ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

"ANÁLISIS DE LA TECNOLOGÍA PKI Y SU APLICACIÓN EN EL ASEGURAMIENTO DE LOS SERVICIOS CORPORATIVOS WWW, FTP Y HTTP"

TESIS DE GRADO

PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES

Presentado por:

TULIO ALEJANDRO VALDIVIEZO ECHEVERRÍA

RIOBAMBA – ECUADOR

2012

DERECHOS DE AUTORÍA

Yo, Tulio Alejandro Valdiviezo Echeverría, declaro que soy el autor del presente trabajo de tesis “ANÁLISIS DE LA TECNOLOGÍA PKI Y SU APLICACIÓN EN EL ASEGURAMIENTO DE LOS SERVICIOS CORPORATIVOS WWW, FTP Y HTTP”, que fue elaborada en su totalidad por mi persona, bajo la dirección del Ingeniero Wilson Baldeon, haciéndome totalmente responsable por las ideas, criterios, doctrinas y resultados expuestos en esta Tesis, y el patrimonio de la misma pertenece a la

Escuela Superior Politécnica de Chimborazo.

_____

Tulio Alejandro Valdiviezo Echeverría

CI. 060312059-3

AGRADECIMIENTO

Agradezco a mis padres por ser quienes me brindaron su apoyo incondicional, mi madre con su amor siempre me ayudo a seguir sin decaer y mi padre con su aliento siempre me impulso a conseguir mis metas.

A mis hermanas, tías y mis amigos que siempre me ayudaron en cualquier adversidad.

Y a todas las personas que trabajan en la ESPOCH que han formado parte durante mis estudios y han logrado llegado a ser más que profesores o trabajadores unos amigos.

Tulio

DEDICATORIA

Dedico este trabajo a mis padres quienes han sido la inspiración que me ha ayudado a seguir adelante y son el motor que impulsa a seguir con todos mis sueños.

Tulio

FIRMAS RESPONSABLES

	FIRMA	FECHA
ING. IVÁN MENES DECANO FAC. INFORMÁTICA Y ELECTRÓNICA	_____	_____
ING. WILSON BALDEON DIRECTOR ESC. ELECTRÓNICA TELECOMUNICACIONES Y REDES	_____	_____
ING. WILSON BALDEON DIRECTOR DE TESIS	_____	_____
ING. MÓNICA ZABALA MIEMBRO DEL TRIBUNAL	_____	_____
TEC. CARLOS RODRÍGUEZ DIRECTOR CENTRO DOCUMENT.	_____	_____
NOTA DE TESIS	_____	

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO I

MARCO REFERENCIAL

1.1. INTRODUCCIÓN

Actualmente la seguridad en redes está en pleno auge, cada día mas a los usuarios les interesa proteger su información en todos los sentidos, desde la integridad de sus datos hasta la seguridad con los que viajan los mismos a través de la red.

El uso del papel como sistema de archivos comienza a ser cada vez una herramienta poca usada, al contrario el rápido crecimiento de las redes la cual conlleva el crecimiento de las interconexiones y los servicios añadidos de lo que es Internet, ya forman parte de nuestra vida cotidiana, al punto que ya sin ella no viviríamos normalmente.

Hoy en día, Internet y el correo electrónico son medios muy utilizados para la comunicación, quizás los más usados; pero, la mayoría de los usuarios no son

conscientes de la facilidad con que sus mensajes pueden ser interceptados en Internet o en Intranets desprotegidos.

La conectividad es fundamental para muchas actividades, sin embargo, la problemática nace, de la necesidad de determinar cómo conducir las transacciones de misión crítica, ya sea sobre Internet o sobre las Intranet o Extranet de manera segura.

Por lo tanto el riesgo de transmitir información delicada y q la misma se encuentre comprometida a ataques o robo de información es muy elevado, todo el tráfico que enviamos a través de la red ya sea Internet, Intranet o una Extranet si no utilizamos ningún mecanismo de seguridad siempre va a ser una forma insegura de transmitir nuestra información.

Entonces el problema de la seguridad en las redes, y por lo tanto en las comunicaciones, viene a ser uno de los inconvenientes más importantes, en lo que son las transacciones administrativas y/o comerciales, cuyos servicios, tienen más desventajas que ventajas, por lo cual el usuario común no confía aun en dichos servicios.

Dicha tendencia de inseguridad ha creado la necesidad de encontrar procedimientos técnicos y disposiciones legales para asegurar que los documentos electrónicos transmitidos por estas vías sean por lo menos tan confiables y reconocidos legalmente como su contrapartida en papel.

Para lograr esto se pretende utilizar una PKI (Infraestructura de Clave Pública) ayudado de herramientas de software libre, este es un procedimiento que a nivel mundial está siendo utilizada por grandes y pequeñas empresas, instituciones públicas y privadas.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS

Hoy en día los medios digitales son susceptibles de substitución, modificación, y replicación, a menos que estén explícitamente protegidos con el objetivo de que se pueda confiar en estas comunicaciones

Un claro ejemplo es nuestro País, en el entorno administrativo, las Entidades Estatales no hacen uso de medidas de seguridad como la criptografía o certificados digitales para aseguran el envío de información a través de la red, si bien por que no lo necesitan o no se dan cuenta de su necesidad, o porque no tienen conocimiento de ello, y claro esta dentro de las entidades estudiantiles principalmente las universidades.

Todas estas necesidades de seguridad son cada vez más exigentes, optando sin duda por el uso de algún mecanismo de seguridad más avanzado, como es una Infraestructura de Clave Pública PKI. Una PKI está considerada en la actualidad como el mecanismo de seguridad más completo

La infraestructura de clave pública es la combinación de herramientas software y hardware que trabajan en conjunto, mediante esta podemos mejorar la seguridad de nuestra red y observar las mejoras que esta nos puede brindar

Bajo las siglas PKI se engloba a una tecnología estándar, impulsada por la ISO, ITU, y el IETF, que pretende llevar a la práctica los conceptos teóricos de la Criptografía de Clave Pública. La Criptografía de Clave Pública permite, entre otras cosas, implementar sistemas de firma digital y el cifrado de datos sin necesidad de compartición de secretos. La firma digital garantiza la Integridad y el cifrado garantiza la Confidencialidad, pero indirectamente la criptografía de clave pública también permite garantizar la Autenticidad del receptor del mensaje cifrado o del emisor del mensaje firmado brindándonos un extra que es el no repudio, donde el emisor no va a poder denegar que él fue quien envió el mensaje. Esto se consigue con el uso de certificados digitales, donde se asigna una identidad a una clave pública. La utilización de claves (públicas y privadas) y certificados digitales para: firmar y cifrar correos electrónicos, autenticarse ante sitios Web, validar transacciones, solamente tienen éxito cuando existe transparencia entre las aplicaciones y los mecanismos que PKI utiliza para garantizar la seguridad

Cada dispositivo de usuario final posee un software de encriptación y un par de claves: pública para distribuirla a otros usuarios y, otra privada, guardada y protegida por su propietario. Por ejemplo, si un usuario quiere enviar un correo electrónico a otro usuario, el usuario emisor cifra el mensaje utilizando la clave pública del receptor; cuando el mensaje se recibe, el receptor lo descifra con su clave privada, por lo tanto resulta crucial contar con algún método para administrarlas y controlar su utilización. Aquí es donde una PKI entra en juego, permitiendo la creación, distribución, seguimiento y revocación centralizada de

claves, siendo este el método de seguridad más completo que existe hoy en día

Por lo tanto la presente nos quiere dar a conocer una de las tecnologías más utilizadas últimamente mediante herramientas como correo electrónico seguro mediante servicios como antispam, autenticación mediante mensajes encriptados. Web segura mediante certificados digitales. Navegación segura mediante la configuración de un Proxy, todo esto dentro de una red prototipo ayudándonos del uso de software libre para el desarrollo del proyecto el que nos permite tener aplicaciones de calidad y con las cuales podemos reducir costos en gastos por utilizar software privativo.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

- Realizar el análisis de la tecnología PKI y su aplicación en el aseguramiento de los servicios corporativos www, ftp y http en una Red Prototipo

1.3.2. OBJETIVOS ESPECÍFICOS

- Estudiar la arquitectura de la infraestructura de clave pública.
- Conocer los principales ataques informáticos a los que están expuestos los usuarios tanto en Internet, Intranet y Extranet

- Implementar una Infraestructura de Clave Publica para el aseguramiento de la transmisión de información en la Red
- Evaluar el resultado de la ejecución de la PKI en aplicaciones www, web y ftp

1.4. HIPÓTESIS

Al analizar la tecnología PKI – Infraestructura de Clave publica y su implementación mediante el uso de software libre mejorará la seguridad en la transmisión de información dentro de las aplicaciones www, email y ftp

CAPÍTULO II

MARCO TEÓRICO

CONCEPTOS GENERALES

Las redes están constituidas por varias computadoras interconectadas por medio de:

2.1. Protocolos

Reglas para suministrar un lenguaje formal que permita que todos los equipos, sin importar su tecnología, puedan comunicarse entre si.

Uno de los más importantes protocolos es el TCP/IP, el cual proporciona transmisión fiable de paquetes de datos sobre redes. El nombre TCP / IP Proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP). Este protocolo hoy en día permite la

utilización de diversos servicios en la red como: transmisión de correo electrónico, transferencia de archivos, Web, etc.

2.2. Los puertos

Un puerto se representa por un valor de 16 bits que indica al servidor a cual servicio se le esta haciendo una petición.

Por convención, los puertos se encuentran divididos en tres rangos:

- Del 0 al 1023: Puertos denominados “Well Known”. Su uso por convención requiere de privilegios de Superusuario.
- Del 1024 al 49151: Registrados y asignados dinámicamente.
- Del 49152al 65535: Puertos privados.

La tabla que se muestra a continuación presenta un listado de algunos de los puertos más utilizados:

Puerto	**Aplicación**	**Descripción**
21	FTP	Control Transferencia Archivos
22	SSH	Servicio Remoto vía SSL
23	Telnet	Servicio Remoto
25	SMTP	Envió de mails
53	DNS	Servicio de Nombres de Dominios
79	Finger	Información de usuarios
80	WWW-HTTP	World Wide Web
110	POP3(PostOffice)	Recepción de mail

137	NetBios	Intercambio de datos en red
443	HTTPS	http seguro vía SSL
779	Kerberos	
5432	PostgreSQL	Base de Datos

Tabla I.I Puertos de red

2.3. Servicios de red

2.3.1. Web Services

Es un estándar de comunicación entre procesos y o componentes, diseñado para ser multiplataforma y multilenguaje, es decir, no importa en qué lenguaje esté programado un Web Service como ser Visual Basic, C# o java, o en qué plataforma esté corriendo, ya sea Windows, UNIX o Linux éstos serán accesibles y utilizables por otras aplicaciones desarrolladas en otras plataformas o lenguajes de programación. Antiguamente se utilizaban otros estándares como DCOM (Distributed Component Object Model) introducido por Microsoft e implementado por otras plataformas, y CORBA (Common Object Request Broker Architecture) introducido por el OMG (Object Management Group) e implementado en distintas plataformas, incluido Windows. Estos estándares tenían bastantes problemas de configuración, especialmente en entornos en que se encontraban firewalls de por medio en los cuales era imposible (debido a estándares de seguridad de muchas compañías) habilitar ciertos puertos de comunicación para que estos componentes funcionaran. De esta manera la preferencia por utilizar el puerto 80 de HTTP, que normalmente

se encuentra habilitado en la mayoría de los servidores y firewalls debido al uso de navegadores y servidores Web, no traería mayores complicaciones el uso de una tecnología que utilice este protocolo y puerto de TCP/IP.

La gran ventaja que trae el protocolo HTTP es su esquema de mensajes especialmente diseñado y optimizado para ser utilizado en redes como Internet, a diferencia de las viejas tecnologías como DCOM o CORBA que necesitaban un tipo de red más estable y local (LAN). Por ello es que el HTTP es el protocolo preferido para el transporte de mensajes de los Web Services.

2.3.1.1. Web 2.0

El término Web 2.0 está asociado a aplicaciones web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos que se ha creado para ellos. Ejemplos de la Web 2.0 son las comunidades web, los servicios web, las aplicaciones Web, los servicios de red social, los servicios de alojamiento de videos, las wikis, blogs.

Aunque el término sugiere una nueva versión de la World Wide Web, no se refiere a una actualización de las especificaciones técnicas de la web, sino más bien a cambios acumulativos en la forma en la que desarrolladores de software y usuarios finales utilizan la Web.

2.3.1.2. Web 3.0

Web 3.0 es una expresión que se utiliza para describir la evolución del uso y la interacción de las personas en internet a través de diferentes formas entre los que se incluyen la transformación de la red en una base de datos, un movimiento social hacia crear contenidos accesibles por múltiples aplicaciones non-browser, el empuje de las tecnologías de inteligencia artificial, la web semántica, la Web Geoespacial o la Web 3D. La expresión es utilizada por los mercados para promocionar las mejoras respecto a la Web 2.0.

Las tecnologías de la Web 3.0, como programas inteligentes, que utilizan datos semánticos, se han implementado y usado a pequeña escala en compañías para conseguir una manipulación de datos más eficiente. En los últimos años, sin embargo, ha habido un mayor enfoque dirigido a trasladar estas tecnologías de inteligencia semántica al público general.

2.3.2. Correo electrónico

Cuando se envía un correo electrónico, el mensaje se enruta de servidor a servidor hasta llegar al servidor de correo electrónico del receptor. Más precisamente, el mensaje se envía al servidor del correo electrónico (llamado MTA, del inglés Mail Transport Agent, Agente de Transporte de Correo) que tiene la tarea de transportarlos hacia el MTA del destinatario. En

Internet, los MTA se comunican entre sí usando el protocolo SMTP, y por lo tanto se los llama servidores SMTP.

Luego el MTA del destinatario entrega el correo electrónico al servidor del correo entrante (llamado MDA, del inglés Mail Delivery Agent, Agente de Entrega de Correo), el cual almacena el correo electrónico mientras espera que el usuario lo acepte. Existen dos protocolos principales utilizados para recuperar un correo electrónico de un MDA:

- POP3 (Post Office Protocol, Protocolo de Oficina de Correo), el más antiguo de los dos, que se usa para recuperar el correo electrónico y, en algunos casos, dejar una copia en el servidor.
- IMAP (Internet Message Access Protocol, Protocolo de Acceso a Mensajes de Internet), el cual se usa para coordinar el estado de los correos electrónicos (leído, eliminado, movido) a través de múltiples clientes de correo electrónico. Con IMAP, se guarda una copia de cada mensaje en el servidor, de manera que esta tarea de sincronización se pueda completar.

Por esta razón, los servidores de correo entrante se llaman servidores POP o servidores IMAP, según el protocolo usado.

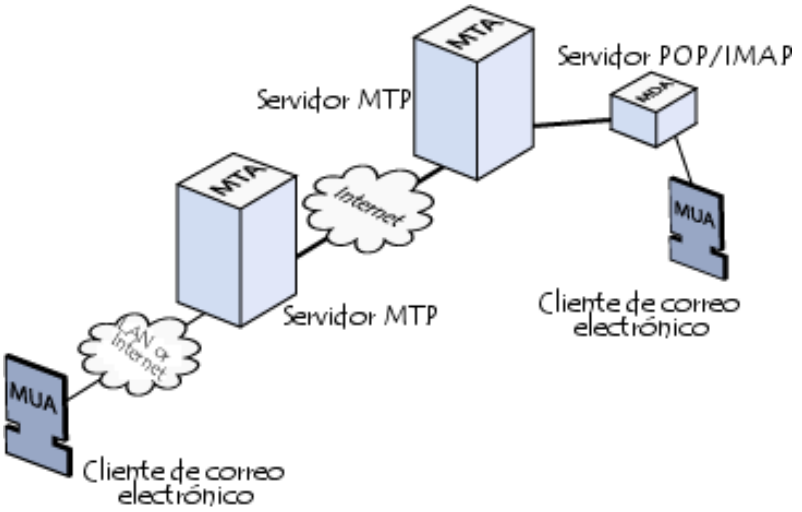

Figura II.1 Correo Electrónico Servidores MTP e IMAP

Usando una analogía del mundo real, los MTA actúan como la oficina de correo (el área de clasificación y de transmisión, que se encarga del transporte del mensaje), mientras que los MDA actúan como casillas de correo, que almacenan mensajes (tanto como les permita su volumen), hasta que los destinatarios controlan su casilla. Esto significa que no es necesario que los destinatarios estén conectados para poder enviarles un correo electrónico.

Para evitar que cualquiera lea los correos electrónicos de otros usuarios, el MDA está protegido por un nombre de usuario llamado registro y una contraseña.

La recuperación del correo se logra a través de un programa de software llamado MUA (Mail User Agent, Agente Usuario de Correo).

Cuando el MUA es un programa instalado en el sistema del usuario, se llama cliente de correo electrónico (tales como Microsoft Outlook, etc.).

Cuando se usa una interfaz de web para interactuar con el servidor de correo entrante, se llama correo electrónico.

2.3.3. Servidor FTP

Un servidor FTP es un programa especial que se ejecuta en un equipo servidor normalmente conectado a Internet (aunque puede estar conectado a otros tipos de redes, LAN, MAN, etc.). Su función es permitir el intercambio de datos entre diferentes servidores/ordenadores.

Por lo general, los programas servidores FTP no suelen encontrarse en los ordenadores personales, por lo que un usuario normalmente utilizará el FTP para conectarse remotamente a uno y así intercambiar información con él.

Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes; o como servidor de backup (copia de seguridad) de los archivos importantes que pueda tener una empresa. Para ello, existen protocolos de comunicación FTP para que los datos se transmitan encriptados, como el SFTP (Secure File Transfer Protocol).

2.3.3.1. Cliente FTP

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar ficheros en un ordenador remoto, se necesitará utilizar un programa

cliente FTP. Un cliente FTP es un programa que se instala en el ordenador del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.

Para utilizar un cliente FTP, se necesita conocer el nombre del fichero, el ordenador en que reside (servidor, en el caso de descarga de archivos), el ordenador al que se quiere transferir el archivo (en caso de querer subirlo nosotros al servidor), y la carpeta en la que se encuentra.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Windows, DOS, Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

2.3.3.2. Modos de conexión del cliente FTP

FTP admite dos modos de conexión del cliente. Estos modos se denominan Activo (o Estándar, o PORT, debido a que el cliente envía comandos tipo PORT al servidor por el canal de control al establecer la conexión) y Pasivo (o PASV, porque en este caso envía comandos tipo PASV). Tanto en el modo Activo como en el modo Pasivo, el cliente establece una conexión con el servidor mediante el puerto 21, que establece el canal de control.

2.4. Seguridad Informática

Seguridad informática es el conjunto de procedimientos, estrategias y herramientas que permitan garantizar el cumplimiento de cinco objetivos importantes:

2.4.1. Confidencialidad

La confidencialidad asegura que la información no esté disponible para personas, procesos o programas no autorizados, por lo cual solo los usuarios autorizados pueden manipularla.

En si la confidencialidad se refiere a la protección de datos frente a la difusión no autorizada, la perdida de confidencialidad puede resultar en problemas legales, perdida del negocio o de credibilidad.

El robo de información, como el robo de contraseñas u otros datos a medida que viaja sin cifrar a través de redes de confianza, es un ataque de confidencialidad, ya que permite a una persona distinta del destinatario para obtener acceso a los datos.

2.4.2. Autenticación

Es el proceso de verificar la identidad de los actores y autorización por parte de la entidad autorizadora en una comunicación.

Los profesionales de la seguridad en redes son responsables de mantener la seguridad de los datos de una organización y garantizar la integridad, disponibilidad y confidencialidad de la información.

Falsificación de direcciones MAC es un ataque de autenticación, ya que permite que un dispositivo no autorizado a conectarse a la red cuando Media Access Control (MAC) de filtrado está en su lugar, como en una red inalámbrica.

2.4.3. Integridad

La integridad garantiza que la información no sea modificada por personas, procesos o programas que no sean debidamente autorizados para ello.

Es necesario asegurar que los datos no sufran cambios no autorizados, la perdida de integridad puede acabar en fraudes, decisiones erróneas o como paso a otros ataques.

2.4.4. Disponibilidad

La disponibilidad significa que la información se encuentra disponible para ser requerida por un usuario, proceso o programa, en todo momento que lo requiera.

También se refiere a la continuidad operativa de la entidad, la perdida de disponibilidad puede implicar, la pérdida de productividad o de credibilidad de la entidad.

En la realización de un ataque de denegación de servicio (DoS), un hacker ataca a los elementos de disponibilidad de sistemas y redes.

2.4.5. No Repudio

Proporciona protección contra la interrupción, por parte de alguna de las entidades implicadas en la comunicación, de haber participado en toda o parte de la comunicación.

- No Repudio de origen: El emisor no puede negar que envío porque el destinatario tiene pruebas del envío, el receptor recibe una prueba infalsificable del origen del envío, lo cual evita que el emisor, de negar tal envío, tenga éxito ante el juicio de terceros. En este caso la prueba la crea el propio emisor y la recibe el destinatario. Prueba que el mensaje fue enviado por la parte específica.
- No Repudio de destino: El receptor no puede negar que recibió el mensaje porque el emisor tiene pruebas de la recepción. Este servicio proporciona al emisor la prueba de que el destinatario legítimo de un envío, realmente lo recibió, evitando que el receptor lo niegue posteriormente. En este caso la prueba irrefutable la crea el receptor y la recibe el emisor. Prueba que el mensaje fue recibido por la parte específica.

Si la autenticidad prueba quién es el autor de un documento y cual es su destinatario, el “no repudio” prueba que el autor envió la comunicación (no repudio en origen) y que el destinatario la recibió (no repudio en destino).

El no repudio evita que el emisor o el receptor nieguen la transmisión de un mensaje. Así, cuando se envía un mensaje, el receptor puede comprobar que, efectivamente, el supuesto emisor envió el mensaje. De forma similar, cuando se recibe un mensaje, el emisor puede verificar que, de hecho, el supuesto receptor recibió el mensaje. Definición según la recomendación X.509 de la UIT-T Servicio que suministra la prueba de la integridad y del origen de los datos ambos en una relación infalsificable que pueden ser verificados por un tercero en cualquier momento.

2.4.6. Ataques y Amenazas

Uno de los factores de seguridad que son muy susceptibles para cualquier usuario, es el que alguna persona no autorizada, tenga acceso a sus datos, o monitoree sus transacciones, esto tiende a definir lo que es una amenaza.

Las categorías generales de ataques o amenazas son las siguientes: Interrupción, Intercepción, Modificación, Suplantación. Los cuales pueden ser pasivos o activos, dependiendo del nivel de intromisión realizada por algún intruso.

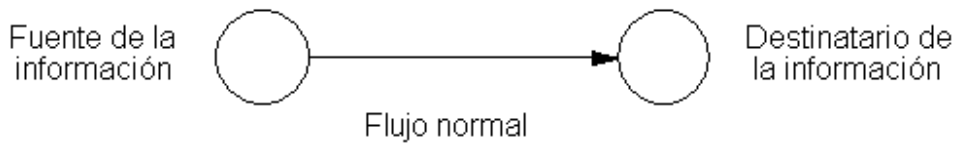

Figura II.2 Flujo Normal de la información

En los ataques pasivos el atacante no altera la comunicación, sino que únicamente la escucha o monitoriza, para obtener la información que está siendo transmitida. Sus objetivos son la intercepción de datos y el análisis de tráfico.

Los ataques pasivos son muy difíciles de detectar, ya que no provocan ninguna alteración de los datos. Sin embargo, es posible evitar su éxito mediante el cifrado de la información.

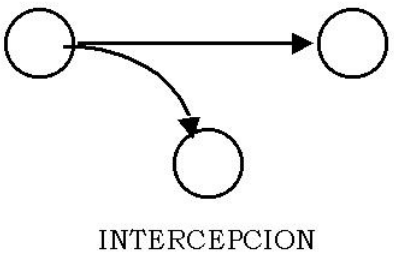

Figura II.3 Ataque Pasivo

Por otra parte los ataques activos, implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos

Estas amenazas y ataques, pueden ser realizados en sistemas aislados, que actualmente vendrían a ser una cantidad mínima, y en sistemas interconectados , que es donde los sistemas presentan más vulnerabilidades.

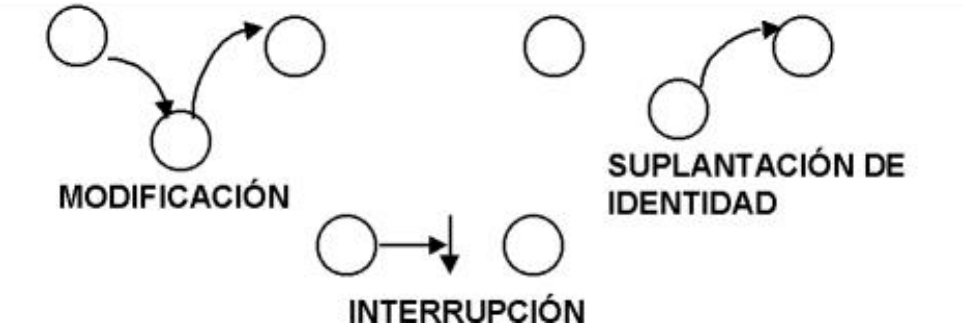

Figura II.4 Ataques Activos

Así mismo dependiendo del entorno, se podrían dar ataques externos o físicos relacionados con la protección de los soportes físicos de la información, más que a la información propiamente.

2.4.7. Mecanismos de Seguridad

No existe un único mecanismo capaz de proveer todos los servicios anteriormente citados, pero la mayoría de ellos hacen uso de técnicas criptográficas basadas en el cifrado de la información, los más importantes son los siguientes:

- **Intercambio de autenticación:** garantiza que una entidad, ya sea origen o destino de la información es la deseada.
- **Cifrado:** consiste en transformar un texto plano, es decir no encriptado mediante un proceso de cifrado en un texto cifrado, gracias a una información secreta o clave de cifrado.

- **Integridad de datos:** implica el cifrado de una cadena comprimida de datos a transmitir, llamada generalmente valor de comprobación de integridad (ICV – Integry Check Value).
- **Firma digital:** implica el cifrado, por medio de la clave secreta del emisor, de una cadena comprimida de datos que se va a transferir.
- **Control de acceso:** sólo aquellos usuarios autorizados acceden a los recursos del sistema o a la red.
- **Tráfico de relleno:** envía tráfico espurio junto con los datos válidos para que el atacante no sepa si se está enviando información, ni qué cantidad de datos útiles se están transmitiendo.
- **Control de encaminamiento:** permite enviar determinada información por determinadas zonas consideradas clasificadas.
- **Unicidad:** consiste en añadir a los datos un número de secuencia, la fecha y hora, un número aleatorio, o alguna combinación de los anteriores, que se incluyen en la firma digital o integridad de datos.

2.5. Criptografía

La palabra criptografía proviene del griego “kryptos” que significa ocultar y “grafos” que significa escribir, literalmente sería “escritura oculta”.

La criptografía es el arte o ciencia de cifrar y descifrar información utilizando técnicas matemáticas que hace posible que la transferencia de información sea segura y que solo pueda ser leída por las personas a quienes va dirigida.

La finalidad de la criptografía es, en primer lugar, garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc.) y, en segundo lugar, asegurar que la información que se envía es auténtica en un doble sentido: que el remitente sea realmente quien dice ser y que el contenido del mensaje enviado, habitualmente denominado criptograma, no haya sido modificado en su tránsito.

2.5.1. Tipos

Se puede distinguir principalmente tres tipos de criptografía: la simétrica, la asimétrica y la híbrida.

2.5.1.1. Sistemas de Cifrado Simétrico

Estos sistemas son también llamados de clave única, ya que utilizan la misma clave tanto para cifrar como para descifrar un mensaje, por lo que la clave debe ser compartida por el emisor y el receptor de éste. En el esquema simétrico para poder mantener la efectividad, la clave debe ser mantenida en secreto por ambos entes, ya que el nivel de seguridad depende exclusivamente del nivel de protección de la clave.

Estos sistemas han tenido muy buena aceptación ya que permiten hacer computaciones elevadas y se caracterizan por ser altamente eficientes con relación al tamaño de su clave, además de ser rápidos y robustos.

