Administración de certificados de seguridad para procesos de red seguros (HTTPS, SSH, SFTP)

Autor: Pablo Adrián Herrera Pérez

Tesis presentada para obtener el título de:
Ing. En Sistemas Computacionales

UNIVERSIDAD VASCO DE QUIROGA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Administración de Certificados de Seguridad para Procesos de Red Seguros (HTTPS, SSH, SFTP)

TESINA

Que para obtener el titulo de

INGENIERO

EN

SISTEMAS COMPUTACIONALES

PRESENTA:

Pablo Adrián Herrera Pérez

Morelia, Michoacán, México
Noviembre del 2008

Dedicatoria

ÍNDICE GENERAL

JUSTIFICACIÓN

Hoy en día resulta indispensable el uso de internet debido a las ventajas que ofrece esta tecnología, tales como, acceso a una gran cantidad de servicios, como son, compras en línea, pago de servicio de televisión por cable, luz, agua, impuestos, transferencias y tramites bancarios, etc. Todos estos beneficios implican el uso de información personal, la cual, deberá permanecer integra y segura para evitar el robo de la misma y el no repudio, o dicho de otra forma, evitar la negación de identidad en la realización de determinado tramite, lo cual proporciona un alto grado de confiabilidad tanto para el usuario como para la entidad prestadora de algún determinado servicio.

Tal como se menciona en un artículo publicado por CNNExpansión.com el día 8 de mayo del 2008 titulado "Cuánto cuesta tu información en Internet"[1], en el estudio realizado por la firma líder en software de infraestructura y seguridad informática, Symantec, sobre Latinoamérica, el dato de una cuenta bancaria se cotizó entre los 10 y 1,000 dólares; el de una tarjeta de crédito entre los 0.40 a 20 dólares; una identidad completa entre uno y 15 dólares y las cuentas de subastas en línea oscilaron entre uno y 18 dólares, al menos hasta el cierre de abril de 2008.

En el reporte se destaca que el sector gubernamental es el que enfrenta mayor problema de identidades expuestas (60%), le sigue el sector financiero (33%) y el de las cadenas minoristas (8%); más de la mitad de las fugas (57%) de información se debieron a pérdidas o robos.

Esto da la pauta para el desarrollo de la presente investigación, mediante la cual se pretende asegurar los servicios brindados por un servidor, el cual permitirá proteger los servicios HTTPS, SSH y SFTP, haciendo uso de un certificado expedido por una Autoridad Certificadora reconocida a nivel mundial, proporcionando al usuario final la confianza de que su información permanecerá libre de amenazas.

Para llevar a buen termino este proyecto, cumpliendo con los alcances establecidos se cuenta con una serie de procedimientos que permitirán concluir exitosamente dicha investigación y de esta manera determinar la importancia que tiene el uso de certificados digitales, como una herramienta tecnológica para brindar de seguridad a los procesos que impliquen intercambio de datos personales y/o confidenciales con cualquier servidor web al momento de hacer uso de los servicios, antes mencionados.

OBJETIVOS

Objetivo general.
Implementar la seguridad en los procesos de red mediante el uso de certificados digitales en diversos ambientes de trabajo, en los cuales al realizar el intercambio de datos en una arquitectura de red cliente-servidor sin medidas de seguridad se expone dicha información.

Objetivos específicos.

- Creación de un certificado digital con el formato X.509, firmado por una Autoridad Certificadora reconocida a nivel mundial.
- Creación de un certificado digital en formato X.509, el cual será autofirmado.
- Creación de un par de llaves privada y pública, para asegurar los servicios SFTP y SSH.
- Instalación del certificado en un servidor de aplicaciones.
- Implementación de las llaves pública y privada en el servicio SSH, lo que permitirá la autenticación de un cliente en un servidor de manera confiable.
- Implementación del certificado en el servicio SFTP para la transferencia de archivos entre un servidor y los clientes.
- Implementación de una aplicación de comercio electrónico segura, la cual permitirá el intercambio de información de manera cifrada, brindando la posibilidad de que no sea interceptada por un tercero y este haga uso de la misma con fines de lucro, esto se llevará a cabo mediante la instalación del certificado en el servicio HTTPS.

Capítulo 1 INTRODUCCIÓN A LA SEGURIDAD DE LA INFORMACIÓN

1.1. Antecedentes.

La evidencia escrita más temprana de conceptos relacionados con la seguridad se encuentra en códigos legales, tales como el sumerio (3000 a.c.) o el de Hammurabi (2000 a.c.). Mas tarde, aparece en obras generalmente refiriéndose al arte de la guerra y gobierno.

Como dicen los antropólogos, las organizaciones sociales primitivas revelan un profundo conocimiento y sofisticada aplicación de los principios y funciones básicas de seguridad. Desde su nacimiento, las personas son instruidas, vía tradición y entrenamiento, y/o vía imitación, en las habilidades para la seguridad.

La evidencia de medidas de seguridad acompaña cada descubrimiento arqueológico. Cerraduras, puertas fuertes, ventanas selladas, trampas, cajas fuertes, sistemas de alarma, barreras físicas y escudos son conocidos y usados desde el principio de la civilización.

De acuerdo con la evidencia anterior, no existe duda de que los conceptos de alertar, evitar, detectar, alarmar y reaccionar son tan viejos como la vida misma, siendo una parte esencial de la pugna diaria por la vida, y están fundados en el instinto básico de supervivencia.

En 1919, un ingeniero de minas y teórico de la administración, Henry Fayol, identificó la seguridad como una de las necesidades fundamentales de la industria, y definió su objetivo:

"...salvaguardar propiedades y personas contra el robo, fuego, inundación, contrarrestar huelgas y felonías, y de forma amplia todos los disturbios sociales que puedan poner en peligro el progreso e incluso la vida del negocio. Es generalmente hablando, todas las medidas para conferir la requerida paz y tranquilidad al personal" [2].

Como se puede ver la filosofía de Fayol estaba enfocada a los bienes físicos, pues en ese entonces no había tiempo para preocuparse por el capital humano, el cual en estos tiempos resulta ser el eslabón más débil de una organización, ya que este hace uso de la información vital de la empresa.

La seguridad en la actualidad se ha convertido en una profesión que requiere de funciones especializadas. Existen sistemas de comunicación, biométricos, de detección de intrusos y tecnologías informáticas que han evolucionado el significado de la seguridad, que hasta los tiempos recientes estaba basado en armas, trampas, cerraduras, cajas fuertes, puertas blindadas y barrotes. Toda esta nueva gama de elementos, son ahora los nuevos ingredientes de los programas de seguridad. Los sistemas de seguridad son cada vez más automáticos, particularmente aquellos de detección y comunicación de siniestros, y en una extensión menor, aquellos relacionados con la valoración, la decisión y la reacción. Los avances en la miniaturización se reflejan en los equipos de seguridad que cada vez son más pequeños, más baratos, más fácilmente instalados y mantenidos, y más confiables. Pero todavía, en concepto de seguridad no se ha añadido ningún nuevo concepto a aquellos ya conocidos anteriormente.

El problema de la seguridad informática ha ido creciendo de manera proporcional a los avances tecnológicos en Hardware y Software. Durante las primeras dos décadas del uso de las computadoras, la seguridad no era tan importante. Las primeras computadoras eran usadas para el manejo de información con un valor relevante, tales como seguridad nacional y para aplicaciones comerciales. Pero el tamaño de las computadoras y la naturaleza de sus aplicaciones, permitían que cualquier problema de seguridad fuera resuelto fuera de la computadora. Si el sistema era para un usuario solo bastaba con recoger sus cintas y tarjetas así como limpiar la memoria del CPU cuando la tarea fuese terminada. Si la información era vital simplemente bastaba con cerrar con llave la habitación donde se encontraba la computadora. Básicamente el usuario tenía en sus manos la propia seguridad de su información así como de sus aplicaciones. La computadora por sí sola no era parte del problema de seguridad o la solución a la misma.

El término SEGURIDAD ha resurgido en estos tiempos, antes de que el problema de la seguridad de la información se publicará extensamente en los medios, en lo que se enfocaban las empresas era en la protección física de las computadoras. Tradicionalmente las instalaciones informáticas se han protegido físicamente por tres razones:

- Para prevenir el robo o daño del Hardware.
- Para prevenir el robo o daño de la información.
- Para prevenir la interrupción del servicio.

El uso de nuevas tecnologías en Hardware y Software esta ofreciendo un nuevo campo de acción para conductas antisociales, delictivas y destructivas de formas inimaginables, vulnerando la seguridad de la información.

Pero la seguridad informática no es solo el conjunto de herramientas tecnológicas que existen en el mercado, ya que el eslabón más débil en una organización es el usuario que hace uso de estos medios para llevar a cabo su labor diaria. La cual se ve vulnerada mediante ataques de ingeniería social, a la mala relación que exista entre los miembros de una organización que pudieran generar el descontento de sus empleados y así llevar a cabo la acción de proporcionar información sensible a terceros que pudieran lucrar con dicha información y así perder la continuidad del negocio.

1.2. Fundamentos de Seguridad de Información.

1.2.1. Información.

La información es el conjunto de caracteres, o elementos que representan una idea, algo que puede ser tan tangible como un oficio en papel o tan intangible como podría ser el flujo de datos por medio de una red, cuya función es permitir la comunicación y realizar la ejecución de dicha información. Entonces se podría manejar el concepto de información dicho de esta forma:

Por información puede entenderse, con carácter general, un conjunto de símbolos que representan hechos, objetos o ideas, (en definitiva son datos,

entendiendo este término en sentido amplio, que nos aporta algún conocimiento) [3].

1.2.2. Informática.

Según la definición de la Real Academia Española, la palabra informática significa: Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras [4].

La informática surge en la preocupación del ser humano por encontrar maneras de realizar operaciones matemáticas de forma cada vez más rápida y más fácilmente. Pronto se vio que con ayuda de aparatos y máquinas las operaciones podían realizarse de forma más rápida y automática.

1.2.3. Seguridad.

Con respecto al término de SEGURIDAD, una definición que tal vez en ámbitos tecnológicos no se vea reflejada pero se puede partir del concepto en crudo. Por esta razón se puede mencionar que la seguridad es una necesidad básica de la persona y de los grupos humanos y al mismo tiempo un derecho inalienable del hombre y de las naciones. Seguridad proviene del latín SECURITAS, que a su vez se deriva del adjetivo SECURUS; implica las nociones de garantía, protección, tranquilidad, confianza, prevención, previsión, preservación, defensa, control, paz y estabilidad de las personas y grupos sociales, frente a amenazas o presiones que atenten contra su integridad [5].

1.2.4. Seguridad de la Información.

La información permite tomar decisiones importantes en los procesos cognoscitivos y sociales del ser humano. Dicha información podría ser sensible, al mismo tiempo debe estar al alcance de quien le pueda dar un buen uso y fuera del alcance de quien pueda darle un mal uso; de ahí que surja la necesidad de proteger la información.

En pocas palabras es la protección de la confiabilidad, integridad y disponibilidad de la información [6].

1.2.5. Confidencialidad.

Nos dice que los objetos de un sistema han de ser accedidos únicamente por elementos autorizados a ello, y que esos elementos autorizados no van a convertir esa información en disponible para otras entidades [7].

1.2.6. Integridad.

Significa que los objetos sólo pueden ser modificados por elementos autorizados, y de una manera controlada. Se refiere al hecho de que los métodos que gestionan la información garantizan un tratamiento sin errores de la misma. La información no debe cambiar mientras se está transfiriendo o almacenando, y nadie puede modificar el contenido de la información o los archivos y aún menos eliminarlos.

Para garantizar la integridad de la información, el remitente debe estar siempre autentificado. La combinación de autenticación e integridad garantiza que la información enviada llega a su destinatario exactamente en la misma forma en que se envió [7].

1.2.7. Autentificación.

La autentificación se utiliza para asegurarse de que las partes involucradas son quienes dicen ser. Por ejemplo, en el comercio electrónico, al hacer trámites con las autoridades o en la comunicación entre personas, es especialmente importante saber quién es la otra parte. Es necesario autentificar tanto el origen de las partes como la fuente de información [7].

1.2.8. Disponibilidad.

Asegurar que los usuarios que tienen acceso autorizado a la información y sus activos asociados tengan acceso a los mismos cuando se requiera. Es la capacidad de tener acceso a un recurso en específico, dentro de un marco de tiempo específico, en base a lo especificado en las tecnologías de información del recurso [7].

1.3. Tipos de Ataque.

1.3.1. Gusanos

Los gusanos tienen ciertas similitudes con los virus informáticos, pero también diferencias fundamentales. Un gusano se parece a un virus en que su principal función es reproducirse, pero por el contrario de cómo lo hacen los virus, en lugar de copiarse dentro de otros archivos, un gusano crea nuevas copias de si mismo para replicarse.

En síntesis, lo que caracteriza a un gusano es que para reproducirse genera nuevas copias de si mismo dentro del mismo sistema infectado o en otros sistemas remotos, a través de algún medio de comunicación, como bien puede ser Internet o una red informática [8].

1.3.2. Virus

Se trata de una rutina o programa capaz de infectar otros archivos ejecutables, como los *.EXE, .COM* y *.SCR* bajo Windows. Para infectar otros archivos ejecutables, estos programas copian su contenido dentro de ellos, y se modifican de manera que, cuando el archivo sea abierto por el usuario, o automáticamente si se trata de un proceso, el propio virus también se ejecute.

Los primeros virus eran de este tipo, y aún hoy en día son de los más peligrosos, dado que su presencia muchas veces no puede ser detectada si no se cuenta con un antivirus actualizado ya que se esconden dentro de programas normales ya existentes en el sistema [9].

1.3.3. Troyano

Programas que, enmascarados de alguna forma como un juego o similar, buscan hacer creer al usuario que son inofensivos, para realizar acciones maliciosas en su equipo.

Estos troyanos no son virus ni gusanos dado que no tienen capacidad para replicarse por si mismos, pero en muchos casos, los virus y gusanos liberan troyanos en los sistemas que infectan para que cumplan funciones especificas,

como, por ejemplo, capturar todo lo que el usuario ingresa por teclado (*keylogger*).

La principal utilización de los troyanos es para obtener acceso remoto a un sistema infectado a través de una puerta trasera [9].

1.3.4. Negación de servicio (Denial Of Service)

Los ataques de negación de servicio requieren generalmente del poder de una red de computadoras que trabajen simultáneamente, hacen que se deshabiliten o desconecten de la red y logran que los servicios no estén disponibles. Estos ataques pueden destruir los servicios del servidor dejando de operar o pueden interrumpir sistemas críticos. Un ataque de negación de servicio ocurre cuando los recursos de la red son tomados por un individuo no autorizado, típicamente el ataque es realizado a los servidores. Esta acción aumenta significativamente el tráfico en la red abrumando los servidores y haciéndolos imposibles para que los usuarios legítimos introduzcan o lean información [10].

1.3.5. Suplantación de identidad (Spoofing)

Es el proceso donde una persona asume la identidad de otra. El acceso físico o electrónico a las terminales se requiere de la identificación de un usuario, la verificación se basa en una cierta combinación, algo que el usuario sabe (ejemplo: una claves de acceso secreto), o algo que el usuario es (una característica fisiológica, tal como huella digital, geometría de la mano o la voz) y algo que el usuario posee, (ejemplo, una llave magnética, una tarjeta), etcétera. Cualquier persona con la combinación correcta de las características de la identificación puede personificar a otro individuo [10].

Este tipo de ataque es el que se utiliza con fines fraudulentos o de investigación y uno de los medios para mitigar dichos ataques es mediante la implementación de certificados digitales (hablando en un ambiente Web), ya que estos se encuentran instalados en el servidor, evitando así la copia, emulación, o suplantación en algún otro servidor desconocido. En este caso el cliente no necesita realizar ninguna operación compleja, basta simplemente

con verificar el certificado de seguridad del sitio, con la finalidad de tener la certeza de que este sitio es quien dice ser.

1.4. Estadísticas

Para tener una idea más clara de lo que se está presentando en la actualidad, se hace referencia a la encuesta realizada por el CSI (Computer Security Institute) correspondiente al año 2007; dicho organismo promueve y fomenta en los Estados Unidos la cultura de la Seguridad Informática. Con bases en la encuesta, este organismo nos permite conocer el estado actual de la seguridad en diferentes ámbitos (empresas públicas, privadas, universidades, etc.) de distinto tamaño que tienen en común el uso de las tecnologías de información.

El informe se realizó en base a la respuesta de 494 empresas, cubriendo un amplio espectro desde agencias de gobierno, instituciones, universidades y sector tecnológico de EE.UU. El perfil de las personas que respondieron la encuesta corresponde a CIOs, CEOs, CSOs, CISOs, Oficiales de Seguridad y Administradores de sistemas.

Como se puede observar en la figura 1.1, referente a los incidentes reportados durante el año. Se puede observar que los encuestados al ser consultados sobre si la organización experimentó incidentes de seguridad durante el año, la cantidad de encuestados se iguala al responder que sí un 46% y que no un 45% (el resto no sabe/no contesta).

Mientras que la figura 1.2 arroja datos que no siguen un patrón ordinario, ya que se puede suponer que los ataques que corresponden al parámetro de número de incidentes de 1 a 5 han disminuido en un 7% aproximadamente, pero contrasta con los ataques referentes a mayores a 10, que han aumentado en un 26%. En general estos números podrían considerarse buenos debido a que los incidentes han decrecido considerablemente (un 25%) en los últimos 4 años.

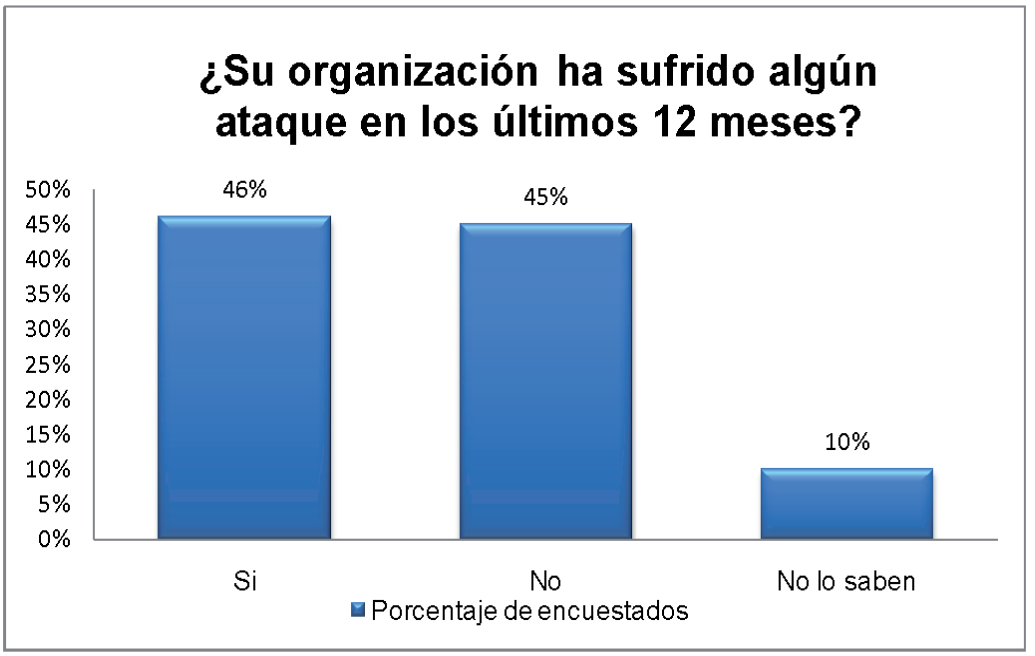

Figura 1.1 Estadísticas de ataques en los últimos 12 meses.

