupación además de su apoyo tanto en el plano personal como en el estudiantil, gracias por guiarme como y aconsejarme en los momentos difíciles de mi vida.

A mi padrastro Félix al cual quiero como un padre, ya que siempre estuviste presente para cualquier necesidad, además de enseñarme muchas cosas que me han sido de gran ayuda para mi vida cotidiana.

Agradezco a mi abuela Lola por la crianza que me dio desde que era niño, y por su ayuda, apoyo y dedicación que siempre tuve de su parte, para mi formación como profesional.

A mi abuelo Manolo por su ayuda y guía en la realización de este trabajo, y en mi vida estudiantil.

A toda mi familia por su apoyo y dedicación que me han brindado desde niño.

A mis amigos de La Habana como de la escuela por la ayuda y apoyo, que me dieron para poder realizar este trabajo. Con ustedes compartí momentos tristes, alegres, divertidos, los aprecio cantidad y muchas gracias por todo lo que vivimos juntos.

Mayrelis

A mis padres por depositar toda su confianza en mí,

por ser mí apoyo, mi soporte, mi inspiración,

las personas más importante en mi vida.

Orlando

A mis padres por siempre estar ahí cuando los necesitaba.

A mi familia que siempre me apoyaron.

A mi abuela Lola por la crianza que me dio.

A mi hijita Danelis que será lo mejor de mí.

A mi abuelo Orlando y mi bisabuela María, siempre los recordaré y querré mucho.

RESUMEN

El presente trabajo se realizó en la Universidad de las Ciencias Informáticas (UCI), centro que posee una amplia infraestructura de red y en el cual se quiere implantar la tecnología Voz sobre IP (VoIP) utilizando el Protocolo de Inicio de Sesión (SIP) para la señalización de las llamadas.

Este trabajo de diploma tiene como objetivo mejorar la seguridad del protocolo SIP para la VoIP. Para esto se presentan dos propuestas que tienen como línea fundamental el cifrado de este protocolo.

En el trabajo se realiza una recopilación de información referente a la tecnología VoIP, profundizando en las características del protocolo SIP y los principales ataques a los que está expuesto. También se realiza un estudio de algunos algoritmos criptográficos que son implementados en la actualidad con el objetivo de garantizar la seguridad de la información. Se exponen además las principales características, propósitos, ventajas y desventajas que se obtienen con la implementación de estas propuestas, así como los procedimientos a tener en cuenta para realizarse.

PALABRAS CLAVES

Protocolo

Señalización

Seguridad

INDICE

INTRODUCCIÓN

Desde la antigüedad el hombre ha tenido la necesidad de comunicarse con sus semejantes y para ello ha utilizado diferentes medios de comunicación, que con el transcurso del tiempo han ido evolucionando. El surgimiento del teléfono fue uno de los grandes avances de las comunicaciones. Con el uso de este y de las redes telefónicas públicas conmutadas (*PSTN*), el hombre pudo hacer posible la comunicación a largas distancias. Hace algunos años, gracias a la evolución tecnológica, además del amplio despliegue de las redes de acceso de banda ancha, sumada a la optimización de los mecanismos de compresión y transmisión, han surgido nuevas tecnologías. Estas brindan la posibilidad de ofertar servicios de voz y video en tiempo real a través de las redes de *conmutación* de paquetes.

La VoIP es una de las tecnologías que está actualmente disponible en el mundo que consiste en la transmisión del tráfico de voz y video sobre una red *IP*. Esta permite que las llamadas telefónicas sean soportadas sobre la redes de datos existentes en lugar de utilizar líneas telefónicas tradicionales, lo cual ofrece muchos beneficios a los proveedores de servicios así como a los usuarios finales.

Muchas son las empresas que están dispuestas a adoptar la VoIP debido al ahorro que representa, siendo irrelevante la distancia y duración de las llamadas desde el punto de vista de los costos. Además ofrece mayor flexibilidad y movilidad a los usuarios, unificando su estructura de comunicación. También brinda la posibilidad de añadir nuevos servicios y funciones no disponibles con el servicio telefónico tradicional.

La VoIP utiliza diferentes protocolos para la señalización, siendo SIP uno de los más utilizados actualmente por las ventajas en cuanto a dispositivos y servicios que este ofrece con respecto a los demás protocolos existentes. Usando SIP es posible implementar servicios telefónicos básicos y avanzados, además de soportar comunicaciones entre usuarios de redes *IP* y, con el empleo de pasarelas, entre usuarios de otras redes, incluyendo terminales de las *PSTN*.

La UCI cuenta con una infraestructura telefónica que le brinda servicios a toda la comunidad universitaria. Actualmente debido al incremento de la matrícula, han ido creciendo los locales de residencia, docencia, producción y servicios, que precisan de los servicios telefónicos, razón por la cual se ha ido incrementando el número de abonados.

Aprovechando la infraestructura de red existente en la universidad se quiere implantar la tecnología VoIP utilizando el protocolo SIP para la señalización de las llamadas como alternativa a la PSTN existente. El problema está en que a pesar de las ventajas que este protocolo ofrece, hay que estar consciente de los riesgos que conlleva su uso, ya que posee grandes vulnerabilidades en cuanto a su seguridad, por lo que está propenso a diferentes ataques. Si bien este problema en la actualidad no preocupa demasiado al usuario, garantizar la seguridad en el entorno VoIP es clave.

De acuerdo a lo planteado anteriormente se determinó que el **problema a resolver** es la necesidad de mejorar la seguridad del protocolo de señalización SIP.

Enmarcado en el **objeto de estudio** la seguridad en la VoIP, teniendo como **campo de acción** los mecanismos de seguridad para el protocolo de señalización SIP.

Para dar solución al problema se plantea como **objetivo general** proponer un mecanismo de seguridad para el protocolo de señalización SIP, definiéndose como **pregunta científica** ¿cómo lograr una mejora en la seguridad del protocolo de señalización SIP?

Para dar cumplimiento al objetivo general de la investigación se llevaron a cabo las siguientes **tareas**:

- Realización de un estudio de la VoIP.
- Realización de un estudio de los principales ataques a la VoIP.
- Realización de un estudio de la seguridad en la VoIP.
- Realización de un estudio del protocolo SIP.
- Realización de un estudio de los diferentes tipos de ataques al protocolo SIP.
- Realización de un estudio de los mecanismos de seguridad existentes para el protocolo SIP.
- Realización de un estudio de los algoritmos criptográficos que se empleen para llevar a cabo la seguridad en la VoIP.
- Realización de un análisis de las diferentes variantes posibles para garantizar la seguridad del protocolo SIP.

El contenido del trabajo está dividido por capítulos, de la forma siguiente:

- **Capitulo 1 Fundamentación Teórica:** Se abordaron temas relacionados con la tecnología VoIP. Se exponen también los elementos, características y funcionalidad de esta tecnología, así como los principales protocolos que soporta, profundizándose en su seguridad y en las vulnerabilidades que esta presenta.

- **Capitulo 2 Análisis General del Protocolo SIP:** Se analiza el protocolo SIP como protocolo de señalización para la VoIP, mostrando sus principales características, entidades, así como su funcionamiento. En el mismo se exponen los principales mecanismos de seguridad que existen para este protocolo, además de los ataques a los cuales está expuesto. Además se analizan varios algoritmos criptográficos que se utilizan para garantizar la seguridad de la información.

- **Capitulo 3 Propuesta de Seguridad para el Protocolo SIP:** Se propone el mecanismo de seguridad para el protocolo SIP, quedando definidas las ventajas y desventajas que se obtendrían una vez implantado. Se muestran también las estrategias utilizadas con el fin de lograr este objetivo.

CAPITULO 1: FUNDAMENTACION TEORICA

1.1. Introducción

La tecnología VoIP permite encapsular la voz en paquetes para poder ser transportados sobre redes de datos sin necesidad de disponer de una infraestructura telefónica convencional, lográndose así una única red homogénea en la que se envía todo tipo de información ya sea voz o video. Esta tecnología permite la realización de llamadas telefónicas a cualquier lugar del mundo, tanto a números VoIP como a personas que poseen equipos con números telefónicos fijos o móviles. Ofrece además nuevos servicios que seguirán aumentando con el desarrollo de nuevas y novedosas tecnologías.

En este capítulo se abordarán diversos aspectos de esta tecnología, como son: funcionalidad, componentes, principales protocolos que soporta, ventajas que proporciona, además de los diferentes ataques que se le realizan a su integridad y algunas consideraciones finales a tener en cuenta para lograr una mejor seguridad.

1.2. Elementos fundamentales de la VoIP

El estándar de VoIP se puede definir con tres elementos fundamentales en su estructura, como se muestra en la figura 1.1.

Figura 1.1 Componentes de la VoIP.

A continuación se describen los elementos principales mostrados en la figura anterior.

- **El cliente:** Es el que establece y termina las llamadas de voz, además de que codifica, empaqueta y transmite la información de salida generada por el micrófono del usuario. Asimismo, recibe, decodifica y reproduce la información de voz de entrada a través de los altavoces o audífonos del usuario. Cabe destacar que el elemento cliente se presenta en dos formas básicas: la primera es una suite de software corriendo en una PC que el usuario controla mediante una interfaz gráfica (*GUI*); y la segunda puede ser un cliente virtual que reside en el gateway.

- **Servidores:** Manejan un amplio rango de operaciones complejas de bases de datos, tanto en tiempo real, como fuera de él. Estas operaciones incluyen: validación de usuarios, tasación, contabilidad, tarificación, recolección, distribución de utilidades, enrutamiento, administración general del servicio, carga de clientes, control del servicio, registro de usuarios y servicios de directorio entre otros.

- **Gateway*:*** Es el que proporciona un puente de comunicación entre los usuarios. La función principal de este, es proveer las interfaces con la telefonía tradicional apropiada, funcionando como una plataforma para los clientes virtuales.

1.3. Principales ventajas de la VoIP

El uso de la VoIP trae consigo grandes ventajas para los usuarios y las empresas que la utilizan, algunas de ellas son:

- Hacer uso de transmisión de la voz a través de una red *IP*, que aporta muchas ventajas respecto al servicio de telefonía tradicional.

- Costo de Tarificación y Mantenimiento. En general, el servicio de telefonía vía *IP* presenta un costo de mantenimiento y de tarificación muy inferior al servicio equivalente tradicional que ofrecen los proveedores de servicio de la *PSTN*.

- Generalmente la capacidad de la red de datos *IP* no suele ser aprovechada en su totalidad, por lo que se puede hacer uso del resto de los recursos para transportar voz en la misma red sin ningún costo adicional.

- Las mejoras de Funcionalidad. Esto permite la integración con otros servicios basados en redes *IP*, incluyendo video-llamadas e intercambio de datos y mensajes con otros servicios en paralelo con la conversación.

- Mediante los paquetes de VoIP se puede obtener servicios extra por los que la red telefónica conmutada tradicional cobra un cargo extra, o servicios que no se encuentran disponibles en algunos países, como son: las llamadas multiconferencia, retorno de llamada, identificación de llamadas, etcétera.

- Las mejoras de Movilidad. Los usuarios VoIP pueden viajar a cualquier lugar en el mundo y seguir realizando y recibiendo llamadas de manera transparente y sin costo adicional mediante un identificador universal de usuario.

- La facilidad de tareas que serían más difíciles de realizar usando las redes telefónicas comunes de la *PSTN*.

- Las herramientas o vías como los puertos físicos de circuitos conmutados para servicios como el correo de voz, ya no son necesarios, el servidor de correo solo necesita tener conexión *IP*.

- Permite que los sistemas de correo de la voz se coloquen en plataformas basadas en estándares (como PC y maquinas *UNIX*).

1.4. Protocolos de señalización para la VoIP

Las necesidades de calidad de servicio hacen que sea necesaria una gestión de recursos que asegure la optimización de la capacidad de transporte de la voz extremo a extremo, para ello surgen los protocolos de señalización. Estos protocolos permiten efectuar operaciones de supervisión, negociación, establecimiento de llamadas, gestión y mantenimiento de la red.

Para cumplir los requerimientos de señalización, la VoIP engloba los principales protocolos:

- **H.323:** Protocolo definido por la *ITU-T*, con el objetivo de proveer un mecanismo para el transporte de aplicaciones multimedia sobre redes, donde no se garantiza la calidad de servicio.

- **SIP:** Protocolo de nivel de aplicación desarrollado por el *IETF* con el objetivo de ser el estándar para la inicialización, modificación y finalización de sesiones interactivas de usuario en las que intervienen elementos multimedia.

- **MGCP:** Protocolo de Control de Medios, es otro estándar de señalización para VoIP desarrollado por la IETF. está basado en un modelo maestro/esclavo donde el Call Agent (servidor) es el encargado de controlar al gateway. De esta forma se consigue separar la señalización de la transmisión de la información.

- **IAX:** Protocolo de Intercambio Inter-Asterisk, fue desarrollado por la empresa Digium para la comunicación entre centralitas basadas en *Asterisk*. El principal objetivo de este protocolo es minimizar el ancho de banda utilizado en la transmisión de voz y vídeo a través de la red IP y proveer un soporte nativo para ser transparente a los *NATs*. El protocolo original ha quedado obsoleto en favor de su segunda versión conocida como IAX2. Permite manejar una gran cantidad de códecs y transportar cualquier tipo de datos.

1.5. Funcionamiento de la VoIP

La VoIP digitaliza la voz en paquetes de datos, las envía a través de la red y la reconvierte a voz en el destino. Básicamente el proceso comienza con la señal análoga del teléfono, que es digitalizada en señales *PCM* por medio del codificador/decodificador de voz (*codec*). Las muestras *PCM* son pasadas al algoritmo de compresión, el cual comprime la voz y la fracciona en paquetes que pueden ser transmitidos para este caso, a través de una red privada *WAN*. En el otro extremo se realizan exactamente las mismas funciones en un orden inverso. El flujo de un circuito de voz comprimido es el mostrado en la figura 1.2. [1]

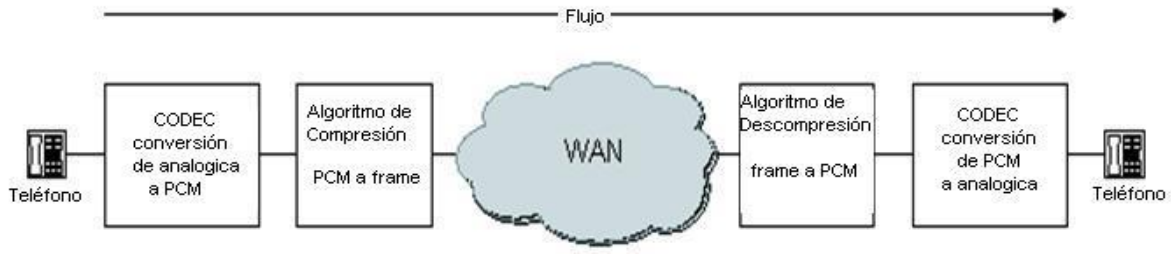


Figura 1.2 Flujo de un circuito de voz comprimida.

Dependiendo de la forma en la que la red este configurada, el enrutador o el gateway puede realizar la labor de codificación, decodificación y/o compresión. Por ejemplo, si el sistema usado es un sistema análogo de voz, entonces el enrutador o el gateway realizan todas las funciones mencionadas anteriormente como se muestra en la figura 1.3. [1]

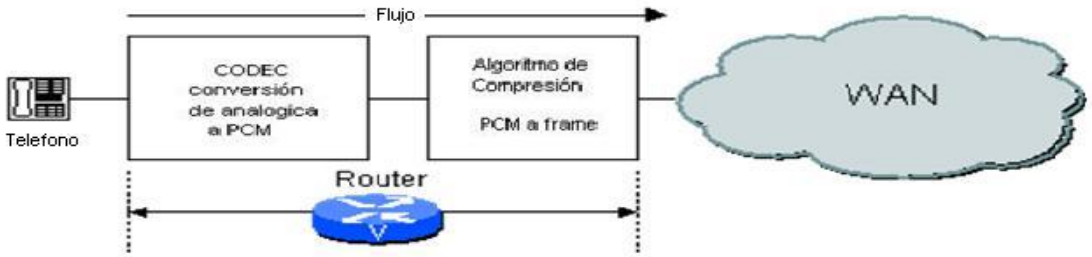


Figura 1.3 Muestra utilizando un sistema análogo de voz.

Si, por otro lado, el dispositivo utilizado es una *PBX*, es entonces este el que realiza la función de codificación y decodificación, y el enrutador solo se dedica a procesar las muestras *PCM* que le ha enviado la *PBX*, como muestra la figura 1.4. [1]

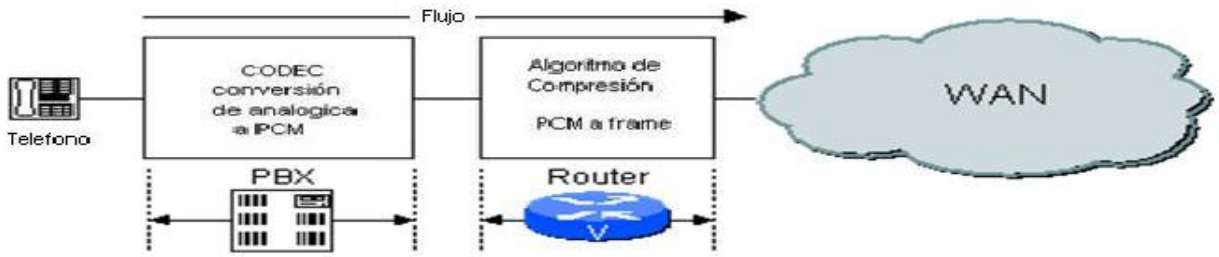


Figura 1.4 Muestra utilizando *PBX* digital.

Para el caso en el que el transporte de la voz, se realiza sobre la red pública Internet, se necesita una interfaz entre la red telefónica y la red *IP*, el cual se denomina gateway y es el encargado en el lado del emisor de convertir la señal analógica de voz en paquetes comprimidos *IP* para ser transportados a través de la red. Del lado del receptor su labor es inversa, dado que descomprime los paquetes *IP* que recibe de la red de datos, y recompone el mensaje a su forma análoga original conduciéndolo de nuevo a la *PSTN*, para ser transportado al destinatario final y ser reproducido por el parlante del receptor. [1]

1.6. Seguridad en la VoIP

VoIP es una tecnología que ha de apoyarse necesariamente en muchas otras capas y protocolos ya existentes de las redes de datos. Por eso en cierto modo va a heredar ciertos problemas de las capas y protocolos ya existentes, siendo algunas de las amenazas más importantes de VoIP los problemas clásicos de seguridad que afectan al mundo de las redes de datos. Por supuesto, existen también multitud de ataques específicos de VoIP que serán vistos próximamente.

La seguridad de la VoIP se construye sobre muchas otras capas tradicionales de seguridad de la información como se muestra en la figura 1.5.

Figura1.5 Representación gráfica de las capas tradicionales de seguridad

En la tabla 1.1 se mencionan algunos de los puntos débiles y ataques que afectan a cada una de las capas. [2]

Capa	Ataques y vulnerabilidades
Políticas y Procedimientos	Contraseñas débiles. Ej: Contraseña del VoiceMail Mala política de privilegios Accesos permisivos a datos comprometidos.
Seguridad Física	Acceso físico a dispositivos sensibles. Ej: Acceso físico al un gatekeeper. Reinicio de máquinas.
Seguridad de Red	DDoS ICMP unreacheable SYN floods Gran variedad de floods
Seguridad en los Servicios	SQL injections Denegación en DHCP DoS
Seguridad en el S.O.	Buffer overflows Gusanos y virus Malas configuraciones.
Seguridad en las Aplicaciones y protocolos de VoIP	Fraudes SPIT (SPAM) Vishing (Phising) Fuzzing Floods (INVITE,REGISTER,etc..) Secuestro de sesiones (Hijacking) Interceptación (Eavesdroping) Redirección de llamadas (CALL redirection) Reproducción de llamadas (CALL replay)

Tabla 1.1 Ataques y vulnerabilidades que afectan a cada una de las capas.

1.6.1. Clasificación de los ataques

Durante los siguientes apartados se va a detallar cuales son los ataques más significativos que afectan a la telefonía sobre redes *IP*, según sus clasificaciones. Se mostraran también, ciertas vulnerabilidades que afectan específicamente a las redes VoIP y a los protocolos que soporta.

- **Accesos desautorizados y fraudes**

Los sistemas VoIP incluyen múltiples sistemas para el control de la llamada, administración, facturación y otras funciones telefónicas. Cada uno de estos sistemas debe contener datos que si son comprometidos, pueden ser utilizados para realizar fraudes. El costo de usar fraudulentamente esos datos VoIP a nivel empresarial pueden ser devastadores. El acceso a los datos telefónicos (de facturación, registros, datos de cuentas, etc.) pueden ser usados con fines fraudulentos. [2]

- **Explotando la red subyacente**

Paradójicamente una de las principales debilidades de la tecnología VoIP es apoyarse sobre una red potencialmente insegura como son las redes *IP*. Puesto que gran cantidad de ataques hacia las infraestructuras *IP* van a afectar irremediablemente a la telefonía, como son ataques de denegación de servicio, inundación de paquetes o cualquier otro tipo de ataque que intente limitar la disponibilidad de la red suponen un gran problema para la telefonía *IP*. Además la VoIP será vulnerable a ataques de bajo nivel como son el secuestro de sesiones, intercepción de paquetes, fragmentación *IP*, paquetes *IP* malformados y suplantación de identidad, más conocido como spoofing. Uno de los mayores problemas es la interceptación o eavesdropping, término por el que se conoce a la captura de información (cifrada o no) por parte de un intruso al que no iba dirigida dicha información. [2]

- **Ataques de denegación de servicio**

Los ataques de denegación de servicio son intentos malintencionados de degradar seriamente el rendimiento de la red, o un sistema, incluso llegando al punto de impedir la utilización del mismo por parte de usuarios legítimos. Algunas técnicas se basan en el envío de paquetes especialmente construidos para explotar alguna vulnerabilidad en el software o en el hardware del sistema, saturación de los flujos de datos y de la red o sobrecarga de procesos en los dispositivos. [2]

Llegan a ser especialmente dañinos los llamados ataques de denegación distribuidos (DDoS). Son ataques DoS simples pero realizados desde múltiples computadores de forma coordinada. Las redes y sistemas VoIP son especialmente vulnerables a los DDoS por diversas razones:

- La dependencia y la necesidad de tener garantías, en la calidad de servicio, hacen que las redes *IP* tengan una tolerancia mucho menor a problemas de rendimiento donde se mantengan llamadas telefónicas.

- En una red VoIP existen varios dispositivos con funciones muy específicas, por lo que un ataque contra cualquier dispositivo de la red, puede afectar seriamente los servicios de telefonía *IP*. Muchos de estos dispositivos, son muy susceptibles de no manejar, priorizar o enrrutar el tráfico de forma fiable, si presenta un consumo alto de *CPU*. Por lo que muchos de los ataques de DoS, se centran en atacar los dispositivos de red e/o

inundar la red de tráfico inservible, para degradar su funcionamiento y que los paquetes pertenecientes a comunicaciones telefónicas se pierdan o se retrasen. [2]

- **Ataques a los dispositivos**

Muchos de los ataques realizados hoy en día por hackers y crackers hacia las redes de datos, tienen como objetivo principal el hardware y el software de los dispositivos. Por lo tanto, en redes VoIP, los gateway, servidores proxy y teléfonos *IP* serían objetivos potenciales a explotar por parte de cualquier intruso.

Hay que tener en cuenta que los dispositivos de la VoIP son tan vulnerables, como lo es el sistema operativo o el *firmware* que ejecutan. Un aspecto que hace muchas veces de los dispositivos un punto débil dentro de la red, son las configuraciones incorrectas. A menudo los dispositivos VoIP trabajan con sus configuraciones por defecto y presentan gran variedad de puertos abiertos. Los servicios por defecto corren en dichos puertos y pueden ser vulnerables a ataques de DoS y desbordamientos de *buffer*. [2]

- **Descubriendo objetivos**

Una vez que el atacante ha seleccionado una red como su próximo objetivo, sus primeros pasos por lo general consistirán en obtener la mayor información posible de su victima. Cuando el intruso tenga información suficiente evaluará sus siguientes pasos eligiendo el método de ataque más adecuado para alcanzar su objetivo. Normalmente el método de obtención de información, se realiza con técnicas de menos a más nivel de intrusión. De este modo, en las primeras etapas el atacante realizará una obtención de toda la información pública posible del objetivo. Otra de las acciones más comunes consiste en obtener la mayor información posible de las máquinas y servicios conectados en la red atacada. Después de tener un listado de servicios y direcciones *IP* consistente, tratará de buscar agujeros de seguridad, vulnerabilidades y obtener la mayor información sensible de esos servicios para poder explotarlos y conseguir una vía de entrada. [2]

- **Explotando el nivel de aplicación**

El nivel de aplicación de la red *IP,* es quizás uno de los más vulnerables, debido en parte a que la VoIP engloba gran cantidad de protocolos y estándares añadiendo cada uno de ellos su propio riesgo de

seguridad. Un ejemplo claro de ellos, es el protocolo SIP, muy discutido desde el punto de vista de la seguridad. Entre los ataques específicos contra el nivel de aplicación de VoIP, se encuentran los ataques de secuestro de sesión, desconexiones ilegales, inundación de peticiones, generación de paquetes malformados, falsificación de llamadas y algunos otros. [2]

1.6.2. Análisis de los principales ataques a la VoIP

1.6.2.1. El fraude

Una de las más importantes amenazas de las redes VoIP, son los fraudes a consecuencia de un acceso desautorizado a una red legal VoIP. Una vez que un usuario no autorizado accede, este realiza llamadas de larga distancia, en muchos casos incluso internacionales. [2]

1.6.2.2. Captura de conversaciones (eavesdropping)

En los términos de la telefonía *IP*, se habla de la intercepción de las conversaciones VoIP por parte de individuos que no participan en la conversación. Dicho ataque requiere interceptar la señalización y los *streams* de audio de una conversación. Los mensajes de señalización utilizan protocolos separados, es decir, *UDP* o *TCP* y los *streams* normalmente se transportan sobre *UDP* utilizando el protocolo *RTP*. Algunas personas podrían pensar que el ataque de eavesdropping podría eliminarse con el uso de switches Ethernet que restringen el tráfico broadcast en la red, porque se limita quién puede acceder al tráfico, pero no ocurre de esta forma. [2]

El impacto de esta técnica es más que evidente, interceptando comunicaciones con lo cual es posible obtener toda clase información sensible y altamente confidencial. Aunque en principio, se trata de una técnica puramente pasiva, razón por la cual, se hace mas difícil su detección, es posible intervenir también de forma activa en la comunicación, insertando nuevos datos (que en el caso de la VoIP se trataría de audio) redireccionar o impedir que los datos lleguen a su destino. [2]

1.6.2.3. Envenenamiento ARP (ARP spoofing)

Envenenamiento de la caché ARP como mecanismo para llevar a cabo un ataque (man in the middle). El atacante envenena la tabla ARP de los sistemas a los que ataca, la tabla ARP sirve para convertir las direcciones lógicas *IP* en direcciones capa 2 del modelo de referencia OSI (direcciones *MAC* de Ethernet). Prácticamente todos los sistemas operativos no protegidos aceptan respuestas ARP no

solicitadas. El atacante rellena la tabla ARP con las direcciones *IP* que necesita, y pone su propia dirección *MAC* detrás de las *IP's* enviando algunas respuestas ARP. [3]

Reenvía cada paquete recibido al destinatario original, que también está envenenado. La comunicación funciona perfectamente, pero la intercepción no es reconocida por los que están hablando, a no ser que utilicen mecanismos criptográficos. Por medio del *ARP* spoofing, un atacante puede capturar, analizar y escuchar comunicaciones VoIP. [3]

1.6.2.4. Spam sobre telefonía IP (SPIT)

El SPIT es uno de los peligros más mencionados para la VoIP, un atacante envía mensajes de voz parecidos al correo basura. El spitter utiliza la dirección de la víctima, en este caso, no su dirección de correo electrónico, sino su dirección SIP. En un contexto de expansión de la telefonía *IP,* sólo es cuestión de tiempo conseguir muchas direcciones SIP válidas, especialmente si se utilizan libretas de direcciones centrales. El spitter llama a un número SIP, el proxy de la víctima procesa esta llamada y la víctima tiene que escuchar la basura del spitter. Como el spammer, el spitter sólo necesita una cosa, ancho de banda, de hecho, los mensajes de voz consumen más recursos que los e-mails. Utilizando troyanos igual que el spam, un usuario desprotegido de Internet podría sufrir el abuso de enviar SPIT a través de su ancho de banda. [3]