En este tipo de sistemas la confidencialidad y la autenticidad se obtienen al mismo tiempo, ya que cuando se descifra un mensaje usando la clave privada, el hecho de que ésta sea tan sólo conocida por el emisor y el receptor garantiza entre comillas las dos propiedades:

- Que el mensaje no es legible para nadie más, o sea confidencialidad.
- Y dado que el texto descifrado es legible, sólo hay un emisor posible, aquel que conoce la clave privada, o sea autenticidad del mensaje.

El problema fundamental de este tipo de criptosistemas es la generación, almacenamiento y sobre todo el intercambio de las claves, debido a que éstas se distribuyen en entornos no seguros. Por tanto las claves deben intercambiarse de un modo eficiente, pues es en ello donde descansan todas las características de seguridad de estos sistemas.

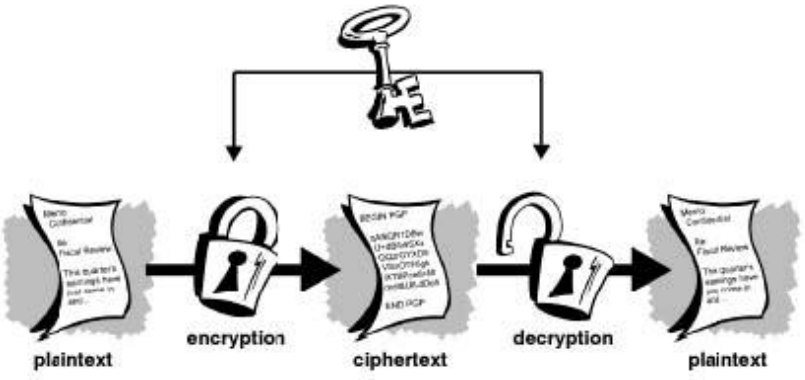

Figura II.5 Proceso de criptografía simétrica

2.5.1.1.1. DES

El algoritmo DES fue diseñado por IBM y es utilizado extensamente dado que se ha convertido en un estándar al estar reconocido por las agencias norteamericanas. Los requisitos que debía cumplir el estándar de cifrado eran:

- Contar con un nivel de seguridad computacional alto.
- Estar especificado en todos sus detalles de forma entendible.
- Su seguridad no debe verse comprometida si se hace público el algoritmo.
- Su implementación en dispositivos electrónicos debía ser de bajo costo.
- Debía poder implementarse en hardware.

La opción más común que se ha elegido para suplantar a DES ha sido usar lo que se conoce como cifrado múltiple, es decir aplicar varias veces el mismo algoritmo para fortalecer la longitud de la clave, esto ha tomado la forma de un nuevo sistema de cifrado que se conoce actualmente como Triple DES o TDES.

2.5.1.1.2. TDES

Debido a que el sistema DES se considera actualmente poco practico por la corta longitud de su clave y para poder continuar utilizándolo, se creo Triple DES. El funcionamiento de Triple DES consiste en aplicar tres veces el sistema DES con claves de 56 bits distintas, con esto se consigue una longitud de clave de 128 bits. Esto se basa en que DES tiene una característica matemática particular, lo que implica que si se cifra el mismo bloque dos veces con claves diferentes, se aumenta el tamaño efectivo de la clave.

2.5.1.1.3. AES

El sistema AES nace en 1997, convoca a un concurso internacional para definir el Estándar de Cifrado Avanzado como sucesor del ya casi obsoleto DES, de manera que el nuevo algoritmo sea capaz de proteger información hasta bien avanzado el siglo XXI.

Dentro de las especificaciones y requisitos mínimos para este nuevo estándar criptográfico, se encontraban:

- Ser de carácter público.
- Ser un algoritmo de cifrado simétrico en bloque.
- Estar diseñado de manera que se pueda aumentar la longitud de clave según las necesidades.
- Ser implementable tanto en hardware como en software.
- Los algoritmos que cumplan con tales requisitos serían juzgados de acuerdo a:
    - Eficiencia computacional.
    - Seguridad o esfuerzo necesario para criptoanalizarlos.
    - Requisitos de memoria.
    - Simplicidad, flexibilidad y requisitos de licencia.

2.5.1.1.4. IDEA

El sistema IDEA a pesar de que lleva solamente unos años en uso es probablemente uno de los mejores algoritmos de bloques existente. Trabaja con bloques de 64 bits y utiliza claves de 128 bits, el número de iteración que utiliza son 8 en las cuales realiza el cifrado sobre la base de operaciones XOR, suma y multiplicación de enteros, es similar a DES, pero las iteraciones son más complejas.

2.5.1.2 Sistemas de Cifrado Asimétrico

En 1976 Whitfield Diffie y Martin Hellman en su trabajo titulado “New Directions in Cryptography” proponen las bases de un nuevo tipo de criptografía denominada de clave pública como solución a los problemas de gestión de claves que posee la criptografía simétrica. Este tipo de sistemas también es conocido como de dos claves, donde cada usuario tiene una clave que es privada sólo conocida por él y otra que es pública que puede ser revelada a todos los otros usuarios de un sistema.

La característica más destacable de estas claves, es que están relacionadas matemáticamente siendo una la inversa de la otra. La fortaleza del sistema depende del tamaño de las claves y la imposibilidad computacional de obtener una de las claves a partir de su inversa, esta imposibilidad se basa en que no se tiene el suficiente tiempo para romper el sistema con los medios técnicos actuales.

La razón del porque a este sistema se le denomina también asimétrico es debido a que no se puede utilizar la misma clave para cifrar y descifrar un mensaje, si se cifra un mensaje con una clave se debe descifrar con la otra.

Por lo tanto si un usuario desea enviar un mensaje secreto a otro debe utilizar la clave pública del receptor para cifrar el mensaje, ya que así sólo éste podrá descifrarlo utilizando la clave privada correspondiente, que sólo él debe conocer. Este procedimiento permite garantizar la confidencialidad de la comunicación, pero no garantiza su autenticidad, dado que todos pueden tener acceso a la clave pública del receptor, por lo que cualquier usuario podría ser el emisor de un mensaje. Por lo tanto si el emisor quiere garantizar la autenticidad del mensaje, debe cifrarlo con su propia clave privada, dado que todo el mundo conoce la clave pública con la cual descifrar el mensaje. Este proceso garantiza el origen del mensaje o sea la autenticidad, pero no garantiza la confidencialidad, ya que cualquiera puede descifrarlo.

Como se puede ver este nuevo sistema logra superar las dificultades de los sistemas simétricos proporcionando algunos de los servicios de seguridad buscados y mencionados desde el principio de este trabajo. Sin embargo la confidencialidad y autenticidad en el sistema asimétrico se logran por separado, entonces para lograr ambos objetivos es necesario combinar los procedimientos de forma apropiada.

Desgraciadamente este tipo de sistema no está libre de dificultades, uno de los principales es que los algoritmos de clave pública son alrededor de mil veces más lentos que sus pares de clave privada, debido a esta razón el

cifrado de la información se suele realizar mediante criptosistemas de clave privada, mientras que los sistemas de clave pública se reservan para el cifrado de la clave utilizada para cifrar la información, ya que ésta posee menos datos. Además, de necesitar enviar un archivo cifrado a varias personas es necesario generar distintos archivos cifrados, porque se debe utilizar para cada una su respectiva clave pública.

Otro de sus inconvenientes se refleja en la necesidad de poder garantizar que el par de claves pertenecen a quien dice ser el dueño de ellas, lo que no nos permite aun garantizar el no repudio.

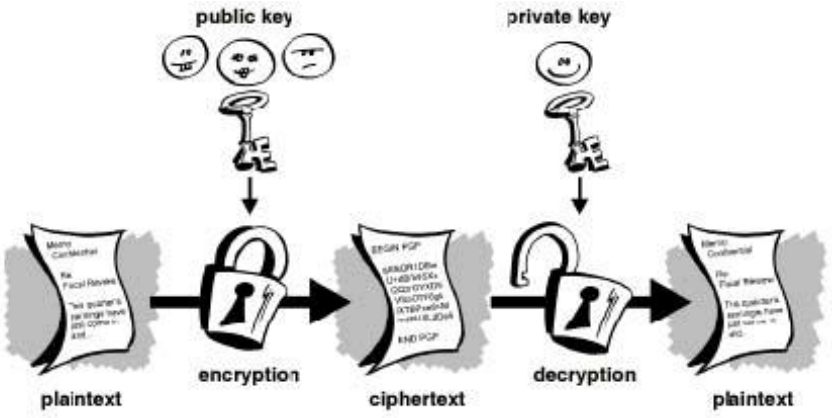

Figura II.6 Proceso de criptografía asimétrica (Encriptación con clave pública)

2.5.1.2.1 RSA

Dentro de los sistemas de clave pública está el RSA, que es quizás el sistema de cifrado más empleado y extendido en la actualidad. El RSA fue creado por Rivest, Shamir y Adleman, de ahí su nombre, quienes lo desarrollaron en el MIT en 1978 basados en los estudios de Diffie y Hellman. Este sistema utiliza dos claves y cualquiera de las dos puede ser pública o privada. Las claves se

generan matemáticamente a partir de la multiplicación de grandes números primos y otras complejas operaciones matemáticas.

Existen dos formas principales de utilizar RSA dependiendo de la aplicación:

- Esquema de cifrado. Se usa principalmente para cifrar claves de sistemas simétricos.
- Esquema de firma digital. Se usa para autentificar mensajes, cuenta con dos partes, una denominada Proceso de Firma o cifrado y la segunda denominada proceso de Verificación de la Firma o descifrado.

RSA es sin duda uno de los sistemas más estudiados hasta el momento y por lo tanto se considera uno de los más seguros ya que ha podido superar todo tipo de controversia, por lo que es uno de los sistemas criptográficos de llave pública más usados en el comercio y en general en toda actividad que requiera para su información un alto grado de seguridad.

Sin embargo, se han estado desarrollando una gran cantidad de sistemas de clave pública con el fin de sustituir o simplemente competir con RSA, los que no han tenido gran éxito. En principio estos nuevos sistemas deben de pasar un riguroso criptoanálisis por parte de la comunidad criptográfica, la prueba es en general proporcionar al menos la misma seguridad que los sistemas existentes con similar facilidad de implementación y que basen su seguridad en problemas muy complejos.

2.5.1.2.2. DSA

El algoritmo DSA es una modificación del sistema ElGamal y fue propuesto como algoritmo estándar de firma digital dentro del DSS por el NIST en 1991. Esta medida fue muy criticada sobre todo por RSA Data Security, ya que esta compañía quería que su algoritmo se convirtiera en el estándar.

Dentro de los inconvenientes de DSA y en los que se basaron las criticas de RSA se encuentran que este sistema es solamente un estándar para firma digital, por lo tanto no se puede utilizar en el cifrado de información y por ende para distribuir claves. Otra desventaja de DSA es su lentitud en comparación con RSA en cuanto a la verificación de la firma digital, una ventaja eso si de DSA es que al momento de generar las firmas digitales, éste realiza dicha función con mayor agilidad.

2.5.1.2.3. ECC

Otro sistema criptográfico de clave pública es el ECC que usa curvas elípticas definidas en un campo finito. Este sistema puede ser usado tanto para cifrar como para firmar digitalmente.

La ventaja más destacable que ofrecen los ECC en comparación con RSA es la longitud de la clave secreta, se puede mostrar que mientras en RSA se tiene que usar una clave de 1024 bits para ofrecer una seguridad considerable, los ECC sólo usan 160 bits para ofrecer lo mismo, así también las claves RSA de 2048 bits son equivalentes en seguridad a 210 bits de ECC. Los requerimientos de memoria y CPU para realizar las operaciones criptográficas son también bastante inferiores por lo que este sistema es muy adecuado para ambientes

restringidos en recursos donde el poder de computo es reducido y requiera una alta velocidad de procesamiento y grandes volúmenes de transacciones, lo que permite su uso por ejemplo en tarjetas inteligentes y celulares.

Su desventaja fundamental es que muchas de sus variantes están patentadas por lo que no pueden utilizarse de forma libre.

2.5.1.3. Criptografía híbrida

Este tipo de criptografía utiliza tanto el cifrado simétrico como el asimétrico. Emplea el cifrado de clave pública para compartir una clave para el cifrado simétrico. El mensaje que se envía en el momento, se cifra usando la clave única (cifrado asimétrico) y se envía al destinatario.

Tanto PGP como GnuPG usan sistemas de cifrado híbridos.

2.6. Certificados

2.6.1. Certificado digital

Es un documento electrónico emitido por una empresa denominada “Autoridad de certificación” que garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.

2.6.2. Autoridad de certificación

Una autoridad de certificación (AC o CA por sus siglas en inglés Certification Authority) o entidad de certificación, es una persona jurídica que presta servicios de emisión, gestión, cancelación u otros servicios inherentes a la certificación digital. Asimismo, puede asumir las funciones de registro o verificación. Las CAs disponen de sus propios certificados públicos, cuyas claves privadas asociadas son empleadas por las CAs para firmar los certificados que emiten. Un certificado de CA puede estar auto-firmado cuando no hay ninguna CA de rango superior que lo firme. Este es el caso de los certificados de CA raíz (autoridad de certificación raíz), el elemento inicial de cualquier jerarquía de certificación. Una jerarquía de certificación consiste en una estructura jerárquica de CAs en la que se parte de una CA auto-firmada, y en cada nivel, existe una o más CAs que pueden firmar certificados de entidades finales (personas, aplicación de software, etc.) o bien certificados de otras CAs subordinadas, plenamente identificadas y cuya Política de Certificación sea compatible con la CA de rango superior.

2.6.3. Proceso para la obtención de un certificado digital

El proceso para obtener un certificado digital es el siguiente:

1. El solicitante se dirige a una empresa o entidad que tenga el carácter de Prestador de Servicios de Certificación y solicita de ellos las claves y el certificado digital correspondiente a las mismas. Este trámite generalmente se puede realizar presencialmente, acudiendo a dicha entidad o virtualmente,

por medio de Internet, utilizando la página Web del Prestador de Servicios de Certificación.

2. El prestador de Servicios de Certificación comprobará la identidad del solicitante, bien sea directamente o por medio de entidades colaboradoras (Autoridades Locales de Registro), para lo cual se deberá mostrar el D.N.I. y si se trata de un representante de una sociedad (administrador, apoderado, etc.) o de cualquier otra persona jurídica, deberá acreditar documentalmente el cargo y las facultades del mismo (vigencia de poderes)

3. El prestador de Servicios de Certificación mediante los dispositivos técnicos adecuados crea las claves pública y privada que le corresponde al solicitante, y genera el certificado digital correspondiente a dichas claves.

Figura II.7 Proceso de obtención de un certificado digital

2.6.4. Firma digital

La firma digital hace referencia, en la transmisión de mensajes telemáticos y en la gestión de documentos electrónicos, a un método de cifrado que asocia la identidad de una persona o de un equipo informático al mensaje o documento. En función del tipo de firma, puede, además, asegurar la integridad del documento o mensaje.

La firma electrónica, como la firma ológrafa (autógrafa, manuscrita), puede vincularse a un documento para identificar al autor, para señalar conformidad (o disconformidad) con el contenido, para indicar que se ha leído o, según el tipo de firma, garantizar que no se pueda modificar su contenido.

2.6.4.1. Terminología

Los términos de firma digital y firma electrónica se utilizan con frecuencia como sinónimos, pero este uso en realidad es incorrecto.

Mientras que firma digital hace referencia a una serie de métodos criptográficos, firma electrónica es un término de naturaleza fundamentalmente legal y más amplia desde un punto de vista técnico, ya que puede contemplar métodos no cifrados.

La firma digital de un documento es el resultado de aplicar cierto algoritmo matemático, denominado función hash, a su contenido y, seguidamente, aplicar el algoritmo de firma (en el que se emplea una clave privada) al resultado de la operación anterior, generando la firma electrónica o digital. El

software de firma digital debe además efectuar varias validaciones, entre las cuales podemos mencionar:

- Vigencia del certificado digital del firmante,
- Revocación del certificado digital del firmante (puede ser por OCSP o CRL), Inclusión de sello de tiempo.

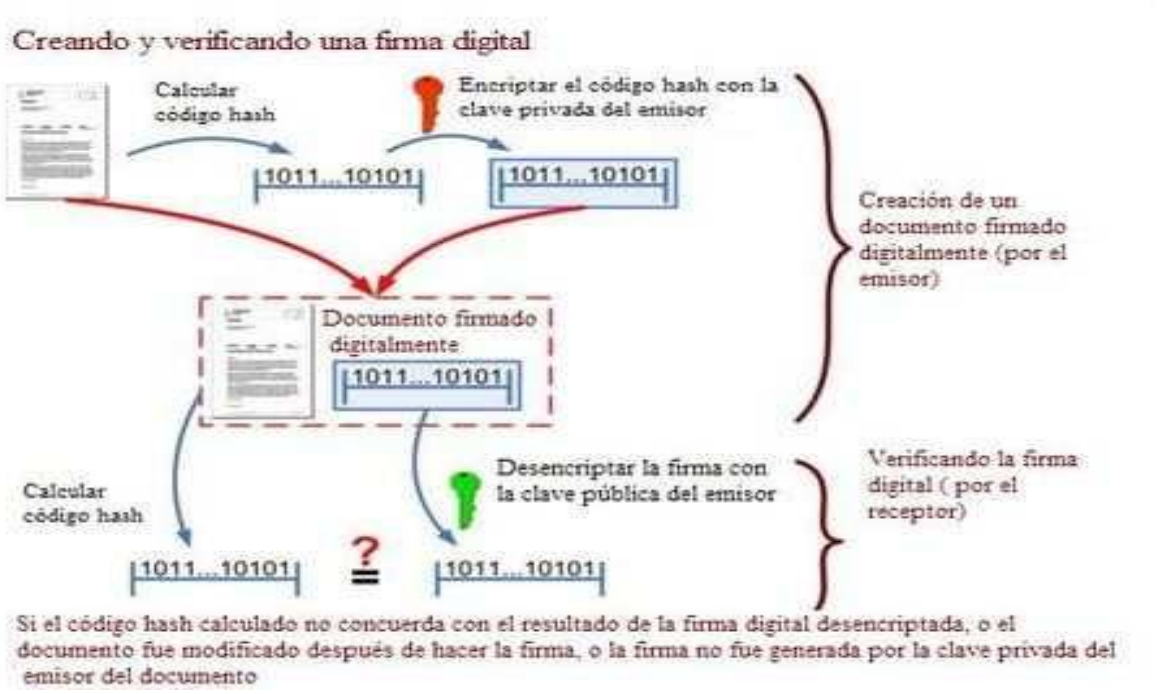

Figura II.8 Creación de una firma digital

2.6.4.2. Aplicaciones

- Mensajes con autenticidad asegurada
- Mensajes sin posibilidad de repudio
- Contratos comerciales electrónicos
- Factura Electrónica
- Desmaterialización de documentos

- Transacciones comerciales electrónicas
- Invitación electrónica
- Dinero electrónico
- Notificaciones judiciales electrónicas
- Voto electrónico
- Decretos ejecutivos (gobierno)
- Créditos de seguridad social
- Contratación pública
- Sellado de tiempo

2.6.5. Funciones Hash

Las funciones Hash o de resumen son una herramienta fundamental en la criptografía moderna y son usadas principalmente para resolver otro de los aspectos de seguridad buscados, la integridad de los mensajes.

Estas funciones son ampliamente utilizadas en conjunto con la firma digital, ya que al ser aplicadas sobre un documento de tamaño variable, producen un pequeño resumen de tamaño constante.

Las funciones de resumen deben poseer las siguientes características para ser consideradas como tales:

- Cualquier cambio en el mensaje, por mínimo que sea, debe producir un resumen distinto.

- Debe tener compresión, o sea a partir de un mensaje de cualquier longitud, el resumen debe tener una longitud fija y lo normal es que sea menor que la del mensaje.
- La función no debe poder invertirse, debe ser unidireccional para que impida obtener el mensaje original a través del resumen.
- Debe ser fácil y rápida de calcular.
- El resumen debe ser una función compleja de todos los bits del mensaje, debe proveer Difusión.

Los algoritmos Hash no emplean claves de ningún tipo, sino que se basan en extraer una determinada cantidad de bits a partir de un texto de longitud variable, o sea cada cierta cantidad de texto elegido de forma arbitraria se procede a realizar una transformación de bits, de esta transformación se obtiene una palabra, esta palabra tiene una extensión de X bits prestablecidos, de esta forma el texto se hace irreconocible, al poder leer sólo números secuénciales que no guardan relación alguna entre si.

2.6.5.1. MD5

Este algoritmo produce un resumen de 128 bits a partir de un bloque de texto de cualquier longitud, para ello divide el texto en bloques de tamaño fijo y luego realiza una serie de operaciones matemáticas en bloques sucesivos. Es uno de los algoritmos de resumen más extendido lo cual lo hace estar presente en variadas especificaciones y aplicaciones de seguridad.

2.6.5.2. SHA-1

El algoritmo SHA-1 fue desarrollado por el gobierno de los EE.UU. a través del NIST con la ayuda de la NSA para ser incluido en el estándar DSS. Su funcionamiento es similar al MD5 con la diferencia que SHA-1utiliza un vector más de 32 bits, con lo que obtiene resúmenes de 160 bits, lo que lo hace más seguro y evita posibles colisiones.

2.6.6. Infraestructura de clave pública (PKI)

Es una combinación de hardware y software, políticas y procedimientos de seguridad que permiten la ejecución segura de operaciones criptográficas (como el cifrado, la firma digital, el no repudio de transacciones electrónicas).

2.7. PKI

2.7.1. Introducción

Buena parte del interés en el desarrollo de las PKIs es el resultado del crecimiento de Internet. Concretamente, el aumento del comercio electrónico ha provocado una toma de conciencia relativa a las cuestiones de seguridad lo que, a su vez, ha incrementado el esfuerzo dedicado a estandarizar todos los aspectos de las PKIs. En 1988 se publica la primera versión de la

recomendación X.509 que proporciona un formato normalizado para certificados y CRLs (Certificate Revocation List). Periódicamente han ido apareciendo nuevas versiones del estándar para incorporar nuevas funcionalidades. La IETF, responsable de los estándares que gobiernan la operación de Internet, ha escrito una serie de RFCs para el uso de una PKI en Internet basada en el estándar X.509. Tal conjunto de documentos se engloban bajo el nombre de PKIX. La misma IETF ha propuesto otro esquema de certificado más simple que el X.509 denominado SPKI, pero que no ha tenido un amplio seguimiento. Cabe decir que en el campo de los estándares se ha progresado significativamente y se ha alcanzado una madurez que permite una sólida implementación, despliegue e interoperación.

Hay un cierto consenso en la actualidad sobre el hecho de que los beneficios de una PKI compensan el coste de su mantenimiento y también sobre el hecho de que las PKIs abren un horizonte de nuevas posibilidades de negocio que antes eran inviables por riesgos de seguridad, legislación, etc.

2.7.2. Aplicaciones que utilizan certificados digitales

Existen varias aplicaciones basadas en protocolos que manejan certificados digitales durante los procesos de autenticación o encripción; entre las principales aplicaciones se pueden mencionar:

- Establecimiento de sitios Web seguros.
- Creación de correos electrónicos seguros.

2.7.2.1. Sitios WEB Seguros

Las aplicaciones que utilizan certificados digitales para la autenticación de sitios Web son tal vez las más difundidas actualmente, estas aplicaciones permiten realizar transacciones seguras a través de un portal sin la necesidad de conocer al propietario del sitio o a sus auspiciantes.

Este tipo de aplicaciones utilizan protocolos que permiten una autenticación del lado del servidor; es decir, el servidor posee un certificado digital que le permite autenticarse y establecer sesiones seguras a través de una red. Entre los protocolos más utilizados para crear sitios Web seguros se tiene a SSL (Secure Sockets Layer) y TLS (Transport Layer Security).

2.7.2.2. SSL

SSL es el protocolo más utilizado para la autenticación de sitios Web, fue publicado por Netscape, con el propósito de proveer privacidad e integridad entre dos aplicaciones que se comunican entres sí.

El protocolo está formado por dos capas, la capa inferior conocida como SSL Record Protocol puede funcionar sobre un protocolo de capa transporte confiable y es la encargada de la encapsulación de varios protocolos de capas superiores.

2.7.3. Estándares utilizados en PKI

2.7.3.1.- Public key cryptography standards (PKCS)

Los Estándares Criptográficos de Clave Pública fueron introducidos por la RSA Data Security para las entidades que desean una interfaz estándar con la criptografía de clave pública. A diferencia con otros estándares que son apoyados por otros organismos internacionales los PKCS son una aproximación a un estándar al mundo de la criptografía.

Muchas organizaciones como Apple, Microsoft, Digital, Lotus, Sun y Massachussets Institute of Technology han participado en su desarrollo pero solo la RSA Data Security toma la última decisión en su promulgación y revisión. Actualmente está constituido por doce normas:

1) PKCS#1 (RSA Encryption Standard) en esta norma se describe un método para utilizar el algoritmo RSA, su finalidad es producir firmas digitales de mensajes y mensajes cifrados, utilizando la sintaxis definida por la norma PKCS#7. Las firmas digitales se producen aplicando la función Hash al mensaje y cifrado de la huella digital que resulta de la clave privada del firmante. Para conseguir la encriptación de mensajes se cifra primero con una clave simétrica y luego ésta clave es cifrada con la clave pública del destinatario del mensaje. PKCS#2 y PKCS#4 se han incorporado a la PKCS#1

2) PKCS#3 (Diffie-Hellman Key-Agreement Standard) en éste se describe un método para implementar el intercambio de claves Diffie-Hellman.

3) PKCS#5 (Password-Based Encryption Standard) en esta norma se describe un método para cifrar los mensajes con la clave secreta. Su objetivo es permitir la transmisión cifrada de claves privadas entre dos ordenadores como se describe en el PKCS#8.

4) PKCS#6 (Extended-Certificate Syntax Standard), ésta describe una sintaxis para certificados extendidos, esto es que se pueden extraer certificados X.509 de un superconjunto. Además se incluyen atributos como la dirección electrónica.

5) PKCS#7 (Cryptographic Message Syntax Standard) proporciona una sintaxis general para los datos que tengan una operación criptográfica asociada ya sea cifrado o firmado. La sintaxis es recursiva de tal modo que se puede anidar mensajes cifrados, también proporciona un método para distribuir certificados o listas de revocación de certificados, con esto se puede decir que el PKCS#7 es compatible con varias arquitecturas de gestión de claves basadas en certificados.

6) PKCS#8 (Private-Key Information Syntax Standard) ésta indica una sintaxis para la información de la clave privada, la que incluye una clave privada, una serie de atributos y una sintaxis para las claves que se utilizarán.

7) PKCS#9 (Selected Attribute Types) en éste se describe algunos atributos para el uso de los certificados extendidos, para los mensajes que son firmados digitalmente, para la información de la clave privada y para las peticiones de firmado de certificados.

8) PKCS#10 (Certification Request Syntax Standard) describe la sintaxis para las peticiones de certificados, ésta petición de certificado consiste en un nombre distinguido o distinguished name, una clave pública y otros atributos que son opcionales, todo esto firmado con la clave privada de la persona que hace la petición. Esta petición se envía a una Autoridad Certificadora, esta autoridad transforma la petición en un certificado X.509 v3 o en un certificado extendido.

9) PKCS#11(Cryptographic Token Interface Standard) especifica una interfaz de programación llamada Cryptoki para utilizarlo con dispositivos criptográficos de cualquier tipo. Cryptoki tiene un enfoque basado en objetos lo que hace que las aplicaciones realicen operaciones criptográficas sin saber la tecnología de los dispositivos.

10) PKCS#12 (Personal Information Exchange Syntax Standard) se describe la sintaxis para almacenar en software las claves públicas del usuario, para proteger sus claves privadas, los certificados y cualquier tipo de información

relacionada con la criptografía. Su finalidad es la utilización de un único fichero de claves que se pueden ser accesibles desde cualquier aplicación.

11) PKCS#13 (Elliptic Curve Cryptography Standard) que describe un método de utilización de algoritmos de curva elíptica, la manera de generar y validar los parámetros, las claves, el procedimiento de firmado y cifrado, etc. Esta es muy similar a la PKCS#1

12) PKCS#15 (Smart Card File Format), surge como una necesidad de cubrir ciertos aspectos que no se contemplan en el PKCS#11. Trata de uniformizar la estructura de directorios y ficheros de las tarjetas inteligentes.