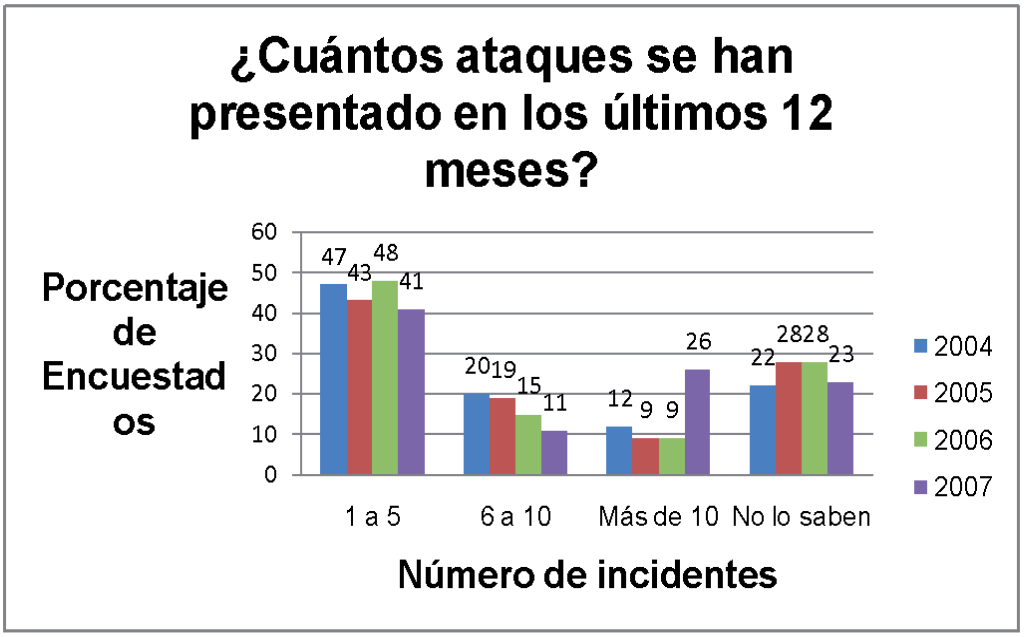

Figura 1.2 Número de incidentes presentados en los últimos 12 meses.

Capítulo 2 SERVICIOS DE RED SEGUROS (SSH, SFTP, HTTPS)

2.1. SSH (SecureSHell)

SSH (SecureSHell) es la herramienta mediante la cual se reemplazan los programas (RSH, RCP y FTP) que fueron utilizados anteriormente para llevar a cabo el inicio de sesión remoto, la copia y transmisión de archivos en un servidor, mismos que no cifraban los datos de los usuarios al hacer uso de los servicios antes mencionados. SSH cumple con la misma finalidad de brindar acceso como cliente a un servidor en modo consola utilizando métodos de autenticación por clave pública para establecer una conexión cifrada segura entre el cliente y el servidor, para poder realizar diversas tareas que van desde las operaciones básicas con archivos o directorios (crear, modificar, consultar, eliminar), incluso gestionar bases de datos, crear escritorios remotos a otros equipos, realizar túneles para redes privadas virtuales (VPN), es decir nos permite tener el control del equipo de manera remota al cual se realiza la conexión, pero con la gran particularidad de que todo se hace mediante una conexión cifrada no como se venía haciendo con RSH, TELNET, RCP, FTP.

Existen dos variedades de SSH actualmente (versión 1 y versión 2). La versión 1 de SSH hace uso de muchos algoritmos de cifrado patentados (sin embargo, algunas de estas patentes han expirado) y es vulnerable a un hueco de seguridad que potencialmente permite a un intruso insertar datos en la corriente de comunicación. La suite OpenSSH bajo Red Hat Enterprise Linux utiliza por defecto la versión 2 de SSH, la cual tiene un algoritmo de intercambio de llaves mejorado que no es vulnerable al hueco de seguridad en la versión 1. Sin embargo, la suite OpenSSH también soporta las conexiones de la versión 1.

SSH es una aplicación cliente servidor que utiliza el puerto 22 de manera predeterminada de TCP, cabe mencionar el concepto propio por parte de una comunidad trascendente como lo es Ubuntu:

SSH (acrónimo del inglés Secure SHell, cuya traducción sería intérprete de comandos seguro) es un protocolo de comunicación para controlar un equipo en remoto a través de una CLI (Command Line Interface -Interfaz de Línea de Comandos- también llamada: shell). Sirve para realizar una conexión con un equipo ante el cual no se encuentra la persona físicamente, bien porque está en una sala de servidores, bien porque no tiene teclado ni pantalla, por ejemplo los que están apilados en un rack.

Es parecido a Telnet, con la gran diferencia de que en el caso de SSH, la información viaja codificada con lo cual es muchísimo más segura, en el caso de conectarnos a un equipo que esté en nuestra LAN no es tan importante, pero si se conecta a través de Internet es fundamental, casi imprescindible, usar un protocolo seguro como SSH [11].

El protocolo SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- Previene ataques hechos mediante IP-Spoofing.
- Utiliza varios algoritmos (RSA para la llave de intercambio e IDEA, DES o triple DES para el cifrado de la sesión)
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio de cifrado de 128 bits, lo cual los hace extremamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de reenviar aplicaciones X11 (X11 se refiere al sistema de visión por ventanas X11R6.7, tradicionalmente llamado Sistema de ventanas X o simplemente X) desde el servidor. Esta técnica, llamada reenvío por X11, proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

SSH cifra todo lo que envía y recibe, se puede usar para tornar seguros los protocolos inseguros. El servidor SSH puede crear túneles para convertir en seguros los protocolos inseguros mediante el uso de una técnica llamada

reenvío por puerto, como por ejemplo POP, incrementando la seguridad del sistema en general y de los datos.

Tabla 2.1 Programas que vienen con la distribución.

Programa	Descripción
sshd	Es el servidor propiamente dicho, escucha a la espera de conexiones.
ssh	Es el cliente, con él se puede conectar a un servidor sshd así como ejecutar comandos.
scp	Copia archivos con seguridad entre hosts. (Sustituto ideal de rcp).
ssh-keygen	Usado para crear RSA keys (host keys y user authentication keys).
ssh-agent	Agente de autenticación. (Usado para manejar RSA keys en la autenticación).
ssh-add	Se usa para añadir nuevas llaves con el agente.
sftp	Subsistema para transferencia segura de archivos.
make-ssh-known-hosts	Usado para crear el archivo /etc/ssh_known_hosts

Este documento se basa en OpenSSH que es la implementación del protocolo SSH de OpenBSD. OpenSSH está disponible para muchas de las distribuciones de Linux y UNIX. De igual forma existen clientes de SSH para Windows como Putty o SecureCRT. [12]

2.2. Configuración básica de SSH.

2.2.1. Archivo sshd_config.

El demonio sshd es el programa que espera conexiones de red de los clientes SSH, controla la autenticación y ejecuta el comando requerido. El puerto por defecto al que escucha es el 22 y el archivo de configuración es */etc/ssh/sshd_config*. Es en este archivo en donde se describe la configuración del servidor SSH. Se analizarán las opciones más importantes:

- **Port:** Esta opción permite especificar el puerto TCP que utilizara el servidor. El valor usual es el 22. Por razones de seguridad es recomendable utilizar algún puerto libre y no el default.

- **Protocol:** Versión del protocolo a utilizar. Se usará solamente el valor 2.
- **HostKey:** Clave privada de RSA o DSA del host. Normalmente las claves de host son generadas en el momento de la instalación de OpenSSH y el valor de esta opción es */etc/ssh/ssh_host_rsa_key*.
- **PubkeyAuthentication:** Autenticación por clave pública. Si el valor de esta opción es **yes**, entonces se permite la autenticación de usuarios mediante clave pública.
- **ListenAdresses:** Es para configurar la dirección IP asignada a alguna interfaz de red en la que el servidor ssh escuchará peticiones, si el servidor tiene varias interfaces de red con diferentes direcciones IP asignadas entonces por default el servidor sshd escuchara peticiones en todas ellas.
- **AuthorizedKeysFile:** Mediante esta opción se indica al servidor en donde están almacenadas las claves públicas de los usuarios. El valor por defecto es *%home%/.ssh/authorized_keys,* esto significa que deben buscarse en el archivo **authorized_keys**, en el directorio **.ssh** del directorio **home** del usuario.
- **PasswordAuthentication:** Permitir la autenticación por password. Si el valor de esta opción es **yes**, se permite la autenticación de usuarios mediante contraseñas.
- **X11Forwarding:** Establece si se permite la ejecución remota de aplicaciones gráficas. Si se va a acceder hacia el servidor desde red local, este parámetro puede quedarse con el valor yes. Si el valor de esta opción es **yes**, se habilita el reenvió de X11 a través de la conexión SSH. Si se va a permitir el acceso hacia el servidor desde redes públicas, resultará prudente utilizar este parámetro con el valor no.
- **AllowGroups:** Grupos de acceso que pueden acceder a SSH, separados mediante comas.
- **AllowUsers:** Usuarios de acceso que pueden acceder a SSH, separados mediante comas.
- **DenyGroups:** Grupos con acceso denegado, separados mediante comas.

- **PermitRootLogin:** Especifica si el usuario root puede hacer uso de SSH, el valor por defecto es **no.**

Configuración por default del archivo sshd_config.

- Port 22
- Protocol 2
- HostKey /etc/ssh/ssh_host_key
- PubkeyAuthentication yes
- ListenAddress 192.168.1.1
- AuthorizedKeysFile .ssh/authorized_keys
- PasswordAuthentication no
- X11Forwarding no
- AllowGroups aadministradores
- AllowUsers admin
- DenyGroups operadores
- PermitRootLogin no

2.2.2. Archivo ssh_config.

En este archivo se describe la configuración del cliente SSH. Las opciones más importantes son:

- **Host:** Esta opción actúa como un divisor de sección. Puede repetirse varias veces en el archivo de configuración. Su valor, que puede ser una lista de patrones, determina que las opciones siguientes sean aplicadas a las conexiones realizadas a los hosts en cuestión. El valor * significa todos los hosts.
- **Port:** Es el puerto de TCP que utiliza el servidor (por default es el 22).
- **Protocol:** Es la versión del protocolo utilizada (por seguridad se tiene que utilizar el 2).
- **IdentityFile:** Archivo que contiene la clave pública (en caso de usar RSA, lo cual es ~/.ssh/id_rsa).
- **PasswordAuthentication:** Autenticación mediante contraseñas (**yes** o **no**).

- **StrictHostKeyChecking:** Define que hará el cliente al conectarse a un host del cual no se dispone de su clave pública. El valor **no** hace que sea rechazada la clave del servidor (y por lo tanto, se aborte la conexión), el valor **yes** hace que se acepte automáticamente la clave recibida, y el valor **ask** hace que pida confirmación al usuario.
- **Cipher:** Algoritmos de cifrado simétrico soportados pasa su uso durante la sesión.
- **ForwardX11:** Reenvió de aplicaciones X11 (los posibles valores son **yes** o **no**).

Configuración por default del archivo ssh_config.

- Host *
- Port 22
- Protocol 2
- IdentityFile ~/.ssh/id_rsa
- PasswordAuthentication yes
- StrictHostKeyChecking no
- Cipher blowfish
- ForwardX11 no

2.2.3. Otros archivos.

Estos archivos son algunos de los que también se utilizan por OpenSSH. Para la variable $HOME se debe interpretar como el directorio HOME del usuario en cuestión.

- **/etc/ssh/ssh_host_rsa_key:** Es la clave privada de RSA del host, la cual tiene permiso de lectura para el usuario root.
- **/etc/ssh/ssh_host_rsa_key.pub:** Clave pública de RSA del host, la cual tiene permiso de lectura para todos los usuarios.
- **/etc/ssh_know_hosts2:** Claves públicas de hosts conocidos del sistema.

- **$HOME/.ssh/config:** Configuración del cliente de SSH para cada usuario. Su contenido es similar al archivo de configuración global /etc/ssh/ssh_config.
- **$HOME/.ssh/id_rsa:** Clave privada de RSA del usuario, con permiso de lectura tan solo para el usuario en cuestión.
- **$HOME/.ssh/id_rsa.pub:** Es la clave pública del usuario en cuestión.
- **$HOME/.ssh/know_hosts2:** Claves públicas de hosts conocidos por el usuario.
- **$HOME/.ssh/authorized_keys2:** Claves públicas del usuario para la autenticación del mismo. Este archivo debe estar en el servidor a conectar.

2.2.4. Servidor SSH y cliente SSH.

El demonio incluido en *OpenSSH* se llama sshd y usualmente está localizado en el directorio /usr/sbin o en /usr/local/sbin. La configuración como ya vimos esta dada por el archivo *sshd_config.* Lo más común es que este servicio se ejecute al iniciar el sistema a través de init.

Mientras que el cliente incluido en *OpenSSH* se llama SSH y por lo general está ubicado en el directorio /usr/bin o en este otro /usr/local/bin/.

2.2.5. Secuencia de eventos al realizar una conexión SSH.

La siguiente serie de eventos lo ayudan a proteger la integridad de la comunicación SSH entre dos host.

- Se lleva a cabo un handshake (apretón de manos) cifrado para que el cliente pueda verificar que se está comunicando con el servidor correcto.
- La capa de transporte de la conexión entre el cliente y la máquina remota es cifrada mediante un código simétrico.
- El cliente se autentica ante el servidor.
- El cliente remoto interactúa con la máquina remota a través de la conexión cifrada.

2.2.6. Comandos más comunes en SSH.

Tabla 2.2 Comandos de navegación.

pwd	Muestra el path completo del directorio en el que se encuentra.
cd	Cambia de directorio, por ejemplo cd directorio/subdirectorio.
cd ~	Lleva a su directorio home.
cd -	Lleva al último directorio en el que estuvo.
cd ..	Sube a un directorio superior.

Tabla 2.3 Listado de archivos.

ls	Lista archivos y directorios de un directorio.
ls –al	Lista archivos y directorios e información sobre los mismos.
ls –aR	Lista archivos e información incluyendo todos los subdirectorios.
ls -aR | more	Lista archivos e información incluyendo todos los subdirectorios por pantallas.
ls -alR > resultado.txt	Lista archivos e información de subdirectorios y lo guarda en un archivo.
cat resultado.txt	Mostraría en pantalla el contenido del archivo.
*ls *.html*	Lista todos los archivos acabados en .html
ls -al directorio/subdirectorio/	Lista archivos e información de ese subdirectorio.

Tabla 2.4 Crear, editar o eliminar archivos y directorios

pico /home/usuario/public_html/index.html	Edita el archivo index.html con el editor pico.
touch /home/usuario/public_html/404.html	Crea el archivo vacio 404.html en ese directorio.
rm archivo.txt	Elimina archivo.txt
rm -rf directorio/	Elimina el directorio indicado, los subdirectorios y todos sus archivos.
mkdir descargas	Crea un directorio llamado descargas.

rmdir descargas	Elimina el directorio llamado descargas.

Tabla 2.5 Otros comandos SSH

cp -a /home/usuario/public_html/origen/* /home/usuario/public_html/destino/	Copia todos los archivos de un directorio a otro manteniendo sus respectivos permisos.
du –sh	Muestra es espacio total ocupado por el directorio en el que se encuentra.
du -sh *	Muestra el espacio ocupado de cada archivo y directorio.
whoami	Muestra su nombre de usuario.

2.2.7. Algunos ejemplos de utilización del servicio SSH.

Iniciando una sesión remota con contraseña.

El primer ejercicio consta de iniciar una sesión remota al servidor 10.185.75.253 (puede ser sustituido por la ip del servidor destino) con el usuario rovskyhp para lo cual se utilizará el cliente para Windows SSH Secure Shell utilizando el puerto por default número 22.