1.6.2.5. Marcadores (Diallers)

El uso de diallers, puede volver a ser una amenaza por el uso de un cliente SIP, ya que tenemos el mismo escenario de un diallers clásico que utiliza el módem para llamar a ciertos números premium. Por ejemplo, un diallers infecta un cliente SIP e instala un número como si fuera un prefijo, o introduce un proxy nuevo, mucho más caro. Las llamadas pasarán a través de estos números caros sin que el usuario se dé cuenta, hasta que reciba la primera factura. [3]

1.6.2.6. Denegación de servicio (DoS)

El DoS es un ataque a un sistema de ordenadores o red, el cual puede ser la causa de que un servicio o recurso sea inaccesible a los usuarios legítimos. Por lo general provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o la sobrecarga de los recursos computacionales del sistema de la víctima. Se genera mediante la saturación de los puertos con flujo

de información, haciendo que el servidor se sobrecargue y no pueda seguir prestando servicios, por eso se le dice denegación, pues hace que el servidor no de abasto a la cantidad de usuarios. [2]

DDoS (Denegación de Servicio Distribuida) es una ampliación del ataque DoS, se efectúa con la instalación de varios agentes remotos en muchas computadoras, que pueden estar localizadas en diferentes puntos. El invasor consigue coordinar esos agentes, para así, de forma masiva, amplificar la saturación de información, pudiendo darse casos de un ataque de cientos o millares de computadoras dirigidas a una máquina o red objetivo. Esta técnica se ha revelado como una de las más eficaces y sencillas a la hora de colapsar servidores. [2]

1.6.2.7. Redirección de llamadas

La redirección de llamadas suele ser otro de los ataques comunes en las redes VoIP. Existen diferentes métodos que van desde comprometer los servidores o el call manager de la red para que redirijan las llamadas donde el intruso quiera. Utilizando una herramienta como RedirectPoison que escucha la señalización SIP hasta encontrar una petición INVITE (indica que el usuario o servicio es invitado a participar en una sección) y responder rápidamente con un mensaje SIP de redirección, causando que el sistema envié un nuevo INVITE a la localización especificada por el atacante. Otro modo de redirección del flujo de datos se consigue con las herramientas: sipredirectrtp y rtpproxy que se basan en utilizar mensajes de la cabecera *SDP* para cambiar la ruta de los paquetes *RTP,* dirigiéndolos a un rtpproxy, que a su vez podrán ser reenviados donde el intruso desee. [2]

1.6.2.8. Inserción de audio

En las llamadas VoIP la transmisión del flujo de datos se realiza por razones de sencillez y eficiencia sobre el protocolo *UDP*. Este protocolo no da garantías en la entrega de sus mensajes y no mantiene ningún tipo de información de estado o conexión. En el encapsulado de *UDP* se encuentra el protocolo *RTP* que transporta verdaderamente los datos de voz. *RTP* no lleva un control exhaustivo sobre el flujo de datos confinando las funciones de recuento de paquetes y calidad de servicios al protocolo *RTCP.* El único método que tiene *RTP* para controlar tramas pérdidas y reordenar las que le llega, es el campo número de secuencia de la cabecera. **[Ver Anexo figura 6]**

El atacante podría realizar ataques de inserción de paquetes dentro de un flujo *RTP* consiguiendo insertar de forma exitosa audio en una conversación telefónica. Incluso se ha comprobado que contra

algunos dispositivos es suficiente bombardear con paquetes *UDP*, para que estos se inserten en la conversación. [2]

1.6.2.9. Vishing: VoIP Phishing

Las amenazas de phishing suponen un gran problema para el correo electrónico. Las denuncias por robo de información confidencial de forma fraudulenta están a la orden del día y exactamente las mismas técnicas son aplicables a la plataforma VoIP. Gracias a la telefonía *IP* un intruso puede realizar llamadas desde cualquier lugar del mundo al teléfono *IP*, con técnicas de ingeniería social y mostrando la identidad falsa o suplantando otra conocida por la victima, puede obtener: información confidencial, datos personales, números de cuenta o cualquier otro tipo de información. [2]

1.6.2.10. Suplantación de identidad en el registro

El registro de usuarios es la primera comunicación que se estable en el entorno VoIP entre el usuario y el servidor de registro. Necesariamente esta comunicación debe de realizarse de forma segura, ya que en caso contrario no hay garantías de que el usuario registrado, sea quien dice ser durante todo el resto de la sesión. A través de los mensajes REGISTER (registro de usuarios), los agentes de usuario SIP informan al servidor de su localización actual, de manera que el servidor sepa dónde tiene que enviar peticiones posteriores. Si un servidor no autentica las peticiones REQUEST cualquiera puede registrar cualquier contacto para cualquier usuario, y por lo tanto secuestrar su identidad y sus llamadas. Cuando un proxy recibe la petición para procesar la llamada (INVITE), el servidor realiza una búsqueda para identificar donde puede ser encontrado el destinatario. [2]

El ataque funciona de la siguiente manera:

- Deshabilitando el registro legítimo del usuario.
- Enviando el mensaje REGISTER con la *IP* del atacante.
- En el servidor de registro queda registrado el usuario X pero con la dirección *IP* del hacker.
- Cuando recibe la llamada, el servidor proxy consulta la dirección del destinatario X, pero obtendrá la dirección *IP* del atacante.

- El ataque ha tendido éxito. El intruso ha suplantado la identidad de X y mientras mantenga el registro todas las llamadas dirigidas a X llegaran a su teléfono *IP*.

Este ataque es posible llevarlo a cabo por el hecho de que lo mensajes de señalización se envían casi siempre en texto plano, lo que permite al intruso capturarlos, modificarlos y retransmitirlos como él quiera. Ver ejemplo en la figura 1.6. [2]

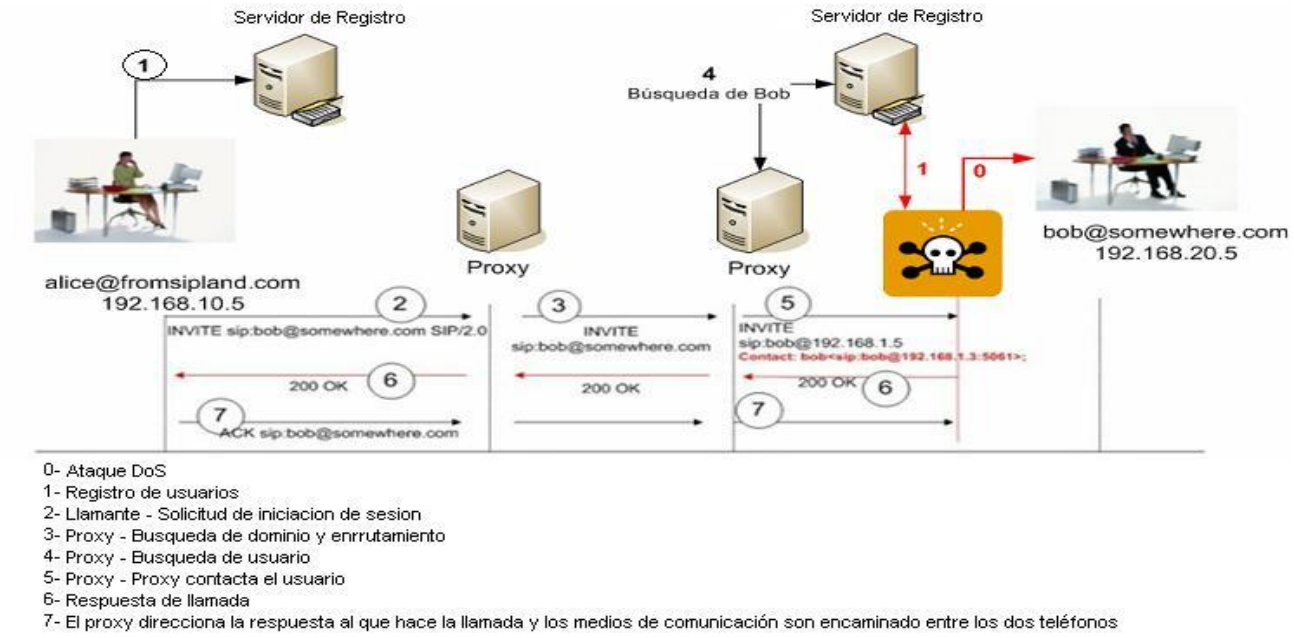


Figura 1.6 Suplantación de identidad en el registro

1.6.2.11. Desregistrar usuarios

El desregistro de usuarios legítimos, es una necesidad para conseguir suplantar su identidad. Básicamente el intruso podrá conseguirlo de la siguiente forma:

- Realizando un ataque de DoS al usuario.

- Generando una condición de carrera, en la que el atacante envía repetidamente peticiones REGISTER en un corto espacio de tiempo, con el objetivo de superponerse a la petición de registro legítima del usuario.

- Desregistrando el usuario con mensajes REGISTER.

El intruso puede ser capaz de desregistrar fácilmente un usuario, enviando al servidor de registro una petición REGISTER (simulando ser la victima) con el siguiente campo “Contact:” y valor del atributo “Expires” a cero. Esta petición eliminará cualquier otro registro de la dirección del usuario (especificada en el campo “To” de la cabecera). El atacante deberá realizar este envío periódicamente para evitar el re-registro del usuario legítimo o en su defecto provocarle un ataque de DoS para evitar que vuelva a registrarse al menos por el tiempo que necesite para realizar el secuestro de la llamada. [2]

1.6.2.12. Desconexión de usuarios

El hecho de que muchos de los protocolos, se utilizan sin encriptación alguna y que los mensajes no se autentican de forma adecuada, es trivial para cualquier intruso desconectar a los usuarios de sus llamadas enviando mensajes BYE con la identidad falsificada, simulando ser el usuario del otro lado de la línea. También se puede realizar un ataque similar utilizando mensajes CANCEL, pero solo afectan cuando se está estableciendo la llamada, es decir, antes de que el destinatario descuelgue el teléfono. Ver el ejemplo en la figura 1.7. [2]

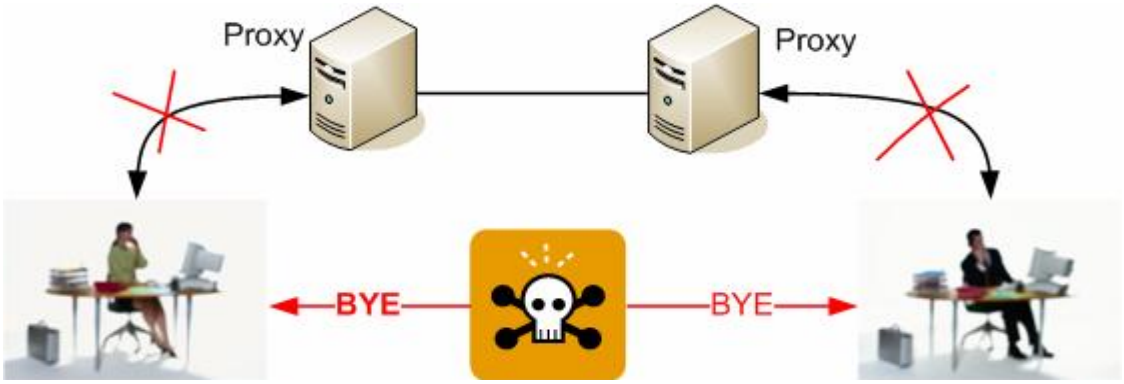


Figura 1.7. Desconexión de Usuarios

1.7. Consideraciones en la seguridad de la VoIP

Como se ha podido apreciar, por lo señalado en los párrafos anteriores, la VoIP tiene grandes problemas de seguridad, por lo que es factible que sea atacada con frecuencia, por lo tanto, se deben tener en cuenta, ciertas consideraciones a la hora de instalar y gestionar una red VoIP. Tratando de evitar así posibles vulnerabilidades o degradación de calidad de servicio de las conexiones VoIP.

Primeramente, se debe tener en cuenta la encriptación, que es una medida de vital importancia para la seguridad de la VoIP, aunque no es sencillo capturar y decodificar los paquetes de voz, encriptar es la única forma de prevenirse ante un ataque. Existen también múltiples métodos de encriptación, la clave está en elegir un algoritmo de encriptación que sea rápido y eficiente, así como emplear un procesador dedicado a la encriptación. Esto deberá aplacar cualquier indicio de amenaza. Otra opción podría ser la Calidad del Servicio (QoS); los requerimientos para QoS aseguran que la voz se utiliza siempre de manera oportuna, se reduce la posible pérdida de calidad.

Cada elemento de una red de VoIP, está integrado en una red de datos *IP*, y por lo tanto es direccionable y alcanzable a través de ella, como cualquier otro elemento. Debe considerarse la seguridad de los elementos que componen una red de datos (*routers*, *firewalls*, etc), así como las medidas generales de seguridad comunes en todas las redes de telecomunicación, (acceso físico, documentación y registro, etc).

Siempre que sea posible, separar las redes de datos de la de voz físicamente, o en su defecto, mediante el uso de *VPN.*

Asegurar la máxima fiabilidad en una red VoIP, aunque sea crítica la protección de los servidores VoIP. Estos deberán estar protegidos físicamente y residir en segmentos de red separados y protegidos por elementos de seguridad perimetral, (*firewalls*) y estar dedicados exclusivamente a este tráfico, aislándolos de los problemas o vulnerabilidades colaterales, debido a la interoperabilidad con otro tipo de servicios.

Se debe tratar de eliminar cualquier tipo de servicio o funcionalidad no requerida por la conexión VoIP en todos los elementos que forman la red, evitando de ese modo, la posible aparición de puertos vulnerables a determinados ataques.

Los *routers* y *switches*, deberán estar siempre configurados adecuadamente, con acceso a las listas de control y a los filtros. Todos los dispositivos deberán estar actualizados en términos de parches y actualizaciones.

Cuando la red de datos y la red de VoIP estén separadas de forma lógica y sea requerida la conexión entre ellas, para ciertos servicios como, por ejemplo, buzones de voz, se recomienda el uso de *firewalls* o *NAT*.

Se debe evitar el uso de contraseñas de administración y gestión de equipos por defecto o estandarizadas (admin/admin, contraseña/secret, etc) para el acceso a cualquier dispositivo de la red VoIP, evitando que el sistema sea fácilmente vulnerable a acceso de personal no autorizado en el estado inicial (tras la instalación) o al reiniciarse.

La disponibilidad de la red VoIP es otra de las preocupaciones porque una pérdida de potencia puede provocar que la red se caiga y los ataques DDoS sean difíciles de contrarrestar. Además de configurar con propiedad el *router*, pues los ataques, no solo irán dirigidos a los servicios de datos, sino también, a los de voz.

1.8. Conclusiones parciales

La VoIP ha ido evolucionando en los últimos años debido a la amplia gama de servicios y funciones que ofrece a sus clientes. Las muchas ventajas que brinda su uso hacen que sea una de las tecnologías que más se está imponiendo en el mundo de las telecomunicaciones. Sin embargo la VoIP es vulnerable a algunos ataques, por lo que la seguridad de la misma es un tema que no se debe dejar pasar por alto.

CAPÍTULO 2: ANÁLISIS GENERAL DEL PROTOCOLO SIP

2.1. Introducción

SIP es un protocolo de señalización que facilita el establecimiento, liberación y modificación de sesiones multimedia. Puede establecer sesiones de dos partes (son las llamadas ordinarias) y las de múltiples partes (es en donde todos pueden oír y hablar), así como las de multidifusión (un emisor y muchos receptores). Las sesiones que se establece entre los participantes pueden contener audio, video o datos.

En el presente capítulo se abordará de una forma más detallada este protocolo. También se describirán los principales ataques a los cuales está expuesto y los mecanismos de seguridad de los que dispone, así como el análisis de algunos algoritmos criptográficos.

2.2. Protocolo de Inicio de Sesión (SIP)

El protocolo SIP, tiene mucha similitud con protocolos clásicos de internet como *HTTP*, utilizado para navegar sobre la WEB y o *SMTP*, utilizado para transmitir mensajes electrónicos (e-mails). Este se basa en el paradigma modelo cliente/servidor al igual que *HTTP*, las sesiones son formadas por transacciones basadas en peticiones y respuestas. Un requerimiento SIP está constituido de encabezamientos al igual que *SMTP*, las direcciones de señalización tienen una sintaxis similar a la de las direcciones de correo electrónico y además presenta códigos de respuestas para cada petición realizada por los participantes, la mayoría de estos códigos son tomados del protocolo *HTTP*.

SIP se vale de funciones aportadas por otros protocolos, las que da por hechas y no vuelve a desarrollarlas. Debido a esto SIP funciona en combinación con otros protocolos con el fin de proporcionar servicios completos a los usuarios, entre estos protocolos se encuentran el *RTP* para intercambiar directamente el trafico de audio/video una vez establecida la sesión, el *RTCP* que trabaja junto con *RTP*, encargado de informar sobre la calidad de servicio ofrecido por *RTP* y el *SDP* para describir el contenido multimedia de las sesiones. Sin embargo a pesar de trabajar en sintonía con estos protocolos, la funcionalidad básica y el funcionamiento de SIP no dependen de ninguno de estos protocolos. Ver figura 2.1.

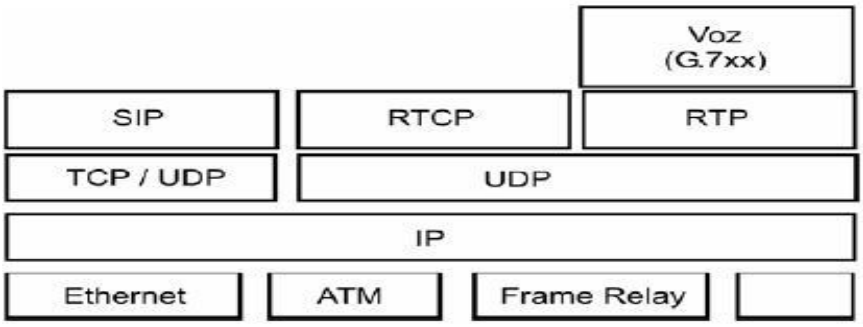


Figura 2.1 Torre de Protocolos de la arquitectura SIP

Como se muestra en la figura 2.2, SIP es un protocolo de nivel de aplicación independiente de las capas de transporte y de red, por lo que puede hacer uso tanto de un nivel de transporte *TCP*, como *UDP*, pero las implementaciones más comunes, usan SIP sobre *UDP*, por su simplicidad y velocidad con respecto a *TCP*.

A continuación en la figura 2.2 se pueden observar al protocolo SIP en la pila de protocolos.

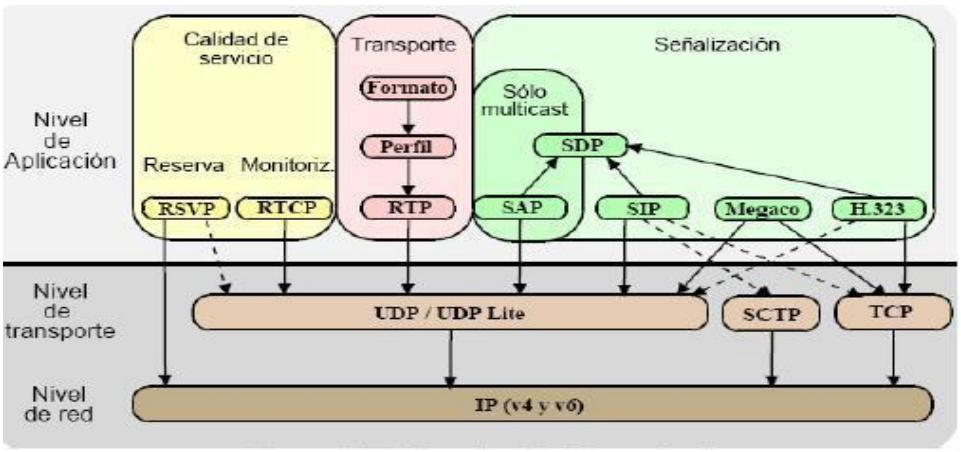


Figura 2.2 SIP en la pila de protocolos

Las funcionalidades que generalmente se le exigen a un protocolo de estas características, son básicamente:

- La traducción de nombres y las ubicaciones de usuarios.
- La negociación de capacidades de cada usuario.

- La gestión de los usuarios que toman parte en una sesión.
- La gestión de los cambios en las capacidades de cada participante.

Una de las mayores potencialidades de SIP es la que incluye entornos sencillos para la programación de servicios, incluso por parte del usuario final. Para los usuarios avanzados, como puede ser un administrador, se ofrece una interfaz *CGI* permitiéndoles tener acceso a todas las funcionalidades del protocolo, con la enorme flexibilidad que eso conlleva. Por otro lado y para usuarios menos expertos, se ofrece un lenguaje de programación: el *CPL*, se trata de una herramienta muy sencilla, que pone una serie de funciones básicas a disposición de los usuarios para que definan sus propios servicios, asegurando que ninguna de sus acciones va a poner en peligro la integridad del sistema.

2.3. Beneficios de SIP

Algunos de los principales beneficios claves de SIP son:

- **Simplicidad:** SIP es un protocolo muy simple. El tiempo de desarrollo del software es muy corto comparado con los productos de telefonía tradicional. Debido a la similitud de SIP a *HTTP* y *SMTP*, el rehúso de código es posible.
- **Extensibilidad:** SIP ha aprendido de *HTTP* y *SMTP* y ha construido un exquisito grupo de funciones de extensibilidad y compatibilidad.
- **Modularidad:** SIP fue diseñado para ser altamente modular. Una característica clave es su uso independiente de protocolos. Por ejemplo, envía invitaciones a las partes de la llamada, independiente de la sesión misma.
- **Escalabilidad:** SIP ofrece dos servicios de escalabilidad:
  - Procesamiento de servidor: SIP tiene la habilidad para ser (*statefull*) o (*stateless*).
  - Arreglo de la conferencia: Puesto que no hay requerimiento para un controlador central multipunto, la coordinación de la conferencia puede ser completamente distribuida o centralizada.

- **Integración:** SIP tienen la capacidad para integrarse con la web, e-mail, aplicaciones de flujo multimedia y otros protocolos.

- **Interoperabilidad:** Es un estándar abierto, SIP puede ofrecer interoperabilidad entre plataformas de diferentes fabricantes. [6]

2.4. Las Entidades SIP

SIP define dos tipos de entidades: los clientes y los servidores. De forma mas precisa, las principales entidades definidas por SIP son:

- **Los Agentes Usuario**

Los usuarios utilizan para establecer sesiones lo que el protocolo SIP denomina Agentes Usuario. Estos no son más que los puntos extremos del protocolo, es decir son los que emiten y consumen los mensajes del protocolo SIP. Un videoteléfono, un teléfono, un cliente de software (*softphone*) y cualquier otro dispositivo similar, es para el protocolo SIP, un agente usuario.

Los agentes de usuario constan a su vez de dos componentes: los agentes de usuario clientes (UAC) y los agentes de usuario servidores (UAS). Son agentes de usuario clientes cuando realizan una petición SIP y mientras que los agentes de usuario servidores están encargados de atender tales peticiones y remitir las correspondientes respuestas.

- **Los Servidores.**

  - **Servidor Proxy**

    El servidor proxy, actúa como servidor por un lado recibiendo y tratando pedidos y como cliente por el otro, encaminando pedidos para otros servidores y también directamente para las terminales. Por ello, a pesar de que SIP es un protocolo de paradigmas cliente/servidor, cualquier elemento asume la función de cliente o de servidor. Su función es similar a la de los proxys *HTTP*: recibir solicitudes y decidir a qué otro servidor deben remitirse, alterando además algunos campos de la solicitud. Por tanto, actúan como intermediarios en las transacciones que procesan. [4]

Normalmente, los proxys actúan como servidores de registro para todos los dispositivos representados por ellos. Para ello, aceptan y procesan peticiones de registro. También, el proxy puede recurrir a un servidor de localización para determinar la dirección en la que actualmente está disponible el destinatario. [4]

El servidor proxy puede *rutear* toda la señalización de las llamadas o simplemente *rutear* la apertura del canal y luego deja que las terminales cambien los mensajes y los medios de señalización directamente. Este último procedimiento, es el ideal para la reducción de alojamiento de recursos en el servidor y en la red para la reducción del tiempo de retención de los mensajes de señalización durante la llamada. Es importante observar, que el proxy también determina la dirección, el puerto y protocolo de transporte cuando encamina cualquier pedido, en dependencia de su configuración. Otra característica importante de operación de este tipo de servidor, es poder operar en dos modos diferentes, o modo *stateless* o el modo *statefull*. [4]

El servidor SIP *stateless* es una entidad lógica que no mantiene el estado de transición de clientes o servidores cuando procesa los pedidos, o sea el proxy recibe las llamadas, realiza cualquiera que sea el traslado del mensaje y lo encamina convenientemente, sin almacenar ningún registro de este evento. En este caso, el mensaje retransmitido por el proxy *stateless* tiene que ser encaminado exactamente de la misma forma que el pedido original. Por tanto, cuando está *stateless*, un proxy actúa como un simple elemento de encaminamiento de pedidos, descartando las informaciones sobre los mensajes, una vez que estos hayan sido retransmitidos. [4]

El proxy *statefull* es una entidad lógica que mantiene el estado de los clientes y servidores durante el procesamiento de un pedido, almacenando la información de llegada del mensaje y el respectivo encaminamiento, de esta manera actúa de forma más inteligente en pedidos subsecuentes y en respuestas relacionadas a una misma sesión, pudiendo modificar el procesamiento de futuros mensajes asociados a ese pedido. Cualquier pedido que sea encaminado para más de una localidad tiene que ser tratado por ese servidor, esto sucede si una terminal, está registrada en más de un servidor de Registro. [4]

Cuando un ambiente real se toma en consideración, el servidor proxy siempre va a ser de tipo *statefull*, pues actúan como *routers* de mensajes de señalización y posiblemente de canal de los medios, además de realizar la función de tarifas y de almacenador del estado de las

llamadas y de los terminales, y practican todos los mensajes de señalización que serán enunciadas a continuación. En contrapartida, un ambiente con una gran cantidad de llamadas simultáneas, puede implicar un problema grave de rendimiento. Por eso, un servidor proxy de tipo *stateless*, si no puede ser usado en toda la red SIP, por lo menos en algunos sectores de la red, que por el contrario del servidor *statefull*, no es capaz de generar ninguna decisión propia y no altera cualquier encabezamiento de los mensajes de señalización. Un servidor proxy puede bloquear la operación en modo *stateless* a cualquier tiempo durante el procesamiento de un pedido, mientras que no haya realizado ninguna función previa como servidor *statefull*. [4]

- **Servidor de Redireccionamiento**

A diferencia de los servidores proxys, los servidores de redireccionamiento no inician transacciones, sino que, cuando reciben solicitudes desde un agente usuario cliente, remiten al mismo agente un mensaje indicando el o los servidores con los que debe ponerse en contacto, en un procedimiento similar al de búsqueda iterativa del sistema *DNS*. [4]

El servidor de redireccionamiento se caracteriza por no aceptar llamadas, ni procesar o encaminar pedidos SIP, solo retorna información de localizaciones del terminal de destino o del servidor proxy correspondiente, basado en una tabla de rutas. El terminal de origen, de poseer esa información, es capaz de contratar directamente la entidad requerida. Normalmente, estos servidores gestionan mayor número de mensajes que los proxys, pero con menores necesidades de procesamiento. Nótese que, puesto que en sesiones controladas por SIP la redirección se realiza mediante mensajes SIP, las respuestas se pueden generar con flexibilidad y adecuación a servicios de conferencia multimedia. [4]

En algunas arquitecturas, se puede desear contar con servidores de redireccionamiento para reducir el proceso de sobrecarga en los servidores proxy, con el objetivo de aumentar la robustez de la red de señalización. [4]

- **Servidor de Registro**

El uso más común de estos servidores es registrar un dispositivo después de su arranque, o sea establecer la ubicación física de un usuario determinado, esto es en qué punto de la red

está conectado, de modo que cuando lleguen invitaciones destinadas a él, los servidores SIP puedan proporcionar su dirección.