2.7.4. Certificados

En pocas palabras, los certificados de clave pública son estructuras de datos firmadas que se usan para asociar el nombre de una entidad (y otros atributos adicionales relacionados con la entidad) con la correspondiente clave pública.

A lo largo del tiempo se han estandarizado diferentes formatos de certificados de clave pública:

- Certificados de clave pública X.509
- Certificados SPKI (Simple Public Key Infrastructure)
- Certificados PGP (Pretty Good Privacy)
- Certificados CV (Card Verifiable)

Entre todos ellos, los certificados de clave pública X.509 son los que han sido más ampliamente adoptados. Estos certificados han ido evolucionando con el tiempo originando diferentes versiones. Los certificados de clave pública de la versión 1 son un subconjunto de los certificados de la versión 2 y éstos a su vez son un subconjunto de los de la versión 3. Debido a que la versión 3 incluye numerosas extensiones opcionales, estos certificados se han instanciado a su vez para adaptarse a aplicaciones específicas dando lugar, por ejemplo, a los certificados PKIX que se usan en Internet y a los certificados SET usados para pagos con tarjetas de crédito. La práctica totalidad de las PKIs usa certificados de clave pública X.509 v3 (versión 3) y es por ello que este trabajo se centra en dicho tipo de certificados.

Originalmente el documento X.509 especificaba los mecanismos de autenticación para el directorio X.500. El directorio X.500 requería mecanismos fuertes de autenticación para asegurar que sólo usuarios autorizados podían modificar o acceder a sus datos. La versión 1 de los certificados presentaba problemas relativos a la renovación de certificados de CAs y una notable inflexibilidad para soportar atributos adicionales, por lo que en 1993 apareció la versión 2 que agregaba dos campos al formato antiguo. Los intentos de usar este nuevo formato en una PKI para el Internet Privacy Enhanced Mail resultaron infructuosos y revelaron deficiencias. En respuesta a los nuevos requisitos, la ISO junto con IEC e ITU desarrollaron la versión 3 de los certificados X.509 en 1997. Esta versión incluye el poderoso mecanismo de las extensiones que, de modo flexible, permite incluir en los certificados información no soportada en los campos básicos. Las nuevas versiones del documento

X.509 simplemente han añadido nuevas extensiones y han especificado los Certificados de Atributos sobre los que se construye las Privilege Management Infrastructures (PMI), pero no han alterado las ideas de la versión 3. La última versión del documento X.509 es de Noviembre del 2008.

La IETF (Internet Engineering Task Force) ha adaptado los certificados X.509 v3 para su uso en Internet promoviendo el uso de algunas extensiones y desaconsejando el uso de otras, entre ellas las que dieron lugar a los certificados X.509 v2. La IETF ha generado las recomendaciones PKIX para adaptar una PKI al entorno de Internet. De hecho, aunque los estándares PKIX están dirigidos principalmente a la comunidad de Internet, algunas de sus recomendaciones pueden aplicarse al entorno de la empresa y mantener de este modo la consistencia.

2.7.4.1. Estructura y semántica de los certificados

A continuación se van a detallar los diferentes campos que constituyen los certificados X.509 v3.

- Version. Este campo indica la versión a la cual se ajusta el formato del certificado. Puede referirse a la versión 1 (valor 0), versión 2 (valor 1) o versión 3 (valor 2).
- Serial Number. Es un entero asignado por la CA al certificado en el momento de su emisión y debe ser único dentro del ámbito de cada CA.

- Signature Algorithm. Este campo identifica el algoritmo criptográfico usado por la autoridad emisora al firmar el certificado. Para ello especifica el OID (Object Identifier) del algoritmo y sus parámetros asociados. Uno de los más usados hasta la fecha es SHA-1 con cifrado RSA. Este campo aparece dos veces en el certificado (dentro y fuera de la porción firmada). Los valores de los dos campos deben coincidir.

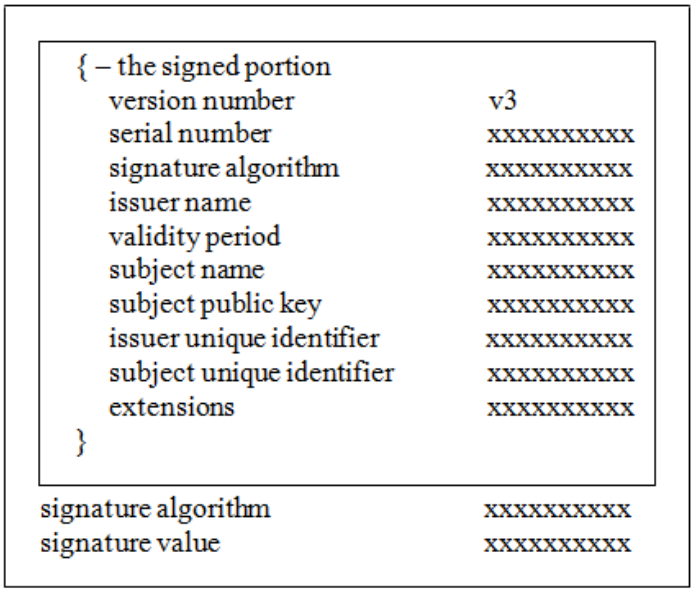

Figura II.9 Elementos de un certificado X.509 v3

- *Issuer Name*. Es el Nombre Distinguido (Distinguished Name, DN) de la CA que emitió el certificado y debe estar siempre presente. Este campo es común a todos los certificados emitidos por la misma CA. La combinación de Serial Number e Issuer Name identifican unívocamente un certificado.
- *Validity Period.* Indica la ventana de tiempo en la que el certificado es considerado válido salvo que haya sido revocado. Se representa como una secuencia de 2 fechas. La primera componente de la fecha marca el

inicio del periodo de validez, mientras que la segunda marca el final del periodo de validez.

- *Subject Name*. Indica el DN del propietario del certificado y puede ser nulo en el caso de que se use un formato alternativo de nombre en las extensiones.
- *Subject Public Key.* Este campo es una secuencia de 2 campos. Uno de ellos representa el valor real de la clave pública certificada, el otro es el OID del algoritmo asociado a la clave.
- *Issuer Unique Identifier.* Es un campo opcional que identifica universalmente la autoridad que firma el certificado. Se usa sólo en las versiones 2 y 3. Su uso no está recomendado.
- *Subject Unique Identifier.* Es un campo opcional que identifica el sujeto del certificado. Se usa sólo en las versiones 2 y 3. Su uso no está recomendado.

Las extensiones son un mecanismo disponible en los certificados de versión 3 que añade flexibilidad y que permite confeccionar diferentes mecanismos de seguridad y protocolos. Cada extensión del certificado se asocia con un flag de criticidad y, casi siempre, son los propios estándares los que imponen la criticidad de las diversas extensiones. Una extensión que se ha marcado como crítica debe ser procesada y comprendida cuando se valida el certificado, de lo contrario se descarta el certificado. Una extensión marcada como no crítica que no es comprendida cuando se valida el certificado se ignora y se procede como si no estuviera presente. En cambio, si se comprende la extensión no crítica

debe ser procesada del mismo modo que si se hubiera marcado como crítica. A continuación se pasa a describir las extensiones más importantes.

- *Authority Key Identifier.* Es un identificador único de la clave que se debe utilizar para verificar la firma digital del certificado. Distingue entre las múltiples claves del mismo emisor de certificados.
- *Subject Key Identifier.* Es un identificador único asociado con la clave pública contenida en el certificado. Distingue entre las múltiples claves del mismo propietario de certificados.
- *Key Usage.* Es una cadena de bits usada para identificar las funciones que soporta la clave pública del certificado. Por ejemplo: firmas digitales, cifrado de claves, cifrado de datos, firma de certificados, firma de CRLs.
- *Extended Key Usage.* Es una secuencia de OIDs que identifican usos específicos de la clave pública certificada. Por ejemplo: protección de e-mail, firma de código, autenticación de servidor TLS, autenticación cliente TLS.
- *CRL Distribution Point.* Indica la ubicación donde reside la CRL asociada al certificado.
- *Private Key Usage Period.* Indica la ventana de tiempo en la que la clave privada correspondiente a la clave pública del certificado puede ser usada. Está pensado para claves de firmas digitales. El uso juicioso de esta extensión puede proporcionar un margen de tiempo entre el instante en el que la clave privada expira y el instante en que el certificado expira. Esto ayuda a eliminar casos en los que firmas

digitales válidas son cuestionadas debido a que los periodos de vida de las claves son muy próximos o incluso idénticos.

- *Certificate Policies*. Indican una secuencia de uno o más OIDs de políticas asociadas con la emisión y uso subsiguiente del certificado. Si esta extensión se marca crítica, la aplicación debe adherirse al menos a una de las políticas indicadas o el certificado es descartado.
- *Policy Mappings.* Indica la equivalencia entre OIDs de políticas definidos en dos dominios de CAs. Sólo están presentes en certificados de CAs.
- *Subject Alternative Name*. Indica nombres alternativos del poseedor del certificado. Los formatos comúnmente usados de nombres alternativos son direcciones de e-mail, direcciones IP, nombres de servicios DNS.
- *Issuer Alternative Name.* Indica nombres alternativos asociados al emisor del certificado. Algunos formatos usados son, como en la extensión anterior, direcciones de e-mail o direcciones IP.
- *Basic Constraints.* Permite distinguir el certificado de una CA del de una entidad final. Debe aparecer en todos los certificados de CAs para afirmar que se trata de un certificado de CA. Adicionalmente esta extensión puede contener un campo opcional para indicar el máximo número de certificados de CAs que pueden seguir a este certificado en un camino de certificación.
- *Name Constraints.* Una extensión sólo presente en certificados de CA. El propósito de esta extensión es restringir el espacio de nombres de los

sujetos que emanan del certificado de CA y es aplicable tanto al campo Subject Name como a la extensión Subject Alternative Name.

- *Policy Constraints.* Esta extensión está sólo presente en certificados de CAs y puede usarse para prohibir Policy Mappings o para requerir que cada certificado en el camino de validación tenga unos OIDs de políticas determinados.
- *Inhibit Any Policy.* Esta extensión está sólo presente en certificados de CAs e indica que el OID correspondiente a Any Policy no debe ser considerado como un identificador de políticas.
- *Freshest CRL Pointer.* Proporciona un puntero a la información de CRL más reciente. En la práctica se trata de un puntero a una Delta CRL.

Según el estándar X.509 se pueden introducir extensions privadas para campos de aplicación específicos. El grupo de trabajo del PKIX ha introducido extensiones privadas para su uso en Internet. A continuación se detallan algunas de ellas.

- *Authority Information Access.* Especifica cómo se puede obtener información o servicios ofrecidos por el emisor del certificado. Se usa, por ejemplo, para servicios de validación on-line basados en OCSP.
- *Subject Information Access.* Especifica cómo se puede obtener información o servicios ofrecidos por el sujeto del certificado. Se usa, por ejemplo, para identificar la ubicación del repositorio donde la CA publica información de certificado y CRL.

Cualquier cambio de la información contenida en un certificado antes de que expire obliga a que el certificado sea revocado y se emita un nuevo certificado. Por tanto, debe procurarse que la información de los certificados sea estática para evitar una innecesaria revocación y remisión de certificados.

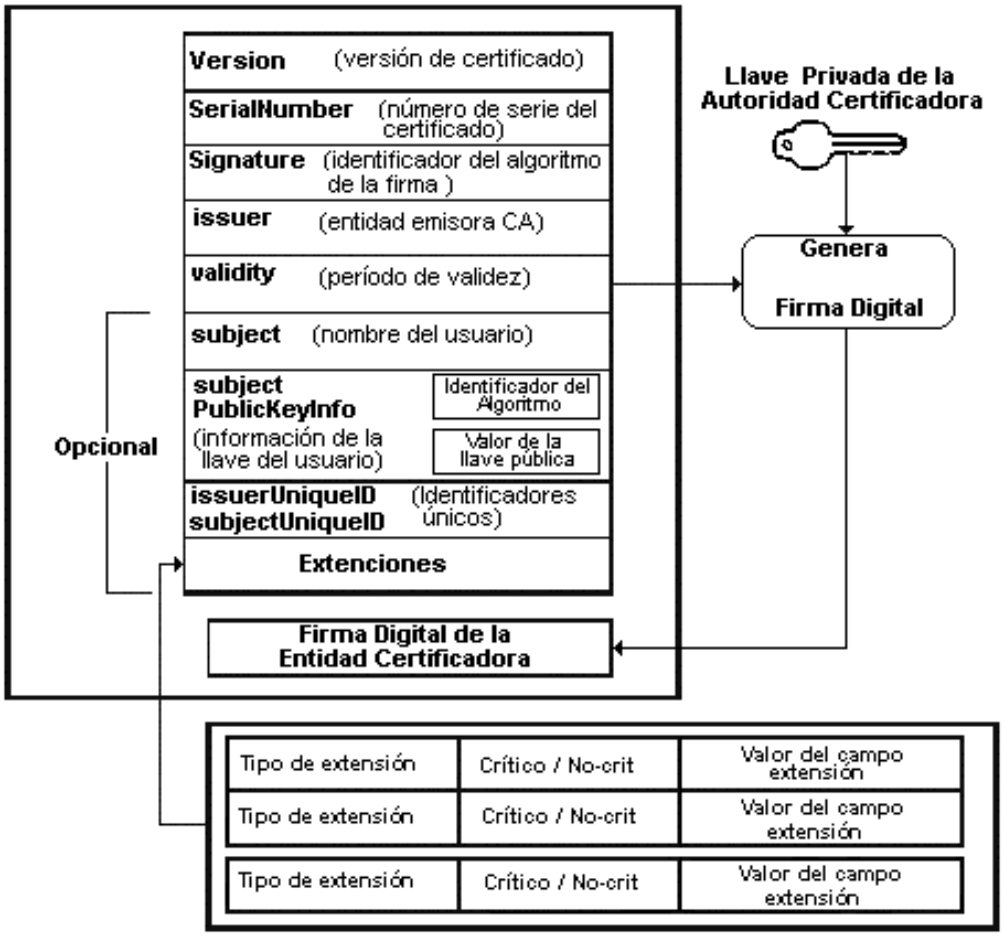

Figura II.10 Campos que componen el Certificado X.509 v3.

2.7.5. Revocación de certificados

En ocasiones puede ser necesario deshacer la asociación entre entidad y clave pública que establece un certificado antes de que expire. En tal caso se procede a revocar el certificado. Las circunstancias que obligan a revocar un certificado son muy variadas y, entre ellas, se pueden citar el compromiso de la clave privada, compromiso de la CA, el empleado abandona la compañía. Las formas más comunes de implementar la revocación de certificados son mecanismos de publicación periódicos, tales como Listas de Revocación de Certificados (Certificate Revocation Lists, CRLs), o mecanismos de consulta on-line tales como el Online Certificate Status Protocol (OCSP).

2.7.5.1. CRLs

En pocas palabras, las CRLs son estructuras de datos firmadas que contienen una lista de certificados revocados. La firma digital agregada en la CRL proporciona mecanismos de autenticidad e integridad. Siempre que las políticas lo permitan las CRLs pueden ser almacenadas en memoria y facilitar la verificación de certificados off-line. Normalmente el emisor del certificado y de la CRL son la misma autoridad, pero no siempre ocurre así.

Actualmente hay definidas 2 versiones de CRLs en el estándar X.509. La versión 1 presenta varios defectos: problemas de escalabilidad, posibilidad de ataques de sustitución que remplazan una CRL por otra sin ser detectado. La versión 2 corrige estos problemas mediante el mecanismo de las extensiones.

2.7.5.1.1. Estructura y semántica de las CRLs

A continuación se van a detallar los diferentes campos que constituyen las CRLs. La figura ll.11 representa el contenido de una CRL v2.

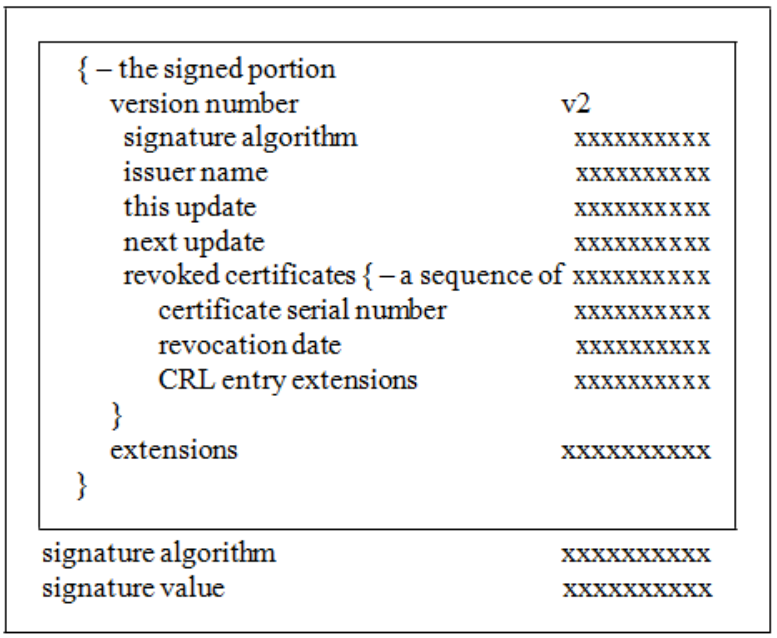

Figura ll.11 Elementos de una CRL X.509 v2

- *Version.* Indica la versión de la CRL. Si el campo no está presente indica que se trata de una CRL v1; si el campo está presente su valor debe ser el entero 1, indicando que se trata de una CRL v2.
- *Signature Algorithm.* Indica el OID del algoritmo usado para calcular la firma digital de la CRL. Debe coincidir con el campo Signature Algorithm perteneciente a la porción no firmada.
- *Issuer Name.* Se trata del DN del emisor de la CRL, es decir, quién firma la CRL. Debe estar siempre presente y ser único.
- *This Update.* Indica la fecha y hora en la que se emitió la CRL.

- *Next Update.* Campo opcional según X.509 que indica la fecha y hora en la que se emitirá la siguiente CRL.
- *Revoked Certificates.* Se trata de la lista de los certificados revocados. Cada entrada contiene el Serial Number del certificado revocado, la fecha y hora en la que se revocó el certificado y, opcionalmente, puede incluir extensiones aplicables a la entrada concreta de la lista.
- *Extensions*. Son las extensiones aplicables a la CRL globalmente.

El estándar X.509 define varias extensiones aplicables a una entrada de la lista. Estas extensiones permiten agregar información adicional a cada revocación y las más importantes son:

- *Reason Code.* Razón por la cual el certificado fue revocado (compromiso de la clave privada, compromiso de la CA, cambio de algún dato).
- *Certificate Issuer.* Es el nombre del emisor del certificado.
- *Hold Instruction.* Permite soportar la suspensión temporal de un certificado.
- *Invalidity Date.* Es la fecha y hora en la que el certificado deja de ser válido.

El estándar X.509 define varias extensiones aplicables globalmente a una CRL. A continuación se detallan algunas de ellas.

- *Authority Key Identifier.* Es el identificador único de la clave que debe usarse para verificar la firma digital. Distingue entre múltiples claves del mismo emisor de CRLs.

- *Issuer Alternative Name.* Contiene uno o más nombres alternativos del emisor de la CRL.
- *CRL Number.* Contiene un número de secuencia creciente para cada CRL emitida por un emisor de CRL. Permite detectar fácilmente cuando una CRL remplaza a otra.
- *CRL Scope.* Proporciona un método flexible para particionar CRLs. Las CRLs se pueden partir de muchas maneras (tipo de certificado, razón de revocación, números de serie)
- *Status Referral.* Esta extensión está presente en CRLs que no llevan información sobre certificados revocados. Simplemente transporta información para asegurar que se usa la información de revocación apropiada. Por un lado proporciona información dinámica sobre la partición de CRLs y, por otro lado, publica una lista de CRLs actuales que se utilizan para establecer si ya se dispone de dicha información.
- *CRL Stream Identifier*. Identifica el contexto en el cual el CRL Number es único.
- *Ordered List.* Indica si la lista de certificados revocados está en orden ascendente por Serial Number o fecha de revocación.
- *Delta Information.* Esta extensión proporciona la ubicación de la Delta CRL correspondiente a esta CRL.
- *Issuing Distribution Point.* Esta extensión identifica el CRL Distribution Point para esta CRL en concreto y el tipo de certificado contenido en la CRL (certificados de CA o certificados de entidad final).

- *Delta CRL Indicator.* Indica que esta CRL es una Delta CRL relativa a la CRL base referenciada.
- *Base Update.* Se usa en Delta CRLs para indicar la fecha y hora después de la cual la Delta CRL proporciona revocaciones.
- *Freshest CRL.* Identifica cómo obtener la Delta CRL más reciente para la CRL base que contiene esta extensión.

La mayoría de CAs emiten sus propias CRLs, es decir, la CA emite los certificados y las CRLs. Periódicamente la CA emite un única CRL que cubre toda su población de certificados. Tales CRLs se denominan CRLs completas. Sin embargo, el uso de las CRLs completas tiene importantes limitaciones: el tamaño de las CRLs puede crecer mucho, la periodicidad de publicación de las CRLs debe ser grande para evitar la degradación de los recursos de red. Esto obliga a buscar soluciones alternativas.

2.7.6. Componentes de una PKI

Las funcionalidades deben estar presentes en cualquier PKI, pero las implementaciones específicas pueden agruparlas de forma diferente. Por ejemplo, la Certification Authority (Autoridad de Certificación, CA) y la Registration Authority (Autoridad de Registro, RA) en ocasiones se combinan en un único componente.

La figura Ⅱ. presenta un modelo simplificado de los componentes de una PKI basado en el que se describe en el RFC5280 del PKIX del IETF:

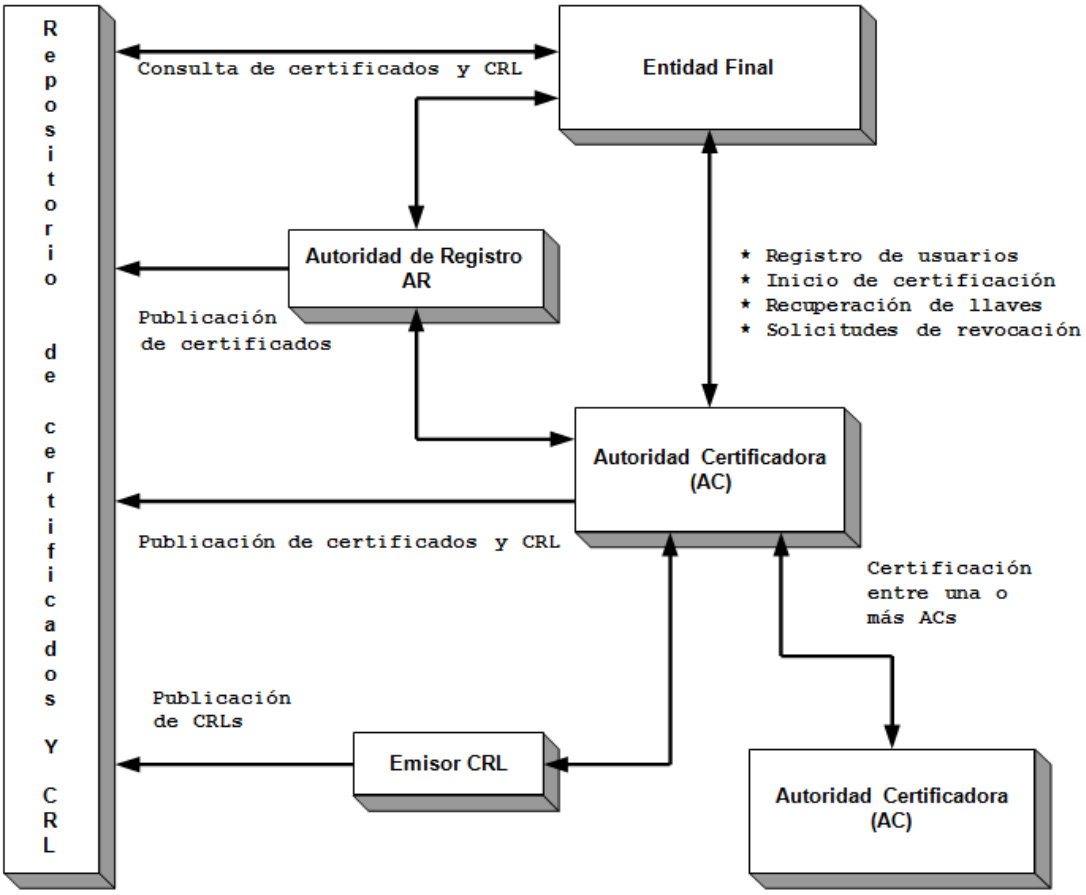

Figura ll. Esquema simplificado PKI

A continuación se explica el esquema. Los usuarios finales (end entities), usando transacciones de gestión, envían su petición de certificado a la RA para su aprobación. Si la petición es aprobada, se pasa a la CA para ser firmada. La CA revisa la petición de certificado y, si supera la revisión, se firma la petición y se genera el certificado. Para publicar el certificado, la CA lo envía al repositorio de certificados. El diagrama también muestra que los usuarios finales pueden comunicarse directamente con la CA de manera que toda la funcionalidad está implementada en la CA. Del mismo modo, el

diagrama muestra que CA y RA envían los certificados al repositorio. La implementación debe escoger una de las alternativas.

La revocación de certificados sigue un curso similar al de la generación. Los usuarios finales piden a la RA que revoque su certificado, la RA decide y renvía la solicitud a la CA. La CA actualiza la Certificate Revocation List (Lista de Certificados Revocados, CRL) y la publica en el repositorio de la CRL.

Finalmente, los usuarios finales pueden verificar la validez de un certificado específico usando un protocolo operativo.

2.7.6.1. Certification Authority

La CA es el componente esencial de una PKI. Una CA es un conjunto de hardware, software y el personal que los opera. Una CA realiza cuatro funciones básicas:

- Emisión de certificados, crea los certificados y los firma.
- Mantener información del estado de los certificados y emitir CRLs.
- Publicar los certificados y CRLs de manera que los usuarios puedan obtener la información que necesitan para implementar los servicios de seguridad.
- Mantener archivos sobre la información de estado de los certificados expirados o revocados que emitió.

Emisión de certificado

Una CA puede emitir certificados a usuarios, otras CAs o ambos. Cuando una CA emite un certificado está afirmando que el sujeto (entidad nombrada en el certificado) tiene la clave privada que corresponde a la clave pública del certificado. Si la CA incluye información adicional en el certificado, la CA está dando por cierto que dicha información corresponde al sujeto. Esta información adicional podría ser información de contacto (por ejemplo, e-mail) o información de política (por ejemplo, los tipos de aplicaciones en los que se puede usar la clave pública). Cuando el sujeto del certificado es otra CA, la CA emisora está afirmando que se puede confiar en los certificados emitidos por la CA sujeto.

La primera responsabilidad de una CA es proteger su clave privada. Si un atacante consiguiera la clave privada de una CA podría suplantarla y emitir certificados como si fuera la misma CA.

La segunda responsabilidad de una CA es verificar que la información del certificado es cierta.

La tercera responsabilidad de una CA es asegurar que todos los certificados y CRLs que emite cumplen con un perfil. Por otro lado, la CA también debe proteger la integridad del perfil y restringir su acceso. La restricción del acceso puede ser física (acceso a través de tarjeta al recinto), lógica (firewall) o procedimental (se requieren dos miembros del personal que atiende la CA para modificar el sistema).

Mantenimiento de la información de estado y emisión CRLs

La cuarta responsabilidad de una CA es mantener una lista de certificados en los que no se debe confiar. Proteger esta información es similar a proteger el perfil, mientras que modificar el estado de un certificado depende de información proporcionada desde el exterior de la CA.

Publicación de certificados y CRLs

La quinta responsabilidad de una CA es distribuir sus certificados y CRLs. En general, la distribución de los certificados es más un problema de rendimiento y disponibilidad que de seguridad; no se requiere restringir el acceso a certificados y CRLs puesto que tal información no es secreta. Sin embargo, en ocasiones, la CA puede querer denegar el acceso a los certificados a personas ajenas a la organización por simples motivos de seguridad.

Mantenimiento de archivos

La sexta responsabilidad de una CA es el mantenimiento de suficiente información de archivo para establecer la validez de certificados después de que hayan expirado. La principal dificultad de esta función es que la información debe mantenerse durante largos periodos de tiempo.