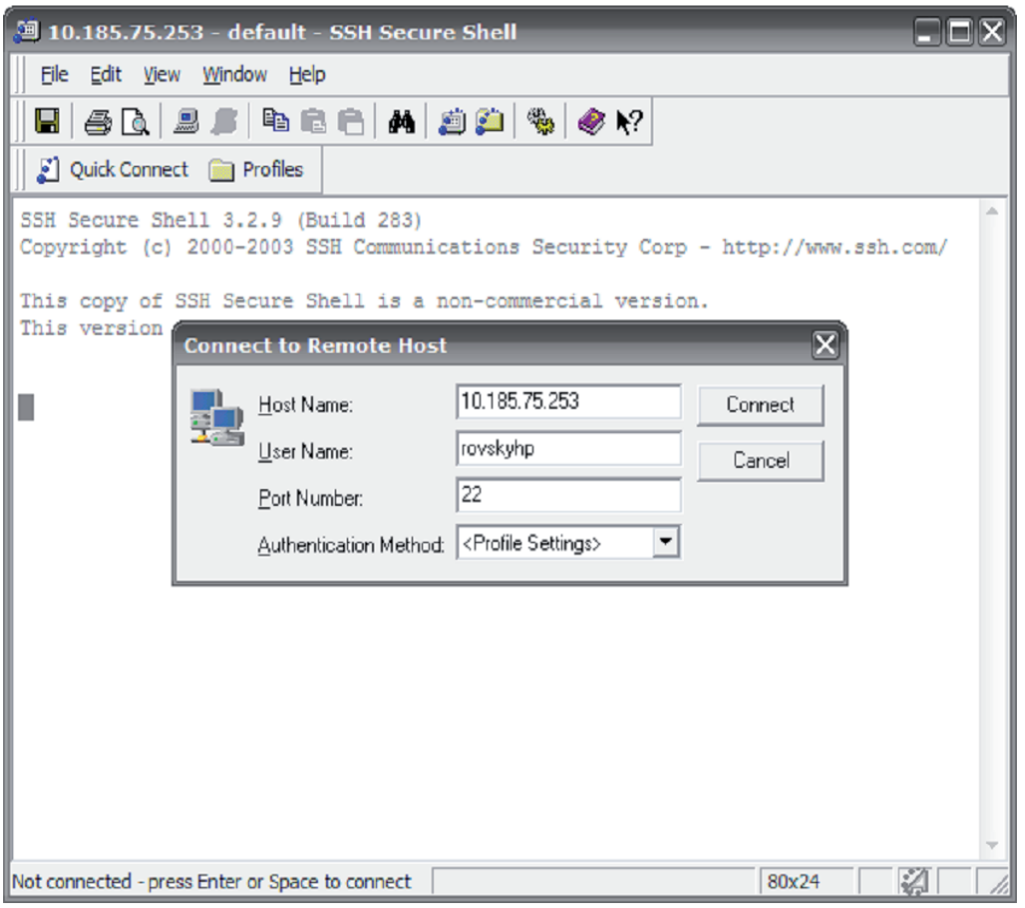

Figura 2.1 Iniciar sesión en un servidor mediante un cliente para Windows

Este ejemplo también puede ser realizado mediante comando de la siguiente forma:

```
ssh rovskyhp@10.185.75.253
```

La salida en pantalla del ejemplo al ser la primera vez que se realiza la conexión a este servidor será:

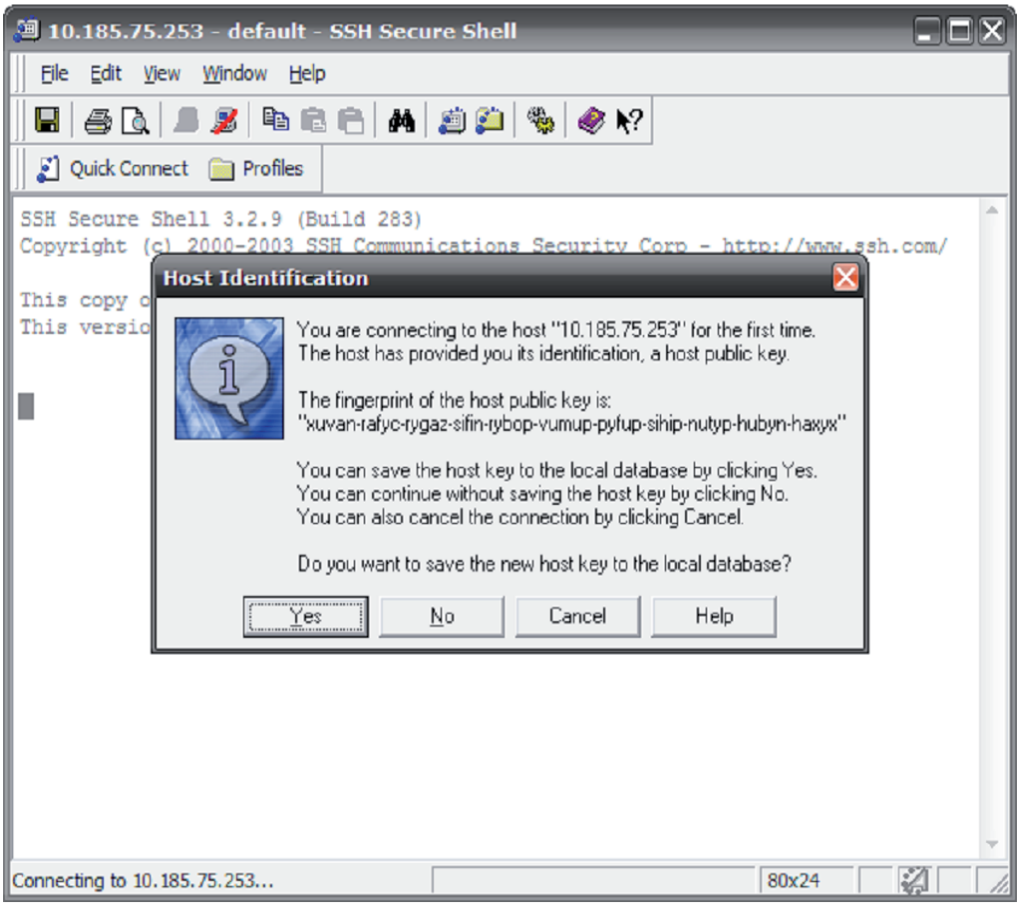

Figura 2.2 Agregar la llave pública del servidor.

En esta etapa se debe de dar clic en el botón *Yes* para agregar el servidor a los hosts conocidos.

Si se realizó mediante línea de comandos se muestra el siguiente mensaje en pantalla:

```
The authenticity of host ' 10.185.75.253 (10.185.75.253)' can't be established.
RSA key fingerprint is d7:7d:c9:b4:09:56:50:bd:9e:78:b8:93:8c:8d:ed:5c.
Are you sure you want to continue connecting (yes/no)? yes
```

Para lo cual se deberá responder tecleando la palabra yes, así se agrega dicho servidor a los hosts conocidos. Esta salida de pantalla no se presentaría si previamente se ha agregado la clave pública del mismo en $HOME/.ssh/know_hosts2.

```
Warning: Permanently added 'servidor.remoto,10.185.75.253 (RSA) to the list of known hosts.
```

Entonces el cliente solicitará el ingreso de la contraseña:

```
rovskyhp@10.185.75.253's password:
```

Finalmente se deberá ingresar la contraseña de dicho usuario y se habrá iniciado una sesión en el servidor 10.185.75.253 con el usuario rovskyhp. Esto se confirma con el mensaje de bienvenida del servidor remoto y mostrando el puntero del prompt.

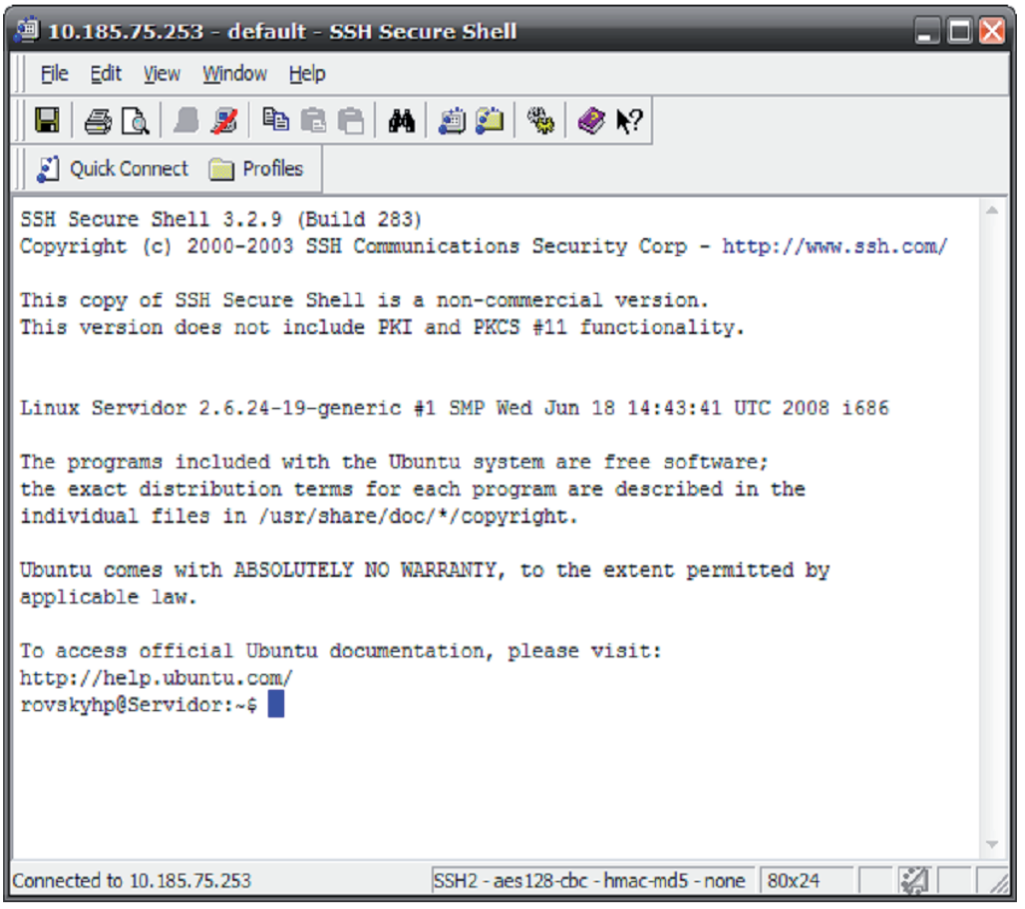

Figura 2.3 Acceso exitoso al servidor mediante el servicio SSH.

```
rovskyhp @10.185.75.253:~$
```

Transferencia de archivos a través de SCP.

SCP (Secure Copy, o Copia Segura) es un protocolo seguro para transferir archivos o directorios entre un host local y otro remoto, a través de SSH. Básicamente, es idéntico a RCP (Remote Copy o Copia Remota), con la

diferencia de que los datos son cifrados durante la transferencia para evitar la extracción potencial de información a través de programas de captura de las tramas de red (sniffers). SCP solo implementa la transferencia de archivos, pues la autenticación requerida es realizada a través de SSH.

El comando SCP permite copiar archivos entre dos máquinas. Utiliza SSH para la transmisión de la información, por lo que ofrece la misma seguridad. De igual forma utiliza los mismos métodos de autenticación.
Para realizar la transferencia de archivos entre dos equipos es necesario conocer las rutas de los directorios origen y destino. Enseguida se muestran las opciones más utilizadas con el comando SCP.

-p Preserva el tiempo de modificación, tiempos de acceso y los modos del archivo original.

-P Específica el puerto para realizar la conexión.

-r Copia recursivamente los directorios especificados.

En el siguiente ejemplo se transferirá un archivo llamado algo.txt preservando tiempos y modos hacia el directorio de inicio del usuario *rovskyhp* en el servidor *192.168.2.186.*

```
notroot@ubuntu:~$  scp -p algo.txt rovskyhp@192.168.2.186:~/
```

Como se puede observar en la figura 2.4, para la transferencia del archivo al utilizar el comando scp, primero alerta sobre la autenticidad del servidor, en cuyo caso se deberá agregar a los hosts conocidos y verificar la autenticidad del servidor.

Además se observa que solicita la contraseña del usuario en el servidor remoto, para lo cual se debe tener conocimiento de un usuario en ese servidor para realizar la operación de copia segura de archivos.

En la figura 2.5 se observa que la ejecución del comando se llevo a cabo con éxito.

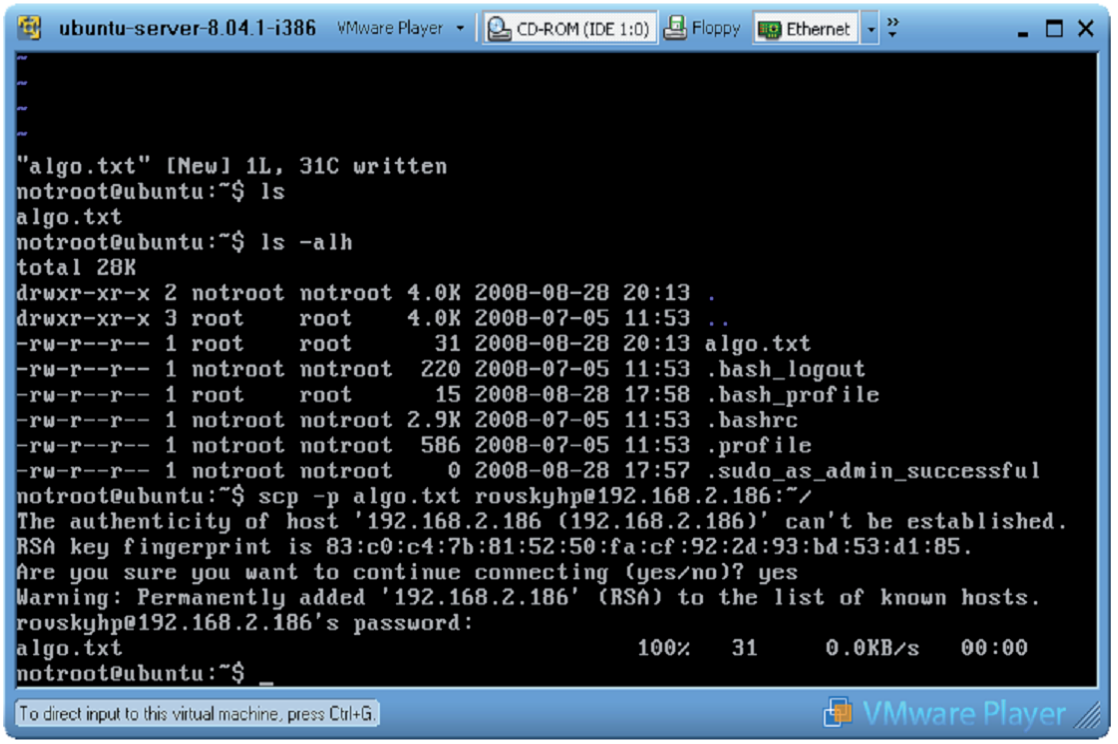

Figura 2.4 Copia segura del archivo archivo.txt a un servidor remoto.

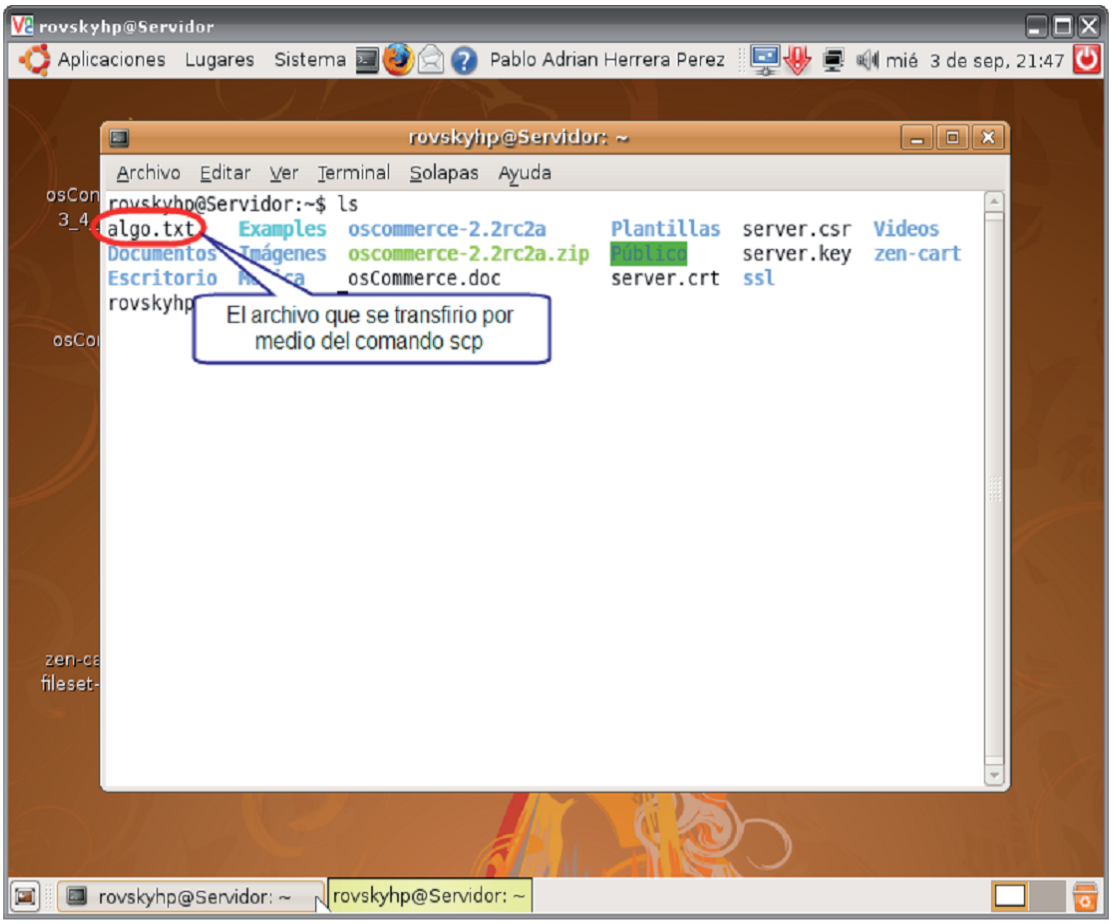

Figura 2.5 Archivo copiado en el home del usuario rovskyhp en el servidor remoto.

El ejemplo siguiente será transferir un directorio y todo su contenido preservando tiempos y modos hacia el directorio de inicio del usuario *rovskyhp* en el servidor *192.168.2.186.*

```
usuario_local@mi_pc:~>  scp -rp Directorio usuario_remoto_1@servidor_remoto_1:~/
```

Como se observa en la figura 2.6 se han transferido el directorio y todo su contenido al directorio de inicio del usuario *rovskyhp* en el servidor remoto *192.168.2.186* satisfactoriamente.

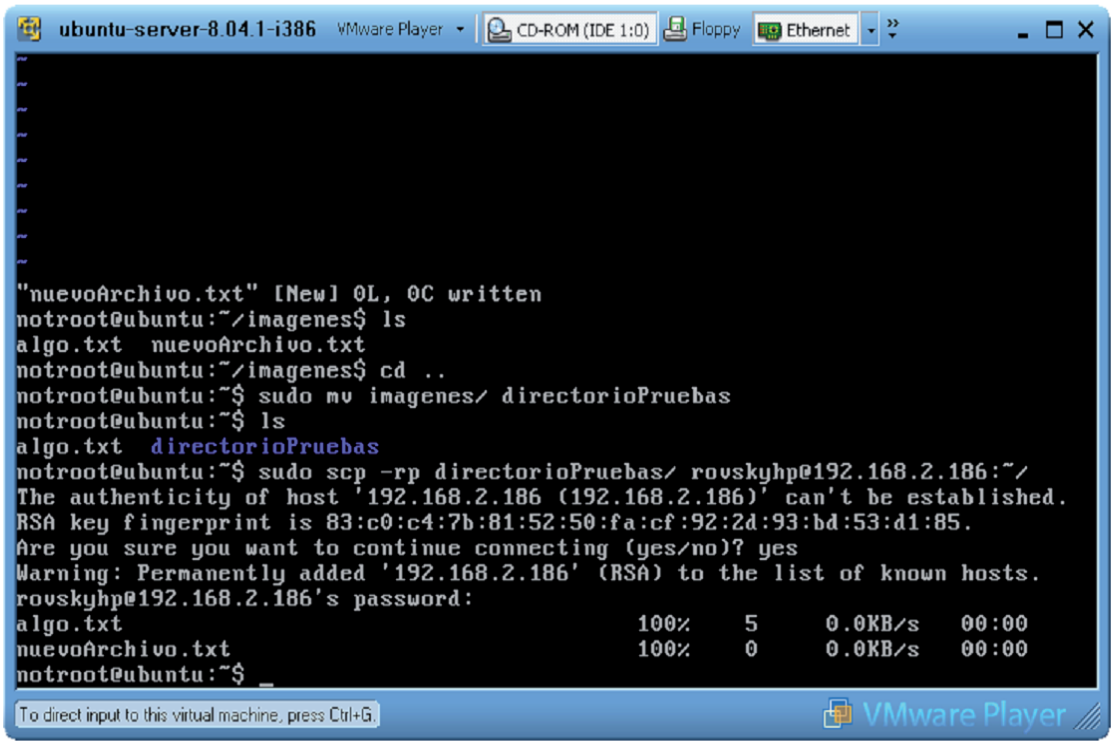

Figura 2.6 Copia segura del directorio y su contenido a un servidor remoto.

En la figura 2.7 se observa que la transferencia se realizo sin ningún problema.