El estado registrado no es permanente, pues se contempla la existencia de un tiempo máximo de validez de cada registro, el cual debe ser actualizado en un periodo de tiempo definido por el servidor. Por otra parte, el registro también puede ser cancelado y es importante notar como el servidor se comporta e interpreta tal pedido.

Cada cliente puede tener su registro en múltiples servidores y un determinado cliente puede tener múltiples registro en un solo servidor. En este último, si el usuario tiene múltiples registros activos y recibe una llamada, todos los destinos que se encuentran registrados recibirán señales de alerta simultáneamente.

- **Servidor de Localización**

Facilita información al servidor proxy o de redireccionamiento sobre la ubicación del destinatario de una llamada.

A continuación en la figura 2.3 se muestran las entidades de una red SIP.

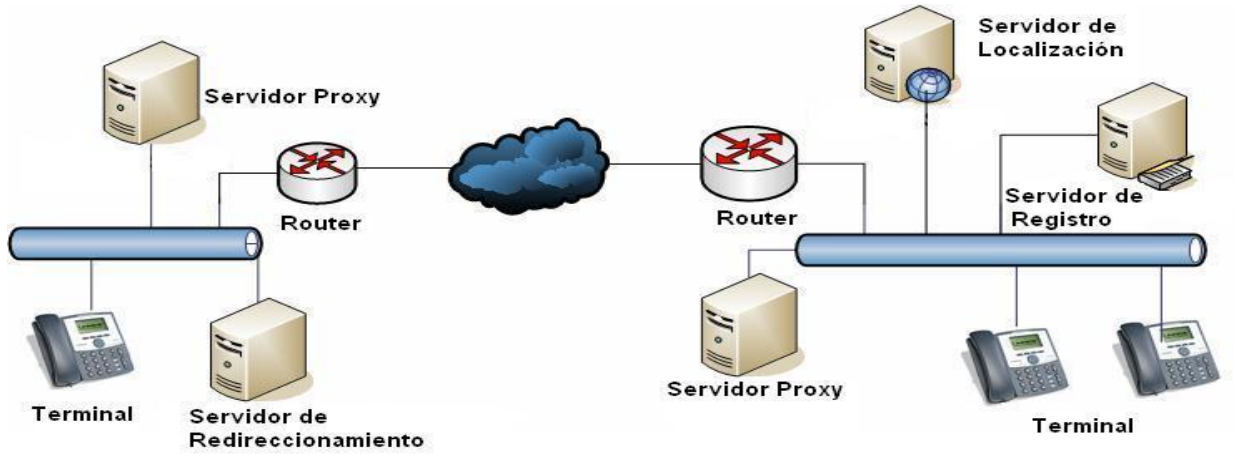


Figura 2.3 Entidades de una red SIP

2.5. Estructura del mensaje SIP

Los mensajes de SIP pueden ser solicitados por un cliente a un servidor o las respuestas de un servidor para un cliente. Cada mensaje contiene una línea inicial, seguida por cero o más encabezamientos y opcionalmente seguido por el cuerpo del mensaje, de acuerdo con la sintaxis mostrada en la figura 2.4.

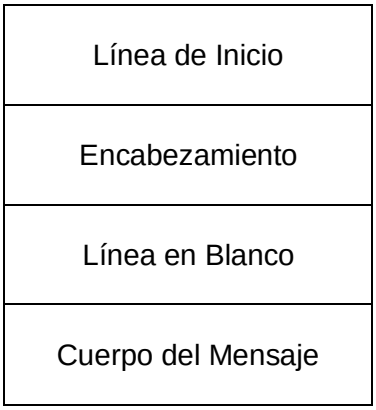


Figura 2.4 Formato del mensaje SIP

El protocolo SIP define dos tipos de mensajes: solicitud y respuesta. Por eso, la línea de inicio puede ser la línea de solicitud que especifica un tipo de solicitud que es enviada, o la línea de respuesta, que indica el suceso o fracaso de una solicitud determinada. En caso del fracaso, la línea indica el tipo o la razón del fracaso. Los encabezamientos del mensaje proveen información adicional relativa a solicitudes o respuestas, también ofrecen los medios para tomar información adicional de los mensajes, como por ejemplo: avisos de cuando los usuarios están disponibles, tema de la conversación, etc. El cuerpo del mensaje, describe el tipo de sección a ser establecida, inclusive una descripción de los tipos de medios que serán cambiados, *códec*, etc., SIP no define la estructura o el contenido del cuerpo del mensaje. [5]

2.6. Direccionamiento SIP

Conocido universalmente como SIP *URI*, el direccionamiento SIP está compuesto por nombres y hacen referencia a una entidad abstracta pudiendo ser un terminal o un servidor. El formato general de la *URL* es sip:user@host o sips:user@host, similar a la dirección del correo. El campo “user” puede representar un nombre de usuario del cliente, el mismo número de teléfono de la red *IP*, o hasta el mismo número valido en el sistema telefónico actual, ya que puede direccionar números telefónicos a

través de gateway *PSTN*. El campo “host” se puede referir tanto al nombre de dominio como a la dirección numérica Ipv4 o Ipv6. [4]

La evaluación de una SIP *URI* se debe hacer por completo, ya que posibilita especificar un número de puerto, protocolo de transporte, dirección *multicast*, asunto del mensaje, tipo de media y también sesiones de emergencia. La forma completa de representarlo, es la siguiente:

sip: user:password@host:port;uri-parameters?headers

El campo “uri-parameters” asume la sintaxis nombre_parámetro = valor_parámetro e incluye los parámetros: transport, maddr, ttl, user, method, lr. Destacando el parámetro maddr, que indica la dirección del servidor que necesita contactar para alcanzar el destinatario, desechando cualquier dirección derivada del campo del host. Cuando un parámetro maddr está presente, los valores del puerto se aplican a la dirección indicada en el campo maddr. Esto permite que el *URL* especifique un servidor proxy, el cual se utilizaría para *rutear* el mensaje hasta su destino final. El parámetro transport determina el mecanismo de transporte a ser utilizado para enviar mensajes. [4]

El campo “headers” también presente en el String de la dirección y separada por el carácter “?” de los demás campos, permite incluir encabezados en cualquier requisito escrito en una estructura *URL*. Los campos del encabezado SIP son similares a los del *HTTP*, tanto sintáctica como semánticamente. [4]

En la siguiente figura 2.5 se muestra una dirección SIP.

Figura 2.5 Direccionamiento SIP

2.7. Señalización SIP

Toda comunicación SIP se basa en apenas dos grandes grupos de mensajes de señalización, los pedidos y sus respectivas respuestas. La sintaxis de los mensajes está basada en texto y es idéntica al del *HTTP*, esto sucede porque ambos son protocolos de la capa de transporte del modelo *OSI* y poseen la objetividad y la rapidez necesaria para soportar sistemas de información distribuidos. La estructura textual de los campos SIP, permite que nuevas características sean incluidas fácilmente y que estas sean compatibles con las versiones anteriores. Los campos o parámetros nuevos pueden ser colocados en cualquier parte del mensaje.

Los mensajes SIP pueden incluir campos con contenidos no estandarizados, para que contengan las especificaciones y las características de cada terminal en la red. Este recurso posibilita a los usuarios intercambiar información con una configuración muy particular, por ejemplo cuando un terminal recibe una llamada y esta se encuentra disponible, es capaz de informar al responsable de la llamada su horario de retorno. Esto permite que el mismo terminal de destino devuelva la llamada de forma automática a la hora programada, informando que el usuario al que llamó está disponible. En este sentido se debe observar y entender estas facilidades personalizadas, con el objetivo de evaluar el impacto en la red y el grado de inter-operatividad con los demás equipos que operan de acuerdo con el patrón del protocolo, o sea, sin funcionalidades propietarias.

La señalización SIP hace simple la apertura de los canales de audio y su fácil adaptación a los patrones de Internet, ya que posee un ambiente que no está totalmente controlado, sobre todo en relación y variación del camino recorrido durante una transmisión, así como las características desconocidas y variables del tráfico, a pesar de que necesite de otros protocolos de reserva, de recursos o políticas de prioridades para garantizar la calidad del servicio. Aún en ambientes controlados como en el caso de una *LAN*, el patrón SIP todavía presenta mayor escalabilidad. [4]

2.7.1. Peticiones SIP

Una Petición SIP tiene una Request–Line (línea de solicitud) cuyo formato es el que se muestra en la figura 2.6.

Método	Espacio en blanco	Pedido *URI*	Espacio en blanco	Versión de SIP

Figura 2.6 Línea de inicio

En la tabla 2.1 se muestra la especificación de cada parte del formato de una petición SIP.

Campos	**Descripción**
Método	Identifica la solicitud que fue emitida
Request-*URI*	Dirección de la entidad para la cual la solicitud está siendo enviada. Este campo tiene un formato *URI*
Versión-SIP	Identifica la versión de SIP que va a utilizarse

Tabla 2.1 Campos de línea de inicio de una solicitud SIP

La comunicación SIP presenta diferentes tipos de peticiones de mensajes. Estas peticiones, a las que también se hace referencia como métodos, permiten que los agentes de usuarios y servidores de red localicen, inviten y administren llamadas. Las principales peticiones SIP son:

- **INVITE:** Estos mensajes indican que el usuario o servicio es invitado a participar en una sección. Incluye una descripción de sección y, para llamadas de dos vías, la parte llamante indica el tipo de medio. Una respuesta con éxito a una invitación INVITE de dos partes incluye el tipo de medios recibidos por la parte llamada. Con este simple método, los usuarios pueden reconocer las posibilidades del otro extremo y abrir una sección de conversación con un número limitado de mensajes e ideas y vueltas. **[Ver Anexo figura1]**

- **ACK:** Los mensajes de este tipo sirven de confirmación para intercambios fiables de mensajes de invitación. Los usuarios deben generar mensajes ACK para confirmar que se ha recibido el mensaje final de aceptación correspondiente a una invitación. Si la parte llamante incluye una descripción de la sección en la petición ACK, no se utilizan más parámetros adicionales en la

misma. De no incluirse una descripción de la sección, los parámetros de la sección en la petición INVITE se utilizan como los predeterminados. **[Ver Anexo figura 5]**

- **BYE:** Estos mensajes indican a los servidores que un cliente desea finalizar la conexión entre dos participantes en una sesión. Se pueden generar tanto en los agentes que iniciaron la llamada como en los que recibieron la invitación. Antes de liberar realmente la llamada, el agente usuario envía esta petición al servidor indicado el deseo de liberar la sesión. **[Ver Anexo figura 8]**

- **OPTIONS:** Este método permite consultar y reunir posibilidades de agentes de usuarios y servidores de red. Los servidores de Redirección y los proxys simplemente los reenvían. Otros servidores pueden responder con un mensaje en el que indiquen sus capacidades o bien con la respuesta que hubiesen dado a una invitación. Sin embargo, esta petición no se utiliza para establecer sesiones.

- **CANCEL:** Esta petición permite que los agentes de usuarios y servidores de red cancelen cualquier petición que este en progreso. Estos mensajes deben contener el mismo valor para los campos Call-ID, To, From, y CSeq que el mensaje de invitación original. En todo caso, estos mensajes nunca finalizan una llamada ya establecida. **[Ver Anexo figura12]**

- **REGISTER:** Los mensajes REGISTER proporcionan la localización de un agente usuario a los servidores de registro. En ellos, los agentes de usuario clientes notifican a los proxys o los servidores de redirección la dirección o direcciones en las cuales se encuentra un usuario. En estos mensajes el campo To de la cabecera indica la dirección que se ha de registrar, mientras que el campo From indica la dirección del usuario responsable del registro. Para el registro de dispositivos durante su arranque se ha reservado una dirección *multicast* a la que pueden enviar mensajes REGISTER.

2.7.2. Respuestas SIP

Las respuestas a los mensajes SIP están basadas en la recepción e interpretación de una petición correspondiente. Se envían como respuesta a una petición e indican si la llamada ha tenido éxito o ha fallado, incluido el estado del servidor.

La línea inicial de una respuesta SIP es una Status- Line. Esta línea contiene un código (un numero de tres dígitos) que indica el resultado de una solicitud y un texto que describirá el resultado. El software del cliente interpretara el código y tomara las medidas necesarias, en cuanto a la descripción pueda ser presentada al usuario para que el mismo pueda entender mejor la respuesta. El Status-Line posee la sintaxis mostrada en la figura 2.7.

Versión SIP	Espacio en blanco	Código	Espacio en blanco	Descripción

Figura 2.7 Línea de inicio de una respuesta SIP

En la tabla 2.2 se muestra la especificación de cada campo de una línea de inicio de una respuesta SIP. [5]

Campos	**Descripción**
Versión-SIP	Identifica la versión de SIP a ser utilizada
Código	El código de la respuesta está compuesto por tres dígitos que permiten clasificar los diferentes tipos existentes.
Descripción	Un texto descriptivo que indica el motivo o comentario sobre la respuesta

Tabla 2.2 Campos de una línea de inicio de una respuesta SIP

Existen seis clases de respuestas:

- **Clase 1xx:** Información, la petición ha sido recibida, comprendida y está siendo tratada
- **Clase 2xx:** Éxito, la petición ha sido recibida, comprendida y aceptada.

- **Clase 3xx:** Redirección, la llamada requiere otros tratamientos antes de poder determinar si puede ser realizada.
- **Clase 4xx:** Error petición cliente, la petición no puede ser interpretada o servida por el servidor. La petición debe ser modificada antes de ser enviada.
- **Clase 5xx:** Error servidor, el servidor fracasa en el tratamiento de una petición aparentemente valida.
- **Clase 6xx:** Fallo global, la petición no puede ser tratada por ningún servidor.

2.8. Cabeceras SIP

Las cabeceras de los mensajes SIP son las responsables de hacer posible la comunicación multimedia en la red y describir los detalles de la transacción. Son informaciones incluidas en los pedidos y respuestas para proveer tanto las informaciones básicas como las más avanzadas y habilitar el tratamiento apropiado a cada mensaje. Cada cabecera tiene sentido para algunos pedidos y respuestas, y en algunos casos la presencia de una determinada cabecera en la respuesta es consecuencia de un determinado pedido. **[Ver Anexo figura11]**

Existen cuatro categorías principales, las cuales se observan en la tabla 2.3.

Categorías de cabeceras	Descripción
Cabeceras Generales	Pueden ser usadas dentro de solicitudes y respuestas. Estas cabeceras contienen informaciones básicas que son necesarias para la administración de las solicitudes y las respuestas.
Cabeceras Solicitud	Son solo aplicados a las solicitudes SIP, y son usados para proveer información adicional para el servidor/cliente del pedido.
Cabeceras Respuesta	Son solo aplicadas a las respuestas SIP y usadas

	para proveer información adicional a respuestas que no pueden ser incluidas en status-line.
Cabeceras Entidad	En SIP, el cuerpo del mensaje contiene información sobre la sección que debe ser presentada al usuario. El propósito de la cabecera entidad es indicar el tipo de formato de la información incluida en el cuerpo del mensaje.

Tabla 2.3 Cabeceras SIP

El encabezado de SIP está conformado por campos, de los cuales los más importantes son:

- **From**: El campo From es el identificador lógico del UA que genera el pedido (UAC). Está compuesto por el *URI* y opcionalmente el parámetro "Display Nombre", el cual es el nombre que se presentara si el usuario tiene el servicio de llamada Id activo. Adicionalmente, dentro del from se encuentra el parámetro "Tag", el cual es un identificador generado por el UAC en el momento de hacer el pedido. Este identificador, junto con el parámetro con el identificador de la llamada (Call-ID), sirven para identificar el dialogo entre el UAC y el UAS. **[Ver Anexo figura 9]** [6]

- **To**: En el campo To se indica el destinatario lógico del pedido. El UAC genera el campo To a partir de lo ingresado por el usuario. En este caso, la llamada se generó marcando la *IP* del destino, por lo que el UAC generó el *URI* automáticamente. **[Ver Anexo figura 10]** [6]

- **CSeq**: Este parámetro define el orden de las transacciones. Consiste de un número de secuencia y de un método. Este número de secuencia se va incrementando de a uno por vez. [6]

- **Call-ID**: Este parámetro es un identificador único que agrupara una serie de mensajes. Es obligatorio que sea el mismo durante todos los mensajes que intercambiados entre UAC y UAS. [6]

- **Via**: El parámetro Via identifica el protocolo de transporte y la ubicación a donde se debe enviar la respuesta. El UAC debe insertar este parámetro obligatoriamente siempre que se genera el request. Es importante notar que este parámetro está presente en los mensajes enviados por el UAC solamente. Dentro del campo Via se incluye el parámetro "Branch", el cual se utiliza para identificar la transacción dentro del UAC. Este parámetro debe ser único. La RFC 3261 especifica que el valor que tomará por defecto el Branco ID comienza con z9hG4bK. [6]

- **Contact**: Este parámetro se utiliza para identificar la instancia específica del UA a donde se puede enviar la request, fuera del dialogo en curso. [6]

- **Max Forwards**: Este parámetro sirve para limitar la cantidad de saltos que un request puede transitar. Por defecto este valor se fija en 70 saltos. Si este parámetro llega a 0, la llamada es terminada con código 483 (demasiados saltos). [6]

- **Allow**: Indica todos los métodos soportados por el UA. Si este parámetro no está presente, no significa que el UA no soporta ningún método, sino que el mismo no los proveyó. Este mensaje busca optimizar la cantidad de mensajes que se necesitan para concluir. [6]

- **Content-Length**: Indica la cantidad de bytes del cuerpo del mensaje. Si no hay información en el cuerpo del mensaje, este parámetro debe ser reseteado en 0. Este parámetro puede ser abreviado utilizando:"l". [6]

- **Content-Type**: Este parámetro indica el tipo de información contenida en el cuerpo del mensaje. [6]

2.9. Cuerpo del mensaje SIP

En el protocolo SIP el cuerpo del mensaje describe el tipo de sesión a ser establecida, para esto utiliza el protocolo *SDP* que establece los parámetros que definirán el establecimiento de la sesión, por ejemplo qué direcciones *IP*, puertos y *códecs* se usarán durante la comunicación. Estos parámetros son los siguientes: **[Ver Anexo figura7]**

- **v** - *SDP* Versión de Protocolo.
- **o** - Propietario/Creador, Sesión Id: Dentro de estos parámetros se puede ver el identificador de la sesión, la *IP* y el propietario.
- **c** - Información de Conexión: En este parámetro se presenta la *IP* donde se debe enviar el stream de audio.
- **t** - Descripción de Tiempo: Es el tiempo que lleva activa el stream de audio.
- **m** - Media Description: Tiene la descripción de todos los *codec* soportados por el UAC que iniciaron la conversación. Se especifica el media type que informa que el contenido de la sesión es audio, el media port, el cual define el puerto *UDP* que se asignó para recibir el stream de RTP.
- **a** - Descripción Media: Este tag especifica los diferentes parámetros para cada *codec* en particular.

La figura 2.8 muestra un ejemplo de mensajes SIP, el primero es un mensaje de petición y el segundo es de respuesta, en este se puede ver como está estructurado y compuesto el mensaje SIP, además de los parámetros que contiene.

Figura 2.8 Mensajes de petición y respuesta SIP

2.10. Funcionamiento del protocolo SIP

2.10.1. Registro de una Terminal

El protocolo SIP permite establecer la ubicación física de un usuario determinado, es decir en qué punto de la red está conectado. Para ello, se vale del mecanismo de registración, el cual funciona de la siguiente manera:

Cada usuario tiene una dirección lógica que es invariable respecto de la ubicación física del usuario. Una dirección lógica del protocolo SIP tiene la forma usuario@dominio, es decir tiene la misma forma que una dirección de correo electrónico. La dirección física (denominada dirección de contacto) es dependiente del lugar en donde el usuario está conectado (de su dirección *IP*). Cuando un usuario inicializa su terminal, por ejemplo conectando su teléfono o abriendo su software de telefonía SIP, el agente usuario SIP que reside en dicho terminal envía una petición con el método REGISTER a un servidor de Registro informando a qué dirección física debe asociarse la dirección lógica del usuario. El servidor de Registro realiza entonces dicha asociación. La forma en que dicha asociación es almacenada en la red no es determinada por el protocolo SIP, pero es vital que los elementos de la red SIP accedan a dicha información. En la figura 2.9 se puede observar el proceso de registro de un UA con autentificación y sin autentificación.

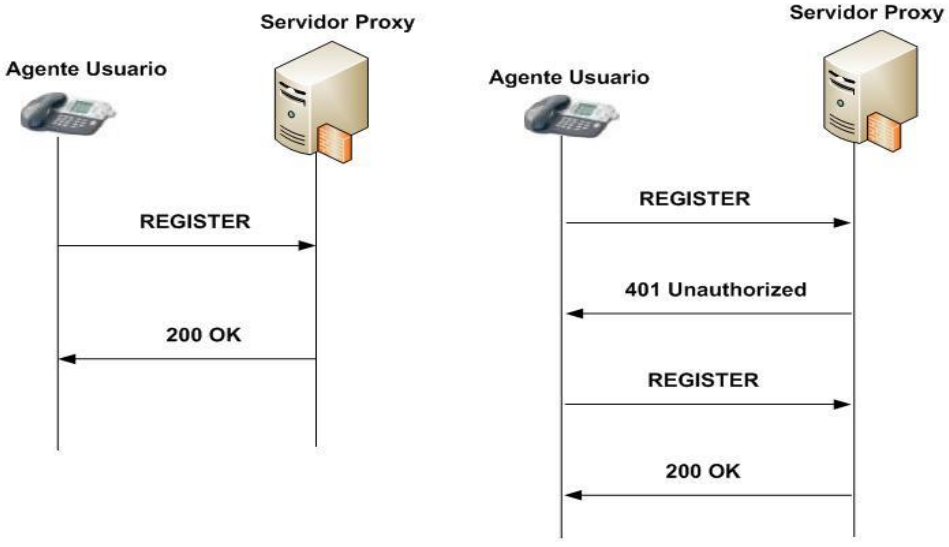


Figura 2.9 Proceso de registro de un Agente Usuario

2.10.2. Establecimiento y liberación de una sesión SIP

El flujo habitual del establecimiento de una sesión mediante el protocolo SIP es el siguiente:

Un usuario ingresa la dirección lógica de la persona con la que quiere comunicarse, además puede indicar al terminal las características de la sesión que quiere establecer (voz, voz y video, etc.), o estas pueden estar implícitas por el tipo de terminal del que se trate. El agente usuario SIP que reside en el terminal, actuando como agente usuario cliente envía la petición (en este caso con el método INVITE) al servidor que tiene configurado. Este servidor se vale del sistema *DNS* para determinar la dirección del servidor SIP del dominio del destinatario. El dominio lo conoce pues es parte de la dirección lógica del destinatario. Una vez obtenida la dirección del servidor del dominio destino, encamina hacia allí la petición.

El servidor del dominio destino establece que la petición es para un usuario de su dominio y entonces se vale de la información de registración de dicho usuario para establecer su ubicación física. Si la encuentra, entonces encamina la petición hacia dicha dirección, mientras le envía a usuario o servidores un mensaje de que lo está intentando 100 (Trying) **[Ver Anexo figura2]**. El agente usuario destino si se encuentra desocupado enviara señales de tono con el código 180 (Ringing) **[Ver Anexo figura3]** hasta llegar al agente usuario origen. Cuando el usuario destino finalmente acepta la invitación, se genera una respuesta de 200(OK) **[Ver Anexo figura4]** que indica que la petición fue aceptada. La recepción de la respuesta final es confirmada por el agente usuario cliente originante mediante una petición con el método ACK, esta petición no genera respuestas y completa la transacción de establecimiento de la sesión.

Durante el proceso de inicio de sesión, llega un momento en el que ambos teléfonos se comunican directamente, sin pasar por el proxy ya que han aprendido las rutas gracias a los encabezados de los mensajes SIP. Después comienza la comunicación y la transferencia de información hasta que alguna de las parte decida colgar el teléfono y terminar la comunicación. La parte que decida finalizar la comunicación manda un mensaje BYE al otro agente usuario sin pasar por los proxys. El otro agente usuario envía un reconocimiento del mensaje BYE (200 OK) y la sesión se da por terminada. En la figura 2.10 se muestra el establecimiento y liberación de una sesión SIP.

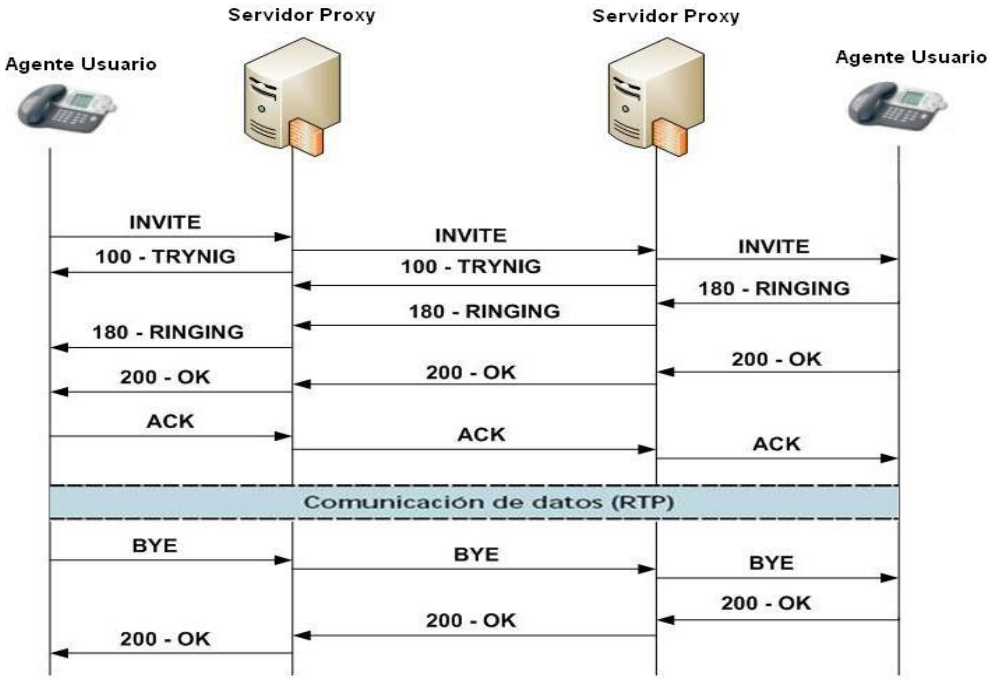


Figura 2.10 Establecimiento y Liberación de una sesión SIP

2.11. Ataques al protocolo SIP

SIP no se diseñó teniendo como objetivo principal la seguridad, pese a lo cual, es posible conseguir niveles de seguridad satisfactorios utilizando diferentes mecanismos. Sin embargo, conseguir estos niveles puede ser relativamente complejo, debido al número de diferentes tecnologías involucradas, así como que los componentes que formen parte de una sesión SIP puedan no implementar todos estos mecanismos, lo cual hace que en algunos casos haya que reducir estos niveles.

En cualquier caso, al igual que en otras tecnologías, los puntos principales de ataque y donde hay que centrar los esfuerzos de seguridad, son en la autenticación y en la confidencialidad/integridad de los datos. En los puntos siguientes vamos a ver algunas de las formas de ataque de las que puede ser susceptible SIP. Estos ataques asumen un ambiente en el que los atacantes pueden leer algunos paquetes en la red, modificar dichos paquetes, también pueden desear robar servicios, escuchar a escondidas las comunicaciones o afectar las sesiones.

2.11.1. Secuestro de registro

Este ataque consiste en secuestrar la información de registro de un usuario legítimo. Un atacante puede aprovecharse de que normalmente los registros viajan por *UDP*, donde a veces se utilizan autenticaciones básicas enviando los datos del usuario y la contraseña en texto plano. En el caso de encriptar la contraseña, el atacante podría realizar un ataque por fuerza bruta basado en directorio para obtenerla a partir de un intento de conexión escuchado, o bien conociendo el nombre de usuario y realizando intentos de conexión contra el servidor, ayudado por la inoperancia de algunos proveedores que no generan *logs* o si los generan no los controlan. Como resultado, el atacante podría pasar a tener el control de la conexión del usuario, impidiéndole las conexiones, cerrándoselas arbitrariamente y limitándose a escuchar en el medio. En la siguiente figura 2.11 se ilustra este ataque.