Delegación de responsabilidad

Es difícil diseñar un sistema que satisfaga simultáneamente todos los requisitos. Una CA suele cumplir los más prioritarios y delegar el resto. La responsabilidad principal de una CA es proteger la clave o claves privadas usadas para firmar certificados y CRLs. Para satisfacer este requisito la CA debe construir un perímetro con controles de seguridad física, tecnológica y

procedimental. Este perímetro permite también alcanzar los requisitos tercero y cuarto.

Este perímetro de seguridad impide el cumplimiento de las restantes responsabilidades. Los tres componentes restantes de la infraestructura se diseñan para aceptar esas responsabilidades en lugar de la CA. Una entidad que verifica el contenido de los certificados se denomina Registration Authority (RA).

2.7.6.2. Registration Authority

El propósito de una RA es verificar el contenido de los certificados en lugar de la CA. De igual modo que una CA, una RA es una colección de hardware, software y las personas que lo operan. Cada CA mantiene una lista de RAs acreditadas y verificando la firma de una RA en un mensaje una CA puede estar segura de que una RA acreditada proporcionó la información y, por tanto, es fiable. Por consiguiente, es importante que una RA proporcione protección adecuada para su clave privada.

Hay dos modelos básicos para que una RA verifique el contenido de un certificado. En el primer modelo la RA recoge y verifica la información antes de presentar a la CA la solicitud para el certificado. En el segundo modelo, la CA recibe una solicitud de certificado que envía a la RA. La RA revisa el contenido y determina si la información es correcta. La RA responde a la petición de la CA con un simple ’Sí’ o ’No’.

2.7.6.3. Repositorio

Un repositorio acepta certificados y CRLs de una o más CAs y los hace disponibles a las partes que necesitan implementar servicios de seguridad bajo petición. Los repositorios se diseñan para proporcionar máxima disponibilidad y rendimiento, ya que los mismos datos establecen su integridad. Los repositorios necesitan restringir el conjunto de usuarios que pueden actualizar la información, ya que, de lo contrario, un atacante podría sustituir los certificados con basura y provocar un ataque de denegación de servicio.

2.7.6.4. Archivo

Un archivo asume la responsabilidad del almacenamiento a largo plazo de información en lugar de la propia CA. Un archivo declara que la información era correcta en el momento en que se recibió y que no ha sido modificada. El archivo protege la información a través de mecanismos técnicos y procedimientos. Si se suscita una disputa, la información del archivo puede usarse para verificar firmas de documentos viejos en fechas posteriores.

2.7.6.5. Usuarios

Hay dos tipos de usuarios soportados por una PKI. Los poseedores del certificado son los sujetos del certificado y mantienen la clave privada. Y las

partes confiantes usan la clave pública de un certificado para verificar la firma o cifrar datos. En la práctica, la mayoría de entidades soportan ambos papeles. Del mismo modo, CAs y RAs son también usuarios ya que generan y verifican firmas y transmiten claves entre ellas mismas o con los propios usuarios.

Poseedores de certificados

Los poseedores de certificados obtienen certificados de la infraestructura y usan sus claves privadas para implementar servicios de seguridad. Generan firmas digitales, descifran datos (por ejemplo claves simétricas) y usan sus claves privadas para establecer claves simétricas a través de protocolos de acuerdo de claves. Para cumplir estos objetivos el poseedor de un certificado debe realizar las siguientes acciones:

- Identificar la CA que emite los certificados.
- Solicitar el certificado directamente o a través de una RA.
- Incluir el certificado en las transacciones que así lo requieran.

En ocasiones los poseedores de un certificado necesitarán interactuar con el repositorio para obtener su certificado, aunque no de una manera regular.

Partes confiantes

Las partes confiantes usan la PKI para implementar servicios de seguridad utilizando la clave pública en el certificado. Pueden verificar firmas digitales, cifrar datos (claves simétricas) y usar la clave pública para establecer claves simétricas a través de protocolos de acuerdo de claves. Para implementar estos

servicios de seguridad, una parte confiante debe realizar las siguientes acciones:

- Identificar una CA como su punto inicial de confianza.
- Verificar firmas de certificados y CRLs.
- Obtener certificados y CRLs del repositorio.
- Construir y validar caminos de certificación.

Una parte confiante interactúa con el repositorio regularmente. Sus interacciones con las CAs se limitan a la selección de los puntos de confianza y no interaccionan con las RAs.

2.7.7. Arquitecturas PKI

La arquitectura de una PKI describe la organización de sus CAs y sus relaciones de confianza. Cada arquitectura tiene sus ventajas y sus inconvenientes y es apropiada para algunos entornos, mientras que para otros no lo es. Las cuatro primeras arquitecturas son aplicables a una organización, mientras que las tres últimas se refieren a las arquitecturas de la interconexión de organizaciones.

2.7.7.1. CA única

La arquitectura PKI más básica es la formada por una CA única que proporciona todos los certificados y CRLs para una comunidad de usuarios. Todos los usuarios confían en la CA que emitió su propio certificado. Por

definición, no pueden añadirse nuevas CAs a la PKI y puesto que sólo hay una única CA no se establecen relaciones de confianza con otras CA. Es la arquitectura más simple de implementar. Los caminos de certificación constan de un único certificado y hay una única CRL. Por contra, esta arquitectura no es escalable y presenta un único punto de fallo. Si se compromete la CA se invalidan todos los certificados emitidos. Cada usuario debe ser informado inmediatamente. Para restablecer la confianza se debe volver a emitir todos los certificados y la información sobre el nuevo punto de confianza debe ser distribuida a todos los usuarios. Esta arquitectura sólo es aplicable a una empresa que no necesita comunicarse con el mundo exterior. La figura II.12 muestra una CA única.

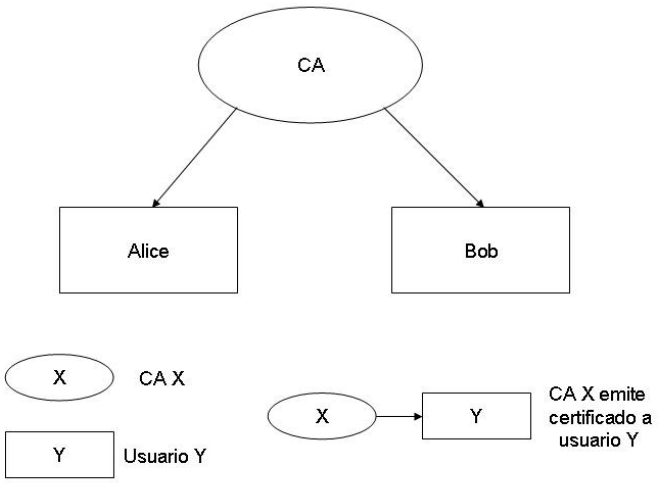

Figura II.12 CA única

2.7.7.2. Listas de confianza simple

La lista de confianza es la forma más simple de soportar más de una CA. En esta arquitectura hay más de una CA pero no hay relaciones de confianza entre

ellas. En este modelo cada usuario mantiene una lista de las CAs en las que confía. Se pueden añadir nuevas CAs a la PKI modificando las listas de los usuarios. Los usuarios aceptan certificados y CRLs emitidos por una CA en su lista de confianza. Un usuario requiere un certificado y una CRL. La construcción y validación de caminos de certificados es muy simple.

La principal ventaja de esta arquitectura es su simplicidad. Los caminos de certificados constan de un único certificado y es fácil añadir una nueva CA a la PKI. Sin embargo, hay importantes desventajas. La nueva CA que se desea añadir a la lista debe investigarse previamente. Si se compromete una CA, probablemente se informará rápidamente a sus propios usuarios, pero no a los usuarios de otras CAs ya que la CA comprometida no tiene manera de saber qué usuarios la tienen en su lista. Un mecanismo muy similar es usado en los navegadores Web. El vendedor del navegador lo pre configura con un conjunto extenso de certificados CA raíz conocidos. Esta característica facilita el uso del navegador, ya que la mayoría de certificados de servidores Web son emitidos por CAs bien conocidas y, por tanto, son automáticamente reconocidos. Muchos navegadores no se han diseñado para verificar la validez de los certificados, como consecuencia una CA comprometida puede seguir usándose por muchos usuarios antes de que la lista sea actualizada o, todavía peor, se distribuya una nueva versión de la aplicación. La figura ll.13 muestra la arquitectura de una lista de confianza simple.

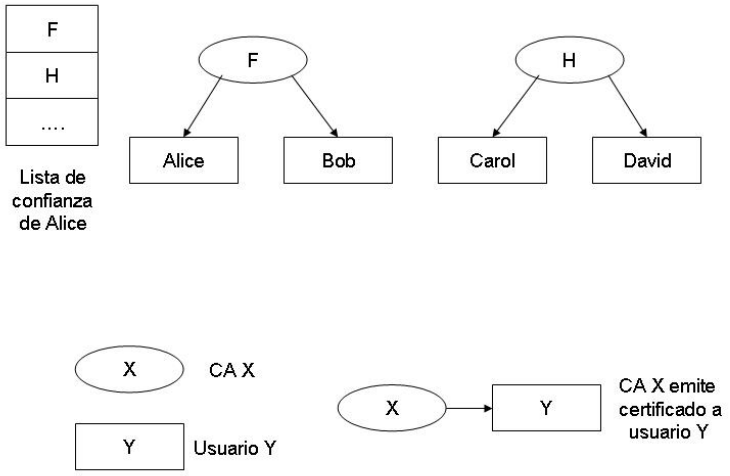

Figura ll.13 Lista de confianza simple

2.7.7.3. Jerárquica

La arquitectura jerárquica es la más tradicional. En esta arquitectura varias CAs, con una relación superior-subordinado, proporcionan servicios a la PKI. En esta arquitectura todos los usuarios confían en la CA raíz. Con la excepción de la CA raíz, todas las CAs tienen una única CA superior. Una CA puede emitir certificados a CAs, usuarios o ambos. Cada relación de confianza entre CAs se representa por un único certificado. El emisor es la CA superior y el sujeto es la CA subordinada. Para añadir una nueva CA a la PKI, una CA existente emite un certificado a la nueva CA. La nueva CA se injerta bajo la CA existente y se convierte en una CA subordinada de la CA emisora. Dos PKI jerárquicas pueden fusionarse de la misma manera. Las CAs superiores pueden imponer restricciones a las CA subordinadas. Estas restricciones pueden implementarse con procedimientos o en los propios certificados.

La arquitectura jerárquica proporciona varias ventajas que contribuyen a hacer de ella uno de los modelos más ampliamente desplegados hasta la fecha. En primer lugar, los caminos de certificados son relativamente cortos. Además, puesto que todos los usuarios confían en la misma CA raíz, hay un único camino para alcanzar un usuario específico. Esto permite a la entidad final (usuario) distribuir los certificados de cualquier CA intermedia en la cadena junto con su propio certificado y dar el camino al usuario del certificado. La desventaja más significativa de este modelo radica en la misma razón que su simplicidad y éxito: la existencia de una CA raíz en la que todos confían. En una comunidad pequeña es posible acordar una única CA raíz, pero en comunidades grandes es imposible que todos acuerden una única CA raíz.

Si se compromete una CA (diferente de la raíz), su CA superior simplemente revoca su certificado. Una vez se ha restablecido, la CA emite certificados a todos sus usuarios. La CA superior emite un nuevo certificado con la comparación con el compromiso de la CA única y es que la CA raíz debe emitir un número mucho menor de certificados por lo que puede operar mucho tiempo offline, reduciendo la posibilidad de un compromiso.

El modelo jerárquico funciona razonablemente bien dentro de los confines de una empresa, particularmente si la empresa tiene una estructura organizativa fuertemente jerárquica. El modelo tiende a dejar de funcionar cuando se atraviesan los confines de la organización. Las razones son usualmente la falta de acuerdo sobre una CA raíz única y las diferentes políticas operativas instituidas en las diferentes organizaciones.

La figura ll.14 es una representación gráfica de una PKI jerárquica.

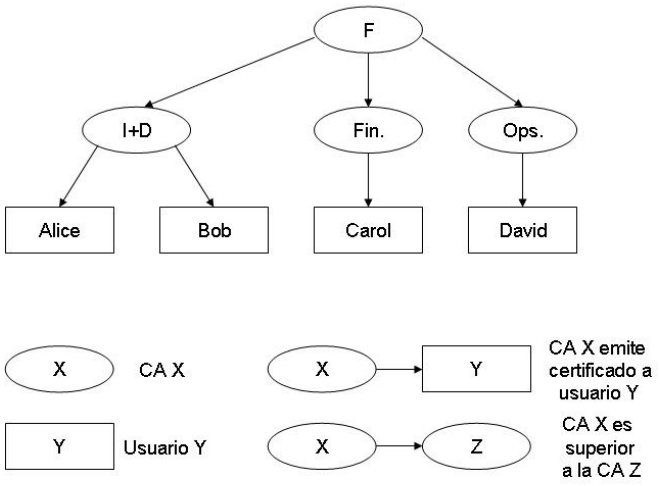

Figura ll.14 PKI jerárquica

2.7.7.4. Malla

La arquitectura en malla es la principal alternativa a la PKI jerárquica. En este modelo varias CAs proporcionan los servicios PKI, pero la relación entre ellas es de igual a igual y no jerárquica. Cada usuario confía en una única CA; sin embargo, no es la misma CA para todos los usuarios. En general, los usuarios confiarán en la CA que emitió su certificado. Las CAs emiten certificados entre ellas, un par de certificados describe una relación de confianza bidireccional. Una nueva CA se añade a la malla simplemente intercambiando certificados con otra CA que ya es miembro de la malla.

La construcción de caminos es particularmente complicada en una malla. El proceso para construir caminos en esta arquitectura no es determinista, conlleva

múltiples elecciones (algunas elecciones llevan a un camino válido y otras no), puede producir lazos y, en general, los caminos son más largos que en una clave pública de la nueva CA insertándola de nuevo en la jerarquía. Durante el periodo de restablecimiento, dos usuarios que no pertenecen a la parte comprometida de la jerarquía pueden seguir operando. El compromiso de la CA raíz tiene el mismo impacto que en la arquitectura de CA única. Hay que informar a todos los usuarios del compromiso, restablecer la CA, emitir todos los certificados y distribuir el nuevo punto de confianza. Sin embargo, todavía hay una ventaja en PKI jerárquica. Los problemas que aparecen al determinar un camino son similares a los problemas de encaminamiento en una red de routers de Internet al enviar un paquete entre terminales de la red. El aumento en la flexibilidad que se deriva de usar una malla tiene su contrapartida en el aumento de la complejidad para formar caminos. Otro inconveniente de las mallas es que una CA puede incluir en la malla otra CA que sea un competidor mío y con el que no me interese tener ninguna relación de confianza. Esto refuerza la necesidad de mecanismos para controlar este tipo de situaciones. La localización de certificados en las mallas es otra cuestión importante. A diferencia de la arquitectura jerárquica, no es posible predeterminar los caminos asociados con cada entidad final. La construcción de caminos en una malla depende fuertemente de la existencia y fácil acceso a directorios para localizar certificados.

Las arquitecturas en malla son muy flexibles. El compromiso de una CA no provoca la completa inutilización de la PKI. Las CAs que emitieron certificados a la CA comprometida simplemente los revocan, lo que conlleva la eliminación

de la CA en la PKI. En el mejor de los casos, la PKI se reduce en una CA y sus usuarios. En el peor de los casos la PKI se fragmenta en PKIs más pequeñas.

La figura ll.15 representa gráficamente una arquitectura en malla.

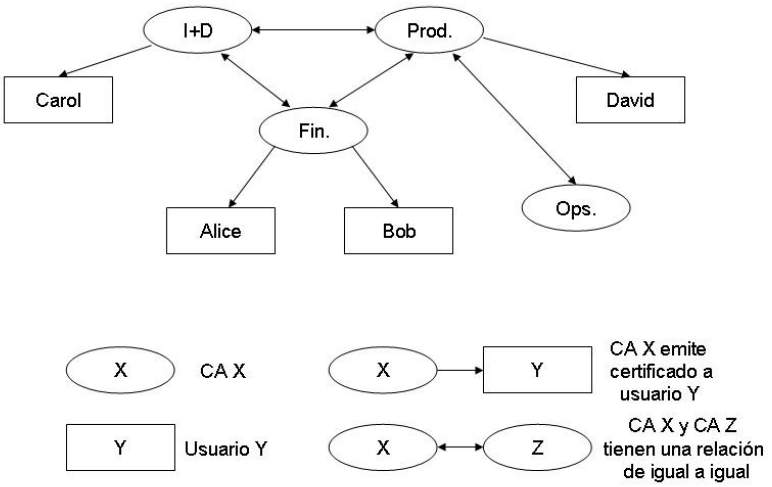

Figura ll.15. PKI en malla

2.7.7.5. Lista de confianza extendida

Esta arquitectura, a diferencia de las anteriores, permite extender la PKI a varias empresas u organizaciones. La arquitectura lista de confianza extendida corrige los defectos de la lista de confianza simple. Cada usuario mantiene una lista de puntos de confianza. Cada punto de confianza identifica una PKI en la que el usuario confía y cuya arquitectura puede ser CA única, jerárquica o malla. En esta arquitectura, el usuario añade una CA por cada PKI en la que confía.

Esta arquitectura conserva la ventaja esencial de la lista de confianza simple la facilidad y rapidez con la que un usuario puede confiar en otras PKIs. Además,

también cuenta con la principal ventaja de las PKI en malla y jerárquica que consiste en que al confiar en una CA extiende su confianza a otras CAs relacionadas con la anterior, con lo que se reduce el número de puntos de confianza que se debe mantener en la lista. No obstante, los problemas relacionados con el mantenimiento de listas extensas de puntos de confianza y con el compromiso de CAs persisten. Esta arquitectura también introduce sus propios inconvenientes: la construcción de caminos de certificados es más compleja ya que el usuario no sabe cuál de las CAs en las que confía le llevará al certificado del usuario. La construcción de caminos suele hacerse partiendo del certificado del usuario hasta llegar a una de las CAs en las que confía.

La figura ll.16 muestra una lista de confianza extendida.

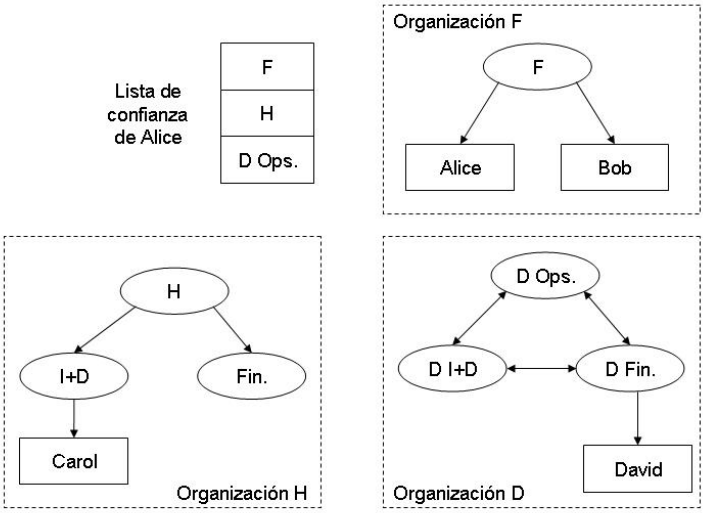

Figura ll.16 PKI Lista de confianza extendida

2.7.8. Beneficios de una PKI

Los beneficios que una PKI ofrece se derivan del propio concepto de PKI y de los servicios que ofrece. Una PKI es una solución global de seguridad, no un conjunto de soluciones puntuales diferentes, y ofrece una única infraestructura de seguridad que puede ser usada por muchas aplicaciones en los entornos más heterogéneos. Específicamente, ofrece servicios de confidencialidad, integridad, autenticación y no repudio en numerosos contextos.

He aquí una lista de los beneficios que ofrece una PKI:

- Gestión de passwords y Single-Sign-On. Hay muchos problemas ocasionados por el sistema tradicional de usernames y passwords. Una PKI resuelve estos problemas de una manera consistente y sencilla para los usuarios y administradores.
- Firmas digitales. Una PKI permite firmar digitalmente documentos. Las firmas tienen un reconocimiento legal. En conjunto se consigue sustituir el papel con formularios electrónicos, más velocidad y trazabilidad en los procesos de negocios y una seguridad mejorada en las transacciones electrónicas.
- Cifrado. Fácil cifrado de datos para cada individuo (sin intercambio previo de información) mediante el acceso al certificado que contiene la clave pública.
- Comodidad del usuario. Menos passwords. Mecanismo consistente de autenticación que basta con aprender una vez. Procedimientos sencillos para cifrar, firmar y autenticar.

- Administración coherente de seguridad en la empresa. Emisión y revocación centralizada de credenciales de usuario. Identificación consistente de usuario cuando se emiten las credenciales. Idéntico mecanismo de autenticación para todas las aplicaciones o servicios de red. Aprovechamiento de la inversión en smart cards o tokens USB al ser usados por muchas aplicaciones.
- Interoperabilidad con otras instituciones. La confianza entre organizaciones y/o empresas permite firmar y cifrar correos, firmar documentos, autenticación en aplicaciones compartidas.
- Solución basada en estándares. Los estándares proporcionan interoperabilidad entre fabricantes diferentes. Muchas implementaciones disponibles. Los estándares permiten que haya código libre.
- Amplio soporte. En sistemas operativos: Windows, Linux, UNIX, Almacenamiento de claves en software y hardware. En aplicaciones:
- Apache, IIS, Oracle, SSL, Código abierto y comercial.

Entornos variados. Las PKI son soluciones ampliamente aceptadas en la industria, organizaciones gubernamentales.

Estas ventajas técnicas se traducen en ventajas de negocio, aunque su cuantificación concreta es difícil:

- Mejoras en la eficiencia del workflow. Se pueden conseguir ahorros significativos de tiempo mediante el manejo electrónico de documentos.

- Optimización de los recursos humanos. El usuario puede centrarse en su trabajo en lugar de gastar tiempo en detalles asociados con la infraestructura de seguridad.
- Reducción de los recursos humanos. La operación de una arquitectura unificada en lugar de múltiples soluciones puntuales requiere menos recursos administrativos.
- Reducción del papel. Se puede ahorrar en costes de material, en el espacio necesario para almacenarlo, en reducción de residuos y en menor intrusión ambiental.
- Menos carga administrativa. Los usuarios finales requieren menos asistencia (help-desk).
- Reducción de pérdidas por robo electrónico. Los datos corporativos están protegidos, lo que reduce el riesgo de que sean revelados sin autorización.
- Ahorros en telecomunicaciones. La capacidad de crear redes privadas virtuales (Virtual Private Networks, VPN) sobre una red pública como Internet es más barata que alquilar líneas privadas.
- Generación de ingresos. Una PKI puede usarse para generar ingresos. Por ejemplo, una organización financiera puede ofrecer servicios de validación de transacciones basados en firmas digitales y certificados.

En cualquier caso, el robo y fraude electrónico van en aumento y deben buscarse soluciones. Las empresas perciben la seguridad como algo necesario y, por tanto, le dedican cierta atención. Cualquier solución global de seguridad

debe contemplar los recursos corporativos y las comunicaciones, tanto externas como internas. El valor de una PKI que protege y permite recuperar información crítica es muy elevado, incluso si sus beneficios cuantitativos son casi imposibles de medir con precisión.

Una PKI es una infraestructura y, como tal, sólo produce beneficios cuando es utilizada. De hecho, son las aplicaciones quienes se conectan a la infraestructura, la utilizan y provocan beneficios. Los tipos de certificados que serán necesarios y las entidades a las cuales se emitirán (usuarios, máquinas) dependen de las aplicaciones que vaya a soportar la PKI. Es importante conocer las principales aplicaciones que pueden aprovechar la PKI y proporcionar una seguridad fuerte ya que la organización o empresa querrá usar algunas de ellas. Algunas aplicaciones están inmersas en el sistema operativo y otras no. He aquí algunas de las principales aplicaciones:

- Web segura. En una intranet corporativa o en una extranet se pueden usar certificados para proporcionar seguridad fuerte mediante los protocolos SSL y TLS. Ambos protocolos proporcionan autenticación de cliente, autenticación de servidor y confidencialidad de datos.
- Correo seguro. El protocolo Secure/Multipurpose Internet Mail Extensions (S/MIME) también está basado en criptografía de clave pública y certificados. Este protocolo permite firmar y cifrar mensajes. Muchas aplicaciones de e-mail proporcionan un sistema dual de claves para firmar y cifrar.

- Cifrado del sistema de ficheros. Proporciona cifrado a nivel del sistema de ficheros. Como medida de seguridad, conviene que permita la recuperación de los datos por una persona adicional.
- Firma de código. Protege contra descargas de código alterado (hackers) de websites.
- Smart card logon. Proporciona autenticación fuerte de 2 factores (posesión de la smart card y conocimiento del PIN (Personal Identification Number)). A diferencia de las passwords, los PINs no se envían a través de la red, lo cual es una medida adicional de seguridad.
- Virtual Private Network. IPSec es un protocolo que funciona a nivel IP con certificados y se usa para autenticar los extremos de la comunicación.
- Cualquier aplicación. Los fabricantes proporcionan APIs con las que adaptar cualquier aplicación para que use la PKI.

CAPÍTULO III

IMPLEMENTACIÓN

3.1. Software a usar

Linux es un sistema de libre distribución al encontrarse los ficheros y programas necesarios para su funcionamiento en servidores conectados a Internet. La tarea de reunir todos los ficheros y programas necesarios, configurarlos e instalarlos, puede ser complicada, por esto se dieron origen las distribuciones de Linux.

Una distribución de GNU/Linux es una variante de ese sistema operativo que incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando origen a ediciones hogareñas, empresariales y para servidores. Pueden ser exclusivamente de software libre, o también incorporar aplicaciones o controladores propietarios.

Para el desarrollo se decidido utilizar la distribución Centos

3.1.1. CentOS

Red Hat CentOS o Comunidad empresarial del sistema operativo, está constituido por una distribución binaria de Red Hat Enterprise (RHEL), la misma que nace a partir de la comunidad de Linux, realizando investigaciones y compilando código fuente liberado por RHEL. Este Sistema Operativo (SO) se encuentra principalmente basado en una herramienta de administración de paquetes (RPM-Ret Hat Package Manager), pero no es asistido ni mantenido por RHEL.

La principal ventaja de este sistema es que los usuarios como organizaciones o personas no necesitan un soporte comercial demasiado fuerte para lograr sus objetivos debido a que CentOS tiene la misma calidad que el mejor GNU/Linux o que le hace 100% compatible con RHEL, pero con la diferencia que no se necesita de una licencia empresarial. Adicionalmente a diferencia de RHEL posee una herramienta de software libre de gestión de paquetes para sistemas Linux (YUM – Yellow dog Updater, Modified) que permite instalar y actualizar los paquetes sin ningún inconveniente, incluso estando de acuerdo con los requerimientos de redistribución RHEL, pudiendo afirmar que está hecho para la gente que necesita un SO de calidad empresarial y alta estabilidad.

3.2. Diagrama de la Red

3.2.1. Diagrama Físico

La red ejemplo esta compuesta de un servidor encargado de brindar la mayoría de los servicios dentro de la intranet, el servidor y dos usuarios, uno dentro de Linux y el otro en Windows.