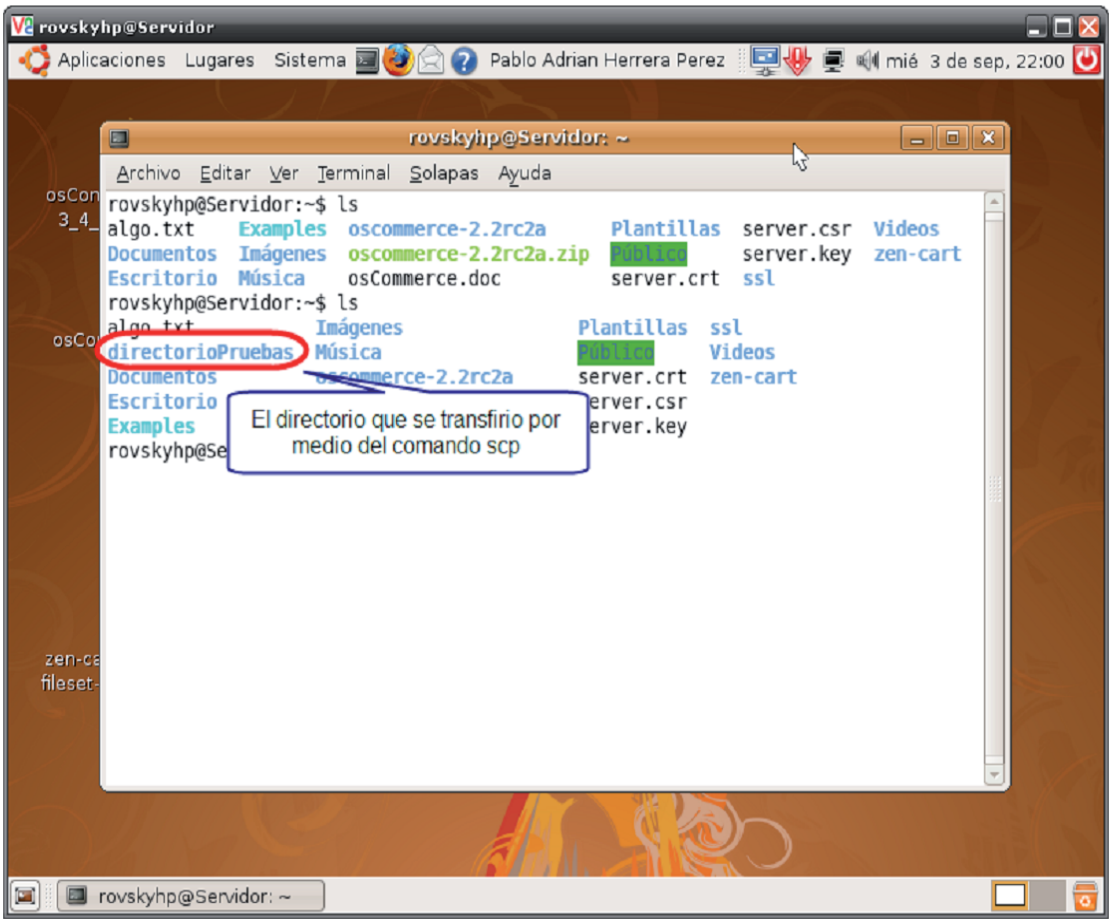

Figura 2.7 Directorio copiado en el home del usuario rovskyhp en el servidor remoto.

2.3. SFTP (Secure Transfer Protocol).

SFTP (SSH File Transfer Protocol) es un protocolo que provee funcionalidad de transferencia y manipulación de archivos a través de un flujo confiable de datos. Comúnmente se utiliza con **SSH** para proveer a éste de transferencia segura de archivos.

2.3.1. Transferencia de archivos a través de SFTP.

El comando SFTP transfiere archivos entre máquinas de forma interactiva y puede ser usado para abrir una conexión segura. Esto es similar a FTP a excepción de que este utiliza una conexión cifrada. La sintaxis general es:

sftp usuario@servidor_remoto

Una vez realizada la autenticación se podrán utilizar un conjunto de comandos similares a los utilizados por el comando FTP.

Para acceder hacia un puerto en particular, sobre el cual esta corriendo el servicio SSH, se hace a través de el parámetro –o Port=Numero de Puerto.

En el siguiente ejemplo se utilizará la cuenta de usuario *rovskyhp*, se accederá por medio de SFTP hacia el servidor ftp *192.168.2.186* el cual tiene corriendo el servicio SSH en el puerto 22.

```
sftp  -o Port=22 rovskyhp@192.168.2.186
```

Con este ejemplo en la figura 2.8 se muestra en pantalla el prompt para el servicio sftp. Antes se solicita la contraseña del usuario *rovskyhp.*

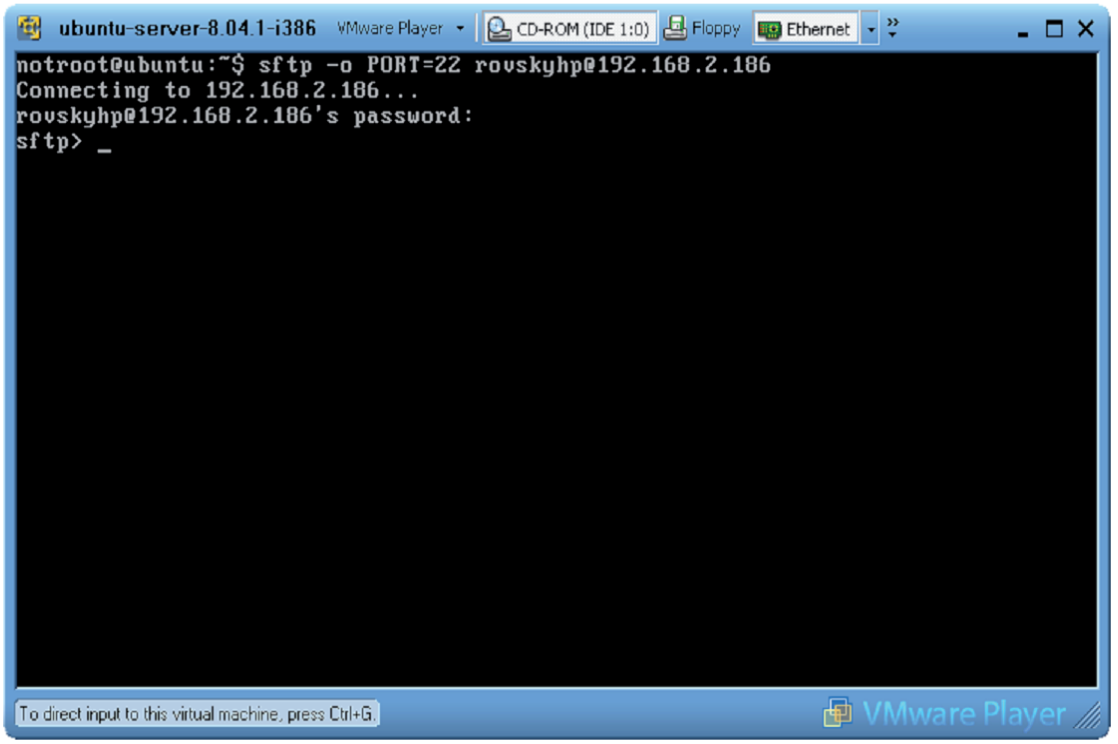

Figura 2.8 Conexión establecida mediante sftp al servidor remoto 192.168.2.186.

El SFTP puede ser usado en forma interactiva, donde se pueden ejecutar comandos desde el equipo local con la finalidad de descargar archivos y subir archivos, todo con un entorno interactivo y seguro ante todo.

En la figura 2.9 se ejecuta el comando get que permitirá la descarga del archivo *ArchivoSFTPPrueba.txt* desde el equipo remoto *192.168.2.186*.

```
sftp> get ArchivoSFTPPrueba.txt
```

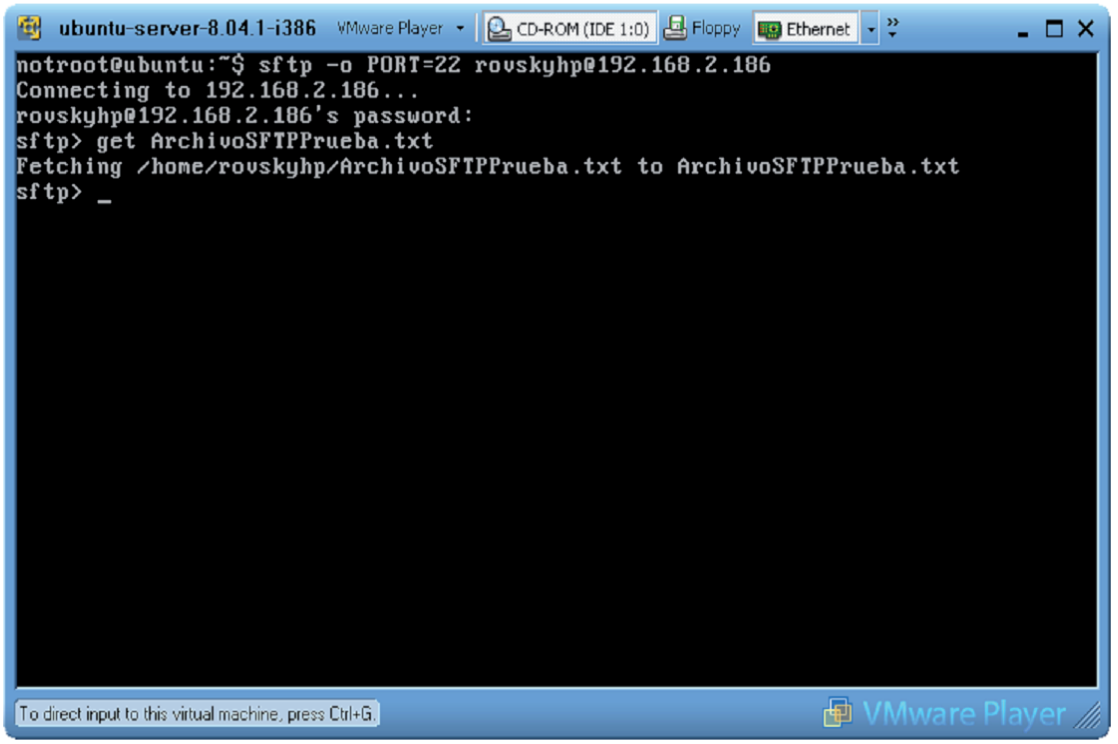

Figura 2.9 Descarga de un archivo, por medio de sftp en servidor remoto.

El archivo será descargado en el directorio en el que se encuentre actualmente en la maquina local como se puede observar en la figura 2.10.

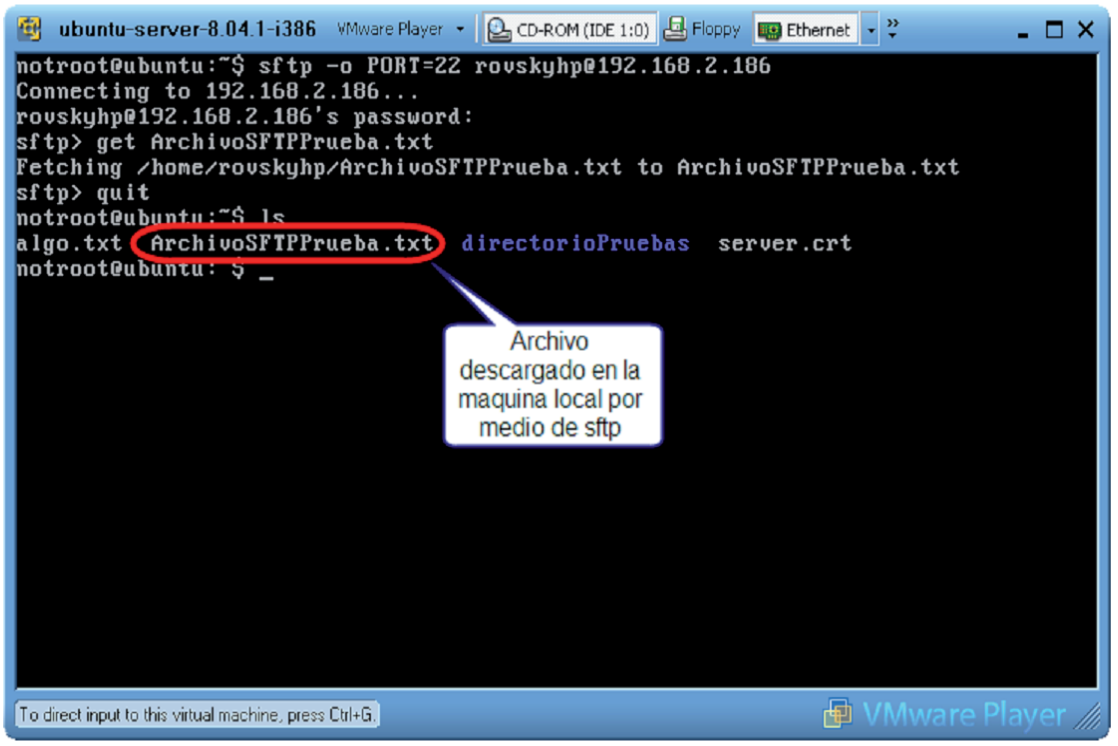

Figura 2.10 Maquina local en la que se ha descargado un archivo por medio de sftp.

En la figura 2.11 se ejecuta el comando put el cual permite subir el archivo *ArchivoSFTPPruebaUpload.txt* desde el equipo local al equipo remoto.

```
sftp> put archivo_upload.txt
```

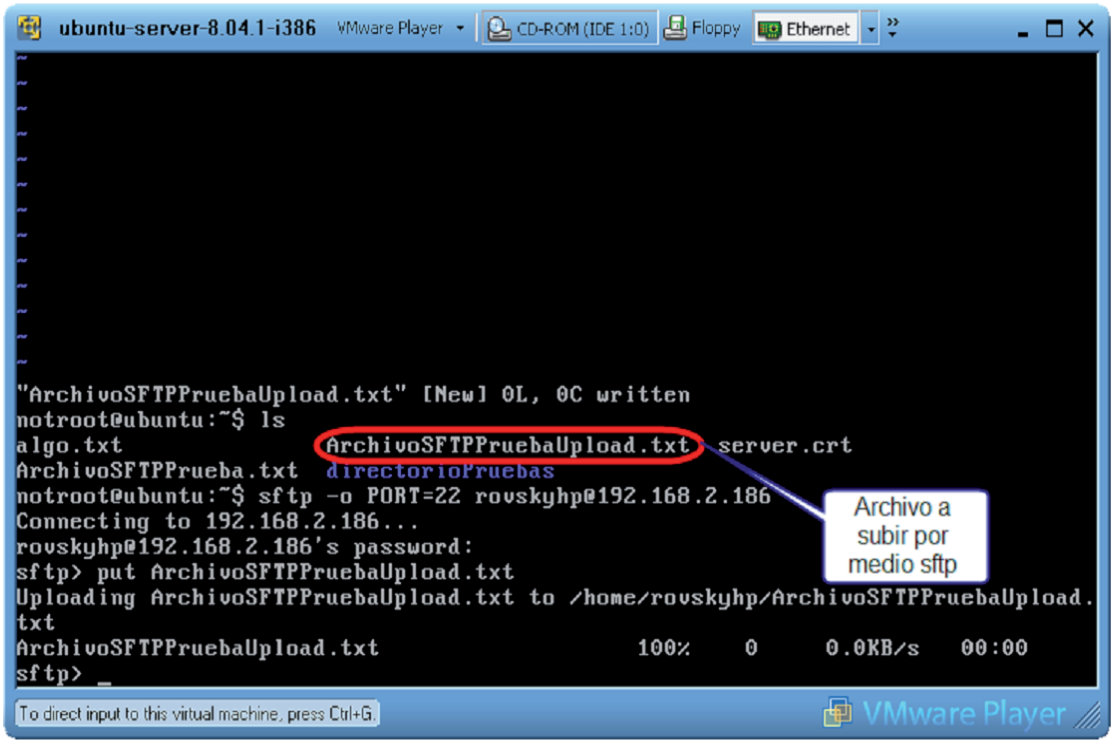

Figura 2.11 Maquina local desde la que se sube un archivo por medio de sftp.

El archivo será colocado en el directorio de inicio del usuario *rovskyhp* en el servidor *192.168.2.186*, como se puede observar en la figura 2.12.

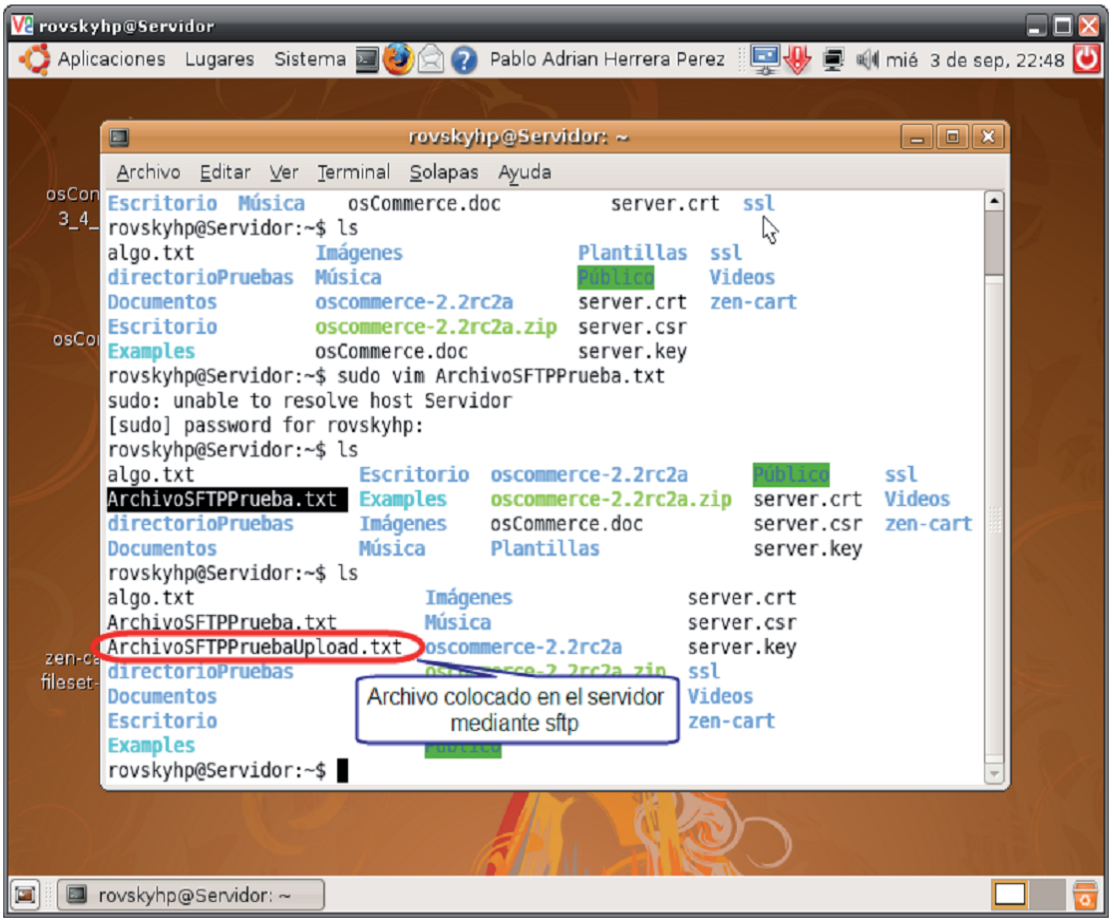

Figura 2.12 Subir un archivo por medio de sftp en el servidor remoto.