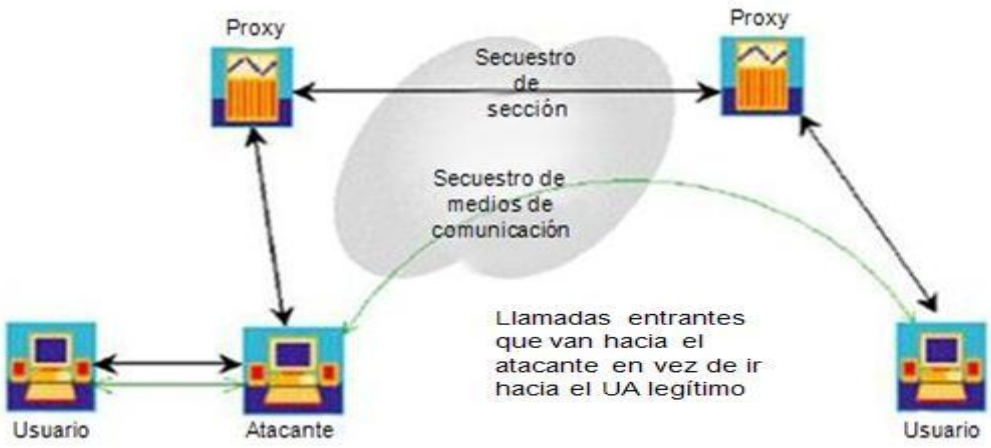


Figura 2.11 Secuestro de registro

2.11.2. Suplantación del proxy

Este ataque ocurre cuando un proxy atacante intercepta las llamadas que se envían desde un UA a un servidor proxy de dominio, ya sea directamente o entrelazados a través de proxy. Con la suplantación del proxy un atacante puede obtener acceso a todos los mensajes SIP, teniendo así el control completo de la llamada. Se puede engañar al UA legítimo y a los servidores proxys para comunicarse con el atacante, y así decidir si las llamadas deben ser enrutadas.

Este ataque puede originarse a través de varios medios, mediante UAs y proxies que comúnmente se comunican a través de *UDP* y no requieren de autenticación fuerte para comunicarse entre si. Esta

situación la podría aprovechar un atacante para ubicarse entre dos proxys o entre un UA y un proxy, realizando otros ataques como *DNS* Spoofing o *ARP* Spoofing, tomando así el control de todas las comunicaciones entre ellos. En la figura 2.12 se muestra dicho ataque.

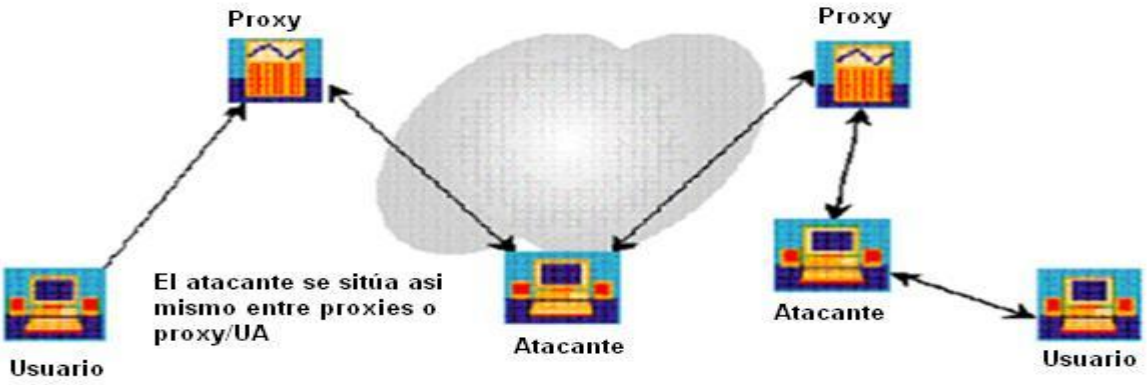


Figura 2.12 Suplantación del proxy

2.11.3. Manipulación del mensaje

La manipulación del mensaje se produce cuando el atacante intercepta y modifica los mensajes que son intercambiados entre los componentes de SIP, debido a que consigue acceso a cualquiera de sus componentes ya sean los UAs o los proxys, mediante otros ataques como son secuestro de registro, suplantación de proxy o un ataque a cualquier componente de confianza que procese mensajes SIP, tales como proxies, gateways o *firewalls*. El atacante al interceptar estos mensajes y modificarlos, puede inhabilitar las conexiones de los usuarios, cambiar las características de las conexiones y producir ataques DoS. En la siguiente figura 2.13 se observa un ejemplo.

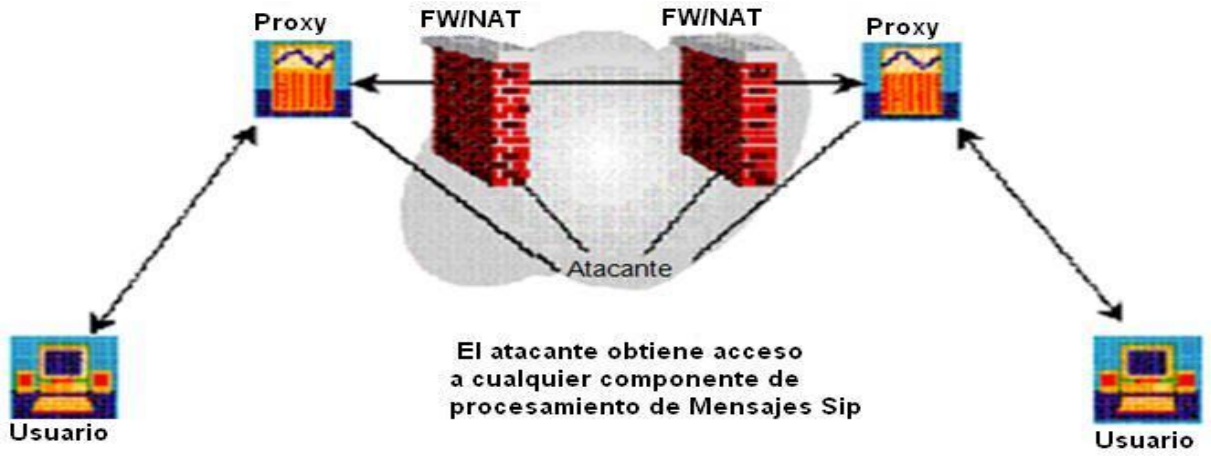


Figura 2.13 Manipulación de mensaje

2.11.4. Derribo de sesiones

El derribo de sesiones consiste en el envío de mensajes BYE que provocan que se cierren las conexiones. Si un atacante conoce la existencia de un UA siempre activo, puede enviarle mensajes BYE para cerrar sus conexiones. Otra modalidad de este ataque consiste en inundar los *firewalls* o proxys con estos mensajes, provocando un ataque de denegación de servicio junto con la desconexión masiva de sesiones. Un aspecto a tener en cuenta es que la desconexión de un UA sin pasar por el proxy supone que la conexión se termine pero sin que el proxy se dé cuenta, por lo que su registro de llamadas, normalmente utilizado luego para la facturación, ya no sería consistente.

Hay otra variante de este tipo de ataque que consiste en enviar mensajes RE-INVITE en vez de BYE para modificar las propiedades de la conexión, por ejemplo para enviar la información multimedia a una dirección de broadcast, produciendo un ataque DoS. En la figura 2.14 se ilustra dicho ataque.

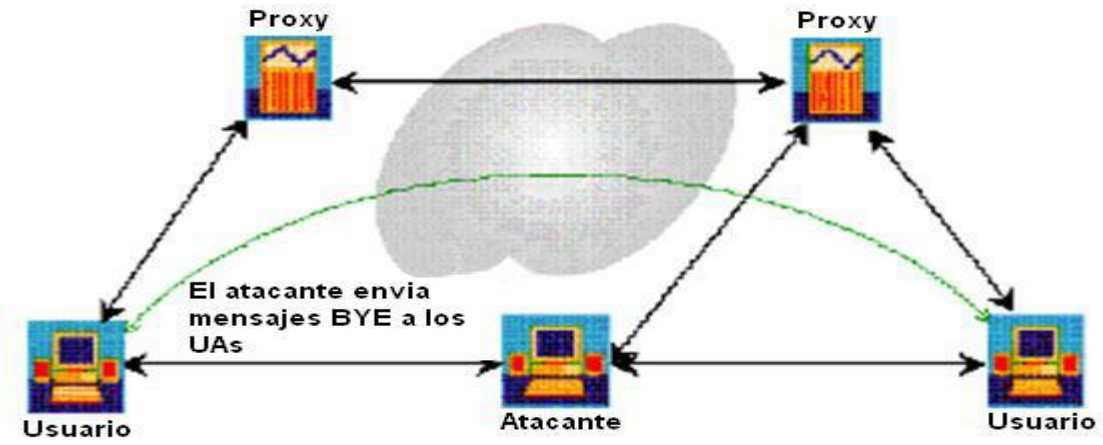


Figura 2.14 Derribo de sesiones

2.11.5. Denegación de servisio(DoS)

Los sistemas de VoIP son más vulnerables a ataques DoS en comparación con otros sistemas de datos, debido a los requerimientos de la Calidad de Servicios. Los ataques DoS contra SIP pueden ocurrir mediante cualquiera de los ataques anteriores donde un usuario malicioso podría provocar un ataque DoS. El principal problema es que al no ser seguras las comunicaciones, los componentes SIP se ven obligados a procesar todos los paquetes, incluso los de posibles atacantes. Los ataques DoS pueden provocarse mediante el envío de paquetes corruptos, manipulación de estados SIP o inundación por mensajes REGISTER o INVITE. A continuación se muestra en la figura 2.15 el ataque.

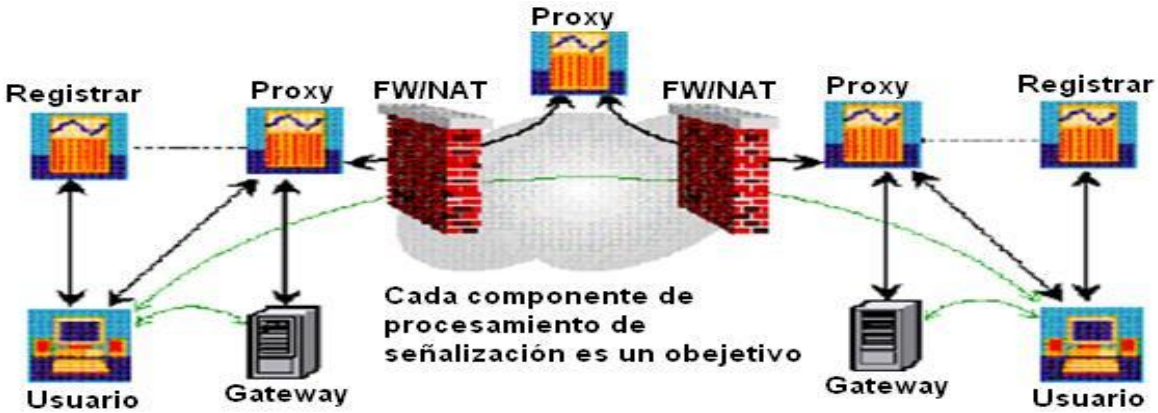


Figura 2.15 Denegación de servicio

2.12. Mecanismos de seguridad existentes para el protocolo SIP

2.12.1. Autenticación en SIP (HTTP Digest)

El protocolo SIP utiliza la autenticación Digest para la autenticación de los clientes. Es un mecanismo simple desarrollado originalmente para *HTTP*, basado en hashes criptográficos que evitan que se envíe la contraseña de los usuarios en texto claro.

La autenticación Digest verifica que las dos partes que se comunican conocen un secreto compartido, que es la contraseña. Cuando un servidor quiere autenticar a un usuario, genera un desafío Digest y se lo manda al usuario. Después de recibir el desafío, el UA le pedirá al usuario el nombre de usuario y la contraseña (si no está preconfigurado), generará la respuesta Digest y la enviará al servidor. Una vez recibida la respuesta Digest, el servidor recalcula el valor de la respuesta para comparar los atributos que da el cliente y el password Si el resultado es idéntico a la respuesta recibida desde el cliente, entonces dicho cliente está autenticado.

Cuando un servidor SIP recibe una petición SIP y quiere verificar la autenticidad del usuario, comprueba si la petición contiene las credenciales de Digest. Si no hay credenciales en la petición SIP, generará una respuesta final negativa e incluirá el desafío Digest en la respuesta. Cuando el cliente recibe la respuesta conteniendo el desafío, debe calcular la respuesta Digest adecuada y enviar la petición de nuevo, esta vez incluyendo las credenciales Digest. El servidor entonces verifica la respuesta Digest y procesa que la petición se ha realizado con éxito. Los Agentes de Usuario SIP utilizan la respuesta “401 Unauthorized” para incluir el desafío Digest.

Una limitación que presenta este mecanismo es que no protege todos los parámetros de la cabecera, lo cual podría provocar fallos de integridad.

2.12.2. Seguridad en la Capa de Transporte (TLS)

TLS es un protocolo que garantiza la privacidad de las comunicaciones entre aplicaciones y sus usuarios en Internet, ya que permite enviar mensajes basados en esta tecnología sobre un canal encriptado de seguridad en la capa de transporte. Cuando se comunican un cliente y un servidor, TLS se encarga de que ninguna tercera parte pueda espiar o modificar la comunicación. TLS se compone de dos capas: el TLS Protocolo de Registro y el TLS Protocolo de Handshake. El TLS Protocolo de

Registro aporta seguridad en la conexión con un método de cifrado como DES (Estándar de Cifrado de Datos), este también puede ser usado sin encriptación. El TLS Protocolo de Handshake permite al servidor y al cliente autentificarse entre ellos y negociar un algoritmo de cifrado y las llaves criptográficas antes de iniciar el intercambio de datos. [4]

Una limitación que presenta TLS es que no puede utilizarse sobre *UDP*, por lo que estamos obligados a utilizar conexiones *TCP* que la mayor parte del tiempo estarán inactivas.

2.12.3. Secure SIP

El mecanismo de seguridad definido en Secure SIP protege los mensajes SIP utilizando canales encriptados con TLS. Originalmente utilizado para securizar las sesiones *HTTP*, TLS también es ahora capaz de proteger la señalización de llamadas SIP contra escuchas no autorizadas o falsificaciones. Secure SIP, que se obtiene ejecutando SIP sobre TLS mediante el método "*hop-by-hop*", proporciona un nivel de seguridad más completo que la autenticación *MD5* básica, sin la carga adicional impuesta por S/MIME.

Secure SIP define el SIPS *URI*, que se emplea del mismo modo que *HTTPS* para proteger las conexiones *HTTP*, garantizando que SIP sobre TLS valide y proteja la conexión entre cada par de saltos, y, por ende, proporciona una conexión segura entre los puntos de la comunicación. En una sesión Secure SIP, el agente usuario SIP del cliente, solicita una sesión TLS al servidor proxy SIP.

Este responde con un certificado público que ha de ser validado por el agente usuario y, a continuación, ambos intercambian claves de sesión para encriptar o desencriptar los datos. En este punto, el servidor proxy SIP contacta con el siguiente salto y negocia de forma similar una sesión TLS, garantizando que se utiliza SIP sobre TLS de principio a fin.

2.12.4. Extensión de Correo Multipropósito Seguro S/MIME

Mediante el uso de S/MIME se pueden encapsular mensajes SIP en cuerpos MIME, aprovechando los mecanismos de seguridad de este para los mensajes del protocolo SIP.

El uso de S/MIME presupone un *PKI* establecido y la implementación de mecanismos necesarios para verificar los certificados. En el caso de SIP, S/MIME normalmente asegura los mensajes *SDP*. La

encriptación de la información en S/MIME se hace de la forma *end-to-end*, encriptando toda la información que no es necesaria que sea vista por los sistemas intermedios.

S/MIME ofrece servicios de seguridad criptográfica en aplicaciones que emplean partes del cuerpo MIME como correo electrónico, *HTTP* y SIP, entre los que cabe señalar los siguientes:

- Autenticación
- Integridad del mensaje y no rechazo del origen (mediante firmas digitales)
- Privacidad y seguridad en los datos (mediante la encriptación).

S/MIME es susceptible a ataques del tipo man-in-the-middle, al igual que otros sistemas basados en clave pública como *SSH*, en caso de que un atacante intercepte el primer intercambio de claves. El algoritmo de cifrado mínimo requerido para la implementación de S/MIME en SIP es el algoritmo 3DES. Se recomienda también, el uso del algoritmo AES.

Hay que tener en cuenta, que incluso con S/MIME hay determinadas cabeceras que deben ir en texto plano ya que son necesarias en puntos intermedios. En estos casos no se puede mantener la confidencialidad de esa información, pero se mantiene la integridad haciendo que esa información vaya también dentro de la parte encriptada, de forma que se pueda comparar posteriormente y detectar posibles modificaciones.

2.12.5. IPSec

Las redes se diseñan normalmente para impedir el acceso no autorizado a datos confidenciales desde fuera de la intranet de la empresa, mediante el cifrado de la información que viaja a través de líneas de comunicación pública. Sin embargo, la mayor parte de las redes manejan las comunicaciones entre los hosts de la red interna como texto sin formato. Con acceso físico a la red y un analizador de protocolos, un usuario no autorizado puede obtener fácilmente datos privados.

IPSec autentifica los equipos y cifra los datos para su transmisión entre hosts en una red, intranet o extranet, incluidas las comunicaciones entre estaciones de trabajo y servidores, y entre servidores. El objetivo principal de IPSec es proporcionar protección a los paquetes *IP*. IPSec está basado en un modelo de seguridad de extremo a extremo, lo que significa que los únicos hosts que tienen que

conocer la protección de IPSec son el que envía y el que recibe. Cada equipo controla la seguridad por sí mismo en su extremo, bajo la hipótesis de que el medio por el que se establece la comunicación no es seguro.

IPSec aumenta la seguridad de los datos de la red mediante:

- La autenticación mutua de los equipos antes del intercambio de datos. IPSec puede utilizar *Kerberos V5* para la autenticación de los usuarios.

- El establecimiento de una asociación de seguridad entre los dos equipos. IPSec se puede implementar para proteger las comunicaciones entre usuarios remotos y redes, entre redes e, incluso, entre equipos clientes dentro de una *LAN*.

- El cifrado de los datos intercambiados mediante Cifrado de datos estándar (DES, 3DES). IPSec usa formatos de paquete *IP* estándar en la autenticación o el cifrado de los datos. Por tanto, los dispositivos de red intermedios, como los enrutadores, no pueden distinguir los paquetes de IPSec de los paquetes *IP* normales.

2.13. Algoritmos criptográficos que se emplean para llevar a cabo la seguridad.

Una de las formas de proteger las comunicaciones basadas en VoIP es la encriptación, tanto de la señal de la llamada (para que las direcciones de teléfono no aparezcan con claridad), como de los paquetes de datos. La encriptación es una técnica que codifica la información de un modo que hace difícil o imposible su lectura, y la decodifica de modo que pueda ser leída nuevamente. A la información codificada se la llama texto cifrado y a la información sin codificar texto claro.

Existen dos tipos fundamentales de algoritmos criptográficos:

Criptografía simétrica o de clave privada: Estos sistemas se caracterizan por emplear la misma clave tanto para cifrar como para descifrar. Ver figura 2.16.

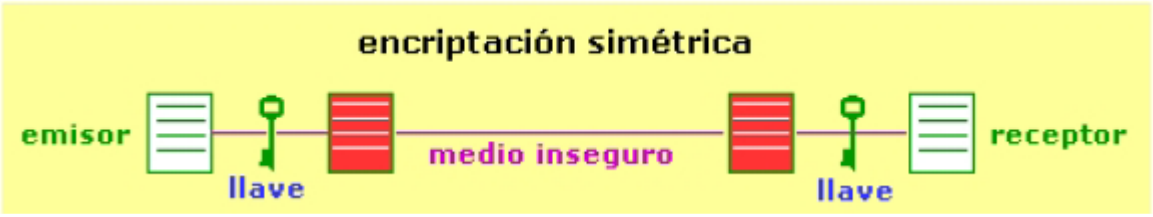


Figura 2.16 Encriptación Simétrica

Toda la seguridad de este sistema, está basada en la llave simétrica, por lo que es misión fundamental tanto del emisor como del receptor, conocer esta clave y mantenerla en secreto. Si la llave cae en manos de terceros, el sistema deja de ser seguro, por lo que habría que desechar dicha llave y generar una nueva. Algunas de las características más destacadas de este tipo de algoritmos son las siguientes:

- A partir del mensaje cifrado, no se puede obtener el mensaje original ni la clave que se ha utilizando, aunque se conozcan todos los detalles del algoritmo criptográfico utilizado.
- Se utiliza la misma clave para cifrar el mensaje original que para descifrar el mensaje codificado.
- Emisor y receptor deben haber acordado una clave común, por medio de un canal de comunicación confidencial antes de poder intercambiar información confidencial por un canal de comunicación inseguro.

Los algoritmos simétricos cifran bloques de texto del documento original, y son más sencillos que los sistemas de clave pública, por lo que sus procesos de encriptación y desencriptación son más rápidos.

Las principales desventajas de los métodos simétricos, son la distribución de las claves, el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.

Criptografía asimétricos o de clave pública: Esta categoría incluye un conjunto de algoritmos criptográficos que utilizan dos claves distintas para cifrar y descifrar el mensaje. Ambas claves tienen una relación matemática especial, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves públicas son diferentes, entonces sus llaves privadas asociadas también lo son, y viceversa, pero esta relación debe

ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la otra. Este es el motivo por el que normalmente estas claves no las elige el usuario, si no que lo hace un algoritmo específico para ello, y suelen ser de gran longitud.

Mientras que la clave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la clave pública es difundida ampliamente por Internet, para que esté al alcance del mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas claves.

La seguridad de esta técnica se basa en que el conocimiento de una de las claves no permite descubrir cuál es la otra clave. En realidad sería necesario conocer todos los números primos grandes para ser capaz de deducir una clave a partir de otra, pero está demostrado que en la práctica se tardarían demasiados años sólo en el proceso de obtención de los número primos grandes. En la siguiente figura 2.17 se puede observar la encriptación asimétrica.

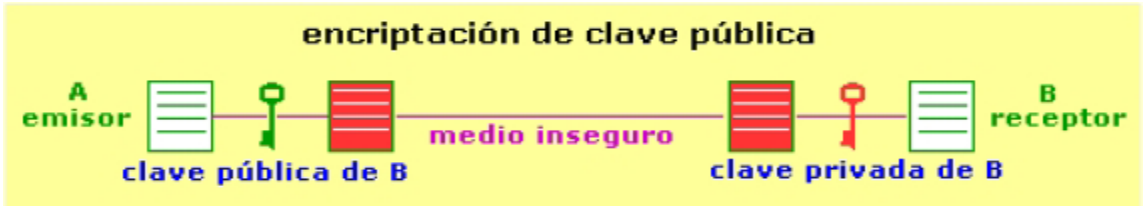


Figura 2.17 Encriptación asimétrica

En este sistema, para enviar un documento con seguridad, el emisor (A) encripta el documento con la clave pública del receptor (B) y lo envía por el medio inseguro. Este documento está totalmente protegido en su viaje, ya que sólo se puede descifrar con la clave privada correspondiente, conocida solamente por B. Al llegar el mensaje cifrado a su destino, el receptor usa su clave privada para obtener el mensaje en claro.

Una variación de este sistema se produce cuando es el emisor A el que encripta un texto con su clave privada, enviando por el medio inseguro tanto el mensaje en claro como el cifrado. Así, cualquier receptor B del mismo puede comprobar que el emisor ha sido A, y no otro que lo suplante, con tan sólo descifrar el texto cifrado con la clave pública de A y comprobar que coincide con el texto sin cifrar. Como sólo A conoce su clave privada, B puede estar seguro de la autenticidad del emisor del mensaje. Este sistema de autentificación de denomina firma digital. Ver figura 2.18.

Figura 2.18 Firma digital

Alguna de las características principales de este tipo de algoritmos son las siguientes:

- Se utilizan una pareja de claves denominadas clave pública y clave privada, pero a partir de la clave pública no es posible descubrir la clave privada.
- A partir del mensaje cifrado no se puede obtener el mensaje original, aunque se conozcan todos los detalles del algoritmo criptográfico utilizado y aunque se conozca la clave pública utilizada para cifrarlo.
- Emisor y receptor no requieren establecer ningún acuerdo sobre la clave a utilizar. El emisor se limita a obtener una copia de la clave pública del receptor, lo cual se puede realizar, en principio, por cualquier medio de comunicación aunque sea inseguro.

La principal ventaja de los sistemas de clave pública frente a los simétricos es que la clave pública y el algoritmo de cifrado son o pueden ser de dominio público y que no es necesario poner en peligro la clave privada en tránsito por los medios inseguros, ya que ésta está siempre oculta y en poder únicamente de su propietario. Como desventaja, los sistemas de clave pública dificultan la implementación del sistema y son mucho más lentos que los simétricos.

En la práctica se emplea una combinación de estos dos tipos de criptosistemas, puesto que los segundos presentan el inconveniente de ser computacionalmente mucho más costosos que los primeros. En el mundo real se codifican los mensajes (largos) mediante algoritmos simétricos, que suelen ser muy eficientes, y luego se hace uso de la criptografía asimétrica para codificar las claves simétricas (cortas).

2.13.1. Estándar de Cifrado de Datos (DES)

DES es un algoritmo de clave privada, tanto de cifrado como de descifrado desarrollado a partir de otro algoritmo conocido como Lucifer que utilizaba claves de 112 bits y que fueron reducidas a 56 bits en el algoritmo DES. La reducción del tamaño de las claves originó controversia debido a que sé pensó que habían debilitado intencionadamente el algoritmo del DES para poder desencriptarlo, pero no fue así, el diseño del algoritmo se realizo de forma que fuera resistente a criptoanálisis, pero lo suficientemente sencillo como para poder ser implementado en un circuito electrónico con la tecnología existente en esos momentos. Por tanto, el algoritmo DES se diseño de forma que no pudiera ser descifrado por criptoanálisis, pero si puede ser descifrado probando todas las claves posibles, asumiendo que se cuenta con el hardware adecuado.

En la figura 2.19 se muestra un esbozo del algoritmo DES. El texto normal se cifra en bloques de 64 bits, produciendo 64 bits de texto cifrado. El algoritmo, que se parametriza con la clave de 56 bits, tiene 19 etapas diferentes.

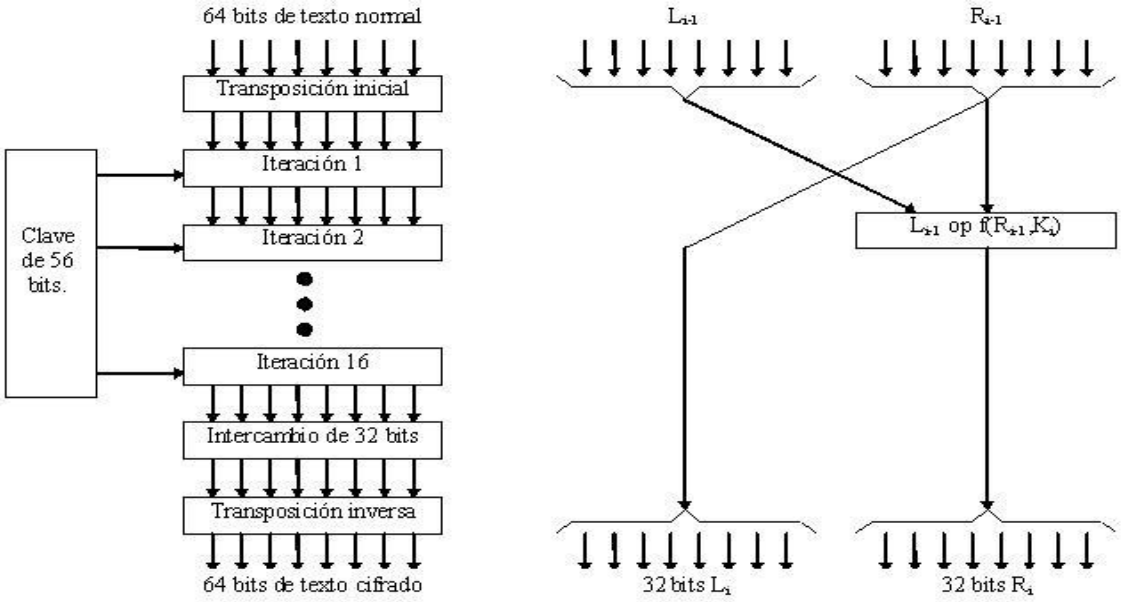


Figura 2.19 Algoritmo del cifrado DES

La primera etapa es una transposición, independiente de la clave, del texto normal de 64 bits. La última etapa es el inverso exacto de esta transposición. La etapa previa a la última intercambia los 32 bits de la izquierda y los 32 bits de la derecha. Las 16 etapas restantes son funcionalmente idénticas, pero se parametriza mediante diferentes funciones de la clave. El algoritmo se ha diseñado para permitir que el descifrado se haga con la misma clave que el cifrado, simplemente ejecutando los pasos en orden inverso. Cada una de las 16 etapas intermedias toma dos entradas de 32 bits y produce dos salidas de 32 bits. La salida de la izquierda es simplemente una copia de la entrada de la derecha. La salida de la derecha es el or exclusivo a nivel de bit de la entrada izquierda y una función de la entrada derecha y la clave de esta etapa Ki. Toda la complejidad reside en esta función.