Para la implementación se va hacer uso de maquinas virtuales dentro del entorno VMware

La maquina que representa al servidor va a brindar los servicios de DNS, DHCP, WEB, MAIL y administrar la PKI

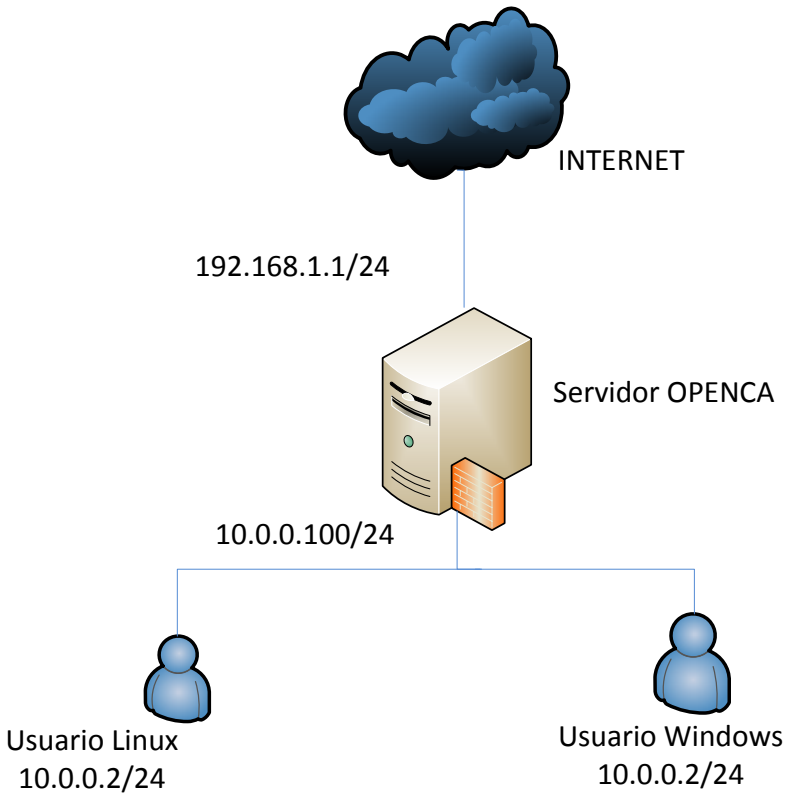

Figura lll.17 Diagrama físico de la red

3.2.2. Direccionamiento.

Para el direccionamiento se decidió usar la red 10.0.0.0 con una mascara de 255.255.255.0, dentro de este rango se van a encontrar todos los usuarios pertenecientes a la intranet, y para simular el internet se decidió usar una dirección 192.168.1.1 con mascara 255.255.255.0

Equipo	Dirección IP	Mascara	Gateway
Servidor OpenCA Intranet Internet	 10.0.0.100 192.168.1.1	 255.255.255.0 255.255.255.0	
Usuario Linux	10.0.0.2	255.255.255.0	10.0.0.100
Usuario Windows	10.0.0.3	255.255.255.0	10.0.0.100

Tabla lll.2 Direccionamiento IP

3.3. Diseño de la PKI

La PKI se encarga de emitir certificados digitales para: usuarios, hosts y servicios. Estableciendo una correspondencia entre la identidad del dueño del certificado con su clave pública.

La elección entre una arquitectura plana, jerárquica o malla, viene dada por las características de la comunidad donde será diseñada una PKI, así también los requerimientos que se hayan impuesto dentro de la organización donde se implementará.

Para el diseño se consideró:

3.3.1. Requerimientos iníciales.

Analizando las necesidades que posee la empresa ficticia y considerando que los servicios de correo electrónico y web se encuentran implementados de forma segura, con el uso de certificados digitales, permitiendo acceder a la información almacenada en el interior de la empresa.

El servicio de navegación squid habilita el acceso al Internet y a los servicios web

3.3.2. Servicios que brinda la empresa.

Manejándose un número reducido de usuarios y todas las aplicaciones se encuentran centralizadas, a través de una solución Open Source

La empresa administra los usuarios para que cumplan con las diferentes políticas internas y puedan manejar los varios servicios que brinda como: DNS, DHCP, correo electrónico, squid y web.

3.3.3. Herramientas de administración de la PKI.

Para la administración de los certificados digitales a través de la PKI, se utilizará la herramienta OpenCA.

OpenCA posee una interfaz web manejable para el usuario, que cumple todas las características para el correcto desempeño de la PKI, integrando su funcionamiento con OpenSSL.

3.3.4. Descripción de AC.

De acuerdo a los requerimientos de la empresa ficticia se estableció que el modelo para la implementación de la PKI es la arquitectura Plana, ya que maneja una sola AC

Este modelo permite manejar empresas pequeñas sin ningún inconveniente, generando autenticación a través de certificados digitales en un solo sentido, brindando confiabilidad a los diferentes servicios de red.

Esta arquitectura permite manejar políticas para crear, gestionar y revocar certificados digitales teniendo una idea básica para el manejo de los datos sensibles, los cuales deben ser protegidos mediante técnicas de encriptación.

La PKI de igual manera trabaja con el estándar X.509 para la implementación de los Certificados Digitales a través de una sola AC y dentro de una misma red, permitiendo el establecimiento seguro de la comunicación y el intercambio de información entre los usuarios que pertenecen al dominio de la empresa.

3.3.5. Interrelación con otras PKI.

De acuerdo al modelo seleccionado, es decir, arquitectura plana, la PKI posee una única AC que es el centro de confianza de las entidades, lo que no permite interrelacionarse con otras ACs.

3.4. Implementación de los servidores.

3.4.1. Configuración IPs

Primero vamos a determinar el actual hostname del sistema, e ingresamos el correcto, en este caso el dominio que vamos a utilizar será “tesis.com”

```
#hostname
#hostname server.tesis.com (servidor)
```

Ahora vamos a ingresar al archivo de configuración

```
#vi /etc/sysconfig/network
```

Y aquí vamos a cambiar el nombre de su hostname y a cambiarlo según corresponda.

Dentro del servidor debemos configurar una IP estática para que pueda servir como default Gateway para el resto de los usuarios, entonces ingresamos a la configuración de los parámetros de red

```
#system-config-network
```

Al final cambiamos el archivo

```
#vi /etc/host
```

Reiniciamos el servicio

```
#service network restart
```

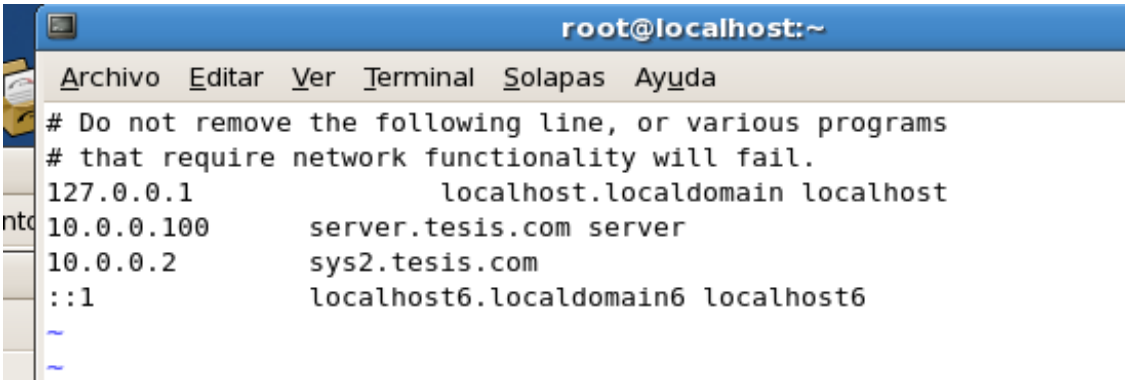

Figura III.18 Archivo hosts

3.4.2. Configuración Routing.

Como vamos a usar el servidor como router para la red 192.168.X.X debemos habilitar el renvío de paquetes para IP versión 4. Esto se realiza editando el archivo /etc/sysctl.conf y estableciendo 1 para activar o bien dejar 0 para mantener inactivo:

```
#vim /etc/sysctl.conf
```

Y cambiando net.ipv4.ip_forward = 0 por net.ipv4.ip_forward = 1:

```
#net.ipv4.ip_forward = 1
```

Para aplicar el cambio, sin reiniciar el sistema, sólo es necesario ejecutar lo siguiente:

```
#sysctl -w net.ipv4.ip_forward=1
```

Ahora para poder simular la salida al internet vamos a crear una IP alias sobre la interfaz del servidor este caso la eth0

```
#ifconfig eth0:0 192.168.1.1
```

Ahora dentro de los usuarios (Linux), añadimos el default route que ruteara a través de la dirección 192.168.1.1 configurada en el servidor

```
#route add –net default gw 192.168.1.1
```

3.4.3. Configuración DNS

Vamos a crear primeramente el archivo named.conf, dentro del directorio /var/named/chroot/etc/:

```
#vi /etc/named/chroot/etc/named.conf
```

Con el siguiente contenido

```
options {
        directory "/var/named";
```

```
};
zone "tesis.com" IN {
        type master;
        file "named.tesis.com";
};
zone "0.0.10.in-addr.arpa" IN {
        type master;
        file "named.10.0.0";
};
zone "localhost" IN {
        type master;
        file "localhost.zone";
};
zone "0.0.127.in-addr-arpa" IN {
        type master;
        file "127.0.0.zone";
};
```

Una vez creado el archivo named.conf vamos a crear el archivo que nos permitirá hacer la resolución de nombres a IPs, entonces creamos un archivo que lo vamos a llamar named.tesis.com dentro del directorio /var/named/chroot/var/named/:

```
#vi /var/named/chroot/var/named/named.tesis.com
```

Con el siguiente contenido

```
$TTL 86400

@     IN     SOA     server.tesis.com.
root.server.tesis.com. (

                    2002060701 ; serial

                    28800      ; refresh

                    7200       ; retry

                    604800     ; expire

                    86400 )    ; minimum TTL

     IN     NS      server.tesis.com.

     IN     NS      sys2.tesis.com.


server    IN     A       10.0.0.100

sys2      IN     A       10.0.0.2

sys3      IN     A       10.0.0.3
```

Creado el archivo de nombres a IPs ahora creamos el archivo de IPs a nombres dentro del mismo directorio:

```
#vi /var/named/chroot/var/named/named.10.0.0
```

Con el siguiente contenido

```
$TTL 86400

@     IN   SOA server.tesis.com.
root.server.tesis.com. (

                          2002060701 ; serial
```

```
                                28800       ; refresh
                                7200        ; retry
                                604800      ; expire
                                86400 )     ; minimum TTL
            IN   NS   server.tesis.com.
            IN   NS   sys2.tesis.com.

      100     IN   PTR  server.tesis.com.
      2       IN   PTR  sys2.tesis.com.
      3       IN   PTR  sys3.tesis.com.
```

Para la zona local localhost creamos el archivo de configuración con el nombre:

```
#vi /var/named/chroot/var/named/localhost.zone
```

Con el siguiente contenido

```
$TTL 86400
@     IN     SOA     server.tesis.com. root.tesis.com. (
                                2002060701 ; Serial
                                10800       ; Refresh
                                3600        ; Retry
                                604800      ; Expire
                                86400 )     ; Minimum TTL
       IN     NS      server.tesis.com.
       IN     A       127.0.0.1
```

Creamos una zona local con la IP

```
#vi /var/named/chroot/var/named/127.0.0.zone
```

Con el siguiente contenido

```
$TTL 86400

@     IN     SOA     server.tesis.com.
root.server.tesis.com. (

                                  2002060701 ; Serial

                                  10800      ; Refresh

                                  3600       ; Retry

                                  604800     ; Expire

                                  86400 )    ; Minimum TTL

  IN NS  server.tesis.com.

1 IN PTR localhost.
```

Ingresamos el nombre del servidor en el archivo de configuración hosts

```
#vi /etc/hosts

          127.0.0.1 localhost

          10.0.0.100 server.tesis.com server
```

Finalmente para que el hostname se quede gravado permanentemente modificamos el archivo de configuración network

```
#vi /etc/sysconfig/network
     NETWORKING=yes
     NETWORKING_IPV6=yes
     #HOSTNAME=localhost.localdomain
     HOSTNAME=server.tesis.com
```

Y el archivo de resolución de direcciones

```
#vi /etc/resolv.conf
     domain tesis.com
     nameserver 10.0.0.100
```

Iniciamos el servicio y para que inicie desde el arranque

```
#named start
#chkconfig named on
```

Para que cada cliente logre comunicarse con el servidor dns cambiamos su hostname para que se encuentre dentro del mismo dominio

```
#vi /etc/hosts
     127.0.0.1      localhost.localdomain localhost
     10.0.0.2  sys2.tesis.com sys2
```

Para que el hostname se quede gravado permanentemente modificamos el archivo de configuración network

```
#vi /etc/sysconfig/network
     NETWORKING=yes
     NETWORKING_IPV6=yes
     #HOSTNAME=localhost.localdomain
     HOSTNAME=sys2.tesis.com
```

Y el archivo de resolución de direcciones

```
#vi /etc/resolv.conf
     domain tesis.com
     nameserver 10.0.0.100
```

3.4.4. Servidor DHCP

Para configurar el servidor DHCP primero evaluamos el servicio de DNS, ingresamos al archivo de configuración del DHCP y nos aseguramos q el dominio sea tesis.com y verificamos la dirección del servidor

```
#vi /etc/dhcpd.conf
     option domain-name "tesis.com";
     option domain-name-servers 10.0.0.100;
```

y agregamos el rango de direcciones que se va a designar mediante DHCP

```
     subnet 10.0.0.0 netmask 255.255.255.0 {
          option routers 10.0.0.100;
```

```
        option subnet-mask 255.255.255.0;
        range 10.0.0.10 10.0.0.20;
    }
```

Para configurar dinámicamente el DNS en un cliente, añadimos al archivo

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
    DHCP_HOSTNAME=”sys2”
```

En el servidor para configurar el DNS dinámico seguimos el siguiente procedimiento

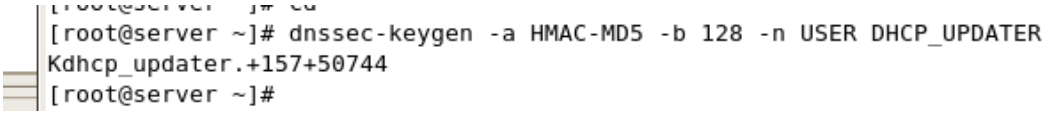

Figura lll.19 DNS dinámico, servidor

Vemos la clave que se genero y la copiamos

```
#cat Kdhcp_updater.*.private
```

Para luego copiarla en el archivo de configuración dhcpd.conf

```
#vi /etc/dhcpd.conf
    max-lease-time 3600;
    default-lease-time 600;
    ddns-update-style interim;
    ddns-updates on;
    key DHCP_UPDATER{
        algorithm hmac-md5;
```

```
        secret "j32mzJ6urHL6UhbqnkGRsQ==";
    }
    zone tesis.com{
        primary 10.0.0.100;
        key DHCP_UPDATER;
    }
    zone 0.0.10.in-addr.arpa.{
        primary 10.0.0.100;
        key DHCP_UPDATER;
    }
    option domain-name "tesis.com";
    option domain-name-servers 10.0.0.100;
    subnet 10.0.0.0 netmask 255.255.255.0 {
        option routers 10.0.0.100;
        option subnet-mask 255.255.255.0;
        range 10.0.0.10 10.0.0.20;
    }
```

Nos vamos al servidor DNS para que acepte las actualizaciones del servidor DHCP, añadimos

```
#vi /var/named/chroot/etc/named.conf

options {
        directory "/var/named";
```

```
};
zone "tesis.com" IN {
        type master;
        file "named.tesis.com";
          allow-update {key DHCP_UPDATER;};
};
zone "0.0.10.in-addr.arpa" IN {
        type master;
        file "named.10.0.0";
allow-update {key DHCP_UPDATER;};
};
zone "localhost" IN {
        type master;
        file "localhost.zone";
};
zone "0.0.127.in-addr-arpa" IN {
        type master;
        file "127.0.0.zone";
};
controls {
        inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "rndckey"; };
       };
       include "/etc/rndc.key";
```

```
Key DHCP_UPDATER {
     Algorithm hmac-md5;
     Secret “j32mzJ6urHL6UhbqnkGRsQ==”;
     };
```

Reiniciamos el servicio

```
#service named restart
#service dhcpd restart
#chkconfig dhcpd on
```

3.4.5. Servidor correo electrónico (Sendmail)

Para empezar la configuración del servidor de correo modificamos el archive sendmail.cf para que acepte correos de el dominio

```
#vi /etc/mail/sendmail.cf
```

Y agregamos al document

```
Cwlocalhost server.tesis.com tesis.com
```

Descomente la línea DaemonPortOptions line, si es que existe, y guardamos

Si se desea utilizar macros:

```
#vi /etc/mail/sendmail.mc
```

Des comentamos dnl, tal que quede así:

```
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1,
Name=MTA')dnl
```

```
#m4 sendmail.mc > sendmail.cf
```

Ingresamos los dominios para que acepte el mail

```
#vi /etc/mail/local-host-names

     server.tesis.com

     tesis.com
```

```
#vi /etc/mail/Access

     Connect:localhost.localdomain     RELAY

     Connect:localhost                 RELAY

     Connect:127.0.0.1                 RELAY

     tesis.com                         RELAY
```

```
#make
```

Reiniciamos el servicio de correo

```
#service sendamil restart
```

Para hacer un test de PoP3 verificamos en el archivo dovecot.conf lo siguiente

```
protocols = imap imaps pop3 pop3s
```

Hacemos que el servicio arranque cuando inicie la maquina

```
#chkconfig dovecot on
```

Para configurar ya la interfaz para el correo electrónico mediante sendmail ingresamos con el usuario que vayamos a usar y seguimos el siguiente procedimiento dentro de centos

Aplicaciones -> Internet -> Correo electrónico -> Evolution

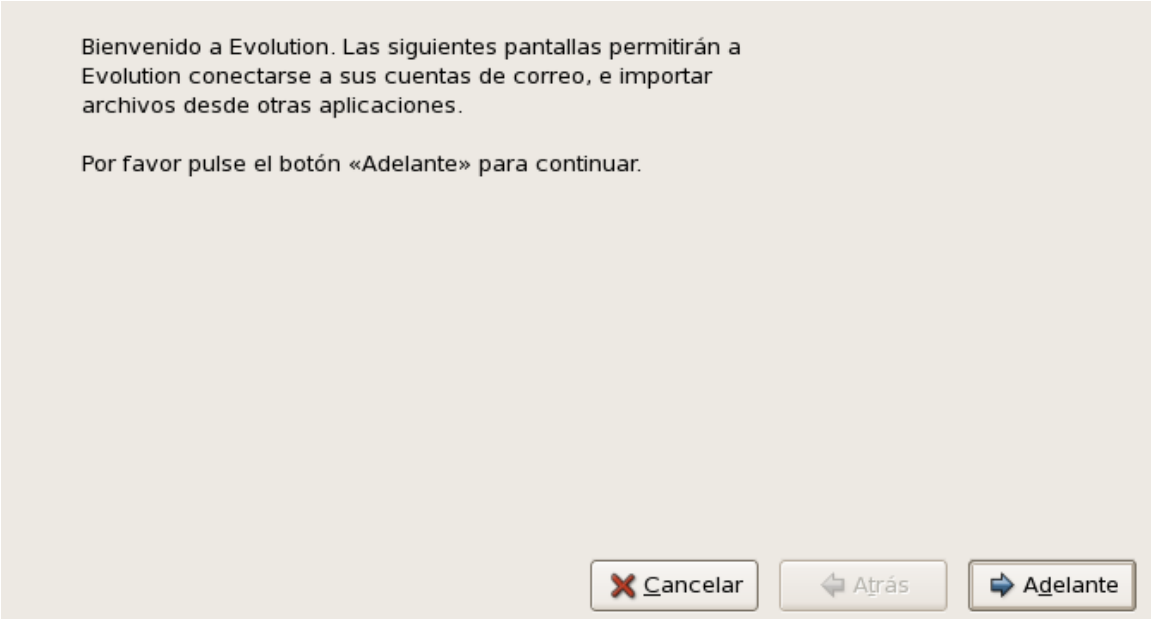

Figura lll.20 Correo electrónico Evolution

Siguiente

Identificamos

usuarioX@tdominio.com

tux1@tesis.com

Figura lll.21 Identificación correo electrónico

Siguiente

Ahora vamos a seleccionar POP como servidor, y configuramos el host y el usuario que vamos a usar, luego seleccionamos tipo de conexión y ponemos usar como conexión segura SSL

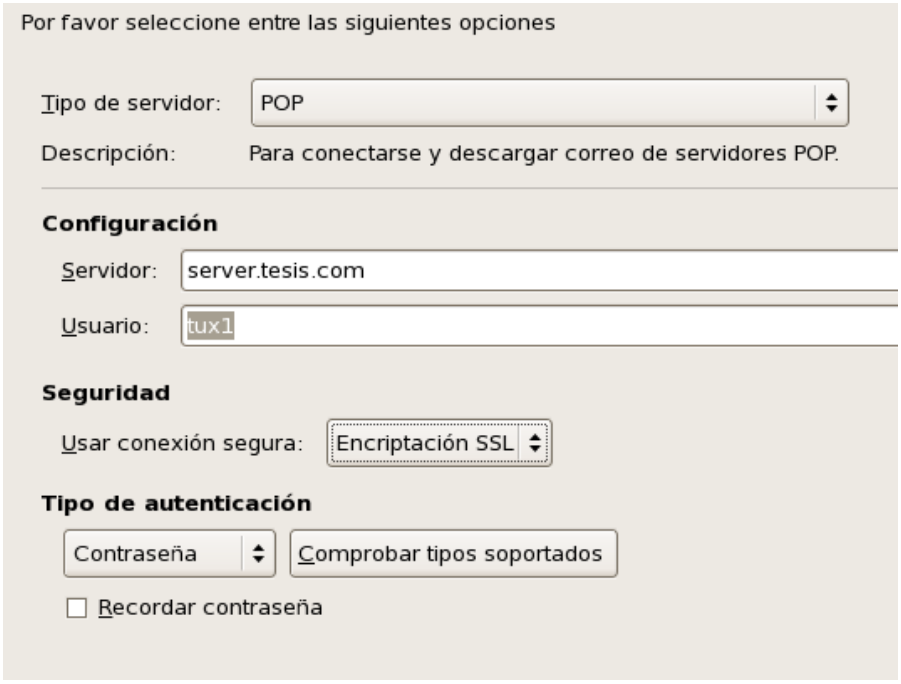

Figura lll.22 Configuración recepción de correo

Siguiente

Se configura las opciones de recepción de acuerdo a su preferencia en este caso se configuro de la siguiente manera.

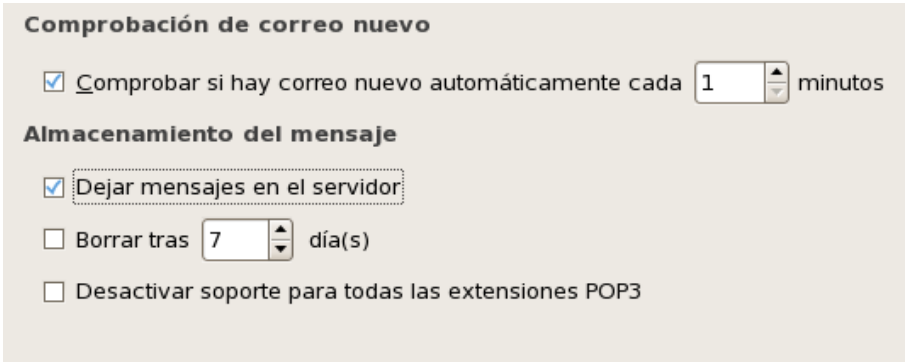

Figura III.23 Configuración opciones de recepción

Siguiente

Configuramos el envío de correo

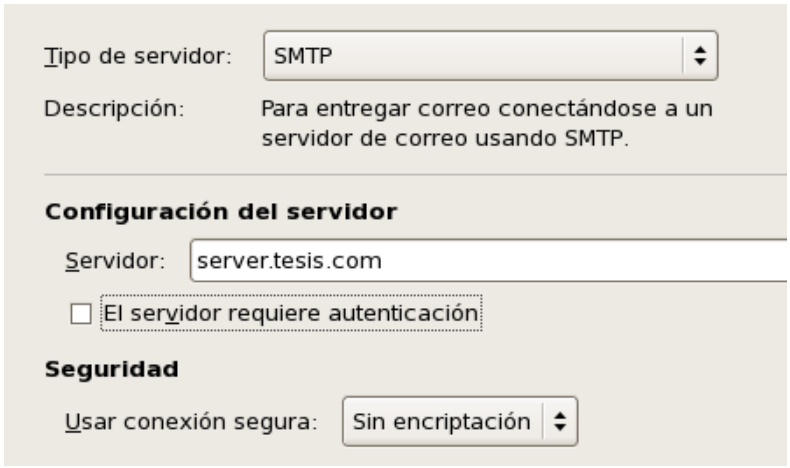

Figura III.24 Configuración envío de correo

Siguiente

Finalizar

3.4.6. Configuración Proxy

Como todos los servicios se están configurando en el servidor seguimos con el procedimiento, ahora vamos a configurar el servidor proxy, entonces iniciamos el servicio

```
#service httpd start
```

Podemos chequear si el servicio esta instalado mediante el comando

```
#rpm –qa squid
```

Si no esta instalado lo podemos instalar usando el cd de instalación de Centos, para ello se sigue el siguiente procedimiento:

Para montar el cd

```
#mount /mnt/cdrom
```

Accedemos al cd,

```
#cd /mnt/cdrom
```

Nos cambiamos al directorio donde están los paquetes rpm

```
#cd Red Hat/RPMS
```

E instalamos el paquete

```
#rpm –ivh squid-version.ARCH.rpm
```

Y desmontamos el cd

```
#umount /mnt/cdrom
```

Muy bien ahora ya instalado el servicio vemos en el servidor y decidimos que interfaz va hacer la que da la cara a la intranet y cual al internet, luego procedemos a la configuración para ello editamos el archivo /etc/squid/squid.conf

```
#vi /etc/squid/squid.conf
     http_port 3128
     acl our_networks src 10.0.0.0/24
     http_access allow our_networks
```

Reiniciamos el servicio

```
#service squid restart
```

Como demostración se va a crear dos reglas para el funcionamiento del proxy, en el mismo archivo squid.conf las agregamos

```
acl porno dstdom_regex "/etc/squid/porno.txt"
http_access allow our_networks all_users !porno
```

Ahora se necesita crear el archivo al cual nos referimos dentro del directorio/etc/squid porno.txt, en este archive vamos a poner las palabras que queremos bloquear

Reiniciamos el servicio

```
#service squid restart
```

Vamos a autenticar los usuarios para ello seguimos el procedimiento:

Modificamos el archivo squid.conf, indicando donde se encuentran los usuarios y sus claves e ingresamos una regla para que el proxy autentique

```
#vi /etc/squid/squid.conf

     auth_param basic program /usr/lib/squid/ncsa_auth
     /etc/squid/passwd

     acl all_users proxy_auth REQUIRED
```

Ingresamos las claves para que los usuarios puedan ingresar

```
#htpasswd –c /etc/squid/passwd USUARIO
```

Reiniciamos el servicio

```
#service squid restart
```

3.4.7. Configuración Apache

Se verificó el directorio donde se almacenaron los certificados digitales, el mismo que debe ser solo accesible para el usuario root. Se cambió los permisos del directorio certs/, que se encuentra almacenado en /etc/pki/tls:

```
# chmod –R 0700 /etc/pki/tls/certs
```

Se creó el archivo tesis.com con los permisos de acceso para el usuario root dentro del directorio etc/pki/tls/certs/ y se ingresó al directorio creado.

Dentro del directorio anterior, se generó la clave a través del algoritmo RSA de 1024 octetos con arquitectura X.509, que contiene cinco ficheros comprimidos con gzip (GNU zip), utilizando como semillas aleatorias que mejoran la seguridad de la clave creada, con el comando -out se le indicó que la clave pública de la AC se almacene en el fichero server.key

```
# openssl genrsa -des3 –rand
fichero1.gz:fichero2.gz:fichero3.gz:fichero4.gz:fichero5.
gz -out server.key 1024
```

Se generó la clave a través del algoritmo Triple DES, permitiendo una inicialización normal con el servidor httpd, el mismo que es enviado a una AC para su validación:

```
# openssl rsa -in server.key -out server.pem
```

Se generó la clave pública a través del comando openssl, que permitió generar una solicitud de firma de certificados creando el archivo server.csr:

```
# openssl req –new –key server.key –out server.csr
```

Se generó el certificado digital autofirmado por la el ficticio tesis.com, donde se especificó a través de la opción –days el número de días que será válido dicho certificado, con la opción –in se indicó, como parámetro de entrada el archivo sever.csr creado anteriormente, con la opción –signkey se ingresa el archivo que contiene la clave encriptada y con la opción –out se creó el certificado digital server.crt:

```
# openssl x509 -req -days 730 -in server.csr -signkey
server.key - out server.crt
```

Se generó la estructura para la red virtual creando los siguientes archivos: cgi-bin, html, logs, etc, var, se creó los directorios dentro de la ruta /var/www/ollva.com/, a través de mkdir desde la línea de comandos se ejecutó:

```
# mkdir -p /var/www/tesis.com/cgi-bin

# mkdir -p /var/www/tesis.com/html

# mkdir -p /var/www/tesis.com/logs

# mkdir -p /var/www/tesis.com/etc

# mkdir -p /var/www/tesis.com/var
```

Se creó el archivo tesis.conf, en el directorio /etc/httpd/conf.d/, para permitir el acceso a través de HTTP, desde la línea de comandos se ejecutó:

```
# vim /etc/httpd/conf.d/tesis.conf
```

Se ingresó al archivo y se añadió la siguiente información:

Se estableció la dirección IP del servidor web y el puerto a utilizar para tráfico normal a través del 80.