2.4. HTTPS (Hypertext Transfer Protocol Secure).

Hypertext Transfer Protocol Secure (en español: *Protocolo seguro de transferencia de hipertexto*), más conocido por su acrónimo HTTPS, es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP. El puerto estándar para este protocolo es el 443 [13].

El sistema HTTPS utiliza un cifrado basado en el Secure Socket Layer (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. Cabe mencionar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP.

HTTPS no es un protocolo separado, pero se refiere a la combinación del protocolo HTTP interactuando sobre una conexión cifrada mediante SSL. Para tener un servidor Web seguro es necesario crear un certificado de llave pública para el servidor Web, dicho certificado puede ser creado mediante diversas herramientas. El certificado deberá ser firmado digitalmente por una autoridad certificadora de una forma u otra, la cual confirma que el servidor es realmente quien dice ser.

También es posible auto firmar el certificado (es decir, ser uno mismo la autoridad certificadora) esto con la finalidad de poder tener páginas seguras dentro de una intranet, en el caso de manejar información confidencial. La página podría ser utilizada para la autenticación del cliente, con el fin de tener acceso restringido a usuarios, los cuales previamente deberán tener su certificado el cual será registrado en el navegador.

2.4.1. SSL (Secure Socket Layer).

SSL (Secure Socket Layer) es un protocolo cliente-servidor que permite conexiones seguras (es decir confidencialidad, autenticación e integridad) a cualquier protocolo basado en TCP/IP, lo que se denomina "*tunneling*". Se usa normalmente con HTTP (formando HTTPS), para ser utilizado principalmente

por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos sensibles. [14]

En 1995, Netscape Communications Corp, atendió la demanda creciente con la creación del protocolo SSL (Capa de Sockets Seguros).
SSL construye una conexión segura entre los dos sockets, incluyendo:

- Negociación de parámetros entre el cliente y el servidor.
- Autenticación tanto del cliente como del servidor.
- Comunicación secreta.
- Protección de la integridad de los datos.

El funcionamiento de SSL es muy similar a SSH:

- El cliente solicita una conexión al servidor SSL, que escucha por defecto en el puerto 443 TCP.
- Comprueban las versiones del protocolo soportadas.
- Acuerdan los algoritmos a utilizar.
- El servidor envía su clave pública al cliente.
- El cliente genera la "clave de sesión" (válida sólo para esa sesión) y la envía al servidor cifrada con la clave pública del servidor.
- A partir de ese momento, todo el tráfico se protege con cifrado simétrico con la "clave de sesión".
- Las claves públicas admiten certificados en los que se basa la autenticación.
- Normalmente sólo se autentica el servidor, mientras que el cliente se mantiene sin autenticar.
- Se utiliza el valor hash de los datos transmitidos para garantizar la integridad de los mismos.

2.4.2. S-HTTP (Secure Hypertext Protocol).

S-HTTP es otro protocolo que proporciona servicios de seguridad sobre internet. Fue diseñado para proporcionar confidencialidad, autenticidad, integridad y no repudio mientras que soporta mecanismos de gestión de

múltiples claves y algoritmos de cifrado mediante opción de negociación entre las partes implicadas en cada transacción. [15]

Usualmente se confunde HTTPS con S-HTTP, pero ambos fueron diseñados para diferentes fines, por lo tanto los dos protocolos pueden utilizarse al mismo tiempo. Mientras que HTTPS fue diseñado para establecer conexión segura entre dos computadoras, S-HTTP fue diseñado para enviar mensajes individuales de manera segura.

Los mensajes SHTTP se basan en tres componentes:

- el mensaje HTTP
- las preferencias criptográficas del remitente
- las preferencias del destinatario

Así, para descifrar un mensaje S-HTTP, el destinatario analiza los encabezados del mensaje para determinar el tipo de método que se utilizó para cifrar el mensaje. Luego, basándose en sus preferencias criptográficas presentes y pasadas, y en las preferencias criptográficas pasadas del remitente, el destinatario puede descifrar el mensaje.

Cuando HTTPS y S-HTTP competían, muchas personas se dieron entonces cuenta de que estos dos protocolos de seguridad eran complementarios, ya que no trabajaban en el mismo nivel. El HTTPS garantiza una conexión segura a Internet, mientras que el S-HTTP garantiza intercambios HTTP seguros.

Como resultado, la compañía Terisa Systems, especializada en protección de red y formada por RSA Data Security y EIT, desarrolló un kit de programación que permitió a los administradores desarrollar servidores Web implementando los protocolos HTTPS y S-HTTP (SecureWeb Server Toolkit) así como clientes Web capaces de soportar estos protocolos (SecureWeb Client Toolkit).

Capítulo 3 FIRMAS DIGITALES

3.1. Aspectos Básicos.

La autenticidad de muchos documentos legales, financieros y de otros tipos se determina por la presencia o ausencia de una firma manuscrita autorizada (firma ológrafa). Las fotocopias de los documentos originales no cuentan con dicha firma. Entonces para que los sistemas de mensajes computarizados reemplazaran a los esquemas tradicionales de papel y tinta, se requería de un método que permitiera que los documentos llevasen una firma infalsificable.

De esta forma el concepto en si de firma digital nació como una oferta tecnológica para acercar la firma tradicional (ológrafa) que se presenta en documentos impresos para brindar autenticidad, a un esquema totalmente distinto como lo son los sistemas de información.

Entonces el problema de inventar un sustituto para las firmas manuscritas es difícil. Básicamente, de lo que se trata es desarrollar un sistema mediante el cual una parte pueda enviar el mensaje firmado a otra logrando que:

- El receptor pueda verificar la identidad del transmisor.
- El transmisor no pueda repudiar (negar) después el contenido del mensaje.
- El receptor no haya podido elaborar el mensaje él mismo.

El primer requisito es necesario, por ejemplo, en los sistemas financieros. Cuando la computadora de un cliente ordena a la computadora de un banco que compre una tonelada de oro, la computadora del banco necesita asegurarse que la computadora que se lo ordena realmente pertenece a la compañía a la que se le aplicara el debito. En otras palabras el banco tiene que autenticar la identidad del cliente (y este a su vez autenticar la identidad del banco).

El segundo requisito es necesario para proteger al banco contra fraude. Un cliente deshonesto podría demandar al banco, alegando que nunca emitió una orden para comprar el oro. Cuando el banco presenta el mensaje en la corte, el

cliente niega haberlo enviado. La propiedad de que ninguna parte de un contrato pueda negar haber firmado se conoce como no repudio.

El tercer requisito es necesario para poder proteger al cliente en el caso de que el precio del oro suba mucho y que el banco trate de falsificar un mensaje en el cual el cliente solicitó un lingote de oro en lugar de una tonelada. En este escenario fraudulento, el banco simplemente mantiene el resto del oro para sí mismo.

La firma digital, consiste en la transformación de algún mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posee el mensaje original y la clave pública del firmante, pueda establecer de forma segura, que dicha transformación se efectuó utilizando la clave privada correspondiente a la pública del firmante, y si el mensaje es el original o fue alterado desde su elaboración.[16]

La persona que cuenta con la clave pública y mensaje inicial puede determinar con certeza 2 aspectos:

- Si la transformación se creó usando la clave privada que corresponde a la clave pública del firmante.
- Si el mensaje ha sido modificado desde que se efectuó la transformación.

En cuanto al término transformación se refiere, esta definición es importante ya que decide la tecnología a utilizar, que recae fundamentalmente en la criptografía. La firma digital utiliza un sistema de cifrado asimétrico. Esto significa que comprende dos procesos.

- La creación de la firma por parte del suscriptor utilizando la clave privada, que es sólo conocida por el suscriptor y el es el único responsable de cuidarla y guardarla.
- La verificación de la firma por la otra parte: el receptor del mensaje comprueba su autenticidad utilizando la clave pública que surge del certificado del suscriptor, comunicándose con el repositorio o registro donde el referido certificado se encuentra registrado.

La finalidad de la firma digital, es el mismo que el de la firma ológrafa: dar confirmación y compromiso con el documento firmado; por esto es que a través del marco regulatorio, se pretende concientizar en su uso y hacer valer los documentos u elementos que contengan firmas digitales, exigiéndose ciertos requisitos de validez.

3.2. Ventajas Ofrecidas por la Firma Digital.

Integridad de la información: la integridad del documento es una protección contra la modificación de los datos en forma intencional o accidental. El emisor protege el documento, incorporándole a ese un valor de control de integridad, que corresponde a un valor único, calculado a partir del contenido del mensaje al momento de su creación. El receptor deberá efectuar el mismo cálculo sobre el documento recibido y comparar el valor calculado con el enviado por el emisor. De coincidir, se concluye que el documento no ha sido modificado durante la transferencia.

Autenticidad del origen del mensaje: este aspecto de seguridad protege al receptor del documento, garantizándole que dicho mensaje ha sido generado por la parte identificada en el documento como emisor del mismo, no pudiendo alguna otra entidad suplantar a un usuario del sistema. Esto se logra mediante la inclusión en el documento transmitido de un valor de autenticación (MAC, Message Autentication Code). El valor depende tanto del contenido del documento como de la clave secreta en poder del emisor.

No repudio del origen: el no repudio de origen protege al receptor del documento de la negación del emisor de haberlo enviado. Este aspecto de seguridad es más fuerte que los anteriores ya que el emisor no puede negar bajo ninguna circunstancia que ha generado dicho mensaje, transformándose en un medio de prueba inequívoco respecto de la responsabilidad del usuario del sistema.

Imposibilidad de suplantación: el hecho de que la firma haya sido creada por el signatario mediante medios que mantiene bajo su propio control (su clave privada protegida, por ejemplo, por una contraseña, una tarjeta inteligente, etc.) asegura, además, la imposibilidad de su suplantación por otro individuo.

Auditabilidad: permite identificar y rastrear las operaciones llevadas a cabo por el usuario dentro de un sistema informático cuyo acceso se realiza mediante la presentación de certificados.

Mediante la siguiente tabla es posible detectar algunas otras ventajas que brinda la firma digital frente a la firma ológrafa.

Tabla 3.1 Ventajas de la firma digital sobre la firma ológrafa.

Propiedad	Firma ológrafa	Firma digital
Puede ser aplicada a documentos electrónicos y transacciones electrónicas.	No	Si
La verificación de la firma puede ser automatizada.	No	Si
La firma permite detectar alteraciones que se hayan hecho al documento.	No	Si
Puede utilizarse para comprometer a las partes involucradas.	Si	Si
Tiene reconocimiento en el aspecto legal.	Si	Si

3.3. Aspectos Técnicos.

A diferencia de la firma manuscrita, que es un trazo sobre un papel, la firma digital consiste en el agregado de un elemento al texto original, siendo este elemento, en definitiva, la firma digital; al conjunto formado por el documento original más la firma digital se le denominará mensaje.

El elemento o firma digital es el resultado de un cálculo que se realiza sobre la cadena binaria del texto original.

En este cálculo están involucrados el documento mismo y una clave privada (que, generalmente, pertenece al sistema de clave pública-privada o sistema asimétrico) la cual es conocida sólo por el emisor o autor del mensaje, lo que

da como resultado que para cada mensaje se obtenga una firma distinta, es decir, a diferencia de la firma tradicional, la firma digital cambia cada vez con cada mensaje, porque la cadena binaria de cada documento será distinta de acuerdo a su contenido.

A través de este sistema se puede garantizar completamente las siguientes propiedades de la firma tradicional:

- Quien firma reconoce el contenido del documento, que no puede modificarse con posterioridad (integridad).
- Quien lo recibe verifica con certeza que el documento procede del firmante. No es posible modificar la firma (autenticidad).
- El documento firmado tiene fuerza legal. Nadie puede desconocer haber firmado un documento ante la evidencia de la firma (no repudio).

Este sistema utiliza dos claves diferentes: una para cifrar y otra para descifrar. Una es la clave pública, que efectivamente se pública y puede ser conocida por cualquier persona; otra, denominada clave privada, se mantiene en absoluto secreto ya que no existe motivo para que nadie más que el autor necesite conocerla y aquí es donde reside la seguridad del sistema.

Ambas claves son generadas al mismo tiempo con un algoritmo matemático y guardan una relación tal entre ellas que algo que es cifrado con la privada, solo puede ser descifrado por la clave pública.

Resumiendo, la clave privada es imprescindible para descifrar criptogramas y para firmar digitalmente, mientras que la clave pública debe usarse para cifrar mensajes dirigidos al propietario de la clave privada y para verificar su firma.

Es conveniente aclarar a la par de la existencia del par de claves, pública y privada, podría intervenir otra clave que es la de la Autoridad Certificante (como se verá más adelante), que provee la garantía de autenticidad del par de claves generadas, así como también, su pertenencia a la persona cuya propiedad se atribuye.

Este esquema se utiliza en intercambios entre entidades cuando se trata de transferencias electrónicas de dinero, órdenes de pago, etc. donde es

indispensable que las transacciones cumplan con los requisitos de seguridad mencionados anteriormente (confidencialidad, integridad, no repudio, etc.).

3.4. Tipos de Claves (Simétricas y Asimétricas).

Las técnicas criptográficas, tales como el cifrado de datos o la firma digital, son empleadas en todos los sistemas que necesiten garantizar los servicios que brindan los sistemas de información electrónicos. El mecanismo más básico empleado es el denominado algoritmo criptográfico, el cual define dos transformaciones:

- El **cifrado:** es la conversión del texto plano (*plaintext*) en el texto cifrado o criptograma (*ciphertext*) mediante el empleo de una determinada clave.
- El **descifrado**: que es el proceso inverso.

La aplicación primordial de un algoritmo criptográfico (más no la única) es asegurar el servicio de confidencialidad: en lugar de transmitir el texto plano se envía el cifrado, de forma que un atacante no podrá descifrar el contenido de la información transmitida a no ser que conozca la clave de descifrado. La seguridad de un sistema de cifrado por ende radica casi totalmente en la privacidad de las claves secretas. [17]

3.4.1. Métodos simétricos o de clave privada.

La criptografía simétrica (por ejemplo DES, AES, IDEA) usa la misma clave para cifrar y para descifrar un mensaje tal como en la figura 3.1 y su seguridad se basa en el secreto de la clave (el algoritmo es públicamente conocido). Generalmente se utilizan dos funciones: una para realizar el cifrado y otro para el descifrado. Su principal desventaja es que hace falta que el emisor y receptor compartan la clave.

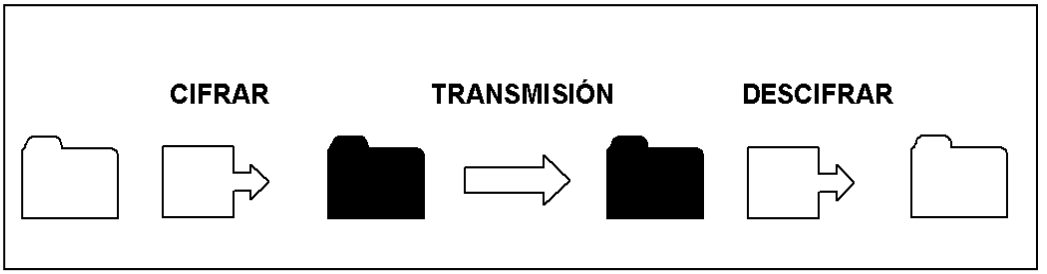

Figura 3.1 Método simétrico.

El cifrado convencional (criptografía simétrica) tiene como principal ventaja, que es muy rápido de ejecutar, es muy útil cuando se tienen que cifrar una gran cantidad de datos, sin embargo tiene la desventaja de que se tiene que transmitir por un canal seguro. Cuando un emisor y un receptor se tienen que comunicar de forma segura, entonces ellos tienen que quedar de acuerdo en que llave han utilizar y mantenerla en secreto, pero si ellos se encuentran en lugares distintos necesitan de un mecanismo seguro para poder intercambiar la llave secreta, ya que cualquier persona que intercepte el mensaje, podría descifrar toda la comunicación entre las dos personas. [18]

3.4.2. Métodos asimétricos o de clave pública.

El concepto de criptografía asimétrica fue introducido por Whitfield Diffie y Martin Hellman en 1975, aunque hay evidencia de que el servicio secreto británico lo había inventado algunos años antes, solo que lo mantienen como secreto militar. La criptografía asimétrica utiliza un par de llaves, una llave que puede ser pública, utilizada generalmente para cifrar, y la correspondiente llave privada, utilizada para descifrar la información cifrada con la pública. [19]

En este tipo de método, cada usuario del sistema criptográfico será poseedor de un par de claves.

- **Clave pública:** esta clave será conocida por todos.
- **Clave privada:** será guardada por su propietario y no deberá de darse a conocer a ningún otro usuario.

Este tipo de algoritmos se pueden utilizar de dos formas, dependiendo de si la clave pública se emplea como clave de cifrado o de descifrado. En el primer caso (figura 3.2), cuando un usuario, A, quiere enviar información a otro usuario, B, utiliza la clave pública de B, *KpuB*, para cifrar los datos. El usuario B utilizará su clave privada (que sólo él conoce), *KprB*, para obtener el texto en claro a partir de la información (cifrada) recibida. Si otro usuario, C, quiere enviar información al usuario B, también empleará la clave pública *KpuB*. Este modo se suele emplear para proporcionar el servicio de confidencialidad, pues sólo el usuario B es capaz de descifrar los mensajes que los usuarios A y C le han enviado.