La función consiste en cuatro pasos, ejecutados en secuencia. Primero se construye un número de 48 bits, E, expandiendo el Ri-1 de 32 bits según una regla fija de transposición y duplicación. Después, se aplica un OR exclusivo a E y Ki. Esta salida entonces se divide en ocho grupos de 6 bits, cada uno de los cuales se alimenta a una caja S distinta. Cada una de las 64 entradas posibles a la caja S se transforma en una salida de 4 bits. Por último estos 8x4 bits se pasan a través de una caja P.

En cada una de las 16 iteraciones, se usa una clave diferente. Antes de iniciarse el algoritmo, se aplica una transposición de 56 bits a la clave. Justo antes de cada iteración, la clave se divide en dos unidades de 28 bits, cada una de las cuales se gira hacia la izquierda una cantidad de bits dependiente del número de iteración Ki se deriva de esta clave girada aplicándole otra transposición de 56 bits. En cada vuelta (o iteración) se extrae y permuta de los 56 bits un subgrupo de 48 bits diferente.

Como la clave efectiva es de 56 bits, son posibles un total de 2 elevado a 56 = 72.057.594.037.927.936 clave posible, es decir, unos 72.000 billones de claves, por lo que la ruptura del sistema por fuerza bruta o diccionario es sumamente improbable, aunque no imposible si se dispone de suerte y una gran potencia de cálculo.

Este algoritmo es muy rápido y fácil de implementar, pero una de las desventajas que tiene es que emplea una clave demasiado corta, lo cual hace que con el avance actual de los ordenadores, los ataques por la fuerza bruta se puedan llevar a cabo. Hoy en día se considera un algoritmo poco robusto, no permite longitud de clave variable, con lo que sus posibilidades de configuración son muy limitadas, además la seguridad del sistema se ve reducida considerablemente si se conoce un número suficiente de textos elegidos, ya que existe un sistema matemático, llamado Criptoanálisis Diferencial, que puede en ese caso romper el sistema en 2 elevado a 47 iteraciones.

2.13.2. Cifrado Triple DES

Como se ha mencionado anteriormente el sistema DES se considera poco practico, debido a la corta longitud de su clave. Para solventar este problema y continuar utilizando DES se creó el sistema Triple DES basado en 3 iteraciones sucesivas del algoritmo DES, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con DES simple. Se ilustra en la figura 2.20.

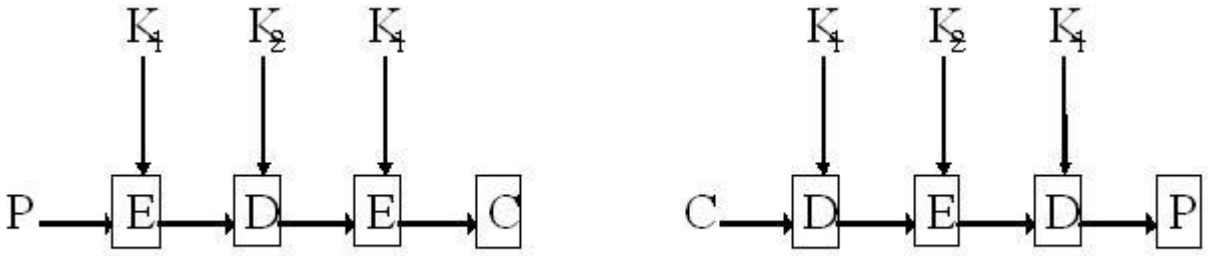


Figura 2.20 Esquema del cifrado DES triple

Aquí se usan dos claves y tres etapas. En la primera etapa, el texto normal se cifra con K1. En la segunda el algoritmo DES se ejecuta en modo de descifrado, usando K2 como clave. Por último, se hace otro cifrado usando K1. El hecho de que se usen dos claves y en modo EDE (cifrado, descifrado, cifrado) en lugar de EEE (cifrado, cifrado, cifrado) es debido a dos motivos:

- En primer lugar, los criptógrafos admiten que 112 bits son suficientes para las aplicaciones comerciales por ahora. Subir a 168 bits (3 claves) simplemente agregaría carga extra innecesaria al administrar y transportar otra clave.

- En segundo lugar, la razón de cifrar, descifrar y luego cifrar de nuevo es la compatibilidad con los sistemas DES de una sola clave. Tanto las funciones de cifrado como de descifrado son correspondencias entre números de 64 bits. Desde el punto de vista criptográfico, las dos correspondencias son igualmente robustas. Sin embargo, usando EDE en lugar de EEE, una computadora que usa cifrado triple puede hablar con otra que usa cifrado sencillo simplemente estableciendo K1=K2. Esta propiedad permite la introducción gradual del cifrado triple.

2.13.3. Algoritmo Internacional de Cifrado de Datos (IDEA)

Es un algoritmo de encriptado de clave privada, usa criptografía de bloque y se suele considerar como muy seguro. Opera con bloques de 64 bits usando una clave de 128 bits, lo que por el momento lo hace inmune a los ataques de fuerza bruta así como al criptoanálisis diferencial para su desencriptado.

La estructura básica del algoritmo se asemeja al algoritmo DES en cuanto a la alteración de bloques de entrada de texto normal de 64 bits en una secuencia de iteraciones parametrizadas para producir bloques de salida de texto cifrado de 64 bits, como se puede ver en la figura 2.21.

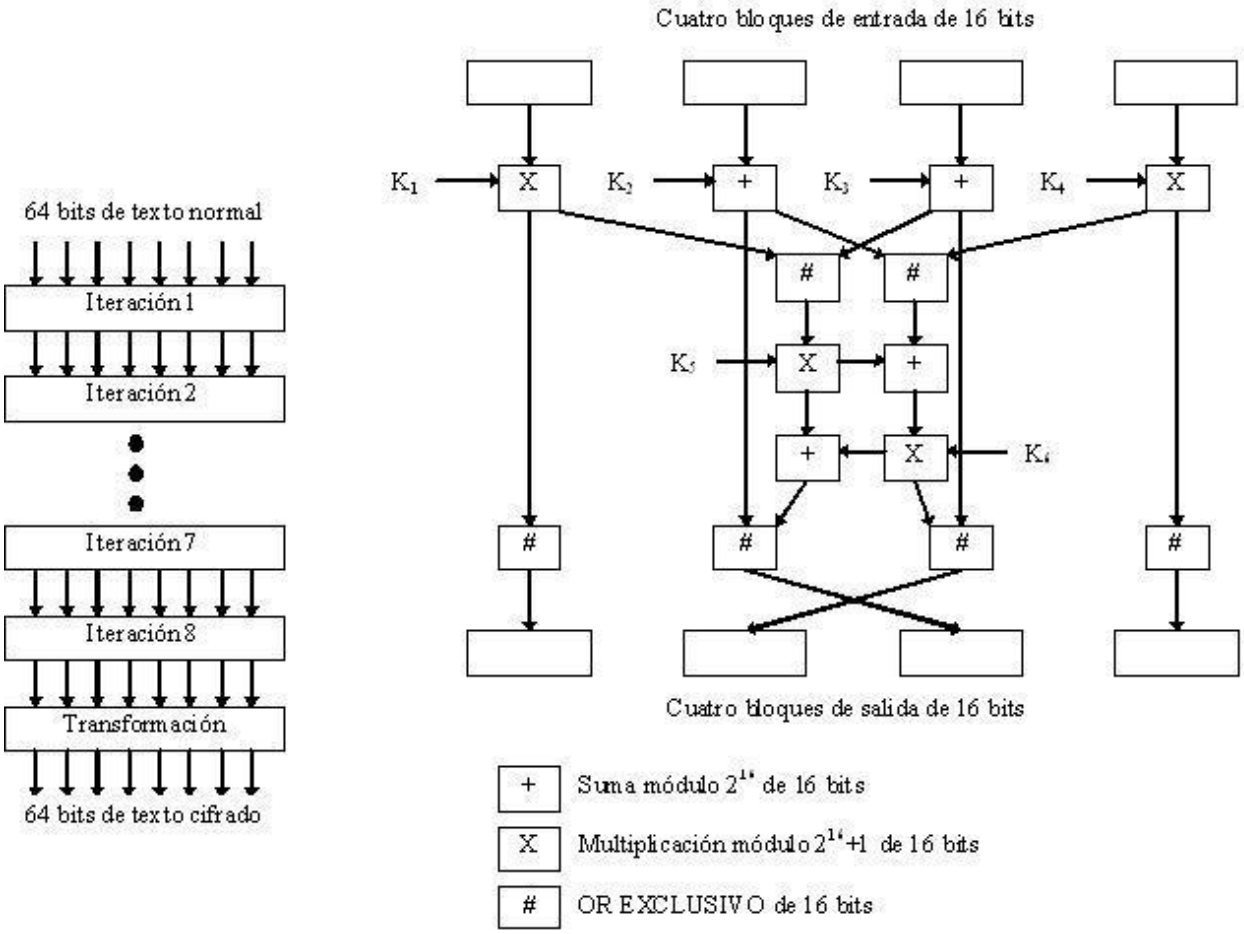


Figura 2.21 Algoritmo del cifrado IDEA

Dada la extensa alteración de bits estos bloques de entrada se separan en 4 subbloques, cada una de 16 bits. Dado que por cada iteración, los bits de salida dependen de cada uno de los bits de entrada, basta con sólo 8 iteraciones. Como con todos los cifrados de bloque, el algoritmo IDEA también puede usarse en el modo de realimentación de cifrado.

El algoritmo IDEA usa tres operaciones, todas sobre números sin signo de 16 bits. Estas operaciones son un or exclusivo, suma módulo 2^16 y multiplicación módulo 2^16+1. Las tres operaciones se pueden efectuar fácilmente en una microcomputadora de 16 bits ignorando las partes de orden mayor de los resultados. Las operaciones tienen la propiedad de que ningunos de los dos pares obedecen la ley asociativa ni la ley distributiva, dificultando el criptoanálisis.

La clave de 128 bits se utiliza para generar 52 subclaves para encriptar y 52 para desencriptar, cada una de 16 bits. Se utilizan 6 subclaves por cada una de las 8 iteraciones y 4 para la transformación final. El método de generación de las subclaves es muy regular, lo que puede presentarse como una debilidad del algoritmo.

Una de las ventajas de trabajar con IDEA, es que el algoritmo para desencriptar es el mismo que para encriptar. Se utilizan en cada caso diferentes subclaves. Las claves de desencriptado se hallan a partir de las subclaves de encriptado, pues no hay ningún método directo de cálculo de las mismas a partir de la clave del algoritmo. Es fundamental para el correcto desencriptado de los datos, agrupar correctamente las palabras que se reciben de la misma forma en que fueron agrupados a la hora de encriptar.

2.13.4. Estándar Avansado de Encriptación (AES)

El algoritmo AES es un estándar de criptografía simétrica. Su diseño permite la utilización de claves de sistemas con longitud variable siempre que sea múltiplo de 4 bytes. La longitud de las claves utilizadas por defecto son 128,192 y 256 bits. De la misma manera el algoritmo permite la utilización de bloques de información con un tamaño variable siempre que sea múltiplo de 4 bytes, siendo el tamaño mínimo recomendado de 128 bits.

La descripción de AES consiste de dos partes, en describir el proceso de cifrado, y el proceso de generación de las subclaves o extensión de la clave K. La estructura del algoritmo, está formado por un conjunto de rondas, o sea un conjunto de reiteraciones de 4 funciones matemáticas diferentes e

invertibles. Por tanto, el algoritmo se basa en aplicar un número de rondas determinado a una información en claro para producir una información cifrada. La información generada por cada función es un resultado intermedio, que se conoce como Estado.

El algoritmo representa el Estado como una matriz rectangular de bytes, que posee 4 filas y Nb columnas. Siendo el número de columnas Nb en función del tamaño del bloque.

Nb= tamaño del bloque utilizado en bits/32

De esta manera, si se tiene un bloque con 128 bits se tendría una matriz de cuatro filas y Nb = 128/32= 4 columnas quedando una matriz de la forma mostrada en la tabla 2.4.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Tabla 2.4 Ejemplo de matriz de estado con *Nb*=4 (128 bits)

La clave del sistema se representa con una estructura análoga a la del Estado, es decir, se representa mediante una matriz rectangular de bytes de 4 filas y Nk columnas. Siendo el número de columnas Nk en función del tamaño de la clave.

Nk = tamaño de las claves en bits/32

De esta manera, al igual que ocurría con el tamaño del bloque, si se tiene una clave con 128 bits se dispondría de una matriz de cuatro filas y Nr = 128/32 = 4 columnas, quedando una matriz de la forma mostrada en la tabla 2.5.

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Tabla 2.5 Ejemplo de clave con *Nk*=4 (128 bits).

El número de iteraciones o vueltas de las cuatro transformaciones sobre la matriz de Estado Intermedio depende de la versión del algoritmo que se utilice. Para tamaños de bloques y claves entre 128 y 256 bits (con incrementos de 32 bits) el número de vueltas Nr es determinado por la siguiente expresión: Nr = máx (Nk, Nb) + 6.

Proceso de Cifrado: El proceso de cifrado se realiza de la siguiente manera, como se puede observar en al figura 2.22.

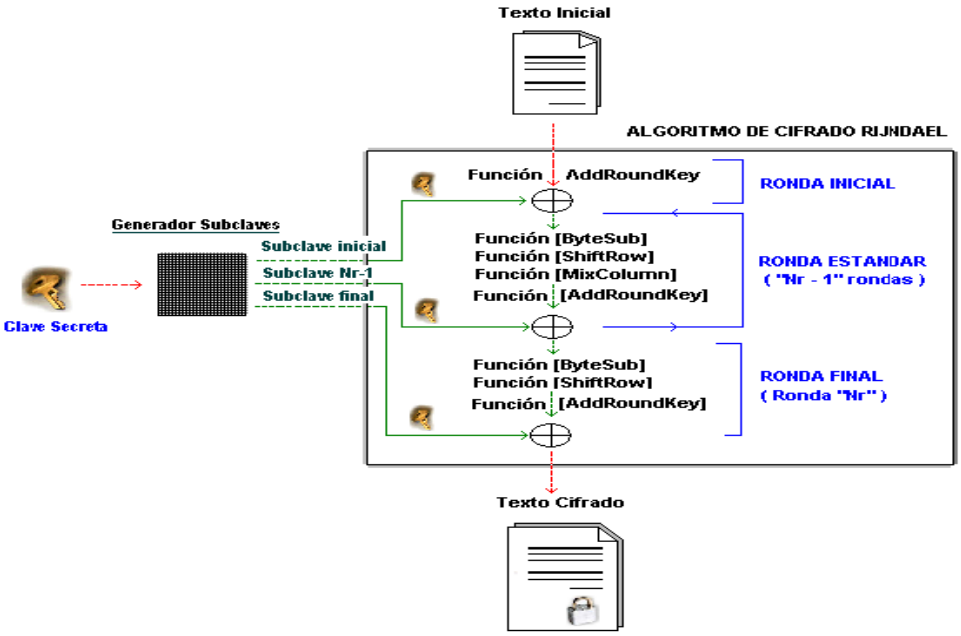


Figura 2.22 Cifrado del algoritmo AES

Se puede ver en el gráfico que el proceso de cifrado consiste en la aplicación de cuatro funciones matemáticas invertibles sobre la información que se desea cifrar. Las transformaciones se realizan de forma reiterativa para cada ronda o vuelta definida.

Las funciones matemáticas realizadas están diseñadas para proporcionar resistencia frente a criptoanálisis lineal y diferencial. Estas son:

- Función AddRoundKey.
- Función ByteSub.
- Función ShiftRow.
- Función MixColumn.

La información a cifrar se va transformando en la matriz de estado. Esta matriz de estado se introduce al cifrador, y sufre una primera transformación en la ronda inicial, que consiste en una operación or-exclusiva (AddRoundKey) entre una subclave generada y la matriz de Estado. A continuación, a la matriz de Estado resultante se le aplican cuatro transformaciones invertibles, repitiéndose este proceso Nr - 1 veces, en lo que se conoce como Ronda Estándar. Finalmente, se le aplica una última ronda o vuelta a la matriz de Estado resultante de las Nr - 1 rondas anteriores, aplicando las funciones ByteSub, ShiftRow y AddRoundKey en este orden. El resultado de la ronda final produce el bloque cifrado deseado.

Proceso de Descifrado: El proceso de descifrado es muy similar al cifrado, sólo hay que hacer el proceso inverso, es decir, invertir el orden de todas las operaciones realizadas, y hacer las transformaciones inversas. Destacar que en este proceso, las subclaves utilizadas, van desde la última generada en el proceso de cifrado hasta la primera (que corresponderá con bytes de la clave elegida para cifrar). En la figura 2.23 se muestra el proceso de descifrado del AES.

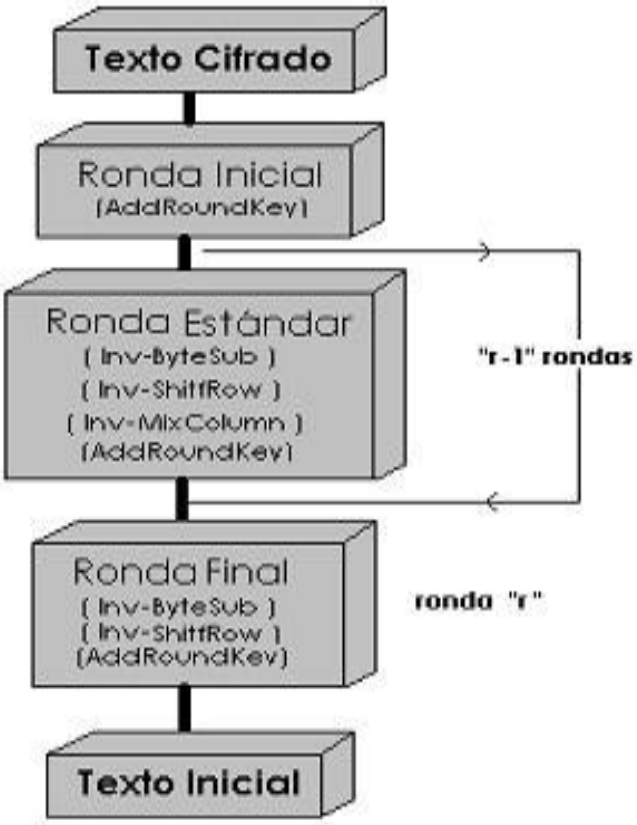


Figura 2.23 Proceso de Descifrado de AES

La fortaleza del sistema AES dependerá en gran medida de la longitud de la clave. Un atacante que desee realizar un ataque por fuerza bruta deberá tener en cuenta las siguientes consideraciones:

Si el atacante dispone solo de un bloque cifrado, debería cifrar con todas las claves posibles todos los bloques en claro posibles, para ir comparando con el bloque cifrado. Si se considera que el tamaño del bloque es de v bits y el tamaño de la clave es n bits. El atacante debería probar para cada clave 2^v bloques posibles y repetir este proceso para todas las claves 2^n. De esta forma aunque fuera factible el cálculo, se encontraría un número muy elevado de diferentes parejas clave-texto en claro que producen el mismo bloque cifrado. Así el atacante no sabría como identificar cual de todas esas parejas es la valida.

En el caso de que el atacante posee el texto en claro y el texto cifrado, puede hacer lo que se conoce como un ataque de fuerza bruta. Es decir cifrar el texto en claro con todas las claves posibles hasta producir un resultado que coincida con el texto cifrado.

En esta situación el atacante debería aplicar el algoritmo n veces al texto en claro, para así estar seguro de que ha obtenido la clave. Para una clave de n bits el atacante necesita aplicar 2^(n-1) veces el algoritmo. [7]

2.13.5. CAST-128

Es un algoritmo del mismo tipo que DES. Es un criptosistema *SPN* que proporciona una buena resistencia contra ataques diferenciales, lineales y related-key, Además de poseer algunas otras propiedades criptográficas interesantes (avalancha, propiedad de no complementación, ausencia de claves débiles y semidébiles), lo que hace que sea un buen candidato para su uso con propósito general para la comunidad de Internet donde se necesite un algoritmo criptográficamente fuerte y disponible libremente. Pertenece a la clase de algoritmos denominada como cifrados Feistel, y su mecanismo, de cuatro pasos, es similar al DES.

CAST-128 es un cifrador de 12 o 16 rondas, que se basa en la red de Feistel con bloques de 64 bits y tamaños de clave entre 40 y 128 bits (pero con solo incrementos de 8 bits). Las 16 rondas completas se usan cuando la clave tiene un tamaño mayor de 80 bits. Incluye unas largas Cajas S de 8x32 bits basadas en funciones bent, rotaciones dependientes de clave, adición y sustracción modular y operaciones xor. Hay tres tipos alternativos de funciones de ronda, pero son de una estructura similar y se diferencian sólo en la elección del tipo exacto de operación (xor, adición o sustracción) en varios puntos.

2.13.6. Blowfish

Es un algoritmo de encriptación que puede usarse como sustituto de DES y de IDEA. Es simétrico y encripta en bloques, con una clave de longitud variable, desde 32 bits hasta 448 bits. Fue diseñado como una alternativa a los algoritmos existentes entonces, y con procesadores de 32 bits lo que lo hace significativamente más rápido que DES.

Este algoritmo, que cifra datos en bloques de 64 bits al mismo tiempo, es dividido en dos partes: claves de expansión y cifrado de datos. Las claves de expansión convierten una clave de más de 448 bits en varias subclaves que totalizan 4168 bytes. El cifrado de datos consiste en una función simple que permite 16 iteraciones, en cada una de ellas se realiza una permutación de la clave y una sustitución de la clave y los datos. Este algoritmo realiza operaciones básicas como el or exclusivo y suma módulo 2^32.

A pesar de utilizar un tamaño de bloque pequeño, que podría facilitar su vulnerabilidad al procesar textos largos, se considera un algoritmo seguro y su fortaleza puede variarse según la longitud de la clave.

2.13.7. Algoritmo Diminuto de Cifrado (TEA)

Es un algoritmo de encriptación rápido y eficiente. Consiste en un cifrado Feistel que usa operaciones de grupos algebraicos mixtos (ortogonales), XOR y sumas en este caso. Encripta bloques de 64 bits usando una clave de 128 bits. Parece altamente resistente al criptoanálisis diferencial y consigue difusión total (una diferencia de un bit en el mensaje original causa aproximadamente 32 bits de diferencia en el mensaje cifrado) en solamente 6 pasos. Se estima que TEA es tan seguro como IDEA. Usa la misma técnica de grupos algebraicos mixtos que IDEA, pero es mucho más simple y por tanto, más rápido. Además es de dominio público, al contrario que IDEA. Existen además varias variantes de este algoritmo, que solucionan problemas menores de éste, principalmente el de claves equivalentes, que hace que el algoritmo TEA original tenga 126 bits efectivos de seguridad en vez de 128, así como un posible ataque de claves relacionadas que requiere poder encriptar 2^23 textos determinados bajo dos claves relacionadas (la segunda depende de la primera). [4]

2.13.8. Rivest-Shamir-Adleman (RSA)

Es el algoritmo de encriptación y autentificación más comúnmente usado. Los mensajes enviados usando el algoritmo RSA se representan mediante números y su funcionamiento se basa en el producto de dos números primos grandes (mayores que 10100) elegidos al azar para conformar la clave de descifrado, y a través de operaciones adicionales obtener un par de números que constituyen la clave pública y otro número que constituye la clave privada. Una vez que se han obtenido las claves, los números primos originales, ya no son necesarios para nada, y se descartan.

Se necesitan tanto las claves públicas como las privadas, para encriptar y desencriptar, pero solamente el dueño de la clave privada lo necesitará. Usando el sistema RSA, la clave privada nunca necesitará ser enviada. La clave privada se usa para desencriptar el código que ha sido encriptado con la clave pública. Por tanto, para enviar un mensaje a alguien, hay que conocer su clave pública, pero no su clave privada. Al recibir el mensaje, se necesitará la clave privada para desencriptarlo. También se puede usar para autentificar un mensaje, firmando con la clave privada un certificado digital. [4]

La seguridad de este algoritmo, reside en la dificultad que supone la factorización de un número compuesto por factores primos muy grandes. Si un criptoanalista fuera capaz de encontrar los factores primos sería capaz también de determinar la clave privada y, por lo tanto, descifrar el mensaje.

Sin embargo, el algoritmo RSA es demasiado lento para poder cifrar grandes volúmenes de datos, por lo cual suele usarse para distribuir claves de sesión de una sola vez para su uso con los algoritmos DES, IDEA u otros semejantes.

2.13.9. Algoritmo de ElGamal

El algoritmo ElGamal fue diseñado en un principio para producir firmas digitales, pero posteriormente se extendió también para codificar mensajes. Se basa en el problema de los logaritmos discretos, que está íntimamente relacionado con el de la factorización, y en el de Diffie-Hellman.

Para generar un par de claves, se escoge un número primo n y dos números aleatorios p y x menores que n. Se calcula entonces y = p^ x (mod n).

La clave pública es (p, y, n), mientras que la clave privada es x.

Escogiendo n primo, garantizamos que sea cual sea el valor de p, el conjunto {p, p2, p3…} es una permutación del conjunto {1, 2,… n-1} lo suficientemente grande.

Firmas Digitales de ElGamal: Para firmar un mensaje m basta con escoger un número k aleatorio, tal que mcd (k, n-1) =1, y calcula:

a = p^k (mod n)

b = (m - xa) k^-1 (mod (n - 1))

La firma la constituye el par (a, b). En cuanto al valor k, debe mantenerse en secreto y ser diferente cada vez. La firma se verifica comprobando que:

y^a a^b = p^m (mod n)

Codificación de ElGamal: Para codificar el mensaje m se escoge primero un número aleatorio k primo relativo con (n-1), que también será mantenido en secreto. Calculamos entonces las siguientes expresiones

a = p^k (mod n)

b = y^k m (mod n)

El par (a, b) es el texto cifrado, de doble longitud que el texto original. Para decodificar se calcula:

m = b * a-x (mod n).

2.13.10. Privacidad Bastante Buena (PGP)

PGP es un sistema de criptografía híbrido, que usa una combinación de funciones tomadas de la criptografía de clave pública y de la criptografía simétrica. Cuando un usuario cifra un texto con PGP, los datos primero se comprimen. Esta compresión de datos permite reducir el tiempo de transmisión a través del canal de comunicación, ahorra espacio en disco y, lo más importante, aumenta la seguridad criptográfica. La mayoría de los criptoanalistas sacan provecho de los modelos encontrados en formato de sólo texto para descubrir el cifrado. La compresión reduce estos modelos de sólo texto y mejora considerablemente su resistencia a los criptoanalistas.

El cifrado se realiza, principalmente, en dos fases:

- PGP crea una clave secreta IDEA en forma aleatoria y cifra los datos con esta clave.
- El PGP cifra la clave secreta IDEA y la envía usando la clave pública RSA del receptor.

El descifrado también se produce en dos fases:

- PGP descifra la clave secreta IDEA usando la clave privada RSA.
- PGP descifra los datos con la clave secreta IDEA obtenida previamente.