```
NameVirtualHost 10.2.0.5:80
<VirtualHost 10.2.0.5:80>
```

Se estableció la dirección de correo del administrador.

```
ServerAdmin tux1#tesis.com
```

Se estableció el directorio donde se localiza la página de inicio de la empresa.

```
DocumentRoot /var/www/tesis.com/html
```

Se estableció el nombre del servidor web y el alias correspondiente.

```
ServerName www.tesis.com
ServerAlias tesis.com
```

Se redireccionó a la página web segura que utiliza certificados digitales.

```
Redirect 301 / https://www.tesis.com/
```

Se estableció la ubicación de los archivos de almacenamiento de logs generados, para verificar eventos del servidor web.

```
CustomLog /var/www/tesis.com/logs/access_log
combined

Errorlog /var/www/tesis.com/logs/error_log
```

Se configuró el acceso seguro a través del puerto 443 que será validado con los certificados digitales creados previamente.

Se estableció la dirección IP del servidor web y el puerto a utilizar para tráfico seguro 443.

Se habilitó el soporte para SSL.

```
SSLEngine on
```

Se estableció el directorio /etc/pki/tls/certs/tesis.com/, en el que se encuentran creados los certificados digitales.

```
SSLCertificateFile
/etc/pki/tls/certs/tesis.com/server.crt

SSLCertificateKeyFile
/etc/pki/tls/certs/tesis.com/server.pem
```

Se configuró el servicio httpd para que arranque siempre que se encienda la maquina.

El link para ingresar al sitio web de la empresa es: https://www.tesis.com/.

3.4.8. Configuración Samba

Antes de iniciar la configuración del servicio, se creó la carpeta /var/samba/tux1 para alojar los scripts de inicio:

```
# mkdir /var/samba/tux1
```

Se asignaron todos los permisos

```
# chmod 777 /var/samba/tux1
```

Se ingresó y se modificó el archivo de configuración de Samba smb.conf, como se realizaron varios cambios, tenemos aquí el archivo final, desde la línea de comandos se ejecutó:

```
# vi /etc/samba/smb.conf
```

En la configuración global se estableció el grupo de trabajo editando el valor del parámetro workgroup, el parámetro netbios name estableció el nombre para el servidor, el parámetro server string es de carácter descriptivo para el controlador de dominio.

```
[global]

workgroup = TESIS

netbios name = TESIS-DC

server string = TESIS PDC Version %v

socket options = TCP_NODELAY IPTOS_LOWDELAY
SO_SNDBUF=8192
```

```
SO_RCVBUF=8192
```

El parámetro host allow valida el rango de direcciones IP que permitirá el acceso a las máquinas dentro del dominio, el parámetro wins support convierte al PDC en servidor WINS, el parámetro time server sincroniza las máquinas con la hora del servidor.

```
security = user
guest ok = no
encrypt passwords = yes
null passwords = no
hosts allow = 127.0.0.1 10.0.0.0/255.255.255.0
wins support = Yes
name resolve order = wins lmhosts host bcast
dns proxy = no
time server = yes
```

Para este caso demostrativo vamos a usar la carpeta creada llamada tux1 para que los usuarios puedan ingresar a ver su contenido

```
[compartidatux1]
comment=Carpeta compartida desde Samba
path=/var/samba/tux1
;    admin users=administrador
```

```
write list=tux1

valid users=tux1

create mask=0777
```

Ponemos una papelera dentro de este usuario

```
vfs objects = recycle

recycle:repository = Papelera

recycle:versions = yes

recycle:keeptree = yes

recycle:exclude =
*.tmp|*.temp|*.o|*.obj|~$*|*.~??|*.log|*.trace|*.TMP

recycle:excludedir = /tmp|/temp|/cache

recycle:noversions = *.doc|*.ppt|*.dat|*.ini

recycle:minsize = 10

recycle:maxsize = 5120
```

3.4.9. Configuración PKI

3.4.9.1. Instalación de paquetes necesarios

OpenCA es un sistema que para su funcionamiento necesita interactuar con varias aplicaciones de la comunidad Open Source, entre ellos validamos que se encuentren instalados los paquetes que se detallan en la tabla lll.3.

MÓDULO	VERSIÓN	COMENTARIO
Authen::SASL	2.04	Requerido para autenticación
CGI::Session	3.95	Requerido para manipulación de cuentas
Convert::ASN1	0.18	

Digest::HMAC	1.01 1,01	Requerido para autenticación
Digest::MD5	2.24	Fundamental para el funcionamiento de Perl
Digest::SHA1	2.02	Fundamental para el funcionamiento de OpenCA
Encode::Unicode		Requerido para codificación decodificación
IO::Socket::SSL	0.92	
IO::stringy	2.108	
MIME::Base64	2.20	Requerido para codificación decodificación
MIME::Lite	3.01	Requerido para manipulación mediante OpenCA
MIME-tools	5.411	Requerido para manipulación mediante OpenCA
Mailtools	1.58	Requerido para manipulación mediante OpenCA
Net-Server	0.86	Requerido para el demonio[67] de OpenCA
URI	1.23 1,23	
Xml::twig	3.09	Usado por XML
libintl-perl	1.10	
perl-ldap	0.28	Interface Perl's LDAP

Tabla lll.3. Requerimientos para la instalación de OpenCa

Se instalaron de los módulos Perl detallados en la tabla lll.3

Con los paquetes previamente instalados se procedió a descargar la herramienta OpenCA y OpenCA Tools de la página home http://www.openca.org/projects/openca/downloads.shtml, los archivos fuente. Una vez descargados, se descomprimieron los paquetes en el directorio.

```
# cd /usr/src/
```

Se instaló OpenCA tools previo a la instalación de OpenCA, se descomprimió el paquete tar.gz, posteriormente se ingresó al directorio, ejecutándose el archivo de configuración mediante ./, finalmente con él se ejecutó make y make install.

Se descomprimió la herramienta OpenCA, ingresando al directorio, con el comando touch se creó el archivo vacío config_ca.

```
# touch config_ca
```

Se proporcionaron los permisos de ejecución.

```
# chmod 755 config_ca
```

Se editó el archivo de configuración para config_ra, ingresando mediante el comando vim:

```
# vim config_ra
```

Se editó, para compilarlo posteriormente, ./configure realizó algunos ajustes para no compilar la configuración por defecto.

Se realizó la configuración de ruta de acceso mediante prefix. Se configuró el servidor web relacionado, ya que OpenCA utiliza una interfaz web, mediante with-httpd.

```
./configure \
--prefix=/usr/local/openca \
--with-web-host=localhost \
--with-httpd-user=apache \
--with-httpd-group=apache \
--with-ext-prefix=/usr/local/openca \
--with-htdocs-fs-prefix=/var/www/cgi-bin/pki \
--with-cgi-fs-prefix=/var/www/cgi-bin/pki \
```

Se añadió la información de la AC, como Nombre de la Organización, localización de la Provincia y País; mediante with-ca.

```
--with-ca-organization="Tesis" \
--with-ca-locality="Chimborazo" \
--with-ca-country="EC" \
```

Se configuró el directorio donde se encuentran los módulos perl, habilitando la base de datos de la interfaz, se estableció una dirección de correo para la PKI.

```
--with-module-prefix=/usr/local/lib/openca/perl_modules \
--enable-dbi \
--disable-db \
--disable-rbac \
--with-hierarchy-level=ca \
--with-service-mail-account="pki@tesis.com" \
```

Se configuraron las opciones de la base de datos, tipo, nombre de la tabla, nombre del servidor que almacenó los datos, puerto, usuario y contraseña de acceso.

```
--with-db-type=mysql \
--with-db-name=openca \
--with-db-host=localhost \
--with-db-port=3306 \
--with-db-user=openca \
--with-db-passwd="openca"
```

Se grabaron los datos y se ejecuto el scrip

```
# ./config_ca

make

make install-ca
```

Se ingresó a la base de datos MySQL, ingresando el usuario y contraseña, desde la línea de comandos se ejecutó:

```
#mysql -u root –p

Enter password:
```

Se creó la base de datos llamada openca.

```
mysql> create database openca;
```

Se estableció una contraseña para la base de datos.

```
mysql> GRANT create,drop,select,delete,insert,update ON
openca.* TO 'openca_u'@'localhost' IDENTIFIED by
'database_password';
```

Se confirmaron los privilegios para el acceso.

```
 mysql> FLUSH privileges;
```

Finalizada la instalación de la base de datos, a continuación se realizaron varios cambios en el archivo de configuración que administra OpenCA, modificando el archivo config.xml,

```
# vim /usr/local/openca/etc/openca/config.xml
```

Donde se editaron los siguientes parámetros.

Se configuraron los parámetros de la AC.

```
ca_organization Tesis

ca_locality Riobamba ca_state Chimborazo ca_country EC
send_mail_automatic yes

service_mail_account tux1@tesis.com
```

Se configuró el Servidor Web, a través del puerto seguro.

```
httpd_host localhost

httpd_port :443
```

Se configuró la base de datos.

```
dbmodule DBI

db_type mysql

db_name openca

db_host localhost

db_port 3306

db_user openca

db_passwd openca
```

A continuación se modificó las plantillas de acuerdo a la configuración establecida de la AC

```
# cd /usr/local/openca/etc/openca

# ./configure_etc.sh
```

Se ejecutó el script para la creación de dichos archivos.

Se inició el servicio openca, en ese momento nos solicitó la contraseña de ingreso a la aplicación.

```
# service openca start
```

Se instalaron los certificados digitales de Apache, se ingresó un periodo de 3650 días.

```
# cd /etc/httpd/

# openssl req -x509 -newkey rsa:2048 -keyout cakey.pem -
days 3650 - out cacert.pem –nodes
```

Con este comando se creó un certificado digital X.509 para la AC, con el algoritmo de encriptación RSA de 2048 bytes. A través del comando -keyout se le indica que la clave privada de la AC se almacene en el fichero cakey.pem. Con el comando -out se le indica que la clave pública de la AC se almacene en el fichero cacert.pem.

Adicionalmente se ingresó los datos del certificado digital como: País, Provincia, Nombre de empresa para identificación como AC y la dirección de correo electrónico.

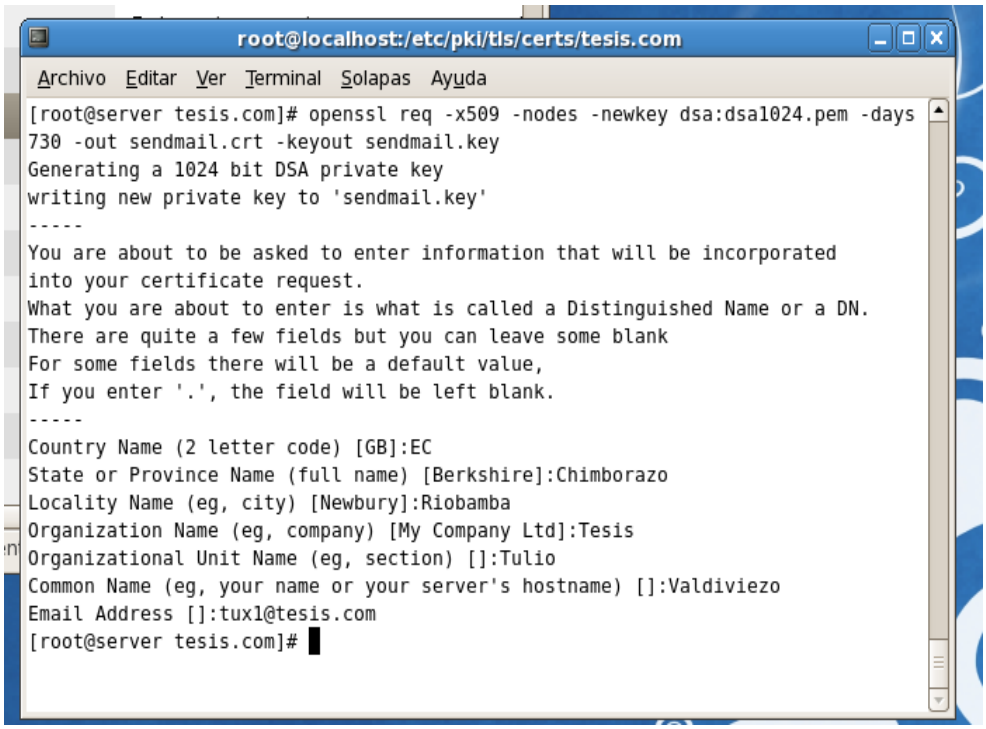

Figura lll.25 Generación del Certificado Digital para la PKI

Una vez generado el certificado digital se configuró el Servidor Web, editando el archivo ssl.conf, que se encuentra almacenado en el directorio /etc/httpd/conf.d/, donde se colocó el path donde se crearon los certificados digitales.

```
#cd conf.d/
#vi ssl.conf

    Listen 443
    <VirtualHost *:443>
    SSLEngine On
    SSLCertificateFile /etc/httpd/cacert.pem
```

```
SSLCertificateKeyFile /etc/httpd/cakey.pem
</VirtualHost>
```

Se configuraron los servicios openca y httpd para que arranquen siempre que se encienda el SO, posteriormente se iniciaron los servicios.

Para probar que el OpenCA quedó correctamente instalado, se abrió el navegador de internet y se colocó a la siguiente dirección: https://10.2.0.3/ca.

CAPÍTULO lV

FUNCIONAMIENTO

El funcionamiento se realiza en el entorno desarrollado mediante maquinas virtuales, en este caso las pruebas se ejecutaran en el servidor y usuario que usan la distribución Centos

4.1. Conectividad

Se probara que exista conectividad entre los usuarios y el servidor y la salida al internet que en este caso esta simulada por la dirección 192.168.1.1

Para ello se realizo un ping desde el usuario Linux hacia el servidor

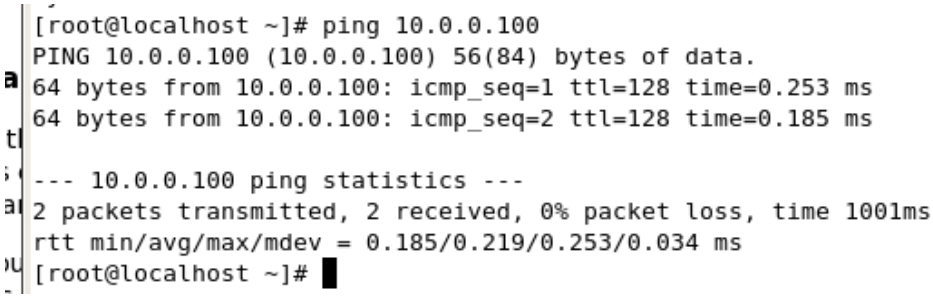

Figura IV.26 Ping hacia el servidor

Y al contrario igual desde el servidor hacia el usuario

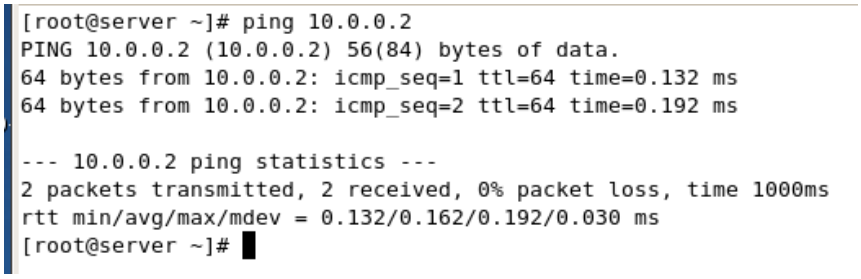

Figura IV.27 Ping desde el servidor

Finalmente comprobamos que haya salida al internet

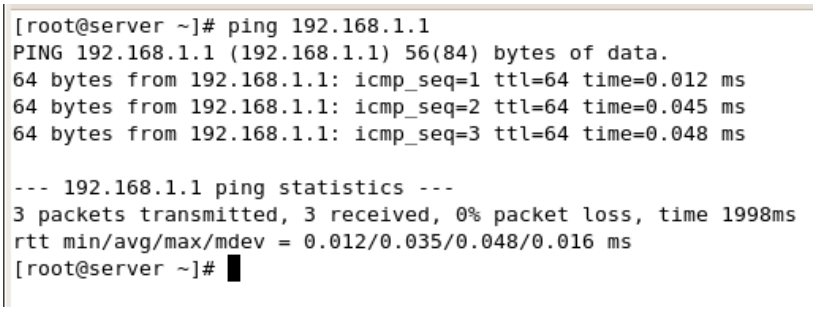

Figura IV.28 Ping desde el servidor hacia internet

4.2. Resolución de nombres (DNS)

Para empezar se comprueba que el servicio de DNS este corriendo satisfactoriamente en el servidor

```
#tail -50 /var/log/messages | grep named
```

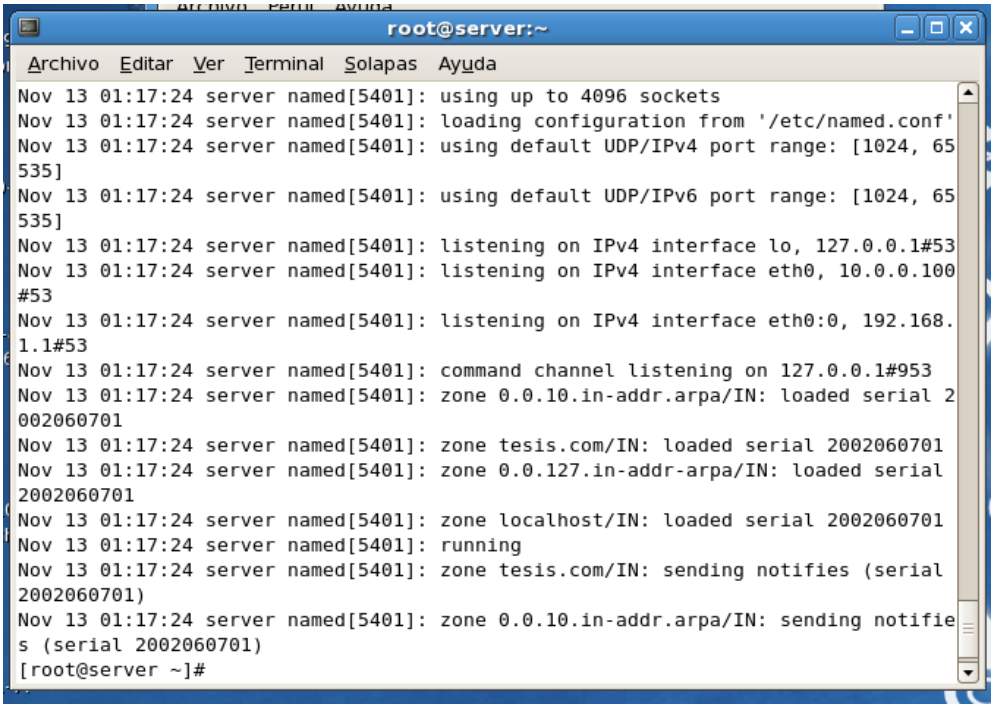

Figura IV.29 Verificación del funcionamiento del DNS

Aquí podemos observar que los servicios se están ejecutando y las zonas estas funcionando para la resolución de nombres tanto de nombres a IPS y viceversa

Ahora dentro del cliente vamos a observar si la resolución de nombres esta funcionando correctamente mediante el comando ping, demostrándolo de varias maneras como se indica en la figura lV.30

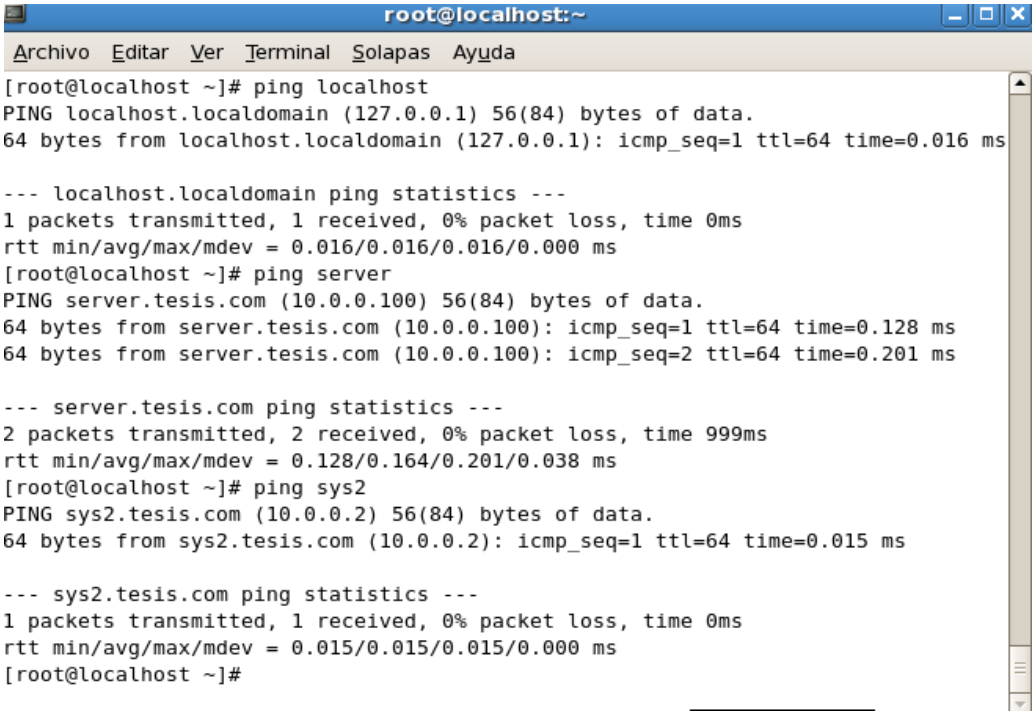

Figura IV.30 Ping para comprobación del DNS

Finalmente para comprobar que los nombres corresponden a las direcciones asignadas en el diagrama de red desde el usuario comprobamos la identidad de los nombres

```
[root@server ~]# host server
server.tesis.com has address 10.0.0.100
[root@server ~]# host localhost
localhost has address 127.0.0.1
[root@server ~]# host 10.0.0.100
100.0.0.10.in-addr.arpa domain name pointer server.tesis.com.
[root@server ~]#
```

Figura IV.31 Resolución del nombre del host de un host en el dominio

4.3. Servidor de DHCP

Para la comprobación del servicio de DHCP en los usuarios Windows simplemente configuramos la interfaz para que obtenga la IP automáticamente

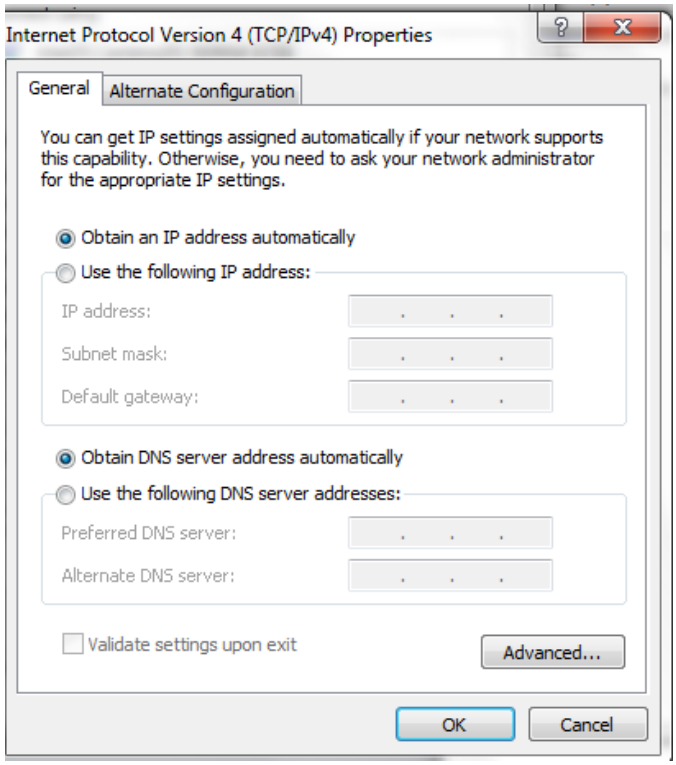

Figura IV.32 Obtener una IP automáticamente Windows

En Linux es parecido dentro de las opciones de configuración de la tarjeta de red ponemos para que acepte automáticamente las direcciones IPs

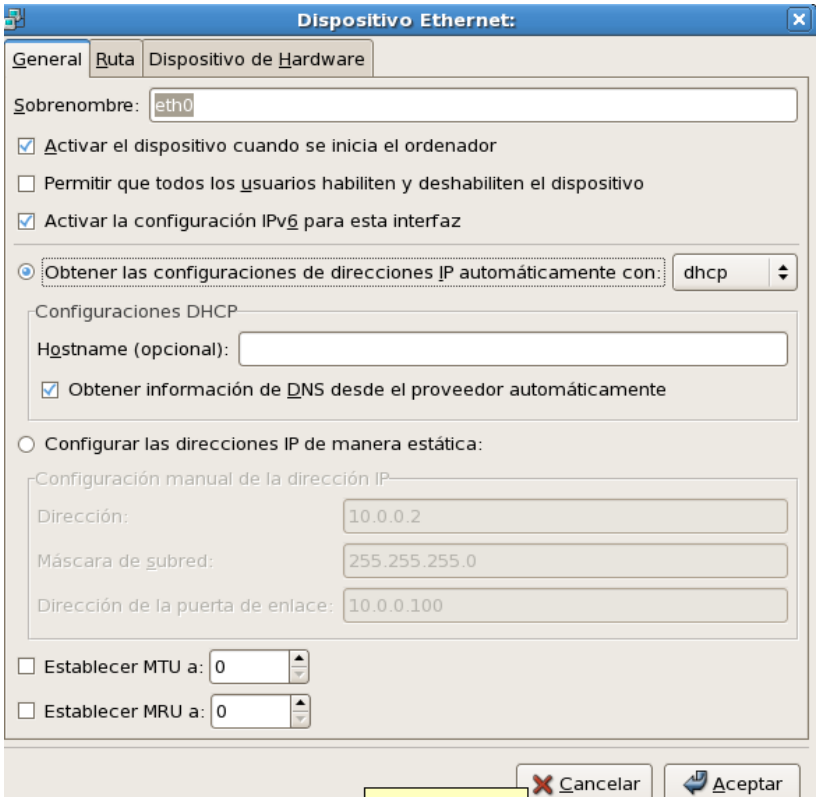

Figura IV.33 Obtener una IP automáticamente Linux

4.4. Servidor mail (Sendmail)

Sendmail nos puede servir de varias maneras, para comprobar su funcionamiento se lo puede hacer desde línea de comandos como se indica a continuación:

```
# echo Este es una prueba2 | mail -s "prueba2"
tux1@tesis.com
```

De esa manera enviamos un mensaje al usuario tux1 y comprobamos q el mensaje haya llegado