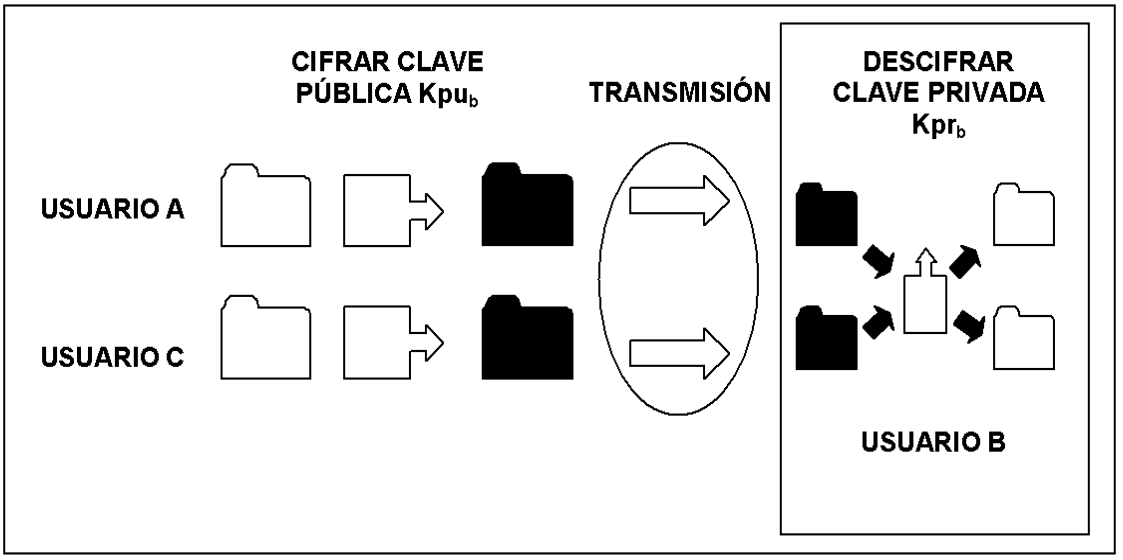

Figura 3.2. Clave pública con enfoque de confidencialidad.

En el otro modo de operación (figura 3.3), es el usuario B quien cifra la información utilizando su clave privada, *KprB*, de forma que cualquiera que conozca *KpuB* podrá descifrar la información transmitida. Este método se puede emplear para proporcionar el servicio de autenticación, ya que la obtención del texto en claro a partir del texto cifrado es una garantía de que el emisor del mensaje es el propietario de *KpuB* (lógicamente, para saber que el mensaje obtenido del descifrado del texto cifrado es el texto en claro original, éste se ha de obtener por otros medios para realizar una comparación – esto se verá más adelante). También es la base para la construcción de los mecanismos de firma digital.

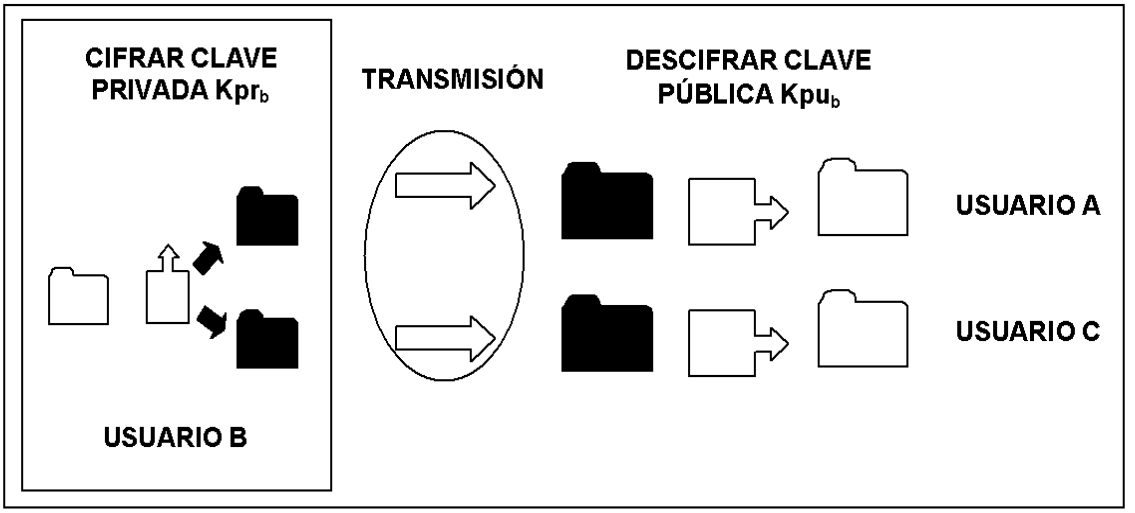

Figura 3.3 Clave pública con enfoque de Autenticación.

El beneficio principal de la criptografía asimétrica, es que permite que personas que no tienen un acuerdo de seguridad previo, puedan intercambiar información de forma segura, y por lo tanto la necesidad de enviar y recibir las llaves secretas únicamente a través de algún canal seguro es eliminada, puesto que todas las comunicaciones involucran únicamente llaves públicas, y ninguna llave privada es transmitida o compartida. Algunos ejemplos de sistemas criptográficos de llave pública son Elgamal, RSA, Diffie-Hellman y DSA (Digital Signature Algorithm).

3.4.3. Función HASH.

Una función hash toma una entrada de longitud variable, es decir algún mensaje de cualquier tamaño, incluso de algunos miles de millones de bits, y produce una salida de longitud fija, con la condición de que si alguno de los bits del mensaje original es modificado, la salida producida por la función de hash, va a ser completamente diferente, a esta salida se le conoce como la firma o resumen del mensaje, (message digest). Actualmente los algoritmos hash mas utilizados son el MD5 (sucesor del algoritmo MD4) y SHA en sus versiones SHA-1, SHA-224, SHA-256 y SHA-512. [20]

Algunas propiedades de la función criptográfica por donde se pasa el mensaje son:

- Su algoritmo es conocido.

- Son de un solo sentido, es decir, a partir del valor hash no se pueden obtener los datos originales.
- El valor hash es obtenido de tal forma que es muy poco probable obtener el mismo valor a partir de otros datos.

La robustez de una función hash se basa en las características mencionadas anteriormente. Por ejemplo, en una situación ficticia, si un atacante conoce el mensaje y su valor hash y pudiese encontrar otros datos que generarán el mismo valor hash, este sería capaz de realizar una sustitución sin que esta pueda ser detectada.

Para el caso de la firma digital se puede tomar como referencia la figura 3.4, este método consiste en la obtención de un valor hash del mensaje y su posterior cifrado con la clave privada del emisor. En recepción se descifra el hash con la clave pública del emisor y se compara con otro valor hash obtenido en recepción de forma independiente a partir del mensaje.

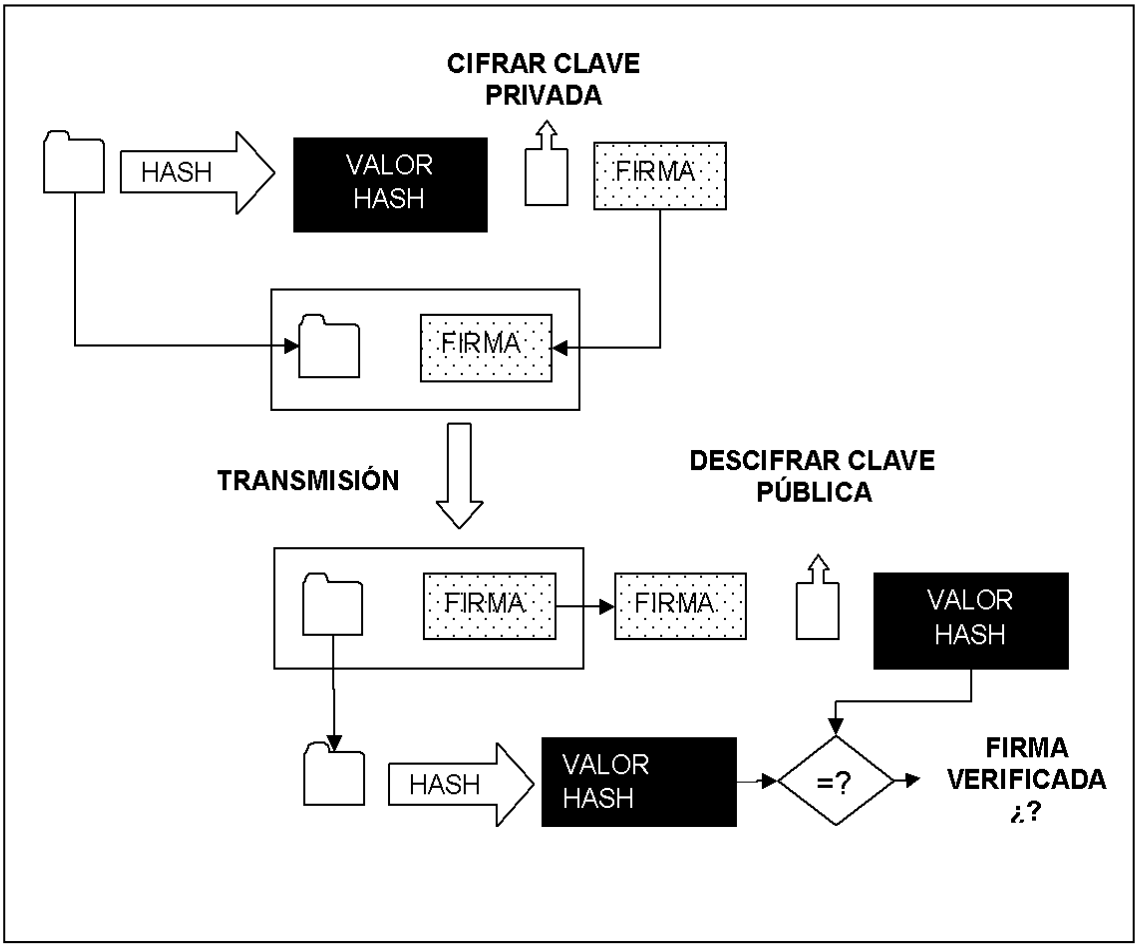

Figura 3.4 Esquema de una firma digital.

A la vista, una firma digital se representa por una extensa e indescifrable cadena de caracteres, dígitos y letras, que representa en realidad un número, el cual es el resultado de un procedimiento matemático aplicado al documento, como se muestra en la figura 3.5.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.4 (SunOS)
Comment: For info see http://www.gnupg.org