El método de cifrado, combina la fácil utilización del cifrado de la clave pública con la velocidad del cifrado convencional. El cifrado convencional es aproximadamente 1000 veces más rápido que los algoritmos de cifrado de clave pública. El cifrado de clave pública resuelve el problema de la distribución de la clave. Combinados, estos dos métodos, mejoran el rendimiento y la administración de las claves, sin poner el peligro la seguridad.

PGP ofrece las siguientes funciones:

- **Firmas digitales y verificación de la integridad de los mensajes:** Función que se basa en el uso simultáneo de la función hash (MD5) y del sistema RSA. La función MD5 condensa el mensaje y produce un resultado de 128 bits que después se cifra, gracias al algoritmo RSA, por la clave privada del emisor.

- **Cifrado de archivos locales:** Función que utiliza el algoritmo IDEA.

- **Generación de claves públicas o privadas:** Cada usuario cifra su mensaje mediante las claves privadas IDEA. La transferencia de las claves electrónicas IDEA utiliza el sistema RSA. Por lo tanto, PGP ofrece dispositivos para la generación de claves adaptados al sistema. El tamaño de las claves RSA se propone de acuerdo con varios niveles de seguridad: 512, 768, 1024 ó 1280 bits.

- **Administración de claves:** Función responsable de la distribución de la clave pública del usuario a los remitentes que desean enviarle mensajes cifrados.

- **Certificación de claves:** Esta función permite agregar un sello digital que garantice la autenticidad de las claves públicas. Es una característica original de PGP, que basa su confianza en una noción de proximidad social en lugar de una entidad de certificación central.

- **Revocación, desactivación y registro de claves:** Función que permite producir certificados de revocación. [8]

2.14. Conclusiones parciales

El protocolo SIP se está consolidando como la opción preferida de los proveedores de telefonía *IP* por su simplicidad, escalabilidad y flexibilidad en el desarrollo de servicios frente a otras alternativas, además de que ofrece ventajas en cuanto a dispositivos y servicios que incorpora con respecto a los demás protocolos existentes para la señalización. Lograr su seguridad es un punto primordial que no se debe tomar a la ligera, ya que este protocolo es susceptible a algunos ataques, que pueden comprometer la integridad de los mensajes.

Los mecanismos de seguridad existentes presentan limitaciones que hacen que no se garantice en gran medida una alta seguridad en el establecimiento de una sesión SIP, por lo que es necesario mejorar la seguridad de este protocolo.

CAPITULO 3: PROPUESTAS DE SEGURIDAD PARA EL PROTOCOLO SIP

3.1. Introducción

La seguridad del protocolo SIP es un aspecto fundamental a la hora de establecer y finalizar una sesión. Los epígrafes que a continuación se presentan, describen dos propuestas para mejorar dicha seguridad ya sea para la comunicación de Terminal a Terminal, como de Proxy SIP a Proxy SIP.

3.2. Selección del algoritmo criptográfico

El algoritmo criptográfico que se propone para garantizar la seguridad del protocolo SIP es el AES. Este fue escogido principalmente por ser un algoritmo simétrico siendo el proceso de encriptar/desencriptar mucho más rápido que en los asimétricos. Este es un detalle que se ha tenido en cuenta ya que la VoIP requiere una alta inmediatez.

En comparación con su predecesor DES, es rápido tanto en software como en hardware, es relativamente fácil de implementar, y requiere poca memoria para efectuar el proceso de encriptar/desencriptar. También proporciona mayor rapidez y menor costo computacional que el 3DES, pudiendo ser implementados en equipos con bajo requerimiento de memoria, como puede ser un teléfono VoIP. Es un cifrado por bloques no una red de sustitución/permutación. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala, es decir es uno de los más potentes y más utilizados a nivel mundial, por su gran seguridad y estabilidad.

Una de las ventajas que presenta el algoritmo AES con respecto a otros algoritmos simétricos es la longitud de la clave, tiene niveles de seguridad con claves de 128, 192, y 256 bits, lo que demuestra que la fortaleza del sistema dependerá de la longitud de esta.

Una desventaja del uso de este algoritmo no está asociada a su seguridad sino a la hora de distribuir la llave secreta, pues como es un algoritmo simétrico, es la misma para encriptar/desencriptar. Debido a esto no se debe utilizar la variante de intercambiar la llave por la red ya que un atacante puede interceptarla y así obtener el texto claro. Para resolver este problema el administrador de la red debe insertar la llave manualmente en cada uno de los dispositivos que hagan uso de dicho algoritmo criptográfico propuesto. En caso de que en algunos de los dispositivos se cambie, se debe actualizar con esta a los demás dispositivos.

3.3. Selección del lenguaje de programación

El lenguaje de programación que se propone a utilizar para la realización de la aplicación es C, debido a las caracteristicas que este presenta en comparación con otros lenguajes, entre las cuales definen a este como un lenguaje versátil. Además C puede compilar el código para varios sistemas operativos haciendo la aplicación multiplataforma. Ofrece el acceso a memoria de bajo nivel mediante el uso de punteros. Tiene también un sistema de tipos que impide operaciones sin sentido, además usa un lenguaje de preprocesador, para tareas como definir macros e incluir múltiples ficheros de código fuente.

Este lenguaje presenta otras ventajas como el hecho de que no requiere de ningún software adicional para ser ejecutado, pues otros como el código de C# requiere el framework de .NET, y Java requiere la maquina virtual. Se pudiera usar otros como Delphy o Turbo Pascal pero C es más robusto y más óptimo que estos ya que es mas orientado al trabajo con el hardware.

3.4. Propuesta 1. Seguridad en el protocolo SIP de Terminal a Terminal desde su inicio hasta la finalización de la sesión.

El objetivo de esta propuesta es tratar de lograr la máxima seguridad posible desde el establecimiento hasta la finalización de una sesión SIP. Esta seguridad estaría presente tanto para la terminal que quiera establecer una sesión con otra terminal dentro de una misma red, como para la que se encuentre fuera de ella.

Para lograr lo expuesto anteriormente, se propone realizar una aplicación implementada en C que va a contener además del algoritmo criptográfico simétrico AES otras funcionalidades, como la de agregarle al protocolo SIP un identificador de 3 caracteres formado cada uno por 8 bits. Este identificador se utilizaría como prefijo del protocolo SIP cada vez que sea encriptado y se encontraría en un fichero de almacenamiento en la aplicación.

La aplicación servidora es la que estaría ejecutándose en cada servidor SIP, ya sea proxy, de registro, de redireccionamiento o de localización. Esta estaría escuchando por un puerto diferente al 5060, que sería el mismo que utilizarían las demás aplicaciones para comunicarse entre sí. La comunicación de la aplicación con las aplicaciones SIP en los servidores y terminales sería por el puerto 5060 que es el nativo para este protocolo. En la aplicación servidora se tendría un registro de las terminales que

tengan activado o no el modo seguro, estando inicialmente todas en modo inseguro. Este sólo se cambiaría cuando en la aplicación terminal sea activado.

La aplicación servidora al recibir un mensaje SIP verificaría si está encriptado o no, comparando los primeros 3 caracteres con el identificador que tiene almacenado, si la comparación es válida desencriptaría a partir del bit 24 y pasaría el mensaje desencriptado al servidor SIP. En caso de que no fuera válida, entonces esta verificaría el modo en que se encuentra el emisor, si está en modo inseguro lo tomaría como texto claro y si no, el mensaje no es compatible con el formato esperado por lo que se desecharía.

En el momento en que el servidor envía un mensaje, este pasaría por la aplicación la cual verificaría si el destino es seguro o no. Así se crearía una condición en esta, para que todos los mensajes de respuesta por parte del servidor SIP a las terminales que estén en modo inseguro, se le envíen sin encriptar y en caso de estar en modo seguro encriptados.

En el caso de las terminales, si es una PC la aplicación se ejecutaría cuando se ejecute el *softphone* **[Ver Anexo figura 14]**, para esto sería necesario modificar el código del mismo. La aplicación utilizaría el mismo puerto de escucha de las aplicaciones servidoras, y el puerto 5060 para comunicarse con el *softphone*. Esta aplicación si está en modo seguro encriptaría los mensajes salientes y desencriptaría los entrantes, para esto brindaría la opción de que sea activado o no, teniendo una funcionalidad, la cual le avisaría a la aplicación servidora en el momento que sea activado o no el modo seguro.

Si la terminal es un teléfono *IP* se propone el uso de SIPSEC **[Ver Anexo figura 13]** que es un dispositivo compuesto por un par de conectores RJ-45 hembra (entrada y salida) por donde deben fluir los datos, además de un microcontrolador capaz de entender el flujo Ethernet, en el que estaría implementada la aplicación. La función de avisar al servidor de que es un medio seguro, sería cuando se conectara el dispositivo a la red, y en caso de que se deje de usar SIPSEC, el usuario no podría comunicarse.

En el momento que se establece la sesión SIP entre las terminales, los participantes de la sesión intercambiarán directamente su tráfico de audio/video a través del protocolo *RTP*. Este protocolo trabaja a nivel de aplicación permitiendo que las aplicaciones VoIP puedan solventar los problemas de retardo y variación de retardo (*jitter*) de los paquetes, proporcionando campos de número de secuencia y estampado de tiempo para la reordenación de los paquetes en los extremos de la comunicación.

Para brindarle seguridad al protocolo *RTP* se propone el uso de *SRTP*, el cual es un perfil del protocolo *RTP,* que es capaz de proporcionar autenticación mediante cifrado, además de confidencialidad, integridad y no repudio tanto para los mensajes de *RTP* como para los de *RTCP*, en aplicaciones *unicast* y *multicast*. Este protocolo consigue una mayor calidad de servicio en una conexión VoIP, una menor sobrecarga, manteniendo la eficiencia de compresión de la cabecera *RTP*. También proporciona alta tolerancia a la pérdida de paquetes y reordenación, y presenta un sistema de cifrado robusto.

En la figura 3.1 se muestra un diagrama de flujo desde la llegada del mensaje hasta su envío.

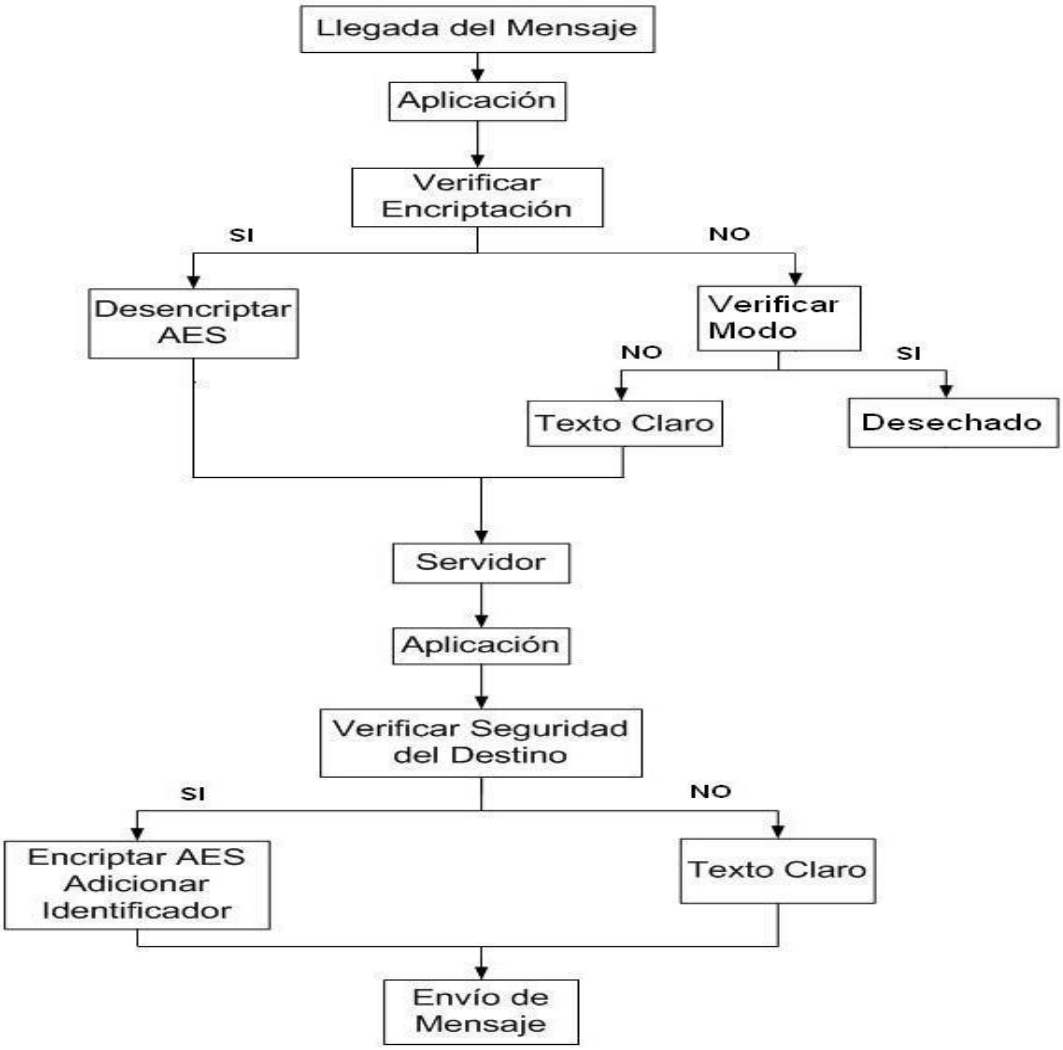


Figura 3.1 Diagrama del flujo del mensaje en un servidor con su aplicación

En la figura 3.2 se muestra un diseño de una arquitectura SIP protegida basada en lo propuesto anteriormente.

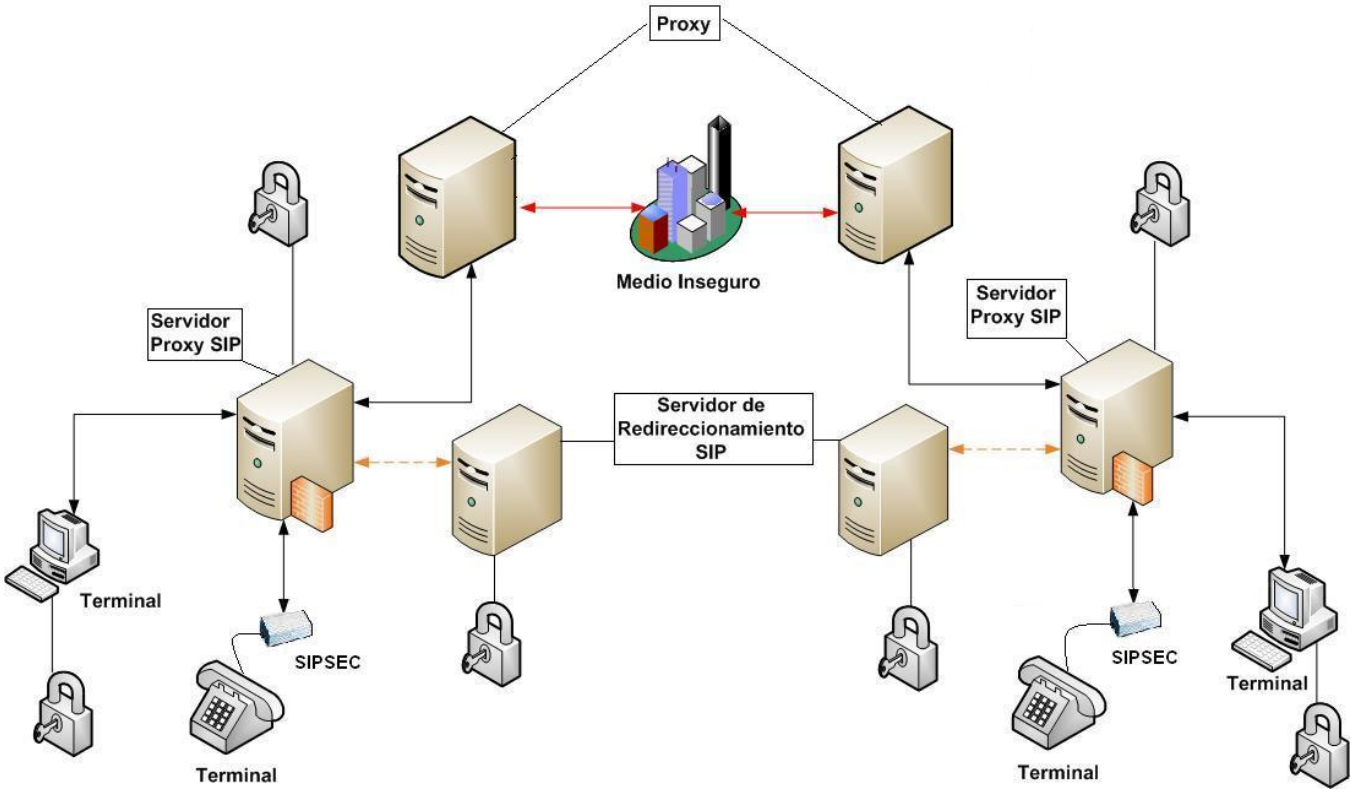


Figura 3.2 Seguridad establecida de Terminal a Terminal para redes SIP

Esta propuesta ofrece las siguientes ventajas:

- Los mensajes no viajarían en ningún momento en texto claro, ni dentro ni fuera de la red SIP. Esto proporciona que en todo momento el proceso de señalización sea seguro evitando la eficacia de los ataques por parte de cualquier intruso, fuera o dentro de la red.
- Permite procesar los mensajes SIP provenientes de las terminales PC, tanto con su identificador como sin él.
- En las terminales PC le brinda al usuario la opción de escoger un medio seguro o no, en el momento de enviar o recibir un mensaje SIP.

La desventaja que presenta esta propuesta es que en el caso de los teléfonos IP, si se desconectara SIPSEC, el usuario no podría comunicarse.

3.5. Propuesta 2 Seguridad al protocolo SIP en la red externa

El objetivo de esta propuesta es garantizar la seguridad del protocolo SIP en el momento que el mensaje es enviado fuera de su red. Dentro de la red interna estarían viajando en texto claro y solo se les daría seguridad al salir al medio externo.

Para lograr lo anteriormente planteado, se propone realizar una aplicación implementada en C que va a contener además del algoritmo criptográfico simétrico AES, otras funcionalidades, como la de agregarle al protocolo SIP un identificador de 3 caracteres formado cada uno por 8 bits. Este identificador se utilizaría como prefijo del protocolo SIP cada vez que sea encriptado y se encontrarían en un fichero de almacenamiento en la aplicación.

Esta aplicación estaría ejecutándose en el último elemento físico de la arquitectura SIP, que estaría escuchando por el puerto 5060 que sería el mismo que utilizarían las demás aplicaciones para comunicarse entre sí. La comunicación de la aplicación con las aplicaciones SIP en el servidor seria por un puerto diferente, que en este caso seria definido al colocar la aplicación en el servidor.

Al llegar un mensaje al servidor, la aplicación verificaría si esta encriptado o no, comparando los primeros 3 caracteres con el identificador que tiene almacenado. Si la comparación es válida desencriptaría a partir del bit 24, y pasaría el mensaje desencriptado al servidor para que este lo procese. Si no es válida se le aplicaría la política de seguridad al mensaje, que consiste en verificar su origen, si el mensaje proviene de una red interna este se procesaría, y en caso de una red externa este se desecharía.

En el momento de enviar un mensaje la aplicación verificaría el destino, para saber si será encriptado o no. Si el mensaje va para la red interna, no seria encriptado y en caso de ir a la red externa entonces este se enviaría encriptado.

En la figura 3.3 se muestra un diagrama de flujo que describe los procesos que se realizarían desde la llegada del mensaje hasta su envío de la red externa.

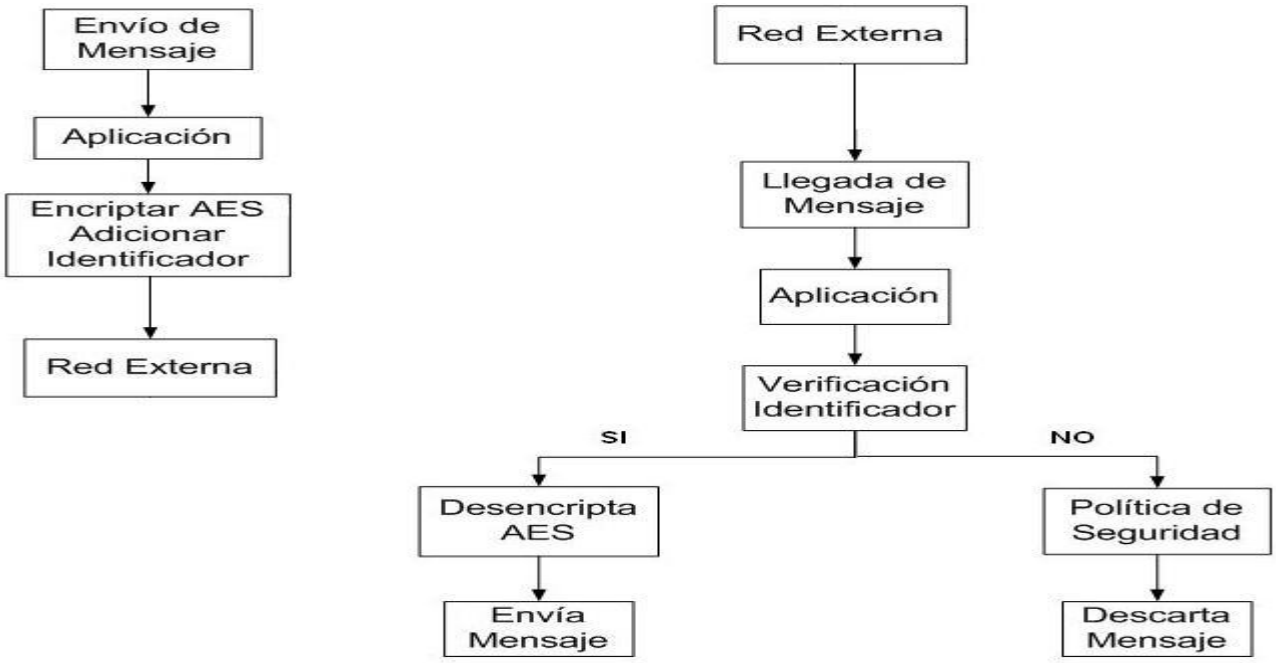


Figura 3.3 Diagrama de Flujo del último elemento físico de la red con la aplicación

En la figura 3.4 se muestra una arquitectura SIP estando protegida al salir a una red externa.

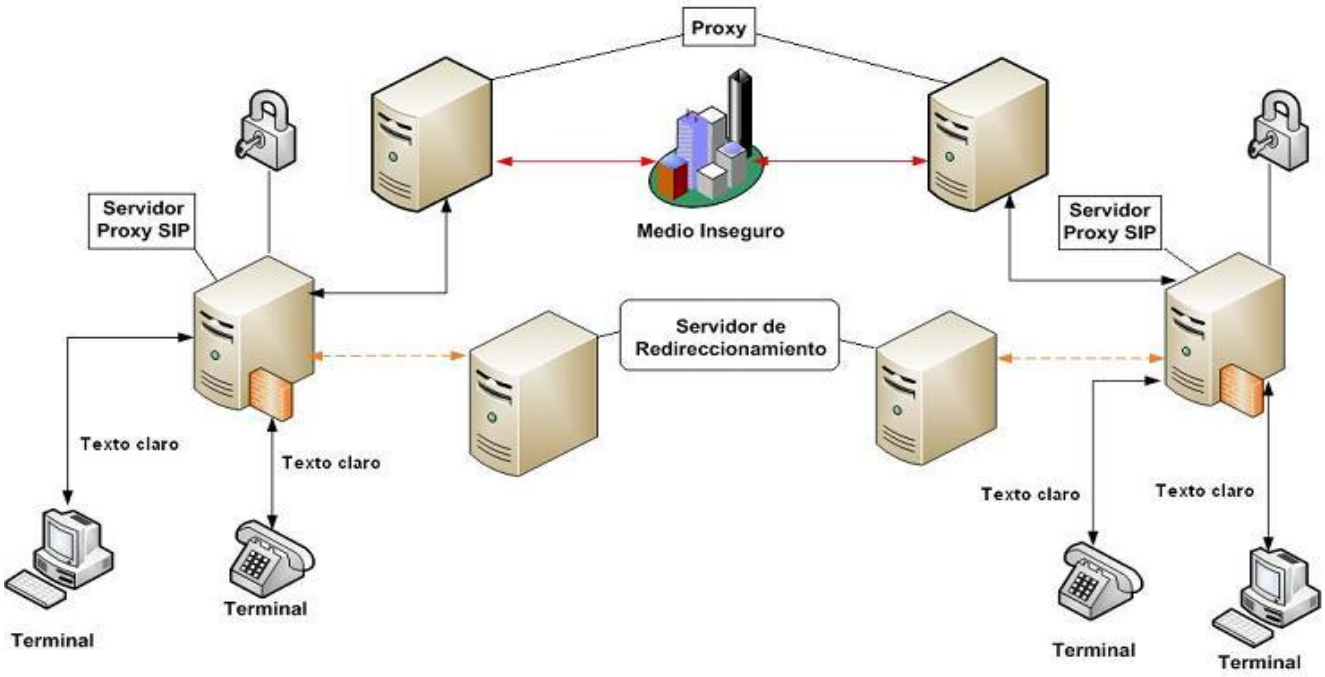


Figura 3.4 Seguridad establecida de Proxy-Proxy

Esta propuesta ofrece las siguientes ventajas:

- El mensaje SIP viajaría encriptado por la red externa, de esta forma se asegura el contenido de cada mensaje al salir al medio externo.
- No implementación de SIPSEC comparado con la Propuesta 1

La desventaja que presenta esta propuesta es que dentro de la red interna los mensajes viajaran en texto claro. Esto da la posibilidad al atacante de realizar cualquier ataque dentro de la red y obtener así el contenido del mensaje.

3.6. Conclusiones parciales

Con las propuestas realizadas se garantiza una mejor seguridad del protocolo SIP de Terminal a Terminal y de Proxy SIP a Proxy SIP. Con la utilización de la aplicación y de SIPSEC, se brinda la posibilidad de encriptar el protocolo SIP, que junto a otras funcionalidades que presentan las aplicaciones servidoras y clientes, se logra una alta seguridad en el envío y recepción de los mensajes.

Se debe tener en cuenta además que de acuerdo a la seguridad que se quiera brindar se escogería una u otra propuesta, debido a que:

- La propuesta 1 tiene como objetivo la integridad de los mensajes SIP en toda su trayectoria, es decir, tanto entre terminal y servidor SIP como entre servidores SIP. Esta presenta como desventaja la complejidad en la construcción de SIPSEC.
- La propuesta 2 tiene el objetivo de asegurar la integridad de los mensajes SIP pero solo entre servidores SIP. Tiene como desventaja que los mensajes viajarían en texto claro en el intercambio de información entre terminal y servidor.

CONCLUSIONES

Con la realización de este trabajo se le dio cumplimiento al objetivo de esta investigación donde se debe destacar que:

- La VoIP es una tecnología que ofrece un conjunto de ventajas y servicios para la comunicación a través de redes *IP*. Para esto utiliza diferentes protocolos, donde su seguridad depende en gran medida de la seguridad de estos.

- SIP, el cual es uno de los protocolos de señalización que utiliza la VoIP, debido a su flexibilidad y ventajas, además de los servicios y dispositivos que este proporciona, es uno de los protocolos más difundidos a nivel mundial. Debido a esto, la seguridad del mismo es de gran importancia por ser el objetivo de muchos tipos de ataques.

- Se realizaron dos propuestas que quedaron listas para ser implementadas las cuales le proporcionan al protocolo SIP una mejor seguridad. En caso de querer garantizar la seguridad de Terminal a Terminal se debe implementar la propuesta 1. Esta posibilita que la información que se transmite por la red este encriptada en todo momento, garantizando una seguridad tanto dentro como fuera de la red. En caso de querer garantizar la seguridad sólo entre los servidores proxy SIP se debe implementar la propuesta 2. En esta situación la seguridad de la información estará garantizada fuera de la red, transmitiéndose en texto claro dentro de esta.

.