```
#cd /var/spool/mail

#cat tux1
```

Y vemos el mensaje que se envió

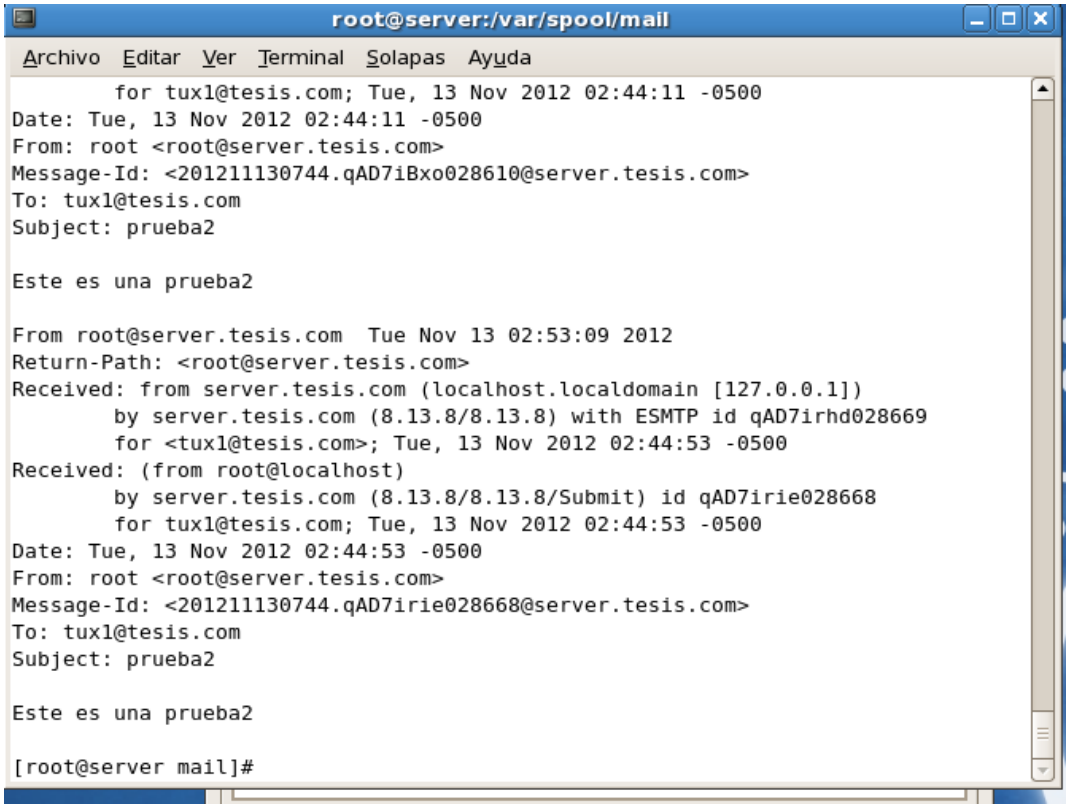

Figura IV.34 Visualización de un mensaje en sendmail

Podemos usar sendmail con SSL para enviar mensaje certificados para probar hacemos lo siguiente:

```
#openssl s_client -connect localhost:995
```

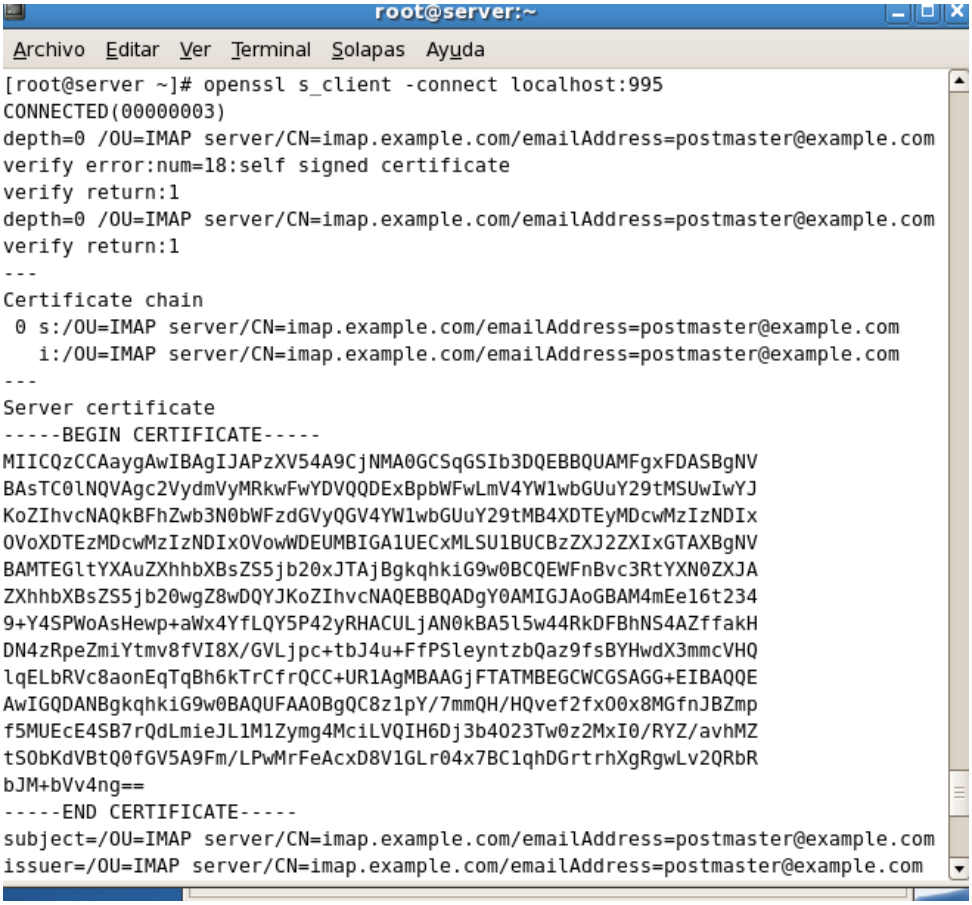

Figura IV.35 Sendmail usando certificado SSL parte 1

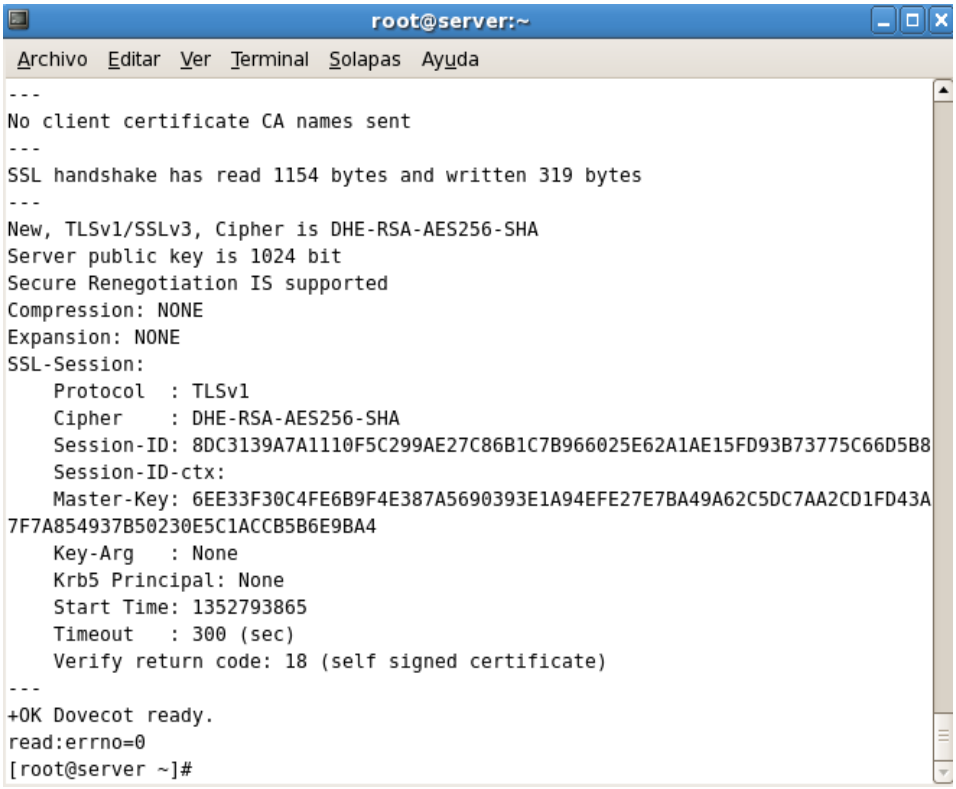

Figura IV.36 Sendmail usando certificado SSL parte 2

Si queremos usar la pagina del squirrelmail nos identificamos e ingresamos a usar normalmente

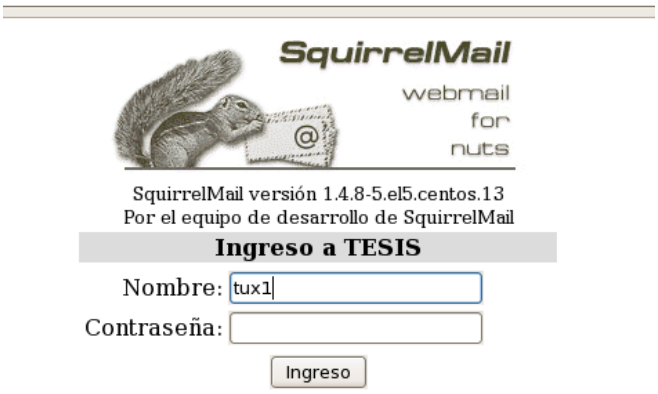

Figura IV.37 Squirrelmail

4.5. Servidor Squid (web seguro)

Para lograr tener una web segura se lo va a realizar mediante un proxy, específicamente Squid, aquí podemos autenticar usuarios para que puedan usar la web, un ejemplo es el siguiente para poder ver una pagina se solicita que el usuario se autentifique como se indica en la figura lV.37

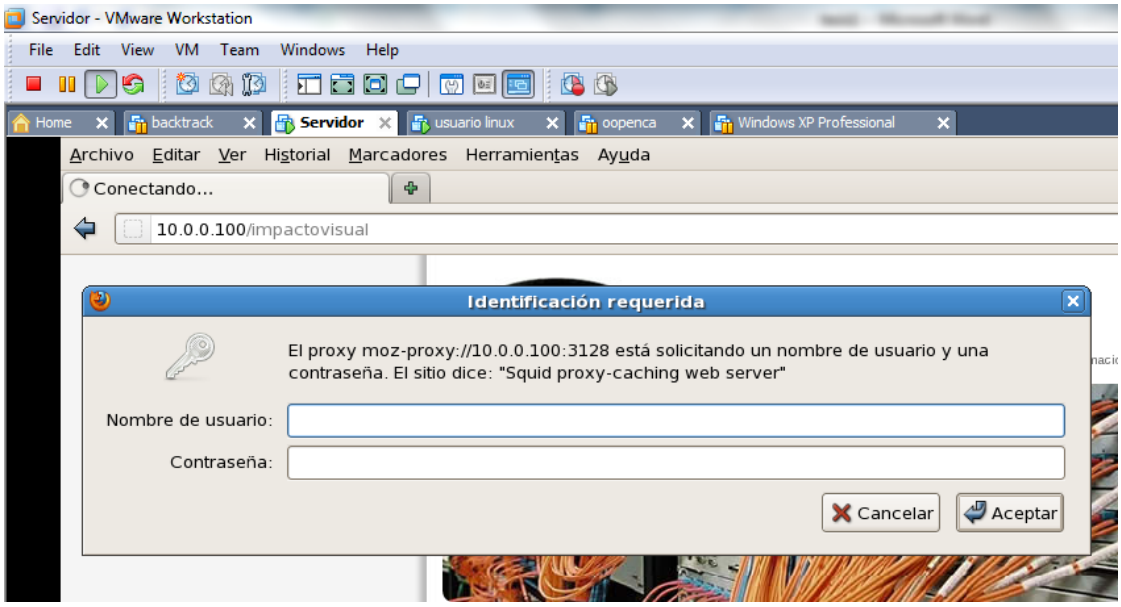

Figura lV.38 Autenticación mediante proxy

Una vez que nos identificamos ya podemos usar el explorador de forma normal, una forma también de controlar los contenidos que los usuarios van a ver es mediante unas listas de control, para este caso se han creado dos uno para bloquear la palabra “sex” y el otro bloqueando algunas paginas que a los administradores no le interesa que los usuarios usen como yahoo.com

Al intentar ingresar una palabra que contenga sex automáticamente el proxy nos va a denegar el acceso

Figura IV.39 Bloqueo palabra "sex", usando un proxy

Ahora que comprobamos una de las listas de control la otra bloquea algunas paginas como ya se explico el funcionamiento es el mismo que con la palabra sex, así podemos ir ingresando varias reglas dependiendo de la necesidad que se tenga en la red.

Figura IV.40 Bloqueo de la dirección www.hotmail.com,usando un proxy

4.6. Servidor Apache

4.6.1. Apache y autenticación de directorios

La configuración de apache lleva varias partes, la primera demostración se va hacer para la autenticación de directorios

En este caso se va a tratar de ingresar al fichero creado en el momento de la instalación para ello vamos a ver que si no nos autenticamos no se nos permite el ingreso, tratamos de ingresar a la dirección server.tesis.com y al no autenticarnos nos sale lo siguiente

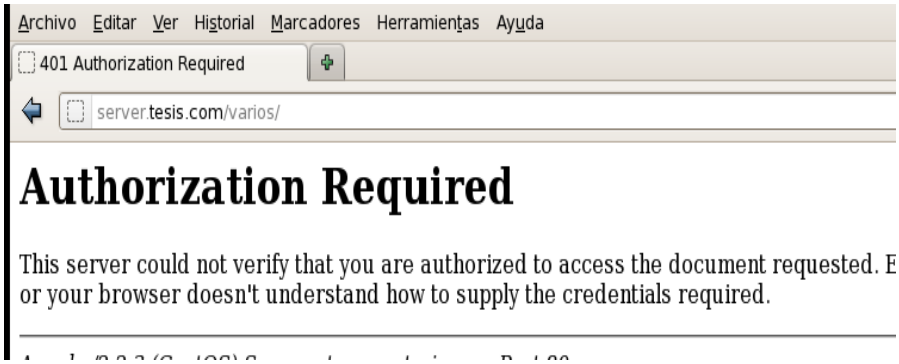

Figura IV.41 Petición de autenticación apache

Como vemos no podemos ingresar actualizamos la pagina e ingresamos los datos

Figura IV.42 Autenticación ingreso a directorios apache

El momento que ya logramos entrar tenemos acceso a las carpetas que el administrador decide que los usuarios van a tener acceso

Figura IV.43 Acceso directorios mediante apache

4.6.2. Apache y SSL

Apache trabaja con uno de los módulos de SSL, la cual permite configurar https, web seguro, para ello hacemos uso de un certificado x509 el cual generamos e

insertamos en el explorador, este procedimiento también nos da protección al demonio http, necesitamos una clave para utilizar el servicio

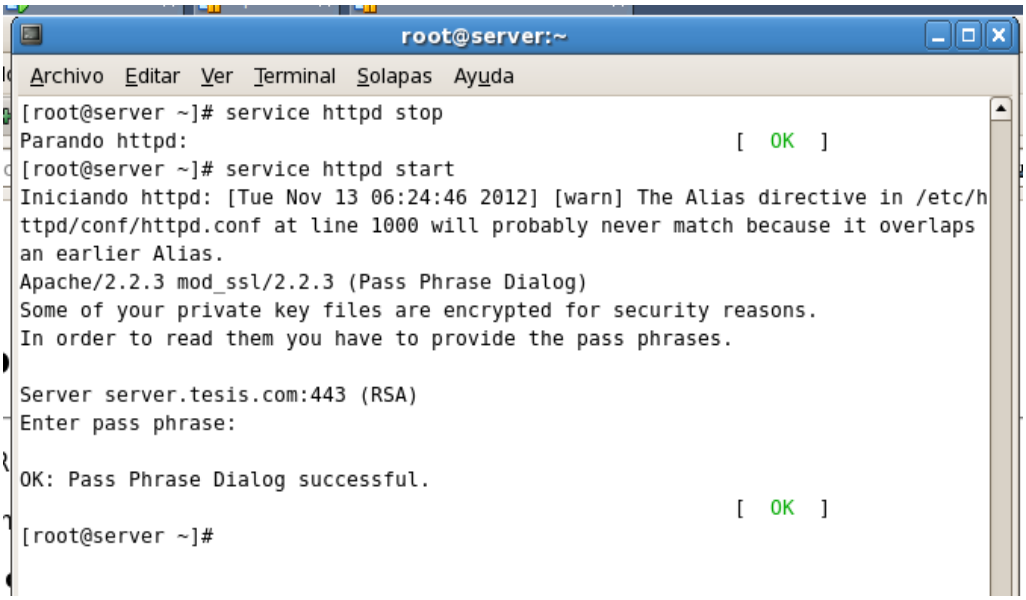

Figura IV.44 Inicio del servicio httpd con autenticación

Como vemos para poder iniciar el servicio necesitamos ingresar la phrase generada por el rsa

Ahora ya podemos usar web segura, https, ingresamos a la dirección ya usando el protocolo https://server.tesis.com, teniendo el siguiente resultado

Figura IV.45 Ingreso a verificar la conexión

Damos clik en entiendo los riesgos y entramos en añadir excepción

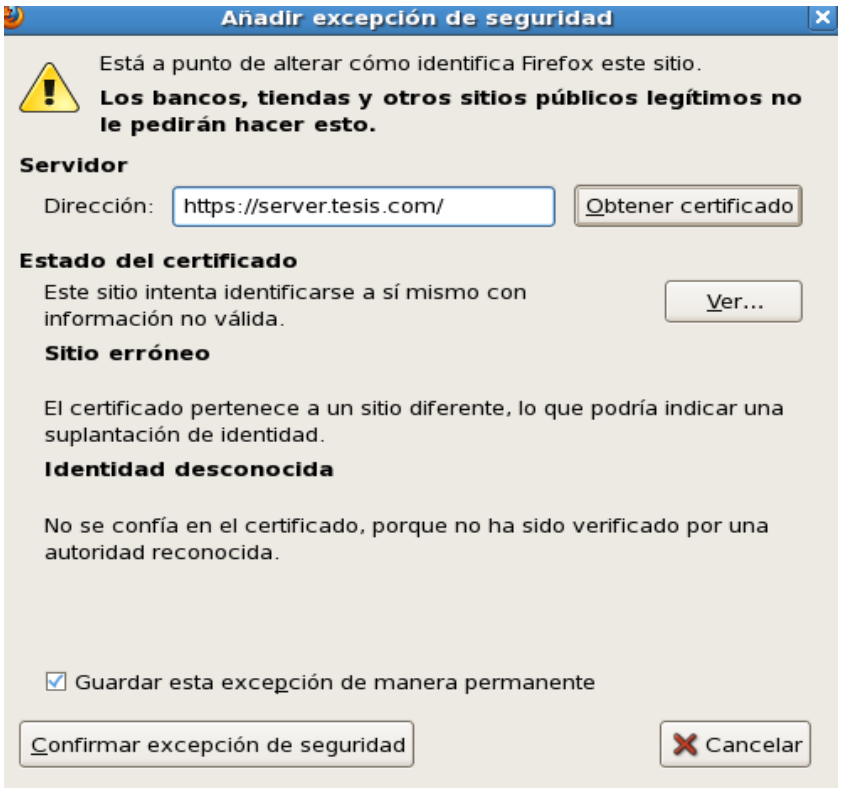

Figura IV.46 Añadir excepción de seguridad

Podemos ver el certificado, en este caso el certificado se llama tesis.com, alguna de la información que se puede ver dentro de la pestaña detalles son los siguientes

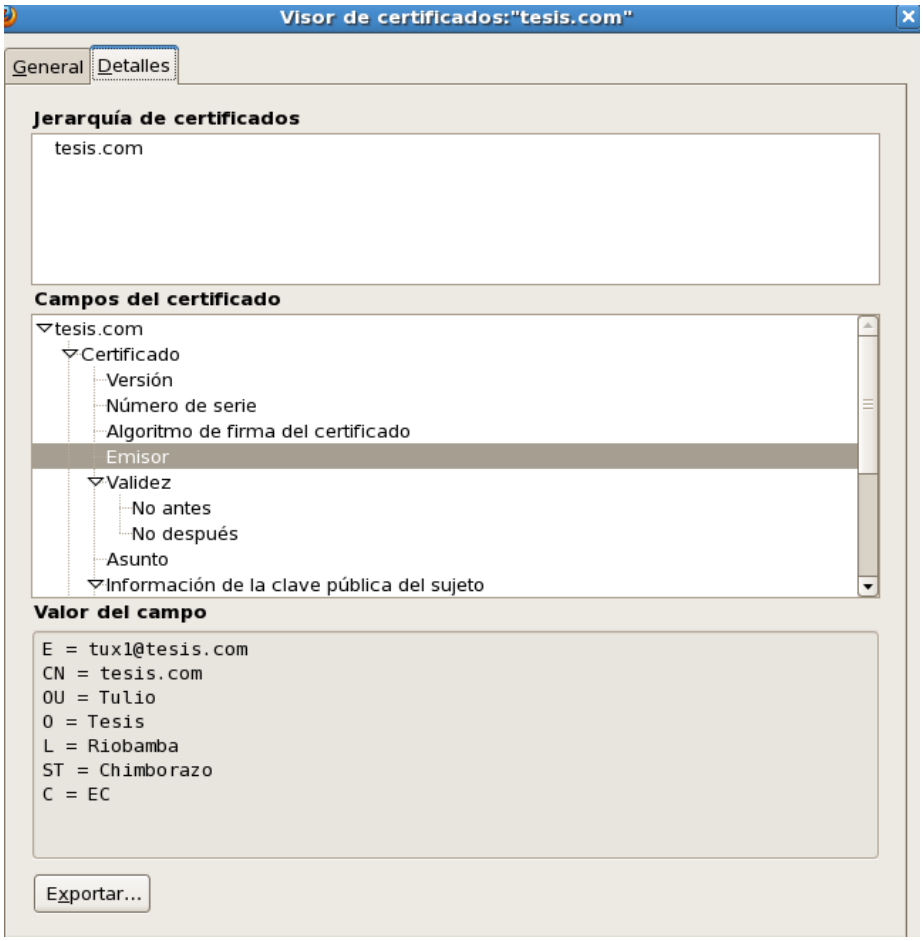

Figura IV.47 Detalles certificado digital

Finalmente una vez que confirmamos la excepción de seguridad podemos navegar en una web segura

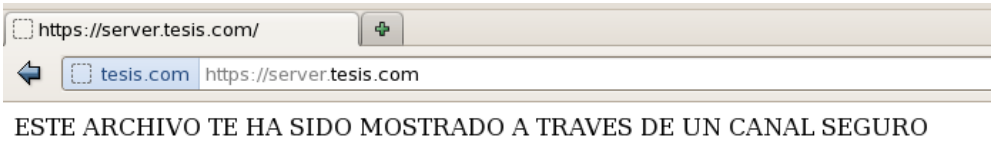

Figura IV.48 Navegación en https

4.7. Servidor OpenCA

Se empezó con la inicialización de la AC, se proporcionó todos los datos necesarios para la solicitud del certificado, ingresando a la pestaña General Initialization

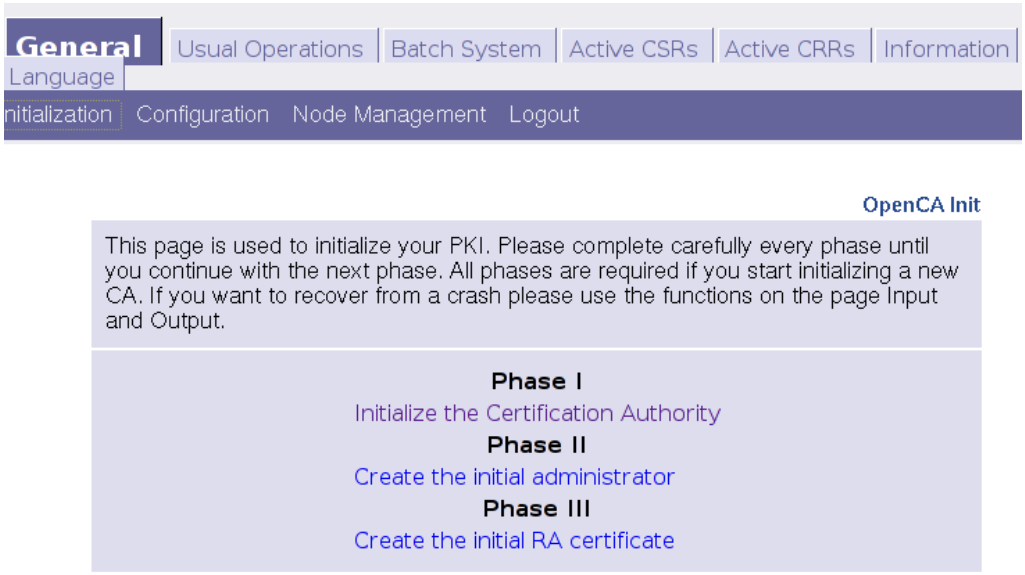

Figura IV.49 Inicialización autoridad certificadora

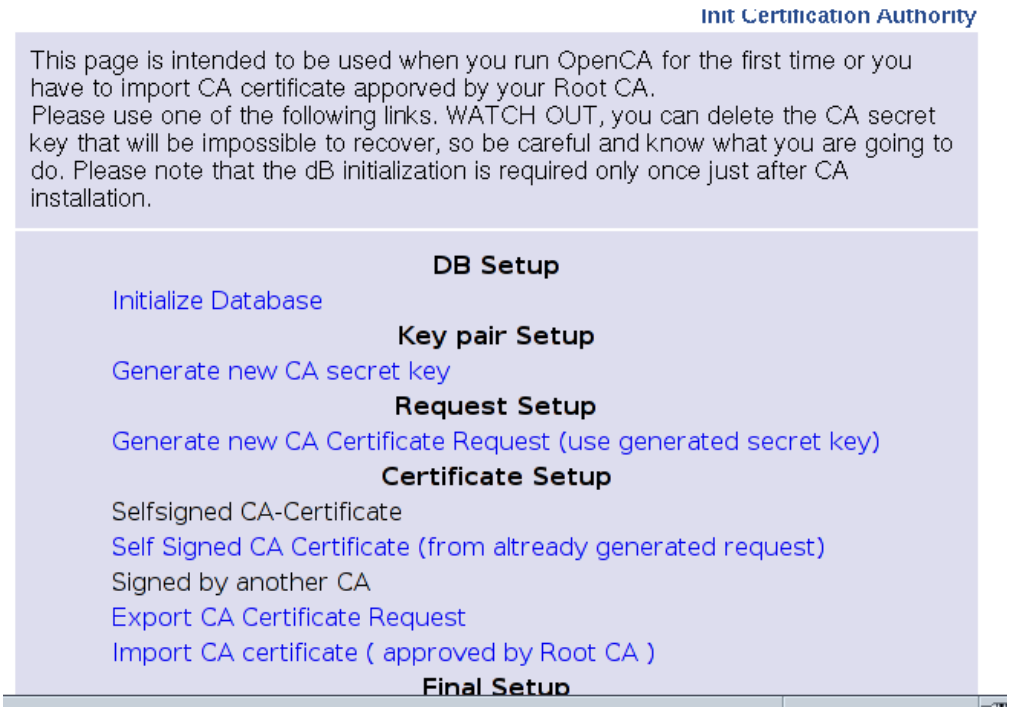

Figura IV.50 Inicialización Base de Datos

Hasta este punto hemos inicializado la base de datos de la CA.

Ahora nos toca generar la pareja de llaves (Ku y Kp) de la CA. Recuerde que esas serán las llaves con las que se firmarán los futuros certificados, entonces ahora generamos la nueva llave secreta CA, la llave vamos a escoger de 1024 bits

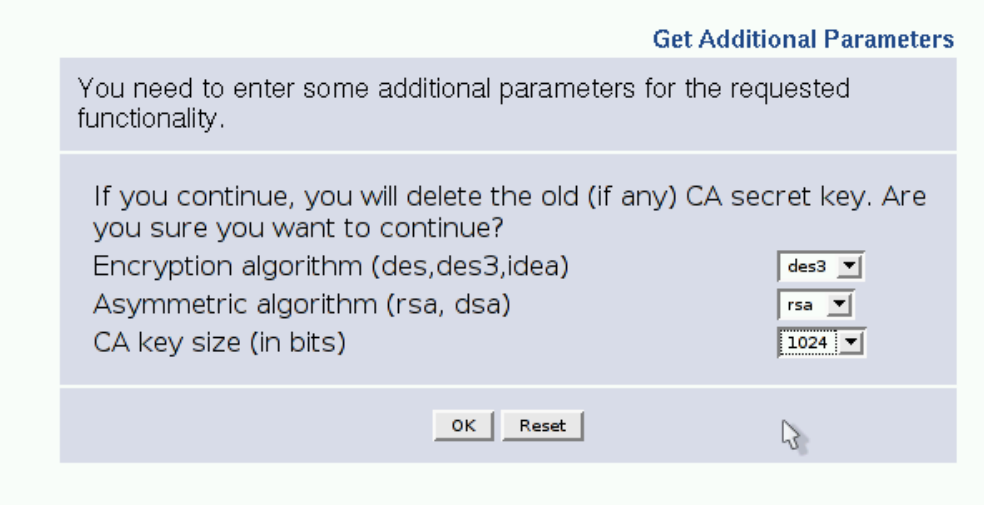

Figura IV.51 Generación par de llaves

Siguiente

Y ahora vamos a ingresar el password de la clave privada. Este será el Password de la Clave Privada de la CA.

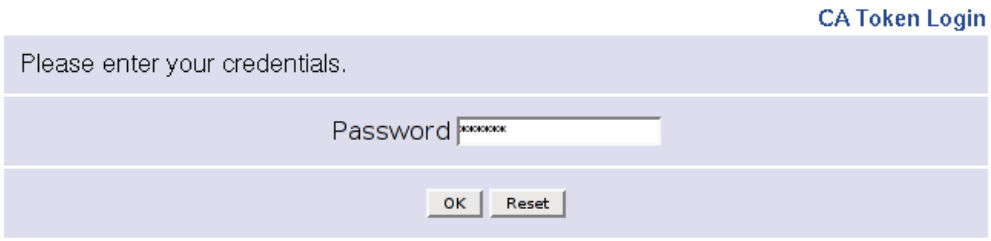

Figura IV.52 Ingreso del password clave privada

Luego de ingresar la clave nos sale la información de la llave cifrada

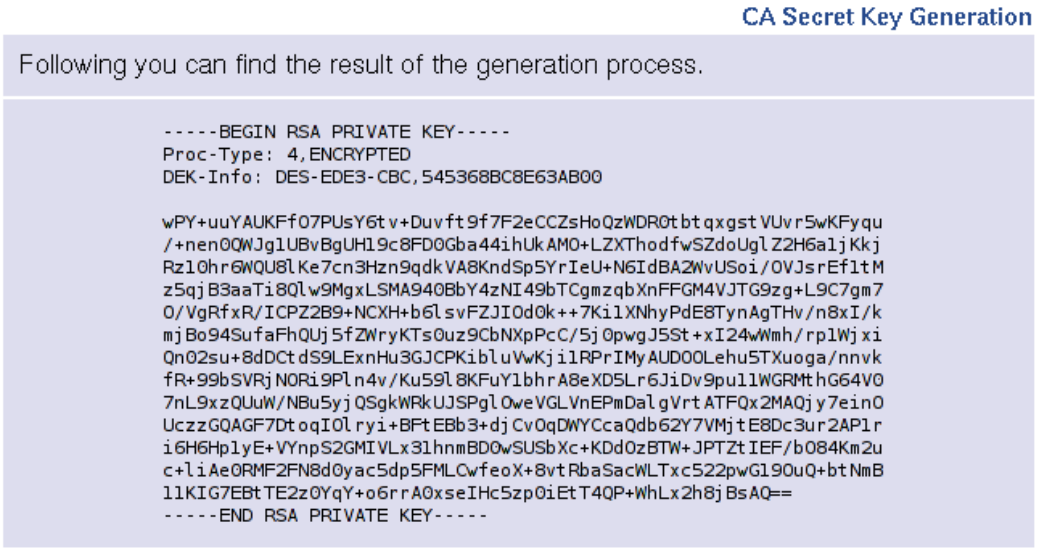

Figura IV.53 Resultado de la generación de la llave

Ya tenemos llaves pero no Certificados. Necesitamos crear los certificados de la CA.

Vamos de nuevo a Inicialización e ingresamos de nuevo a la fase 1 inicializar la autoridad certificadora

Una vez ahí entramos en Request Setup

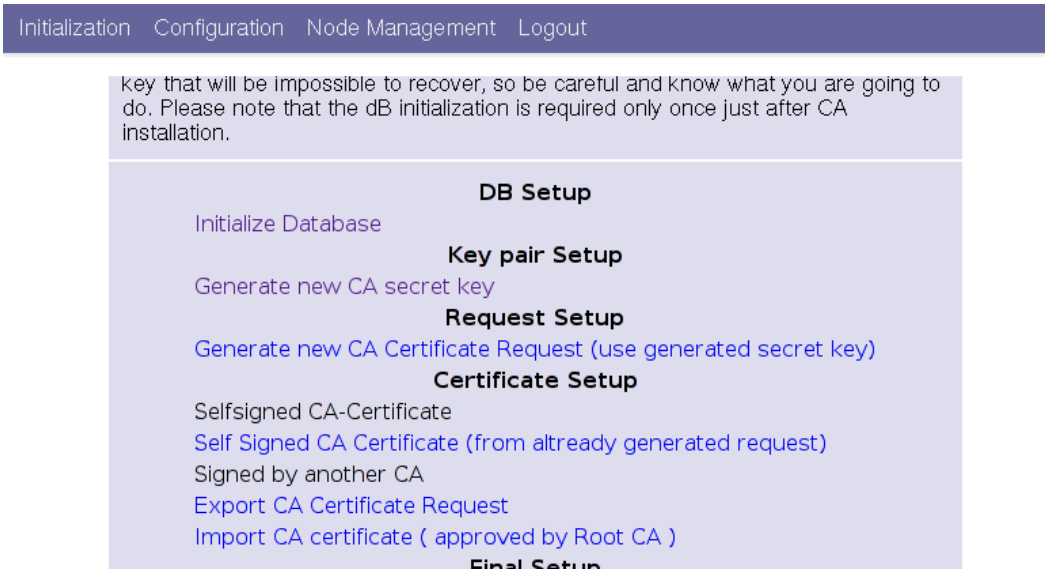

Figura IV.54 Request Setup

Una ves que entremos es aquí donde vamos a configurar los datos que irán de acuerdo a nuestra CA

Figura IV.55 Parámetros de la CA

Una vez que llenamos todos los datos seguimos adelante y vamos a otra ventana donde va hacer la última oportunidad de corregir algún dato

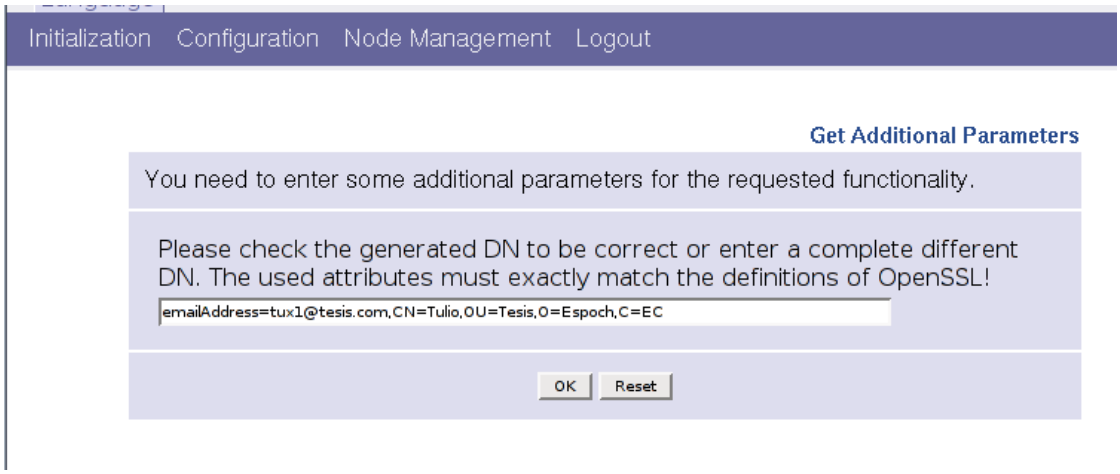

Figura IV.56 Parámetros adicionales de la CA

Ponemos OK y nos va a pedir que ingresemos la clave que ingresamos en el paso anterior, una vez que lo ingresamos nos va a salir el resultado del proceso de generación

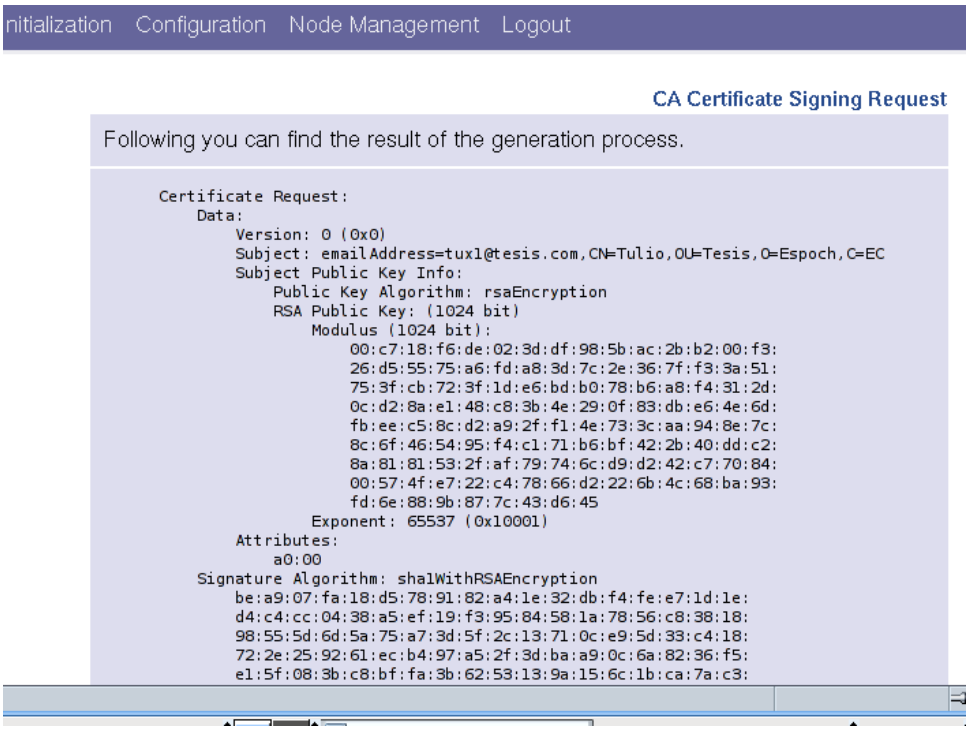

Figura IV.57 Resultado proceso de generación

Para continuar regresamos de nuevo a la fase 1, Hemos hecho una solicitud de certificado. Ser firmada por la CA de jerarquía superior más bien dicho la firmaremos nosotros mismos.

Vamos ya dentro de la fase 1 a Certificate Setup y seleccionamos firmarla nosotros mismos

Figura IV.58 Self Signed CA

Una vez dentro vamos a configurar algunos parámetros más, en este caso se ha configurado que la validez de la CA sea de 90 días

Figura IV.59 Parámetros adicionales CA

Damos ok y de nuevo nos va a pedir que ingresemos el password, para luego finalmente tener el certificado de la CA

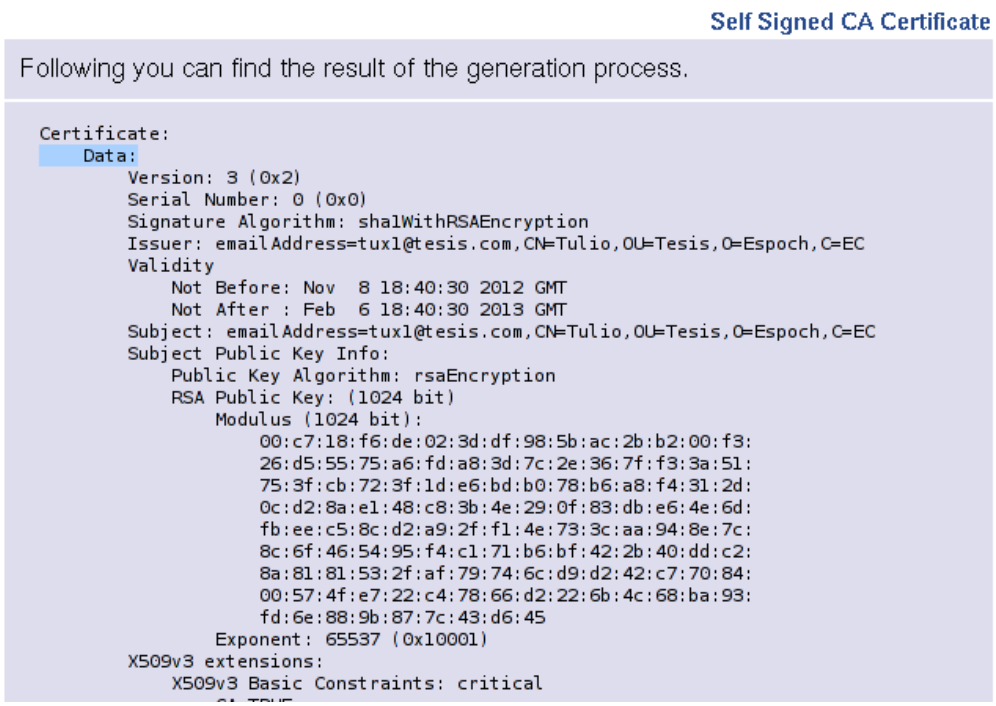

Figura IV.60 Certificado CA resumen

Ya tenemos Certificado Digital de la CA, ahora deberemos exportarla para que todos puedan obtenerlo Para esto secuencialmente realice los últimos 2 pasos de la sección FINAL SETUP.

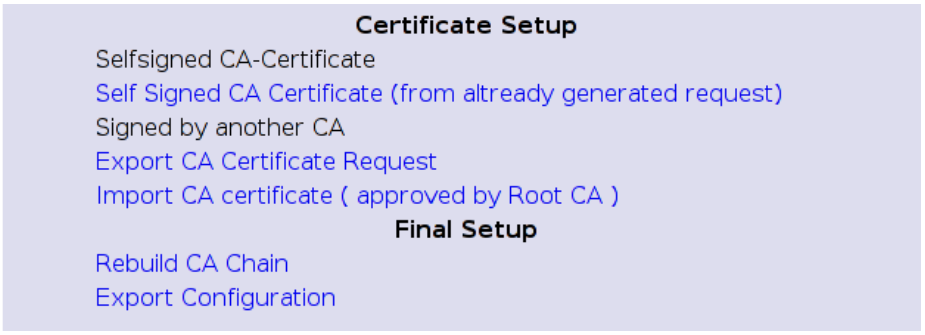

Figura IV.61 Final Setup

Damos clik en Rebuild CA Chain

Figura IV.62 Reconstrucción de la CA

Luego para exportar la configuración en Export Configuration

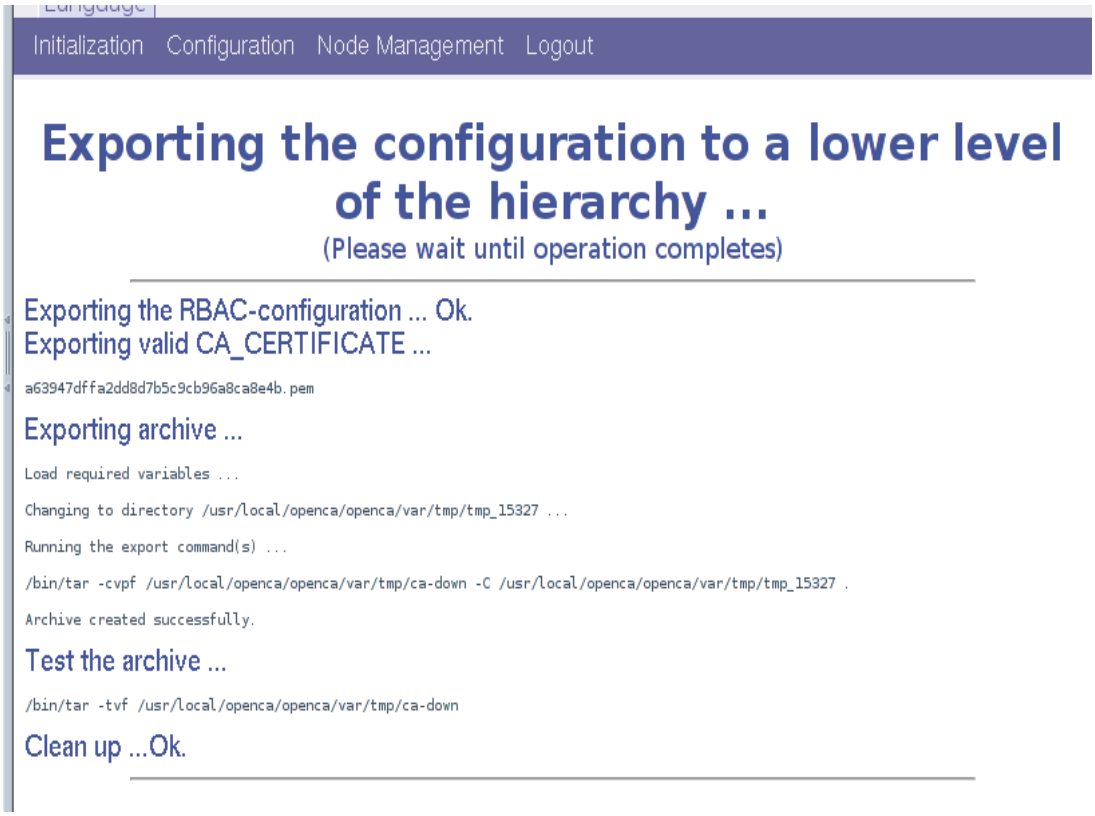

Figura IV.63 Exportación del certificado CA

Con el último paso hemos exportado desde la CA el Certificado de la misma. Pero recuerde que los usuarios no tienen acceso a la CA sino a la RA, por lo

tanto tenemos que importar en la RA el Certificado creado. Entonces seleccionamos OpenCA: ra-node y nos vamos a Dataexchange

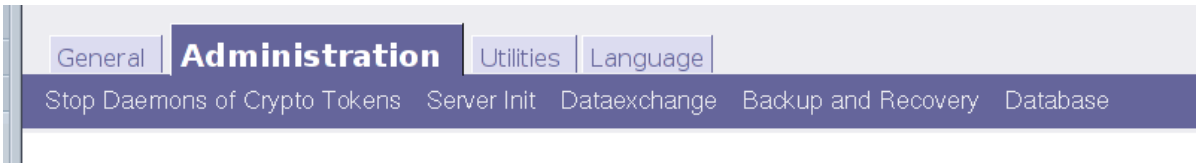

Figura IV.64 Dataexchange

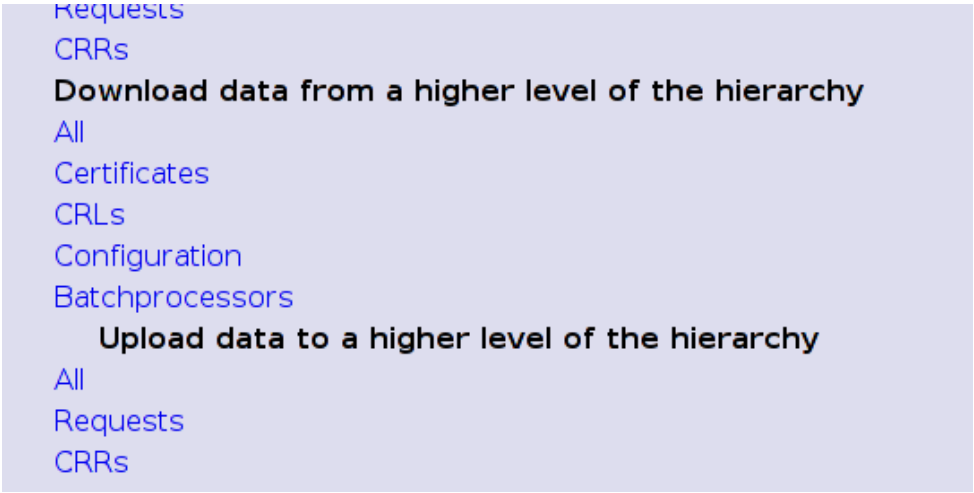

Figura IV.65 Descarga desde un nivel superior

Ahora ya podremos ir a la RA (pub) y descargar el certificado de la CA.

Figura IV.66 Obteniendo un certificado

Revisemos ahora en nuestro browser el certificado instalado.

Finalmente para la administración de los certificados en OpenCA se pueden administrar los certificados viendo cuales son validos, cuales han expirado para tener un completo control sobre los certificados

Figura IV.67 Certificados

CONCLUSIONES

- La seguridad informática es un punto esencial dentro de un grupo de trabajo, nos garantiza que los usuarios y la información que tenemos dentro de la empresa no sea vulnerada por malos usuarios

- Las herramientas se software libre son de gran utilidad, ya que proveen un gran servicio y son de gran calidad sin el costo de programas propietarios

- Se puede usar una maquina en Linux que puede servir como router, donde añadimos las rutas de las redes que necesitamos, claro esta es de utilidad en redes de no gran tamaño.

- El DNS es primordial dentro de una red, porque ayuda a las personas a recordar nombres más no direcciones y dentro de una red con muchas redes el trabajo de recordar tantos números se vuelve casi imposible

- Las herramientas del DNS nos permite trabajar en conjunto con las herramientas de DHCP, ya que posee una opción donde se puede aceptar las actualizaciones de los nombres de las maquinas que se integren al dominio

- El correo electrónico es una de los principales puntos vulnerables dentro de una red por este motivo siempre debemos tener cuidado y debemos configurar los servidores para que trabajen de manera segura mediante SSL

- El proxy se puede convertir en una gran herramienta de seguridad si no disponemos de certificados digitales para los usuarios, ya que nos permite trabajar pidiendo a los usuarios que se autentifiquen

- Se debe siempre cambiar los privilegios de modificación y acceso a los archivos de configuración a los que se puede acceder desde la red para que el único que pueda modificar los mismo sea solo el administrador

- Los certificados digitales es la forma de garantizar la autenticidad de los documentos electrónicos

- La PKI es grupo de herramientas que en conjunto sirven para garantizar la seguridad de donde se la emplee, no es de gran ayuda si cada servicio es independiente y trabaja por su cuenta

- Una PKI es una infraestructura y como tal, sólo produce beneficios cuando es utilizada. De hecho, son las aplicaciones quienes se conectan a la infraestructura, la utilizan y provocan beneficios.

RECOMENDACIONES

- Se recomienda a los administradores implementar siempre una PKI, aunque su costo sea alto el beneficio que ofrece a la seguridad de la red es mucho mayor

- Tener siempre mucho cuidado con la clave privada que posee el administrador, y no permitir el acceso desde la red a la carpeta que posee la misma

- Al momento de instalar los servidores tener en cuenta que algunos servicios necesitan tener antes instalados algunas librerías, debemos ahorrar el tiempo que nos va a tomar darnos cuenta del error por no tener instaladas las mismas

- Tener cuidado al momento de instalar apache que nos sirve para implementar una web, ya que si no instalamos el modulo de seguridad SSL para apache este se vuelve en un gran punto vulnerable a ataques

- Hay que sacar respaldo de los archivos de configuración antes de realizar algún cambio, de esta manera podemos corregir fácilmente algún error que se pueda cometer

- Siempre debe haber comunicación entre los encargados de brindar los servicios dentro de una red y la o las personas encargadas de la seguridad de la misma

RESUMEN

Este trabajo trata del análisis de la tecnología PKI (Infraestructura de Clave pública) y su implementación mediante el uso de software libre, para mejorar la seguridad en la transmisión de información, dentro de las aplicaciones www, email y ftp. Se utilizó el método Inductivo-Deductivo para la recolección y análisis de la información, a través de maquinas virtuales dentro de las cuales se configuraron los servicios de DNS, WEB, MAIL y el principal servidor encargado de los certificados digitales por medio de software libre, utilizando la distribución Centos, ya que por ser una herramienta Linux de alta calidad, proporciona estabilidad

Se comprobó que una infraestructura de clave por medio de todos los servidores efectuados, ayudaron a mejorar la seguridad en el envío de correos electrónicos por la firma electrónica de Sendmail, el uso de una WEB segura con el certificado X.509 y en el envío de paquetes de información con el protocolo SSL, todo esto usando la herramienta Open CA, ofrece seguridad para los usuarios al hacer uso de los servicios, dentro de la red

Como conclusión podemos decir, que una PKI es una infraestructura que cuando las aplicaciones se conectan a la infraestructura y se utilizan producen beneficios.

Se recomienda implementar una PKI a todos los administradores, en vista de que el costo de implementación es igual, al costo beneficio, ya que se administran la mayoría de servicios y brinda un entorno de trabajo confiable

SUMMARY

This work is about the analysis of the PKI technology (Infraestructura de Clave Publica) and its implementation through the use of free software to improve security in the transmission of information within www applications, email and ftp. It was used the inductive-deductive method for collection and analysis of information, through virtual machines within services are configured DNS, WEB, MAIL server and the main server incharged of digital certificates through free software, using the distribution Centos as being a high quality Linux tool, provides stability

It was checked, that a public key infrastructure using all servers effected, helped improve security in sending emails signature of Sendmail, the use of a secure web with the X.509 certificate and sending information packets with SSL, this tool using Open CA provides security for users to make use of services within the network.

Finally, we can say that a PKI is an infrastructure where applications connect to infrastructure and use, produce benefits.

It is recommended to implement a PKI for all administrators, given that the implementation cost is the same to the cost benefit, now that are managed services and provides a reliable working environment .

BIBLIOGRAFÍA

1. **INSTITUTO NACIONAL DE ADMINISTRACIÓN PÚBLICA.,** Firma Digital y Administraciones Publicas., 1a. ed., Madrid-España., Editorial Paraninfo., 2003., Pp. 107-119.

2. **AREITIOBERTOLÍN, J.,** Seguridad de la información: redes, informática y sistemas de información., 2a. ed., Madrid-España., Editorial Paraninfo., 2008., Pp. 368-382.

3. **JULIANVERON PIQUERO.,** Prácticas de Redes., 1a. ed., Caracas-Venezuela., Editorial Caceres., 2010., Pp. 238.

4. **ESPAÑA, M.,** Servicios Avanzados de Telecomunicación., 1a. ed., Madrid-España., Editorial Díaz de Santos, S.A., 2003., Pp. 57-70.

BIBLIOGRAFÍA DE INTERNET

5. CONCEPTOS DE SEGURIDAD EN REDES

http://www.iit.upcomillas.es/palacios/seguridad/

2012/06/25

6. SERVIDORES LINUX

http://www.alcancelibre.org/staticpages/index.php/manuales-indice

2012/09/10

http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento;jsessionid=8C33D8C73C585E3700C6ADC6EE6C6D67

2012/05/29

7. PKI

http://ospkibook.sourceforge.net/docs/OSPKI-2.4.7/OSPKI-html/ospki-book.htm

2012/07/15

http://www.ietf.org/html.charters/pkix-charter.html

2012/07/13

8. OPENSSL

http://csrc.nist.gov/cryptval/140-1/140sp/140sp733.pdf

2012/06/09

http://www.openssl.org

2012/06/09

9. APACHE

http://www.apache.org

2012/06/16

10. OPENCA

http://www.openca.org

2012/08/02

http://www.openca.org/~madwolf/ch03.html

2012/08/02

http://www.dartmouth.edu/~deploypki/CA/OpenCA-LiveCD.html

2012/08/04

http://www.eslared.org.ve/walcs/walc2011/material/track6/Pr%E1ctica%20de%20Laboratorio%20OPENCA.pdf

2012/08/05

http://wiki.cecalc.ula.ve/index.php/Instalaci%C3%B3n_y_configuraci%C3%B3n_de_la_CA

2012/08/06

http://es.scribd.com/doc/55488597/Instalacion-OpenCA

2012/08/10

WIRESHARK

http://seguridadyredes.nireblog.com/post/2008/02/14/analisis-de-red-con-wireshark-interpretando-los-datos

2012/10/28

Glosario

AC	Attribute Certificate
AES	Advanced Encryption Standard
CA	Certification Authority
CAPI	Cryptographic API
CP	Certificate Policy
CPS	Certification Practice Statements
CRL	Certificate Revocation List
DNI-e	Documento Nacional de Identidad Electrónico
EE	End Entity
EFS	Encrypting File System
HTTPS	Hypertext Transfer Protocol Secure (HTTPS)
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IPSec	Internet Protocol Security
ISO	International Organization for Standardization
IT	Information Technology
ITU	International Telecommunication Union
LDAP	Lightweight Directory Access Protocol
NSA	National Security Agency
PGP	Pretty Good Privacy
PKC	Public Key Cryptography
PKI	Public Key Infrastructure

PKIX	Public Key Infrastructure X.509
RA	Registration Authority
RSA	Rivest Sham
SPKI	Simple Publi
SSL	Secure Sockets Layer
TLS	Transport Layer Security