mQEPAznuyr4AAAEIANA4pbwHA1yiYCKHTdnnztM5bXiwIuoSUEQsUbxg+4l2Qzlv
7w7GmREvvrdw3+XVvcIwE72jNI9n8T+1dJ9cMr/MkUSW51iSrRTl/PlOxHqVT+z3
nnPV5bUwPxdY6os3QQkkcyIQiVlK9KiCExUl4FVx741dQISIRS8Pt4NpLzGbPaR1
7WDbzCyzXD+coFbVqFqU2cw/dQ5WdpKPg60728bIX6z9BtKGlqAjNEVyXDrcRZP1
XFLpnjmeYHRFcMlnIkL6nq/36DAPkWSdPjWR3ASxtWKrLWv4tb2/7dT5rR9vswIZ
7byLPoQy3xykBFgGghzKIPONCnheuYRnIyxMZpsAEQEAAbRPUEdQIGNlcnRpZmlj
YXRlIG9mIFJlZElSSVMgV2ViIFNlcnZlciAsIHNlZSBodHRwOi8vd3d3LnJlZGly
aXMuZXMvcGdwL2Zpcm1hd2ViL4kBHwQQAQIACQUCOjaPUQIZAQAKCRC5hGcjLExm
m709CACZ7db7ZR3+4zLcnvfAa9mvoIMmZ1rnfAQqCuHIhu7TcMZHfEjMHjHxL+Lp
juwA5IVy+F48mrTbjjzhXByOPYEtNjPdsNKOHqhRCjtzIBZPBzYDo/XM38hblx2r
pFqtS93IBLhjh+xfBMqvjb0+1+yCwEoWWPrLi3Tf7fsoFEQA4k28Y7xO+e5Ocyk8
1Ej2rKIbSDjOHQ9GzGLonrD9Lsdh83GulR9IgeGesEVUkq1th0N2MrkUKpePRQvT
DmBAf3be4gE8SzXUsGNwUdgLS7fkVdbVkquMiu74dSfRo8p8XStRRb3bfsEdp85c
rnCLQ2g4yW6SBlDBgO0mELifdw8P
=jvIu
-----END PGP PUBLIC KEY BLOCK-----
```

Figura 3.5 Ejemplo de Firma Digital.

Capítulo 4 ADMINISTRACIÓN DE CLAVES PÚBLICAS

4.1. Nociones preliminares.

Antes de que las organizaciones puedan ver los frutos cosechados en lo que a comercio electrónico se refiere, primero deberán tener en cuenta y encarar las siguientes amenazas.

- Pérdidas financieras por fraude.
- Robo de información de propiedad privada como consecuencia de accesos no autorizados.
- Pérdida de la confianza de los clientes por estar comprometida la integridad de los datos de la organización.
- Perdida de negocios ante un negación de servicios (DOS).
- Críticas por parte de los clientes a consecuencia de los controles inadecuados de la privacidad.

Para esto se han creado, mejorado e implementado estrategias que permitirán la protección de los datos, la autenticación, la confidencialidad, la disponibilidad y la integridad de los mismos, mediante herramientas tales como los certificados, las firmas electrónicas y la infraestructura de llave pública (PKI por sus siglas en ingles).

La criptografía de clave pública hace posible que las personas que no comparten una clave se comuniquen con seguridad. También posibilita firmar mensajes sin la presencia física de un tercero confiable. Dando lugar a las Entidades Certificadoras que en su momento se describirán y cuya función de manera general es la de evitar la suplantación de identidad entre usuario y organización.

4.2. Certificados Digitales.

Un Certificado Digital es un documento digital mediante el cual un tercero confiable (una autoridad de certificación) garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública. [21]

Si bien existen variados formatos para certificados digitales, los más comúnmente empleados se rigen por el estándar X.509. El certificado contiene usualmente el nombre de la entidad certificada, número de serie, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital) y la firma digital de la autoridad emisora del certificado de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación.

Los certificados digitales emitidos por las entidades de certificación deben contener al menos los siguientes campos tal como se puede ver en la figura 4.1. [22]

1. Datos que identifiquen al suscriptor.
2. Datos que identifiquen a la Entidad de Certificación.
3. La clave pública.
4. La metodología para verificar la firma digital del suscriptor impuesta a un mensaje de datos.
5. Número de serie del certificado.
6. Vigencia del certificado.
7. Firma digital de la Entidad de Certificación.

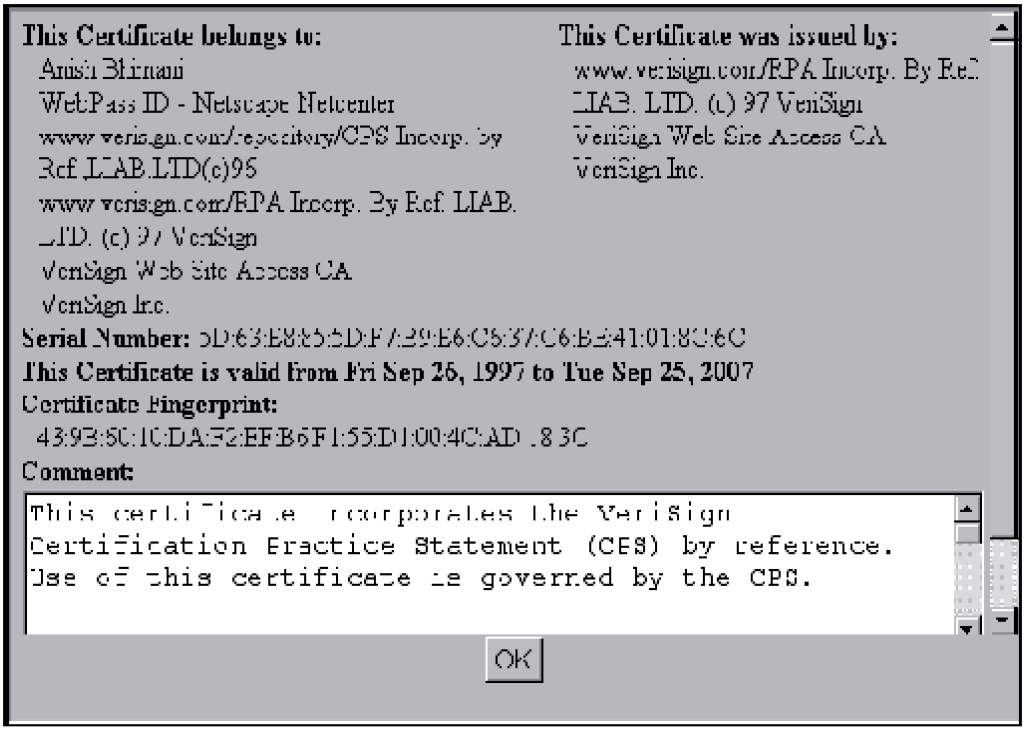

Figura 4.1. Ejemplo de un certificado.

Los certificados digitales permiten verificar que la información que se envía es auténtica, es decir que el remitente sea realmente quien dice ser y que el contenido del mensaje enviado (criptograma) no haya sido modificado en su tránsito.

Así pues los certificados digitales, proporcionan un mecanismo para verificar la autenticidad de programas y documentos obtenidos a través de la Web, como por ejemplo el envío de correo electrónico cifrado, firmas digitales, control de acceso a recursos, la validación oficial de documentos electrónicos, etc.

Los certificados digitales representan el punto más importante en las transacciones electrónicas seguras. Estos brindan una forma conveniente y fácil de asegurar que los participantes en una transacción electrónica puedan confiar el uno en el otro. Esta confianza se establece a través de un tercero llamado Autoridad Certificadora, que como se verá mas adelante, puede ser uno mismo la autoridad certificadora, pero que no seria lo más conveniente pues no confiarían en dicha entidad, lo mejor es utilizar Autoridades Certificadoras reconocidas a nivel mundial.

Como ejemplo, existe un servidor Web seguro el cual ofrece un certificado para cifrar las comunicaciones entre el servidor y el navegador cliente. Cuando el navegador cliente recibe el certificado, comprueba en su configuración, una lista que ya trae por defecto de autoridades certificadoras válidas, si dicho certificado viene firmado por una autoridad de confianza. En caso de que la firma del certificado esté avalada por una autoridad de confianza el navegador continúa normalmente su diálogo cifrando la información. Si la firma no está avalada por ninguna de las autoridades reconocidas, entonces avisa al cliente de que la firma no es de confianza y pregunta qué hacer. Es importante observar que la confianza o no de un certificado, en general, no impide que se pueda utilizar para establecer una comunicación, simplemente plantea dudas sobre la autenticidad del emisor.

Los pasos para establecer una comunicación segura mediante certificados son:

- Se solicita un certificado.

- Se verifica la firma con la Autoridad Certificadora para comprobar que el certificado es correcto, que dicha Autoridad Certificadora fue quien emitió el certificado y que su llave pública es la misma que la del certificado.
- Se recibe la confirmación de la Autoridad Certificadora que el certificado es correcto.
- La información se cifra mediante la llave pública y se envía.

En pocas palabras, los certificados digitales garantizan que dos computadoras que se comunican entre sí puedan efectuar transacciones electrónicas con éxito. La base de esta tecnología reside en los códigos secretos o en el cifrado. Dicho cifrado garantiza la confidencialidad, la integridad y la autenticidad de la información que se desea transmitir y que tiene vital importancia para la persona o empresa.

4.2.1. Generación y distribución de certificados.

Las autoridades de certificación (AC) tienen como misión la gestión de los denominados certificados (de clave pública). Un certificado está compuesto básicamente por la identidad de un usuario (subject), su clave pública, la identidad y la clave pública de la AC emisora (issuer) del certificado en cuestión, su periodo de validez y la firma digital del propio certificado. Esta firma, realizada por la AC emisora, permite que aquellas entidades que deseen realizar comunicaciones con la persona poseedora del certificado, puedan comprobar que la información que éste contiene es auténtica (suponiendo que se confía en la AC emisora). Una vez que los certificados han sido firmados, se pueden almacenar en servidores de directorios o transmitidos por cualquier medio (seguro o no) para que estén disponibles públicamente. [23]

4.2.2. Validación de certificados.

Antes de enviar un mensaje cifrado mediante un método asimétrico, el emisor ha de obtener y verificar los certificados de los recipientes de dicho mensaje. La validación de un certificado se realiza verificando la firma digital en él incluida mediante el empleo de la clave pública de su signatario, que a su vez

ha de ser validada usando el certificado correspondiente, y así sucesivamente hasta llegar a la raíz de la jerarquía de certificación. En el proceso de verificación se ha de comprobar el periodo de validez de cada certificado y que ninguno de los certificados de la cadena haya sido revocado. Esto último se realiza utilizando las CRLs (*Certificate Revocation Lists*). [24]

4.2.3. Revocación.

Los certificados tienen un periodo de vida limitado, el cual está especificado en el propio certificado y que viene determinado por la política de la AC emisora. Sin embargo, en algunas ocasiones especiales la seguridad de la clave privada asociada puede haberse visto comprometida, por lo que la utilización de la correspondiente clave pública ha de ser evitada. También puede ocurrir que el propietario del certificado cambie de nombre, hecho que implica que ha de modificarse el certificado. En tales casos, la AC emisora puede revocar el certificado para prevenir su uso. La decisión de revocar un certificado es responsabilidad de la AC emisora, generalmente en respuesta a la petición de una entidad autorizada, como por ejemplo, el propio dueño del certificado. [25]

4.2.4. Lista de Certificados Revocados o CRL.

Como ya se mencionó, los certificados tienen un periodo de validez definido por la AC. Durante el tiempo que el certificado es válido la AC que lo generó mantiene información sobre el estado del certificado.

Con la finalidad de conocer si un certificado no esté revocado, es evidente la necesidad de contar con algún archivo, directorio o base de datos que contenga los certificados revocados y por cada unos de ellos, la fecha y la hora a la que fueron revocados. Una primera aproximación a este directorio de certificados revocados es la conocida como Lista de Certificados Revocados (CRL por sus siglas en ingles). Un CRL es un archivo, firmado por la Autoridad Certificadora, que contiene la fecha de emisión del CRL y una lista de certificados revocados, cada uno de ellos con la fecha de revocación.

Un CRL puede ser autenticado como cualquier otro documento firmado digitalmente, en este caso con la llave pública de la Autoridad Certificadora. Una vez autenticado, se puede confiar en su contenido y determinar con certeza si un certificado está revocado o no, esto es, hasta la fecha definida por "Última Actualización". El CRL es muy útil en algunos casos, por ejemplo:

1. El sujeto “A” recibió un documento firmado por el sujeto “B” el día 13 de marzo de 2007.
2. Autentica el documento.
3. La AC publica el CRL diariamente, de manera que el sujeto “A” obtiene, al siguiente día, una copia del CRL cuya fecha UTC es: las 0:00 horas del día de 15 de marzo de 2007; la autentica con la llave pública de la AC.
4. El sujeto “A” extrae del certificado del sujeto “B” el número de serie de dicho sujeto.
5. Consulta el CRL para determinar si el número de serie de B se encuentra listado en él.

Existen diversos métodos para la actualización de las listas de certificados revocados:

- **Muestreo de CRL:** Las aplicaciones acceden a la AC, o donde se encuentre el archivo CRL y copian el ultimo registro que existe de certificados revocados en intervalos regulares.
- **Anuncio de CRL:** La AC realiza el anuncio de que ha habido un cambio en el CRL. La desventaja de este método es que seria muy costoso avisar cada que se revoca un certificado.
- **Verificación en la Web:** Deberá existir un sistema que realice la consulta en línea a la AC para determinar el estado de revocación de los certificados.

En la actualidad se cuenta con un formato (estándar) que se ha extendido casi para todas las aplicaciones, este es el llamado X.509. Este formato contiene los datos del poseedor del certificado, la clave pública del propietario, y la firma de una autoridad certificadora. La mejor propiedad del formato X.509 es que contiene el mínimo necesario de información para poder realizar muchas

transacciones, principalmente comerciales y financieras. Sin embargo para otras aplicaciones puede ser un poco robusto.

4.3. X.509.

El formato de certificados X.509 es un estándar del ITU-T (International Telecommunication Union-Telecommunication Standarization Sector) y el ISO/IEC (International Standards Organization / International Electrotechnical Commission) que se publicó por primera vez en 1988. El formato de la versión 1 fue extendido en 1993 para incluir dos nuevos campos que permiten soportar el control de acceso a directorios. Después de emplear el X.509 v2 para intentar desarrollar un estándar de correo electrónico seguro, el formato fue revisado para permitir la extensión con campos adicionales, dando lugar al X.509 v3, publicado en 1996. [26]

El estándar X.509 solo define la sintaxis de los certificados, por lo que no esta atado a ningún algoritmo en particular y contempla los siguientes campos:

- **Versión.** El campo de versión contiene el número de versión del certificado codificado. Los valores aceptables son 1, 2 y 3.
- **Número de serie del certificado.** Este campo es un entero asignado por la autoridad certificadora. Cada certificado emitido por una AC debe tener un número de serie único.
- **Identificador del algoritmo de firmado.** Este campo identifica el algoritmo empleado para firmar el certificado (como por ejemplo el RSA o el DSA).
- **Nombre del emisor.** Este campo identifica la AC que ha firmado y emitido el certificado.
- **Periodo de validez.** Este campo indica el periodo de tiempo durante el cual el certificado es válido y la AC está obligada a mantener información sobre el estado del mismo. El campo consiste en una fecha inicial, la fecha en la que el certificado empieza a ser válido y la fecha después de la cual el certificado deja de serlo.
- **Nombre del sujeto.** Este campo identifica la identidad cuya clave pública está certificada en el campo siguiente. El nombre debe ser único

para cada entidad certificada por una AC dada, aunque puede emitir más de un certificado con el mismo nombre si es para la misma entidad.

- **Información de clave pública del sujeto.** Este campo contiene la clave pública, sus parámetros y el identificador del algoritmo con el que se emplea la clave.
- **Identificador único del emisor.** Este es un campo opcional que permite reutilizar nombres de emisor.
- **Identificador único del sujeto.** Este es un campo opcional que permite reutilizar nombres de sujeto.
- **Campo de Extensión.** Permiten la adición de nuevos campos, a la estructura sin que por ello se tenga que modificar la definición del certificado.

La firma realizada por la AC emisora, permite que aquellas entidades que deseen realizar comunicaciones con la persona poseedora del certificado, puedan comprobar que la información que éste contiene es auténtica (suponiendo que confíen en la AC emisora).

Una vez que los certificados han sido firmados, se almacenan en servidores de directorios, o son transmitidos utilizando algún medio (seguro o no) para que estén disponibles para el público. [27]

4.3.1. Extensiones.

Las extensiones del X.509 v3 proporcionan una manera de asociar información adicional a sujetos, claves públicas, etc. Un campo de extensión tiene tres partes:

- **Tipo de extensión.** Es un identificador de objeto que proporciona la semántica y el tipo de información (cadena de texto, fecha u otra estructura de datos) para un valor de extensión.
- **Valor de la extensión.** Este sub-campo contiene el valor actual del campo.
- **Indicador de importancia.** Es un valor que indica a una aplicación si es seguro ignorar el campo de extensión si no reconoce el tipo. El indicador

proporciona una manera de implementar aplicaciones que trabajan de modo seguro con certificados y evolucionan conforme se van añadiendo nuevas extensiones. [28]

Las extensiones de archivo de certificados X.509 son las que se muestran en la tabla 4.1.

Tabla 4.1 Extensiones de archivo de certificados X.509.

Extensión	Descripción
.CER	Certificado codificado en CER, algunas veces es una secuencia de certificados
.DER	Certificado codificado en DER
.PEM	Certificado codificado en Base64, encerrado entre "-----BEGIN CERTIFICATE-----" y "-----END CERTIFICATE-----"
.P7B .P7C	Estándar de sintaxis para mensajes criptográficos. Define la sintaxis genérica para los mensajes que tienen criptografía aplicada a ellos.
.PFX .P12	Puede contener certificado(s) (público) y claves privadas (protegido con clave)

PKCS #7 es un estándar para firmar o cifrar datos. Dado que el certificado es necesario para verificar datos firmados, es posible incluirlos en la estructura SignedData. Un archivo .P7C es simplemente una estructura SignedData, sin datos para firmar.

PKCS #12 evolucionó del estándar PFX (Personal inFormation eXchange) y se usa para intercambiar objetos públicos y privados dentro de un archivo.

Un archivo .PEM puede contener certificados o claves privadas, encerrados entre las líneas BEGIN/END apropiadas. [29]

El ITU-T y el ISO/IEC han desarrollado y publicado un conjunto de extensiones estándar en un apéndice al X.509 v3:

- **Limitaciones básicas.** Este campo indica si el sujeto del certificado es una AC y el máximo nivel de profundidad de un camino de certificación a través de esa AC.

- **Política de certificación.** Este campo contiene las condiciones bajo las que la AC emitió el certificado y el propósito del certificado.
- **Uso de la clave.** Este campo restringe el propósito de la clave pública certificada, indicando, por ejemplo, que la clave sólo se debe usar para firmar, para el cifrado de claves, para el cifrado de datos, etc. Este campo suele marcarse como importante, ya que la clave sólo está certificada para un propósito y usarla para otro no estaría validado en el certificado.

El formato de certificados X.509 se especifica en un sistema de notación denominado sintaxis abstracta uno (Abstract Sintax One o ASN-1). Para la transmisión de los datos se aplica el DER (Distinguished Encoding Rules o reglas de codificación distinguible), que transforma el certificado en formato ASN-1 en una secuencia de octetos apropiada para la transmisión en redes reales. Siguiendo la notación de ASN.1, un certificado contiene diversos campos, agrupados en tres grandes grupos. [30]

El primer campo es el sujeto (*subject*), que contiene los datos que identifican al sujeto titular. Estos datos están expresados en notación DN (Distinguished Name), donde un DN se compone a su vez de diversos campos, siendo los más frecuentes los siguientes; CN (*Common Name*), OU (*Organizational Unit*), O (*Organization*) y C (*Country*). Un ejemplo para identificar un usuario mediante el DN, es el siguiente: CN=david.comin O=Safelayer, OU=development, C=ES. Además del nombre del sujeto titular (*subject*), el certificado, también contiene datos asociados al propio certificado digital, como la versión del certificado, su identificador (*serialNumber*), la AC firmante (issuer), el tiempo de validez (*validity*), etc. La versión X.509.v3 también permite utilizar campos opcionales (nombres alternativos, usos permitidos para la clave, ubicación de la CRL y de la AC,etc.).

En segundo lugar, el certificado contiene la clave pública, que expresada en notación ASN.1, consta de dos campos, en primer lugar, el que muestra el algoritmo utilizado para crear la clave (ej. RSA), y en segundo lugar, la propia clave pública.

Por último, la AC, ha añadido la secuencia de campos que identifican la firma de los campos previos. Esta secuencia contiene tres atributos, el algoritmo de firma utilizado, el hash de la firma, y la propia firma digital. [31]

4.4. Infraestructura de clave pública (PKI).

PKI (*Public Key Infrastructure*) es el término utilizado para referirse a la infraestructura de seguridad, basada en criptografía de clave pública, que permite la gestión de certificados digitales.

Por tanto, en una Infraestructura de Clave Pública, se tendrá que definir y establecer todos los métodos necesarios para gestionar los certificados digitales de forma óptima. Principalmente, hay que establecer procedimientos para:

- Emisión de certificados digitales.
- Revocación de certificados digitales.
- Consulta de certificados digitales.

La tecnología PKI permite a los usuarios autenticarse frente a otros usuarios y usar la información de los certificados de identidad (por ejemplo, las claves públicas de otros usuarios) para cifrar y descifrar mensajes, firmar digitalmente información, garantizar el no repudio de un envío y otros usos.