RECOMENDACIONES

Se recomienda:

- Realizar la implementación de los elementos que conforman las propuestas de este trabajo, con el objetivo de llevarlas a la práctica.
- Implementar la propuesta 1 para ponerla en práctica en la UCI, con el objetivo de brindarle seguridad tanto dentro como fuera de su red.
- La implantación de estas propuestas de seguridad en todas las organizaciones del país, las cuales en dependencia del nivel de seguridad que deseen, utilizaran una u otra propuesta.

REFERENCIAS

[1] **Jason Sinclair, Paul Fong, Scott M. Harris , Martin Walshaw.** *Configuring Cisco Voice Over IP.* s.l. : Reviews.

[2] **Gil, Roberto Gutiérrez.** Seguridad en VoIP : Ataques, Amenazas y Riesgos. *Seguridad en VoIP : Ataques, Amenazas y Riesgos.* [Online] Enero 11, 2007. [Cited: febrero 20, 2008.] http://www.uv.es/montanan/ampliacion/trabajos/Seguridad%20VoIP.pdf..

[3]**Tobias Glemser, Reto Lorenz.** Seguridad en la Voz sobre IP – Protocolos SIP y RTP. *Seguridad en la Voz sobre IP – Protocolos SIP y RTP.* [Online] Marzo 2005. [Cited: Marzo 3, 2008.] www.compuven.net/Contenidos/Revistas/Hakin9/Hakin9-Seguridad-VoIP-Protocolos-SIP-y-RTP.pdf .

[4] **Baptista, Lourenço Rogério.** Protocolos VoIP para Redes Convergentes. *Protocolos VoIP para Redes Convergentes.* [Online] Agosto 28, 2007. [Cited: Marzo 10, 2008.] http://www.bdtd.ndc.uff.br/tde_busca/arquivo.php?codArquivo=2294.

[5] **Junior, Jucimar Maia da Silva.** Uma aplicação de voz sobre IP baseada no Session Initiation Protocol. *Uma aplicação de voz sobre IP baseada no Session Initiation Protocol.* [Online] Agosto 06, 2003. [Cited: Marzo 10, 2008.] http://www.bdtd.ufpe.br/tedeSimplificado//tde_busca/arquivo.php?codArquivo=945.

[6] **Francois Bounoure, Anibal Coppo, Diego Csernoch, Bruno Pravisani, Daniel Serrano.** SIP SESSION INITIATION PROTOCOL. *SIP SESSION INITIATION PROTOCOL.* [Online] Diciembre 2006. [Cited: Marzo 20, 2008.] http://www.fiuba6662.com.ar/6648/presentaciones/2006/Informe%20SIP.pdf.

[7] **Vázquez, Miguel Fernández.** Algoritmo Criptográfico AES para protección de datos. *Algoritmo Criptográfico AES para protección de datos.* [Online] Septiembre 2007. www.iit.upcomillas.es/pfc/resumenes/46ea7511774d8.pdf.

[8] **Lorin, Sylvain.** PGP - Pretty Good Privacy. *PGP - Pretty Good Privacy.* [Online] http://es.kioskea.net/crypto/pgp.php3.

Bibliografías

1. **Simon Znaty, Jean-Louis Dauphin , Roland Geldwerth.** SIP : Session Initiation Protocol. *SIP : Session Initiation Protocol.* [Online] 2005. [Cited: Marzo 25, 2008.] http://www.efort.com.

2. **Pouzols, Federico Montesino.** SIP: Sesion Initiation Protocol. *SIP: Sesion Initiation Protocol.* [Online] Mayo 2003. [Cited: Marzo 25, 2008.] www.rediris.es/mmedia/gt/gt2003_1/sip-gt2003.pdf .

3. **Camarillo, Gonzalo.** *SIP Demystified.* 2002.

4. **Castañeda, Rodolfo.** Protocolos para voz IP. *Protocolos para voz IP.* [Online] 2005. [Cited: Abril 05, 2008.] http://www.cudi.edu.mx/primavera_2005/presentaciones/rodolfo_castaneda.pdf.

5. **G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler.** RFC 3261. *RFC 3261.* [Online] Junio 2002. [Cited: Marzo 05, 2008.] http://www.rfc.net/rfc3261.html.

6. **Lawton, Opal.** INFORME SOBRE VoIP. *INFORME SOBRE VoIP.* [Online] Julio 2007. [Cited: Febrero 20, 2008.] www.canto.co.cu/documentos/reportes_canto/informe-sobre-el-taller-de-voip - .

7. **Richard Kuhn, Thomas J. Walsh, Steffen Fries.** Security Considerations for Voice Over IP Systems. *Security Considerations for Voice Over IP Systems.* [Online] Enero 2005. [Cited: Febrero 20, 2008.] csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf.

8. **Judith Vivar Mesa, Aisel Guerrero Luis.** Soluciones de movilidad empleando el protocolo SIP: caracterización del comportamiento del tráfico para servicios isócronos. *Soluciones de movilidad empleando el protocolo SIP: caracterización del comportamiento del tráfico para servicios isócronos.* [Online] Diciembre 01, 2006. [Cited: Abril 15, 2008.] http://www.cujae.edu.cu/eventos/cittel/trabajos/trabajos.html#CITTEL61.

9. **Francois Bounoure, Anibal Coppo, Diego Csernoch, Bruno Pravisani, Daniel Serrano.** SIP SESSION INITIATION PROTOCOL. *SIP SESSION INITIATION PROTOCOL.* [Online] Diciembre 2006. [Cited: Marzo 20, 2008.] http://www.fiuba6662.com.ar/6648/presentaciones/2006/Informe%20SIP.pdf.

10. **Jason Sinclair, Paul Fong, Scott M. Harris , Martin Walshaw.** *Configuring Cisco Voice Over IP.* s.l. : Reviews.

11. **Junior, Jucimar Maia da Silva.** Uma aplicação de voz sobre IP baseada no Session Initiation Protocol. *Uma aplicação de voz sobre IP baseada no Session Initiation Protocol.* [Online] Agosto 06, 2003. [Cited: Marzo 10, 2008.] http://www.bdtd.ufpe.br/tedeSimplificado//tde_busca/arquivo.php?codArquivo=945.

12. **Baptista, Lourenço Rogério.** Protocolos VoIP para Redes Convergentes. *Protocolos VoIP para Redes Convergentes.* [Online] Agosto 28, 2007. [Cited: Marzo 10, 2008.] http://www.bdtd.ndc.uff.br/tde_busca/arquivo.php?codArquivo=2294.

13. Seguridad en SIP - Session Initiation Protocol. *Seguridad en SIP - Session Initiation Protocol.* [Online] Mayo 26, 2006. [Cited: Abril 10, 2008.] www.eslomas.com/index.php/archives/2006/05/26/seguridad-en-sip-session-initiation-protocol.

14. **Tobias Glemser, Reto Lorenz.** Seguridad en la Voz sobre IP – Protocolos SIP y RTP. *Seguridad en la Voz sobre IP – Protocolos SIP y RTP.* [Online] Marzo 2005. [Cited: Marzo 03, 2008.] www.compuven.net/Contenidos/Revistas/Hakin9/Hakin9-Seguridad-VoIP-Protocolos-SIP-y-RTP.pdf.

15. **Marco Aguilar Junca, Paola Riaño Castellanos.** TELÉFONO SOFTWARE IP BASADO EN SIP E IMPLEMENTACIÓN DE PESQ. *TELÉFONO SOFTWARE IP BASADO EN SIP E IMPLEMENTACIÓN DE PESQ.* [Online] Mayo 2005. [Cited: Abril 20, 2008.] www.javeriana.edu.co/biblos/tesis/ingenieria/tesis87.pdf.

16. **Gorka Gorrotxategi, Iñaki Baz.** Voz sobre IP y Asterisk. *Voz sobre IP y Asterisk.* [Online] 2006. [Cited: Abril 25, 2008.] www.irontec.com/files/cursoAsteriskVozIP-2-dispositivos-SIP.pdf .

17. **Carlos Ramos, Ana Belén García.** REDES Y SERVICIOS I: Arquitectura y protocolos VoIP. *REDES Y SERVICIOS I: Arquitectura y protocolos VoIP.* [Online] 2007-2008. [Cited: Marzo 15, 2008.] http://asignaturas.diatel.upm.es/rrss1/documentacion_archivos/TEORIA%20ACTUAL/VoIPCurso2007-2008-Completo.pdf.

18. **Gil, Roberto Gutiérrez.** Seguridad en VoIP : Ataques, Amenazas y Riesgos. *Seguridad en VoIP : Ataques, Amenazas y Riesgos.* [Online] Enero 11, 2007. [Cited: febrero 25, 2008.] http://www.uv.es/montanan/ampliacion/trabajos/Seguridad%20VoIP.pdf..

19. **Vázquez, Miguel Fernández.** Algoritmo Criptográfico AES para protección de datos. *Algoritmo Criptográfico AES para protección de datos.* [Online] Septiembre 2007. www.iit.upcomillas.es/pfc/resumenes/46ea7511774d8.pdf.

20. **Vera Delgado, Rafael Palacios.** Introduccion a la ciptografia:tipos de algoritmos. *Introduccion a la ciptografia:tipos de algoritmos.* [Online] 2006 Febrero. www.icai.es/publicaciones/anales_get.php?id=1210.

21. **Muñoz, Alfonso Muñoz.** Algoritmo criptografico Rijndael. *Algoritmo criptografico Rijndael.* [Online] Septiembre 2004. www.kriptopolis.org/docs/rijndael.pdf.

22. **Lorin, Sylvain.** PGP - Pretty Good Privacy. *PGP - Pretty Good Privacy.* [Online] http://es.kioskea.net/crypto/pgp.php3.

GLOSARIO DE TÉRMINOS

- **ARP:** Address Resolution Protocol (Protocolo de Resolución de Direcciones) para la resolución de direcciones en informática, responsable de encontrar la dirección hardware que corresponde a una determinada dirección IP.

- **Asterisk:** Es una aplicación de software libre de una central telefónica (PBX). Como cualquier PBX, se puede conectar un número determinado de teléfonos para hacer llamadas entre sí e incluso conectar a un proveedor de VoIP.

- **Buffer:** Un buffer de datos es una ubicación de la memoria en una computadora o en un instrumento digital reservada para el almacenamiento temporal de información digital, mientras que está esperando ser procesada.

- **CGI:** Common Gateway Interface (Interfaz Común de Pasarela). Interfaz entre servidores de información y programas de aplicación. Define una serie de reglas que deben cumplir tanto las aplicaciones como los servidores, para hacer posible la presentación de resultados de programas ejecutables en tiempo real a través de servicios de información estandarizados.

- **CODEC:** Codificador-Decodificador. Describe una especificación desarrollada en software, hardware o una combinación de ambos, capaz de transformar un archivo con un flujo de datos (stream) o una señal.

- **CPU:** Central Processing Unit (unidad de proceso central). Es donde se producen la mayoría de los cálculos. En términos de potencia del ordenador, la CPU es el elemento más importante de un sistema informático.

- **Conmutación:** Es la conexión que realizan los diferentes nodos que existen en distintos lugares y distancias para lograr un camino apropiado para conectar dos usuarios de una red de telecomunicaciones. Existen dos tipos de conmutación, de paquetes y de circuitos.

- **DNS:** Domain Name System, es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en las redes.

- **DSL:** Digital Subscriber Line (Línea de abonado digital), es un término utilizado para referirse de forma global a todas las tecnologías que proveen una conexión digital sobre línea de abonado de la red telefónica local.

- **FIREWALL:** Cortafuegos (informática), elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones.

- **Firmware:** Programación en Firme, es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria tipo ROM, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

- **GUI:** Graphical User Interface (Interfaz Gráfica de Usuario). Tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos, para representar la información y acciones disponibles en la interfaz. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora.

- **HTTP:** HyperText Transfer Protocol (Protocolo de Transferencia Hipertexto). Protocolo de comunicaciones utilizado por los programas clientes y servidores de WWW para comunicarse entre sí.

- **HTTPS:** Hypertext Transfer Protocol Secure (Protocolo seguro de transferencia de hipertexto), es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

- **IETF:** Internet Engineering Task Force (Grupo de Trabajo de Ingeniería de Internet). Es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad.

- **IP:** Internet Protocol (Protocolo de Internet). Es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

- **ITU-T:** International Telecommunication Union (Unión Internacional de Telecomunicaciones). El Sector de Telecomunicaciones de la ITU, que desarrolla los estándares para la interconexión de equipo de telecomunicaciones entre diversas redes.

- **JITTER:** Es un término que se refiere al nivel de variación de retado que introduce una red. Una red con variación 0 tarda exactamente lo mismo en transferir cada paquete de información, mientras que una red con variación de retardo alta tarda mucho más tiempo en entregar algunos paquetes que en entregar otros. La variación de retardo es importante cuando se envía audio o video, que deben llegar a intervalos regulares si se quieren evitar desajustes o sonidos inteligibles.

- **KERBEROS V5:** Protocolo de seguridad principal para la autenticación dentro de un dominio.

- **LAN:** Local Area Network (Red de Área Local). Red de datos para dar servicio a un área geográfica máxima de unos pocos kilómetros cuadrados con velocidades de transmisión de hasta 100 Mbps (100 megabits por segundo).

- **LOGS:** Registro lógico de las actividades desarrolladas por algún Host.

- **MAC**: Media Access Control Address (Dirección de Control de Acceso al Medio). Es un identificador de 48 bits (6 bytes) que corresponde de forma única a una tarjeta o interfaz de red. Es individual, cada dispositivo tiene su propia dirección MAC.

- **MD5:** Message-Digest Algorithm 5 (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits.

- **MULTICAST:** Multidifusión, es el envío de la información en una red a múltiples destinos simultáneamente.

- **NAT:** Network Address Translation (Traducción de Dirección de Red) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes.

- **OSI:** Open System Interconnection (Modelo de Referencia de Interconexión de Sistemas Abiertos). Es un modelo elaborado por la ISO que define los protocolos de comunicación en siete niveles diferentes. Estos niveles son los siguientes: aplicación, presentación, sesión, transporte, red, enlace y físico.

- **PBX:** Private Branch Exchange (Central Secundaria Privada). Es una central telefónica, propiedad de una empresa privada, en contraposición con la central que es propiedad de un operador de telecomunicaciones o de una empresa de telefonía.

- **PSTN:** Public Switched Telephone Network (Red pública de telefonía conmutada). Es la concentración de las redes públicas mundiales de circuitos conmutados, al igual que Internet es la concentración de redes públicas mundiales de paquetes conmutados basados en IP.

- **PCM:** Pulse-Code Modulation (Modulación en Código de Pulsaciones). Es un procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits.

- **PKI:** Public Key Infrastructure (Infraestructura de clave pública). Una arquitectura PKI o de clave pública proporciona los fundamentos para establecer y mantener un entorno de red seguro, a través de la generación y distribución de claves y certificados digitales.

- **Router:** Dispositivo que distribuye tráfico entre redes. La decisión sobre a donde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento. Es el nodo básico de una red IP.

- **RTCP:** Real Time Control Protocol (Protocolo de Control en Tiempo Real). Es un protocolo de comunicación que proporciona información de control que está asociado con un flujo de datos para una aplicación multimedia (flujo RTP). Trabaja junto con RTP en el transporte y empaquetado de datos multimedia, pero no transporta ningún dato por sí mismo. Se usa habitualmente para transmitir paquetes de control a los participantes de una sesión multimedia

- **RTP:** Real Time Protocol (Protocolo en Tiempo Real). Es un protocolo de nivel de transporte utilizado para la transmisión de información en tiempo real, como por ejemplo audio y vídeo en un video-conferencia.

- **RUTEAR:** En redes es dirigir la información que se transmite a través de una red, desde su origen hasta su destino, eligiendo el mejor camino posible a través de la/s red/es que los separan.

- **SDP:** Session Description Protocol (Protocolo de Descripción de Sesión). Es un protocolo para describir los parámetros de inicialización de los flujos multimedia

- **SMTP:** Protocolo de Transporte de Correo Simple, que se usa para la transferencia de correo electrónico entre computadoras. Es un protocolo de servidor a servidor, de forma que para poder leer los mensajes se deben utilizar otros protocolos.

- **SOFTPHONE:** Software que simula a un teléfono convencional por computadora.
- **SPN:** Substitution-Permutation Networks (Red de Sustitución de Permutación), es un cifrador iterado. La idea general de estos cifradores consiste en dividir el mensaje en bloques de bits, generalmente del mismo tamaño fijo, y aplicar un número de rondas o vueltas de substituciones y permutaciones a cada bloque.
- **SRTP**: Secure Real Time Protocol (Protocolo en Tiempo Real Seguro), define un perfil de RTP, con la intención de proporcionar cifrado, autenticación del mensaje e integridad, y protección contra reenvíos a los datos RTP.
- **SSH:** (Secure SHell) nombre del protocolo y del programa que lo implementa, sirve para acceder a máquinas remotas a través de una red.
- **STATEFULL:** (Estado Lleno), mantiene internamente parte de la información de estado.
- **STATELESS:** (Estado Vacio), no mantiene ninguna información de estado, es más fácil de implementar y es más tolerante a fallas.
- **STREAMS:** La cantidad de transporte de volumen que tiene lugar en un punto particular.
- **SWITCH:** (Conmutador), es un dispositivo electrónico de interconexión de redes de ordenadores que opera en la capa 2 (nivel de enlace de datos) del modelo OSI.
- **TCP:** Transmission Control Protocol (Protocolo de Control de Transmisión). Conjunto de protocolos de comunicación que se encargan de la seguridad y la integridad de los paquetes de datos que viajan por Internet.
- **UDP:** User Datagram Protocol (Protocolo de Datagramas de Usuario). Protocolo de red para la transmisión de datos que no requieren la confirmación del destinatario de los datos enviados.
- **UNICAST:** Unidifusión, es el envío de información desde un único emisor a un único receptor.
- **UNIX:** Es un sistema operativo portable, multitarea y multiusuario.

- **URI:** Uniform Resource Identifier (Identificador Uniforme de Recurso). Es el conjunto genérico de todos los nombres y direcciones en forma de denotaciones cortas que se refieren a un recurso.

- **URL:** Universal Resource Indicators (Localizador Uniforme de Recurso). Cadena de caracteres con la cual se asigna dirección única a cada uno de los recursos de información disponibles en Internet.

- **VPN:** Virtual Private Network (Red Privada Virtual), conecta ordenadores dentro de una red pública como Internet, donde se utilizan sistemas de cifrado en las comunicaciones y la confidencialidad de los datos está garantizada.

- **WAN:** Wide Area Network (Red de Área Amplia), es un tipo de red de computadoras, capaz de cubrir distancias desde unos 100 hasta unos 1000 km, dando el servicio a un país o un continente.

ANEXOS

Capturas con el Ethereal:

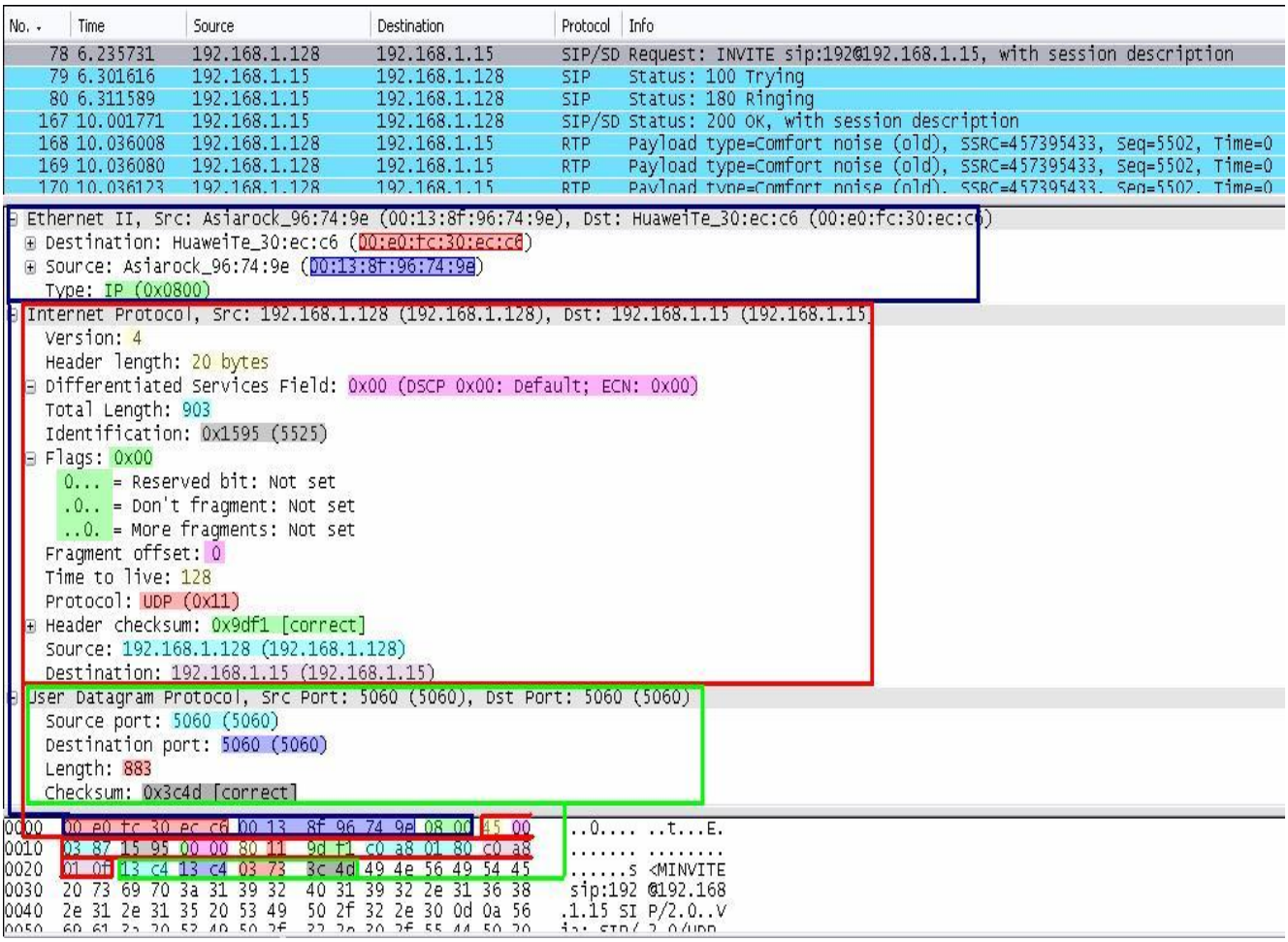


Figura: 1 Mensaje **INVITE**

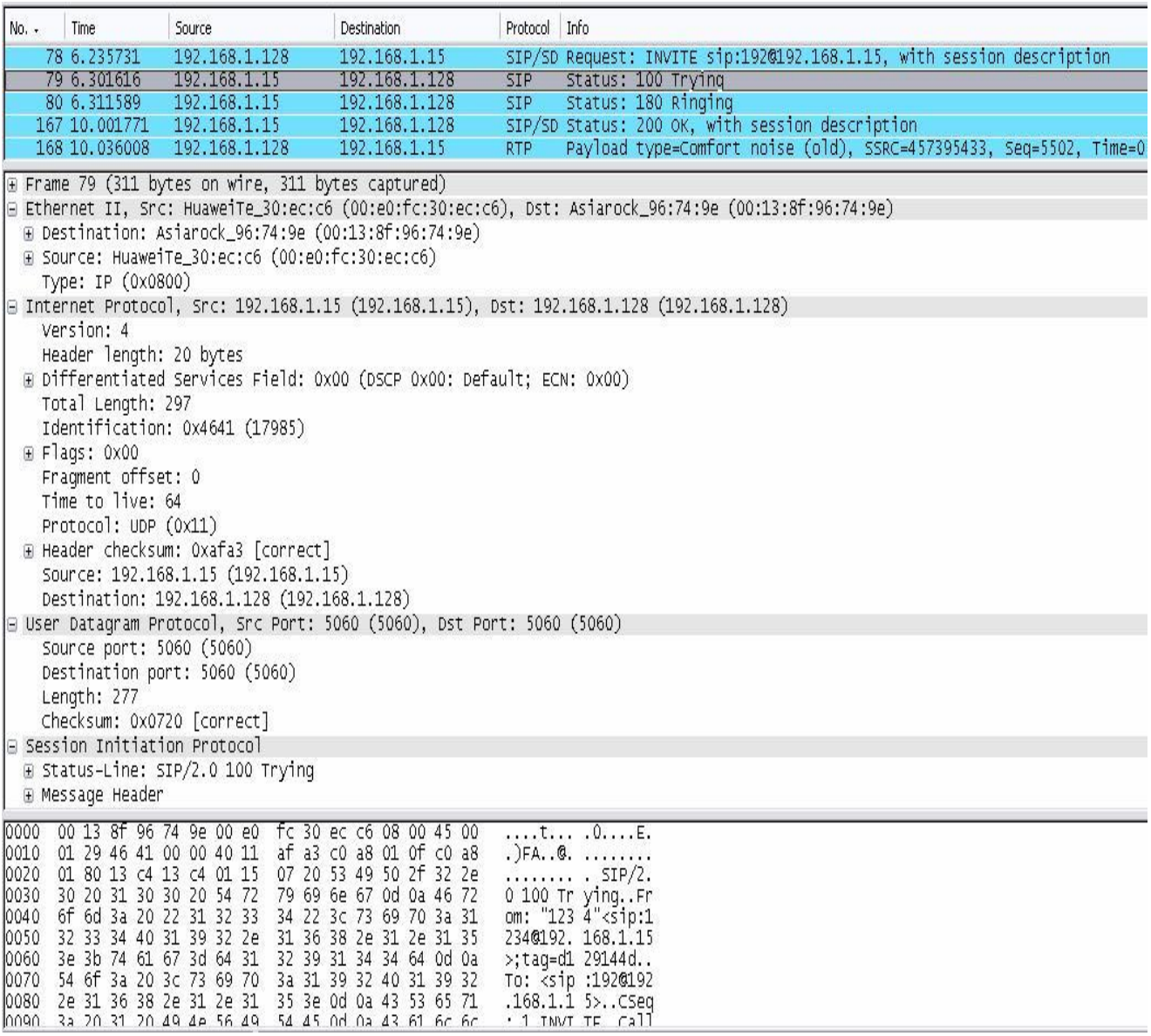
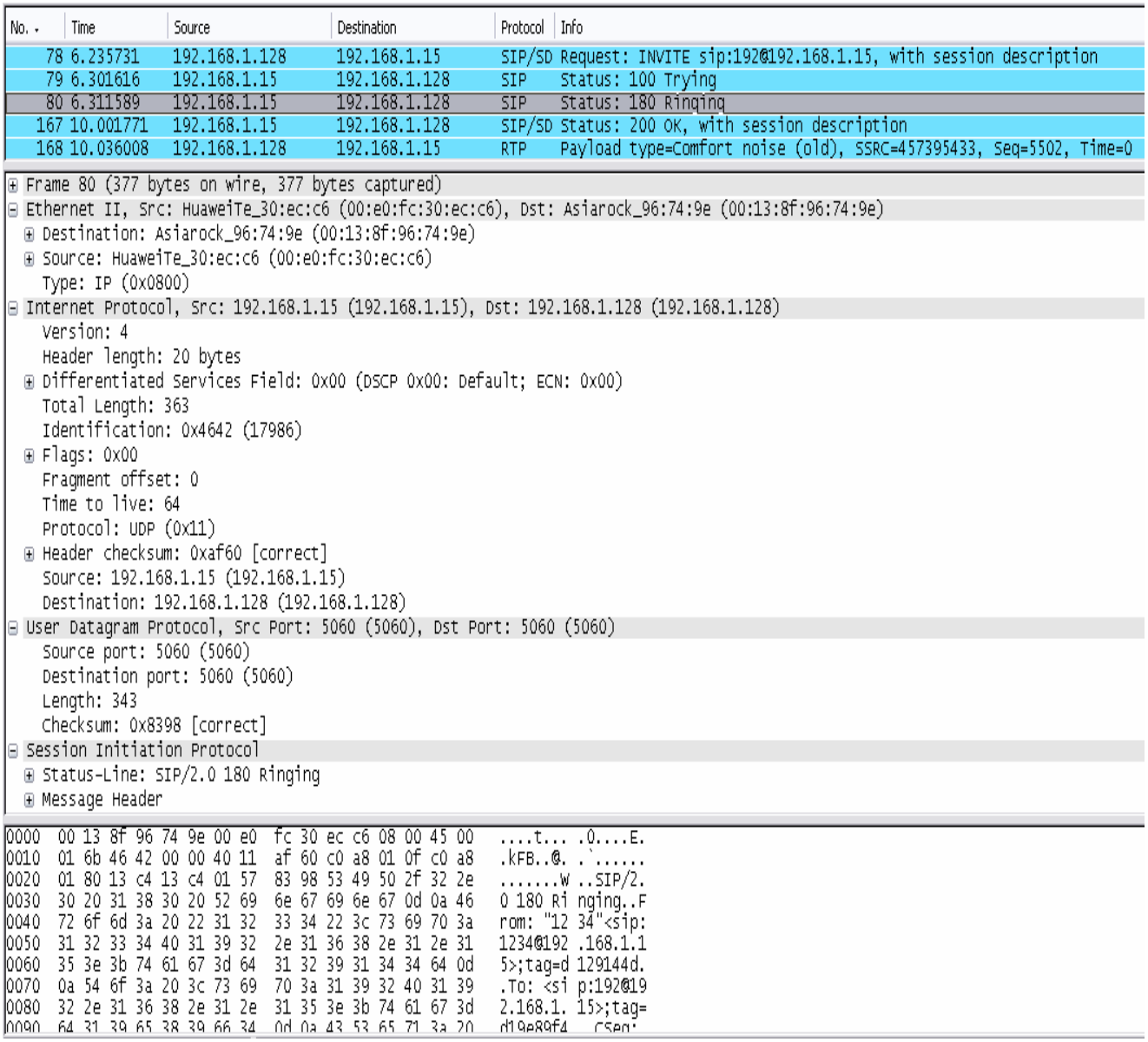


Figura: 2 Mensaje **100 TRYING**



No.	Time	Source	Destination	Protocol	Info
78	6.235731	192.168.1.128	192.168.1.15	SIP/SD	Request: INVITE sip:192@192.168.1.15, with session description
79	6.301616	192.168.1.15	192.168.1.128	SIP	Status: 100 Trying
80	6.311589	192.168.1.15	192.168.1.128	SIP	Status: 180 Ringing
167	10.001771	192.168.1.15	192.168.1.128	SIP/SD	Status: 200 OK, with session description
168	10.036008	192.168.1.128	192.168.1.15	RTP	Payload type=Comfort noise (old), SSRC=457395433, Seq=5502, Time=0

```
⊞ Frame 80 (377 bytes on wire, 377 bytes captured)
⊟ Ethernet II, Src: HuaweiTe_30:ec:c6 (00:e0:fc:30:ec:c6), Dst: Asiarock_96:74:9e (00:13:8f:96:74:9e)
  ⊞ Destination: Asiarock_96:74:9e (00:13:8f:96:74:9e)
  ⊞ Source: HuaweiTe_30:ec:c6 (00:e0:fc:30:ec:c6)
    Type: IP (0x0800)
⊟ Internet Protocol, Src: 192.168.1.15 (192.168.1.15), Dst: 192.168.1.128 (192.168.1.128)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 363
    Identification: 0x4642 (17986)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0xaf60 [correct]
    Source: 192.168.1.15 (192.168.1.15)
    Destination: 192.168.1.128 (192.168.1.128)
⊟ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
    Source port: 5060 (5060)
    Destination port: 5060 (5060)
    Length: 343
    Checksum: 0x8398 [correct]
⊟ Session Initiation Protocol
  ⊞ Status-Line: SIP/2.0 180 Ringing
  ⊞ Message Header
```

```
0000  00 13 8f 96 74 9e 00 e0  fc 30 ec c6 08 00 45 00   ....t... .0....E.
0010  01 6b 46 42 00 00 40 11  af 60 c0 a8 01 0f c0 a8   .kFB..@. .`......
0020  01 80 13 c4 13 c4 01 57  83 98 53 49 50 2f 32 2e   .......W ..SIP/2.
0030  30 20 31 38 30 20 52 69  6e 67 69 6e 67 0d 0a 46   0 180 Ri nging..F
0040  72 6f 6d 3a 20 22 31 32  33 34 22 3c 73 69 70 3a   rom: "12 34"<sip:
0050  31 32 33 34 40 31 39 32  2e 31 36 38 2e 31 2e 31   1234@192 .168.1.1
0060  35 3e 3b 74 61 67 3d 64  31 32 39 31 34 34 64 0d   5>;tag=d 129144d.
0070  0a 54 6f 3a 20 3c 73 69  70 3a 31 39 32 40 31 39   .To: <si p:192@19
0080  32 2e 31 36 38 2e 31 2e  31 35 3e 3b 74 61 67 3d   2.168.1. 15>;tag=
0090  64 31 39 65 38 39 66 34  0d 0a 43 53 65 71 3a 20   d19e89f4 ..CSeq:
```

Figura: 3 Mensaje **180 RINGING**

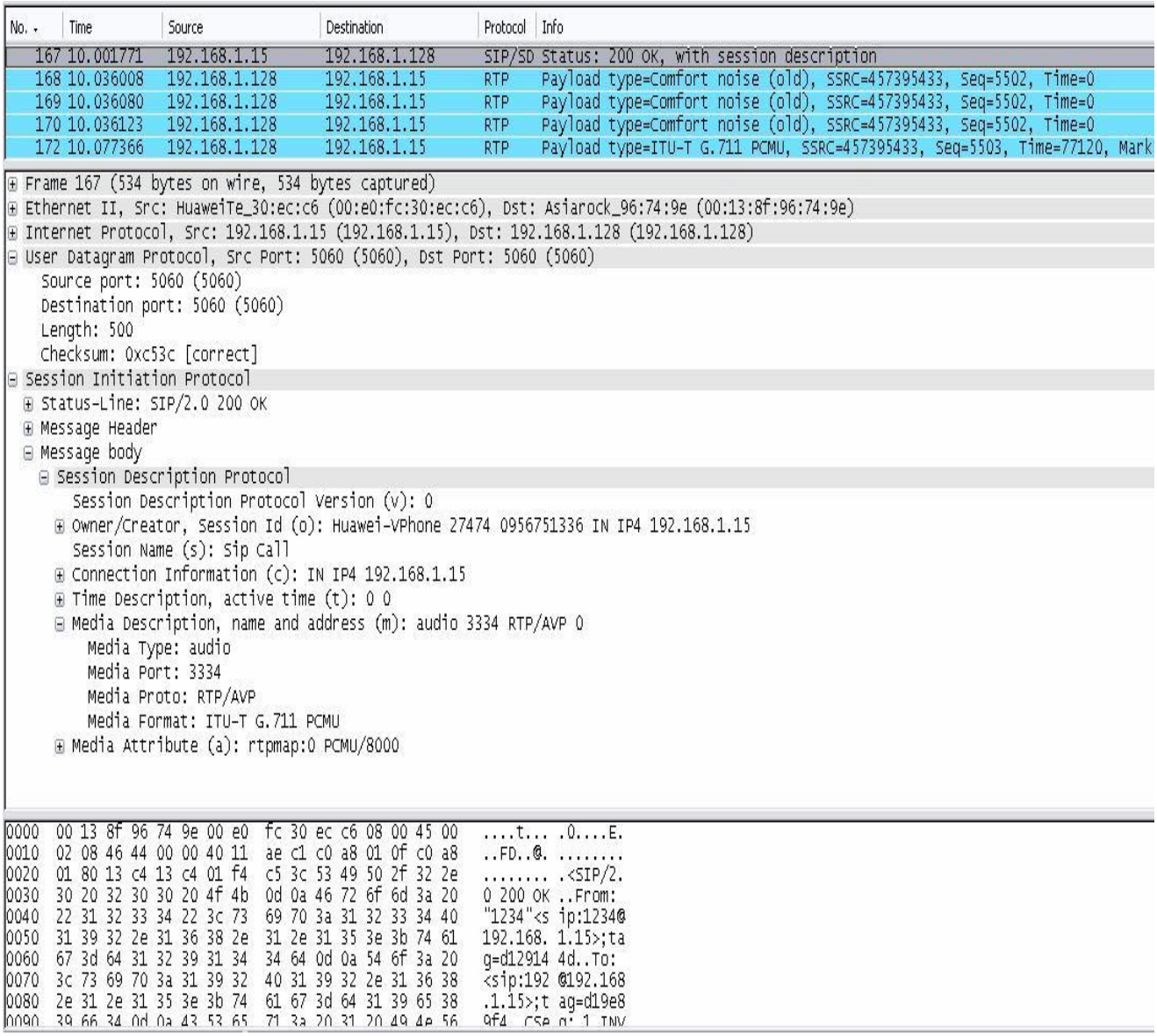


Figura: 4 Mensaje **200 OK**

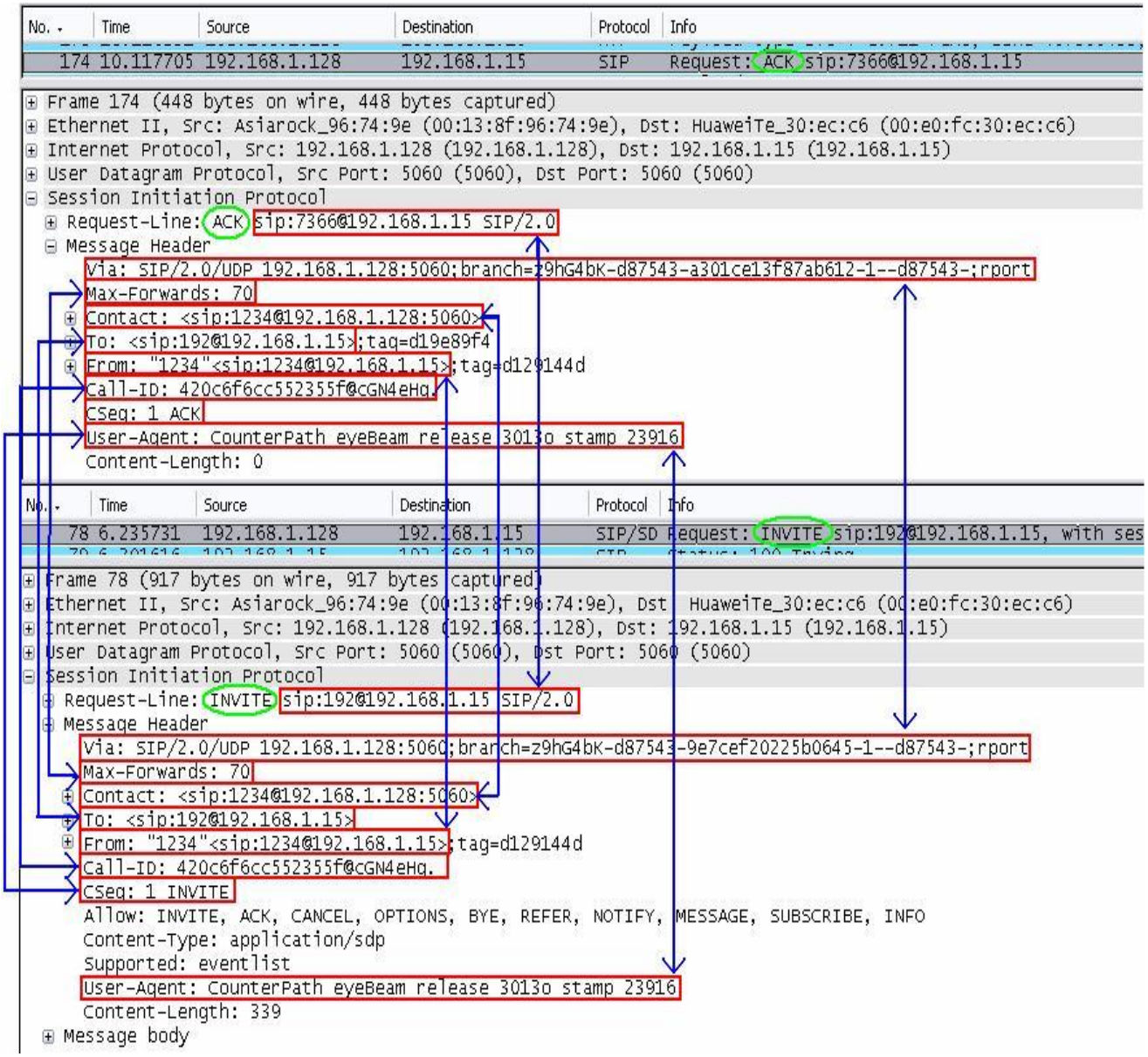


Figura: 5 Mensaje **ACK**

Figura: 6 **RTP**

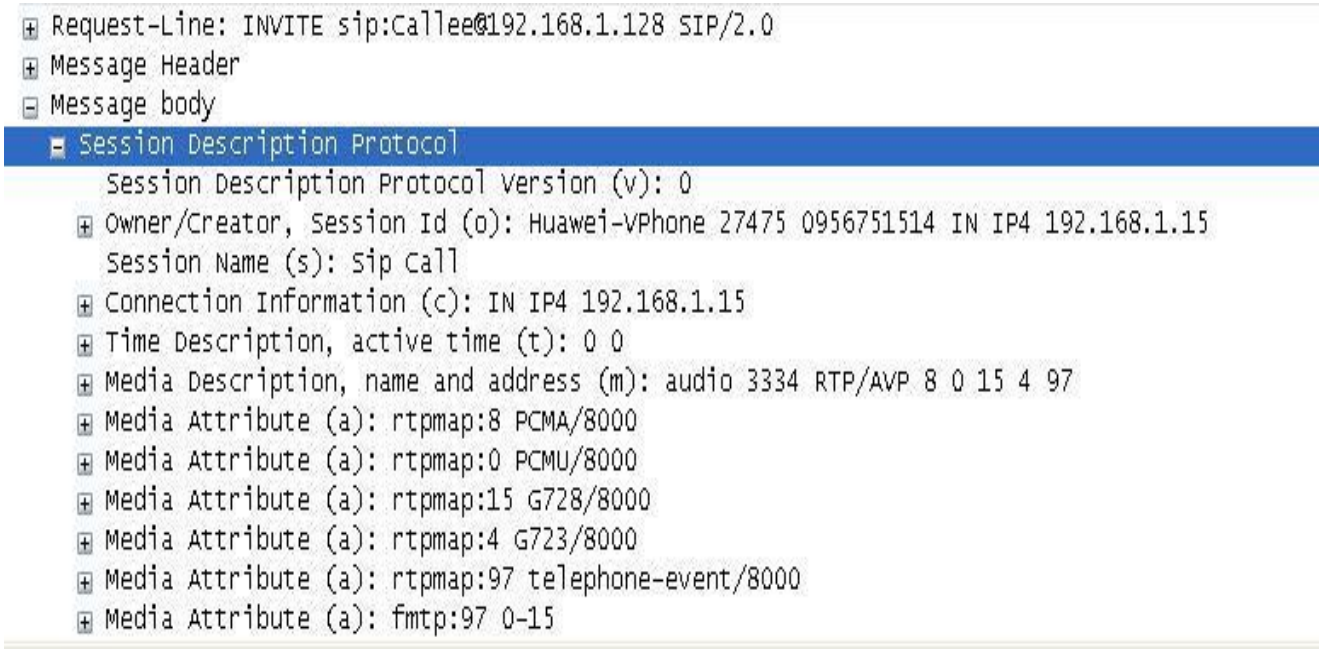


Figura: 7 **SDP**

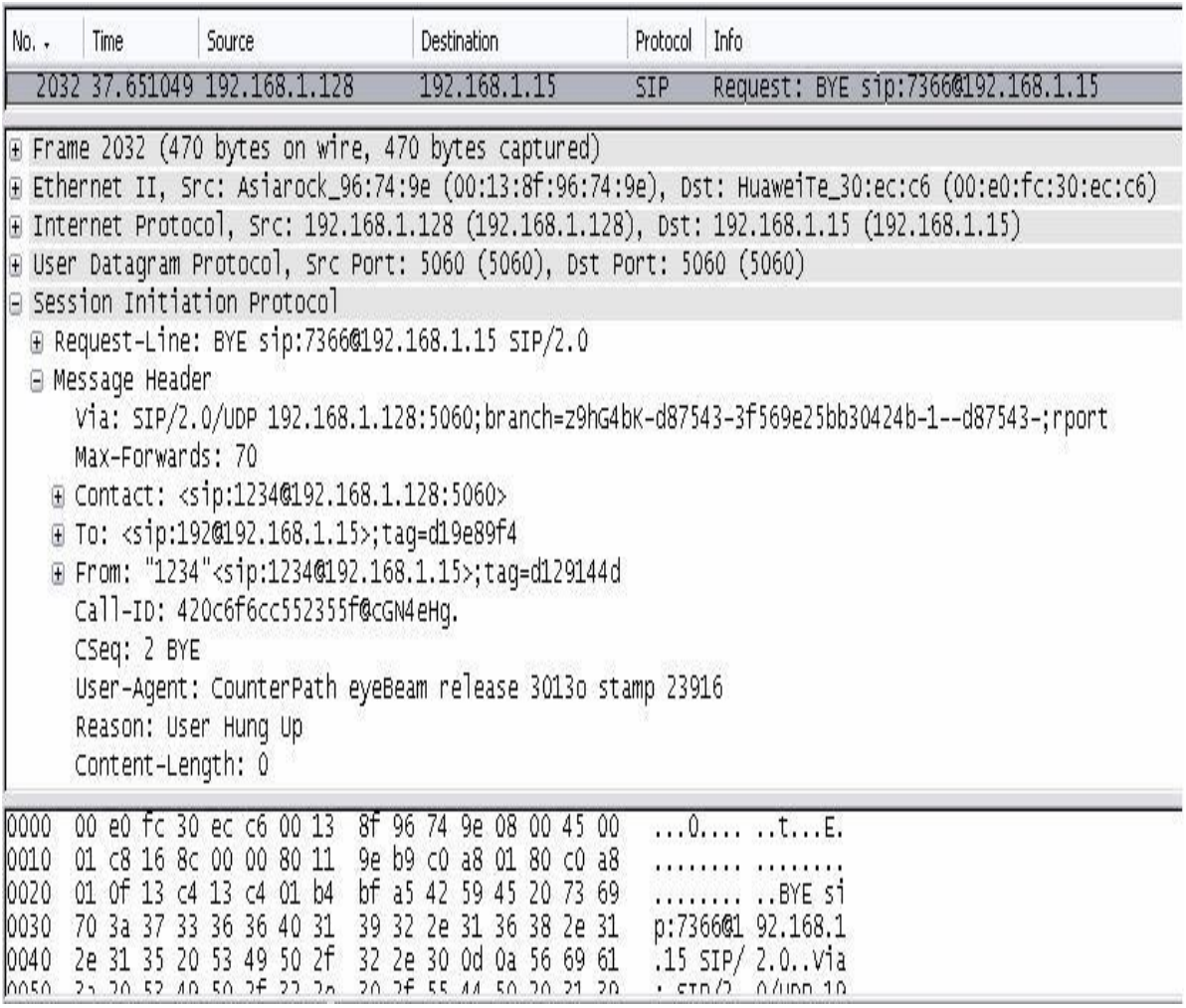


Figura: 8 Mensaje **BYE**

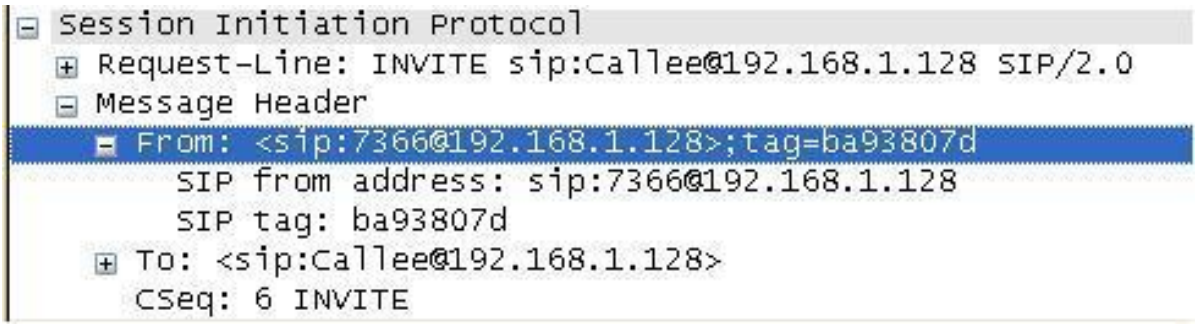


Figura: 9 Campo **FROM** dentro del **INVITE**

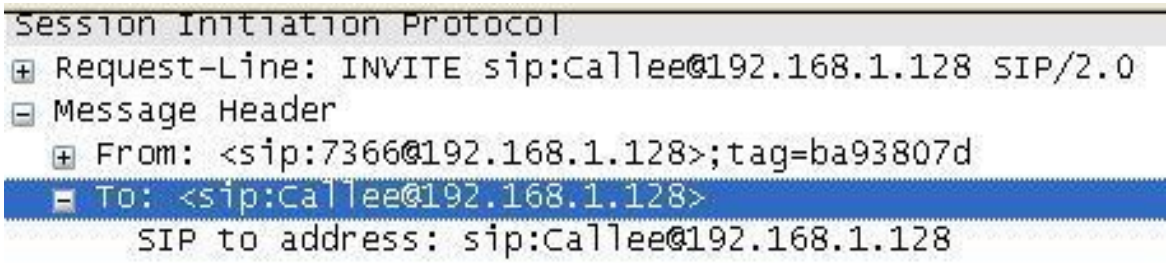


Figura: 10 Campo **TO** dentro del **INVITE**

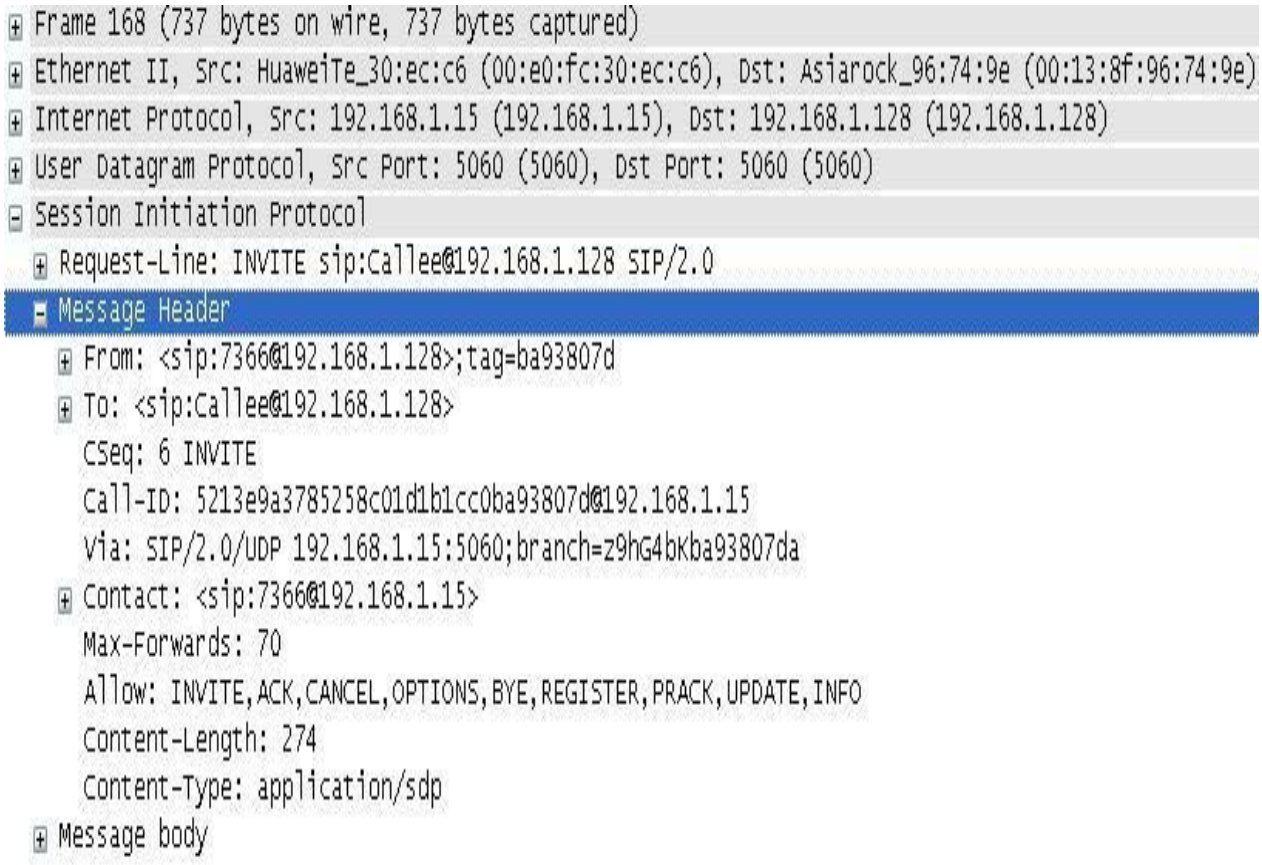


Figura: 11 **HEADER** de SIP

```
⊞ Frame 173 (406 bytes on wire, 406 bytes captured)
⊞ Ethernet II, Src: AsustekC_8f:17:5d (00:0e:a6:8f:17:5d), Dst: Asiarock_96:3e:14 (00:13:8f:96:3e:14)
⊟ Internet Protocol, Src: 192.168.1.200 (192.168.1.200), Dst: 192.168.1.128 (192.168.1.128)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 392
    Identification: 0x1303 (4867)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
  ⊞ Header checksum: 0xa1c9 [correct]
    Source: 192.168.1.200 (192.168.1.200)
    Destination: 192.168.1.128 (192.168.1.128)
⊞ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
⊟ Session Initiation Protocol
  ⊟ Request-Line: CANCEL sip:unknown@192.168.1.128:5060 SIP/2.0
      Method: CANCEL
      [Resent Packet: False]
  ⊟ Message Header
      Via: SIP/2.0/UDP 192.168.1.200:5060;branch=z9hG4bK-d87543-7b39fb7c903eb40b-1--d87543-
    ⊞ To: "192.168.1.128"<sip:unknown@192.168.1.128:5060>
    ⊞ From: "dsd"<sip:dsd@dsd>;tag=7413c868
      Call-ID: ZWIzZDJjY2I1OTk1YzJjMTVjN2IyMmZkYzZiMDI1YzE.
      CSeq: 1 CANCEL
      User-Agent: X-Lite release 1006e stamp 34025
      Content-Length: 0
```

Figura: 12 Mensaje **CANCEL**

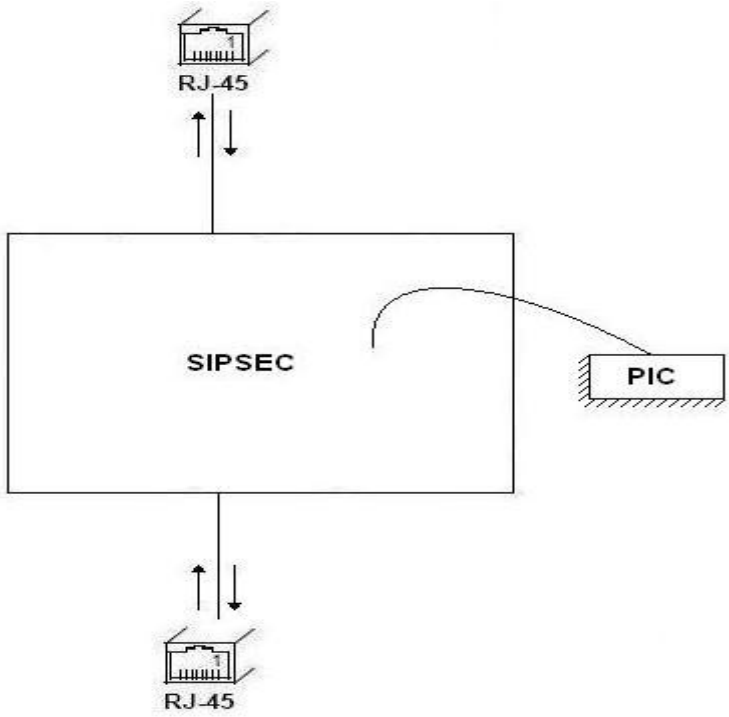


Figura: 13 **SIPSEC**

Figura: 14 **SOFTPHONE**