En una operación criptográfica que use infraestructura PKI, intervienen conceptualmente como mínimo las siguientes partes:

- Un usuario iniciador de la operación.
- Unos sistemas servidores que dan fe de la ocurrencia de la operación y garantizan la validez de los certificados implicados en la operación (autoridad de certificación, Autoridad de registro y sistema de Sellado de tiempo).

Un destinatario de los datos cifrados/firmados/enviados garantizados por parte del usuario iniciador de la operación (puede ser él mismo). [32]

4.4.1. Modelos de PKI.

4.4.1.1. PKI de la Comunidad Europea (EuPKI).

Creado por la Comunidad Europea, con el objetivo de producir un Software Libre orientado a la Infraestructura de Clave Pública. Para proveer acceso libre, alta calidad modular y un software totalmente abierto, para asegurar el intercambio de información, en las aplicaciones electrónicas de administración y negocios. [33]

EuPKI, fue diseñado para la construcción de una Infraestructura de Clave Pública con las siguientes características:

- Un Diseño completo y modular.
- Facilidad de Mantenimiento.
- Adaptabilidad en cualquier tipo de implementación.
- Basado en estándares actuales para asegurar la interoperabilidad.
- EuPKI es liberado bajo dos licencias que dan un alto nivel de modularidad y que son distribuidos mediante diferentes permisos:
- Licencia Pública General de GNUGPL.
- Licencia Pública de Mozilla MPL.

Una de las características principales de esta plataforma es que puede ser aplicado en cualquier tipo de dominio (financiero, académico, comercio, gobierno, etc.).

Arquitectura.

EuPKI, puede operar un número diferente de Autoridades de Certificación, cada AC tiene asociado un CSP (Crypto Service Provider), la cual tiene una llave privada que es utilizada como marca en las operaciones de los procesos de la AC. La plataforma también puede manejar varias RA (Registration Authority), cada uno definiendo un dominio diferente (namespace), como también políticas de certificación diferentes. El KGS (Key Generation Systems) esta conectado al resto del sistema para facilitar la operación del Software en la

entidad final para la Generación de Llaves, Por otro lado el UKGS, interactúa directamente con la AC (si un entorno en línea es utilizado) o indirectamente vía RA's, El KA (Key Archive), puede ser conectado al central KGS para soportar la recuperación de llaves cifradas.

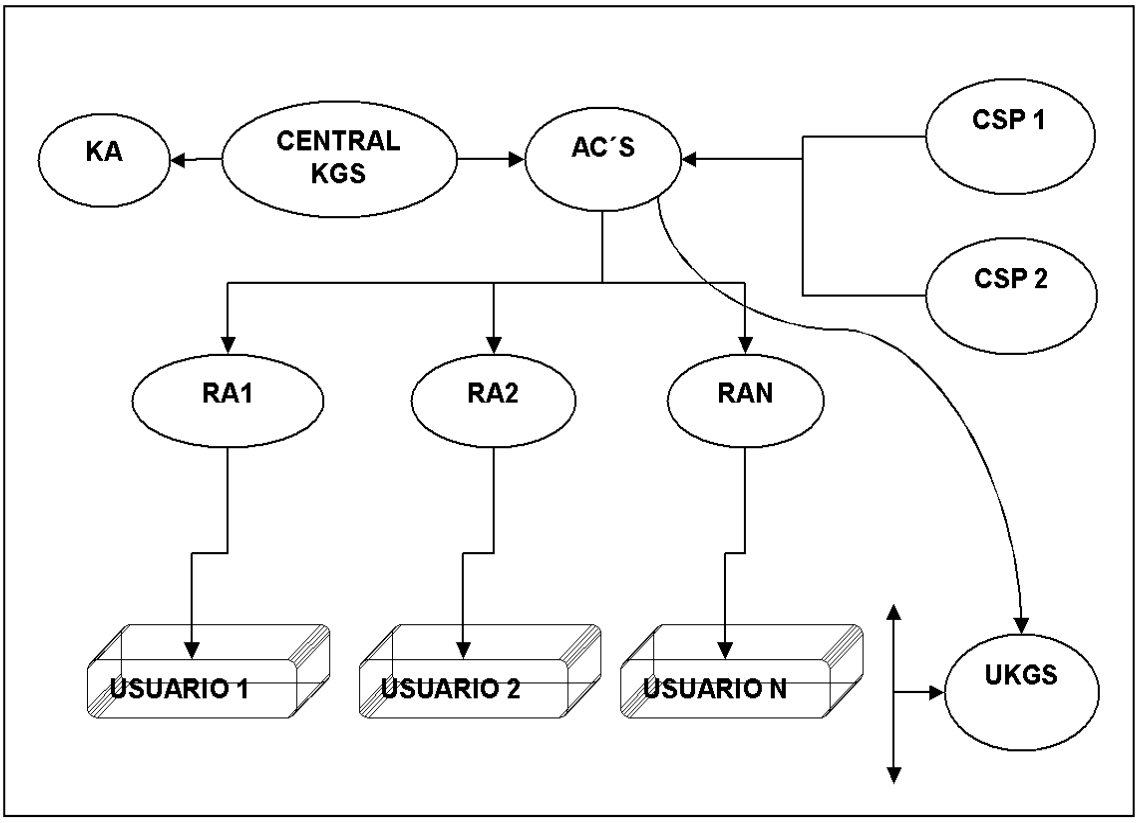

Figura 4.2 Arquitectura del modelo EuPKI

Certificación.

Los sistemas CSP, el sistema AC, puede adquirir las llaves públicas desde la entidad final o desde una central de generación de llaves y distribuye la información en base a directorios, también mantiene un almacén denominado AC DB, para almacenar los certificados, su estado y la información relacionada a la entidad (con una referencia a una entrada en la RA).

La AC para certificar, trabaja además con subsistemas que le ayudan ya sea en el establecimiento de autenticaciones iníciales, o el compartimiento de llaves secretas entre la AC y cualquier entidad.

La AC también soporta requerimientos de registro y certificación en modo de línea de comandos, controlando para esto el KGS, en la obtención de pares de llaves.

4.4.1.2. Pretty Good Privacy (PGP).

Es una aplicación informática de libre distribución, desarrollado por Phil Zimmerman, a inicios de 1990. PGP permite intercambiar archivos y mensajes con confidencialidad, autenticación, integridad y comodidad.

Para realizar esto, PGP utiliza criptografía de clave pública RSA, Diffie-Hellman o DSS para generar firmas y cifrar claves, criptografía de clave secreta IDEA, CAST, TRIPLEDES y AES para el cifrado del documento propiamente dicho y la función *hash* MD5, SHA-1 para crear las huellas digitales que se emplean en la firma digital. [34]

Arquitectura.

PGP es un sistema descentralizado, en el que cada usuario se hace responsable de su archivo de claves públicas, denominado anillo, y es el propio usuario el que establece la confianza en la validez de dichas claves.

Cuando un usuario necesita comunicarse con nuevos usuarios, podrá obtener sus claves a través de personas en las que ya confía, las cuales podrán firmar y enviarle claves públicas de otros usuarios en los que ellos confiaban previamente, estableciendo así una cadena de confianza que se basa en las relaciones personales.

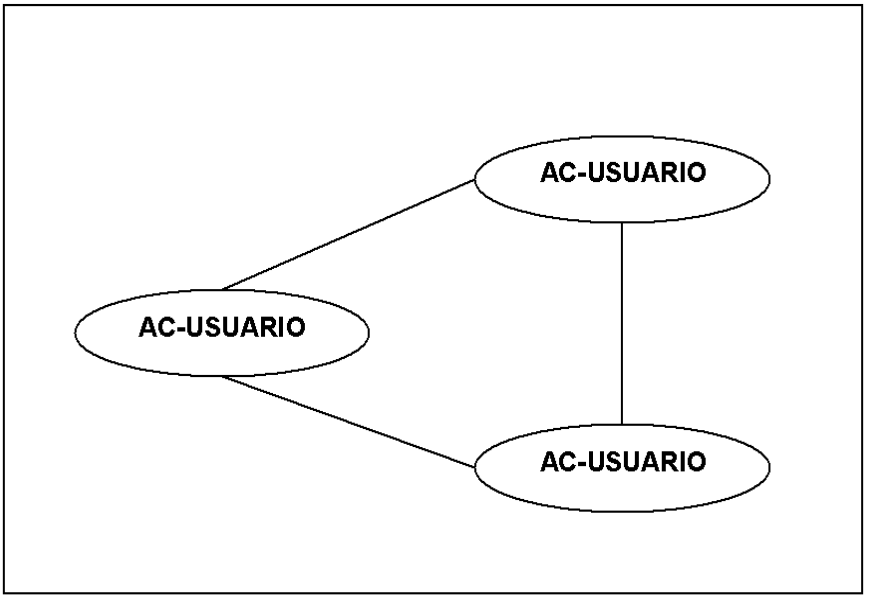

Figura 4.3 Arquitectura del modelo PGP.

Certificación.

PGP utiliza un formato propio de certificado que consta únicamente de un identificador del usuario (que contiene su nombre y algún dato significativo y personal como la dirección de correo electrónico, su número de teléfono o cualquier información que ayude a garantizar la singularidad del identificador), de una fecha de emisión y de la propia clave pública. Como en PGP no existen AC, el certificado es emitido por el propio usuario propietario de la clave pública, y firmado inicialmente por él mismo. A este certificado se le pueden añadir firmas de otros usuarios que se utilizarán como aval de validez. La sencillez del certificado hace que en PGP se hable directamente de claves públicas y no se utilice la palabra certificado.

4.4.1.3. Privacy Enhancement for Internet Electronic Mail (PEM)

Se creó en 1991, a partir de las necesidades de transferir información criptográfica a través del correo electrónico principalmente.

PEM proporciona los servicios de integridad, autenticación y no repudio en origen y opcionalmente confidencialidad.

Lo habitual en PEM es utilizar algoritmos de clave secreta para el cifrado de los datos y algoritmos de clave pública para la administración de claves y la gestión de firmas digitales.

Aunque no es obligatorio el uso de algoritmos de clave pública en PEM, es recomendable hacerlo para aprovechar las ventajas que ofrecen este tipo de algoritmos en cuanto a la administración de claves se refiere, ya que, al contrario de lo que sucede con los algoritmos de clave privada, no es necesario un canal de comunicación seguro para efectuar el intercambio de claves secretas. [35]

Arquitectura.

PEM define una infraestructura de certificación basada en una organización jerárquica arborescente de Autoridades de Certificación.

En PEM, las AC's y los usuarios están organizados en un único árbol, en el cual los usuarios representan las hojas. La raíz del árbol la ocupa la IPRA (*Internet Policy Registration Authority*). El principal cometido de IPRA es el de establecer la política global, la cual se aplicará a todos los procesos de certificación bajo esta jerarquía. IPRA certifica a las Autoridades de Certificación de Políticas, PCAs (*Policy Certification Authorities*), que definen políticas particulares dentro de la política global definida por IPRA. Las PCAs certifican a otras AC que se encargaran de certificar a usuarios finales o a otras AC que se encuentren por debajo en la jerarquía.

Autoridad de Registro de Políticas de Internet (IPRA)

- Registro de PCAs
- Garantía de unicidad de nombres distintivos.
- Corrección de nombres distintivos.
- Convenios sobre nombres distintivos.
- Gestión de CRLs.
- Expedición de licencias de algoritmos de clave pública.

Autoridades de Certificación de Políticas (PCAs)

- Identidad de la PCA
- Alcance de la PCA
- Privacidad y seguridad de la PCA
- Política de Certificación
- Gestión de CRLs
- Convenios de Nombramiento
- Emisión de Negocios

Autoridades de Certificación (AC's)

Las AC's tienen como misión crear y asignar certificados. Aunque la recomendación X.509, impone pocas restricciones sobre las AC's, la implementación práctica del sistema de certificación ha llevado a la necesidad de establecer algún convenio. Por ejemplo, las AC's, deben mantener una base de datos con los DNs de las entidades a las que certifican con el fin de garantizar la unicidad nominal. Las AC's deben establecer medidas de seguridad para proteger su clave secreta, estas medidas pueden estar impuestas por su PCA. Las AC's deberán emitir y enviar CRLs a su PCA en los periodos establecidos por esta. Las AC's podrán definir su propia política de registro de usuarios siempre que éste dentro de lo definido por su PCA.

Dependiendo del tipo de entidades que las AC's certifiquen, en PEM se pueden distinguir tres tipos de AC's: AC's organizacionales, AC's Residenciales, AC's Personas.

Usuarios y agentes de usuario

Como PEM es una especificación para correo electrónico, cuando se habla de usuario PEM, en algunos casos se refiere a un agente de usuario UA (User Agent), término definido por la recomendación X.400, para referenciar al medio por el cual el usuario interactúa con el sistema de manejo de mensajes.

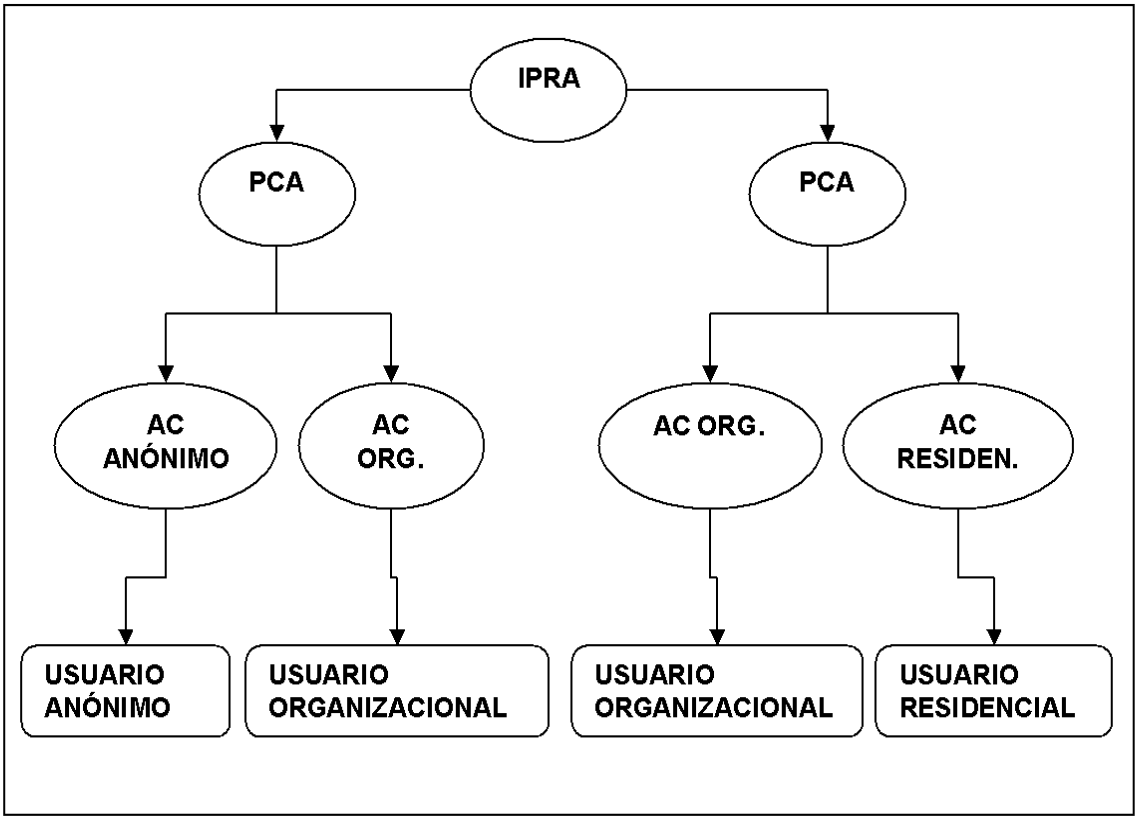

Figura 4.4 Arquitectura del modelo PEM.

Certificación.

PEM utiliza la versión 1 del certificado X.509, y una versión modificada de la CRL versión 1.

Debido a la organización jerárquica de las Autoridades de Certificación definida en PEM en forma de un único árbol, en esta arquitectura existe siempre un punto de confianza común entre dos usuarios, que es la raíz de árbol IPRA. En PEM todos los usuarios están en posesión de la clave pública de IPRA, la cual les es facilitada como parte del procedimiento de registro o en el proceso de instalación del software PEM.

PEM no prevé el uso del Directorio X.500 para el almacenamiento de certificados y CRLs pero deja libertad a las diversas entidades de su arquitectura para utilizarlo.

Cada usuario y AC de PEM conoce el conjunto de certificados que forman su camino de certificación hasta IPRA, ya que cuando una AC emite un certificado lo envía a su propietario incluyendo los certificados que forman el camino de

certificación desde el nuevo certificado hasta IPRA (Internet Policy Registration Authority) y opcionalmente algún certificado cruzado que la AC considere de interés.

El procedimiento habitual en PEM, es que los usuarios en el proceso de envió de mensajes incluyan en campos opcionales el certificado de la clave, pareja de la clave de firma, y en otros todos los certificados que conforman su camino de Certificación hasta IPRA o hasta un nodo común en el árbol. De esta forma el usuario receptor podrá validar la firma comenzando por el certificado de la PCA emitido por IPRA.

Un usuario PEM debería consultar regularmente las CRLs correspondientes, para tener la seguridad de que los certificados que intervienen en el proceso de validación no han sido revocados. PEM también describe procedimientos para la consulta, obtención y almacenamiento de CRLs en las bases de datos de la PCAs.

La tabla 4.2 muestra las características de los diferentes modelos de PKI.

Tabla 4.2 Comparativa de los modelos de PKI.

	PGP	**PEM**	**EuPKI**
Sistema de cifrado utilizado	RSA CAST-IDEA TripleDES MD5 SHA-1	RSA DES	Definido por el Dominio
Soporte Certificados X.509	NO	SI	SI
Tipo Organización Jerárquica	NO	NO	NO
Tipos de Autoridades	Usuarios	IPRA PCA	CA RA

		CA	(*CKGS, UKGS, CSP, KA*)
Aplicaciones Soportadas	Correo Electrónico Transacciones Electrónicas, Administración electrónica	Correo Electrónico	Comercio Electrónico, Administración electrónica

Capítulo 5 AUTORIDADES CERTIFICADORAS

5.1. Nociones preliminares

La Autoridad de Certificación es un usuario privilegiado y único que ha sido reconocido por el resto de los usuarios de un determinado entorno como certificador de las identidades digitales de todos. En forma práctica el reconocimiento de las identidades lo efectúa a través de la emisión de Certificados Digitales. Un Certificado Digital no es más que un Documento Electrónico firmado digitalmente por la Autoridad de Certificación donde consta la Clave Pública del usuario al que hace referencia. Además incluye todos los datos de los Algoritmos utilizados y de los datos del usuario. Esto último se conoce con el nombre de Certificado de Atributos.

En pocas palabras, la Autoridad de Certificación es el órgano responsable de la emisión de los Certificados luego de la verificación, por los métodos que considere en sus Políticas de Certificación, proveedora de la tecnología de cifrado para emisión de las claves y la encargada de publicar las Claves Públicas en los denominados Directorios de Clave Pública.

Para todo ello, la Autoridad de Certificación debe considerarse un órgano perfecto, con medidas de seguridad que infundan la confianza requerida para el éxito de su gestión, proveedor de innovaciones tecnológicas acordes a su gestión y altos niveles de Calidad en lo que hace a la atención y disponibilidad.

La confianza de los usuarios en la AC es importante para el funcionamiento del servicio y justifica la filosofía de su empleo, pero no existe un procedimiento normalizado para demostrar que una AC merece dicha confianza.

Una de las formas por las que se establece la confianza en una AC para un usuario consiste en la instalación en la maquina del usuario (tercero que confía) del certificado auto firmado de la AC en la que se desea confiar.

Si está instalada una AC en el repositorio de AC's de confianza de cada navegador, cualquier certificado firmado por dicha AC se podrá validar, ya que se dispone de la clave pública con la que verificar la firma que lleva el certificado. Cuando el modelo de AC incluye una jerarquía, es preciso establecer explícitamente la confianza en los certificados de todas las cadenas de certificación en las que se confíe. Para ello, se puede localizar sus certificados mediante distintos medios de publicación en internet, pero también es posible que un certificado contenga toda la cadena de certificación necesaria para ser instalado con confianza.

5.2. Funciones de las Autoridades Certificadoras.

5.2.1. Generación y registro de llaves.

Cualquiera que desee firmar digitalmente mensajes o recibir envíos cifrados, debe poseer un par de claves dentro de algún sistema basado en infraestructura de clave pública. Los agentes pueden tener más de un par de claves: uno para el trabajo, otro con efectos administrativos, otro para uso personal, etc. Es más, otras entidades de la red como son las estaciones de trabajo, los servidores, las impresoras, etc., también pueden tener sus pares de claves; de la misma forma que lo harán personas jurídicas e instituciones como pueden ser los departamentos de las empresas, la secretaría de una universidad, la recepción de un hotel, etc.

Una vez, generadas las claves, el usuario debe registrar su clave pública en una Autoridad de Certificación aceptada dentro del escenario en el cual pretende moverse. Para la inscripción sólo tiene que enviar su clave pública y probablemente algún otro documento digital de solicitud firmado con dicha clave. [36]

Satisfechas las condiciones marcadas por la Autoridad de Certificación en su documento público de Política de Emisión de Certificados incluida en su Política de Seguridad, esta autoridad devuelve al solicitante un certificado digital que confirma la validez de su clave pública para actuar dentro del sistema.

5.2.2. Emisión de certificados.

Además de los compromisos que conlleva el ser una Autoridad de Certificación, ésta se compromete a emitir documentos digitales (los certificados) únicos y perfectamente identificables a través de su número de serie. Dichas autoridades también son responsables de mantener un registro seguro y disponible sobre cual es el estado de cada uno de los certificados que emite.

Un certificado digital siempre está en alguno de los siguientes estados:

- ***Activo o Pre-activo***: Por estado pre-activo se entienden aquellos certificados que, generados en un determinado instante, sólo serán válidos en un intervalo de tiempo posterior. Desde el momento en que se genera el certificado y hasta que llega el momento de entrar en vigencia, el certificado está en estado pre-activo. Cuando la fecha en curso cae dentro del intervalo de vigencia de un certificado, en este caso, se dice que está en estado Activo.

- ***Suspendido***: Muchas veces es necesario anular temporalmente la vigencia de un certificado, para ello, la Autoridad de Certificación emisora decide pasarlo al estado de Suspendido. Con ello no se está invalidando de forma irreversible el certificado, sino que se le retira de circulación hasta que se le vuelva a dar el estado de Activo.

- ***Revocado***: Cuando las condiciones que llevaron a la emisión de un certificado cambian antes de que éste expire, y son de importancia suficiente, la Autoridad de Certificación deberá anularlo; para ello, emite un segundo certificado especial, denominado de revocación, por el cual, desde ese instante desautoriza al certificado previo y lo hace de modo irreversible.

- ***Caducado***: Este es el estado final de cualquier certificado y se produce cuando la fecha en curso es posterior a la fecha de caducidad indicada en el propio certificado. El estado de “certificado caducado” no le resta valor histórico ya que, mientras estuvo activo, las operaciones en las que participó eran perfectamente válidas.

5.2.3. Almacenamiento de la clave privada en la AC.

Dado que todo el valor reside en que cada Agencia de Certificación es la única capaz de generar las firmas que llevan su identificador, es muy importante que esas claves privadas se almacenen y gestionen de forma segura. Cualquier fallo en la seguridad de las claves privadas no sólo pone en entredicho a la institución, sino que invalida todos los certificados emitidos por ella.

5.2.4. Mantenimiento de las claves vigentes y revocadas.

Las Autoridades de Certificación pueden, dentro de los servicios que ofrecen al público, almacenar los certificados emitidos durante su periodo de validez. De este modo, en el caso de que uno de los agentes pierda su certificado, siempre podrá pedirle a la autoridad emisora que le envíe de nuevo una copia. También se ofrece este servicio en aquellas Autoridades de Certificación que tienen asociados servidores públicos de certificados mediante los cuales cualquier agente puede solicitar los certificados de cualquiera de los demás agentes.

La disponibilidad pública de los certificados electrónicos, para algunos supone, además de una ventaja, un riesgo. Al mantener expuestas las claves cualquiera podrá obtenerlas y someterlas a un ataque.

Para evitar con cierto éxito (prácticamente total) que estos ataques puedan obtener resultados provechosos para el atacante, todo par de claves pública y privada tienen un tiempo de vida limitado. Este periodo se establece según sea la complejidad computacional del ataque, como se prevé que evolucione la tecnología durante ese tiempo y cual sea el nivel de uso previsto para esa clave.

En cualquier verificación de una firma siempre se debe comprobar la fecha de caducidad y la fecha actual; en ningún caso se deben aceptar mensajes firmados con fecha pasadas.

5.2.5. Servicios de directorio.

En el caso de que alguien quiera encontrar la clave pública de un usuario del sistema, existen diversas formas de conseguirlo: bien por teléfono, por correo de superficie, consultando publicaciones periódicas, etc., sin embargo, estos métodos, aún pudiendo ser muy seguros, adolecen de una lentitud a veces intolerable. Para poder obtener esa misma información a la velocidad habitual de las redes, las Autoridades de Certificación dan Servicios de Directorio mediante los cuales, cualquiera puede obtener la clave pública certificada de cualquier miembro con quien quiere ponerse en contacto o establecer relaciones de algún tipo.

Un servicio de Directorio consiste en una gran base de datos en la que cada entrada de usuario en el directorio contiene los certificados de las claves públicas de las que es titular, y cada entrada de una Autoridad de Certificación contiene todos los certificados emitidos para ella por otras Autoridades de Certificación ante las que está inscrita, y todos los certificados emitidos por ella misma para otras autoridades.

De no existir este tipo de servicios, la distribución de los certificados debería hacerse a través de canales de comunicación ajenos a la red con lo que se tornaría más lenta la velocidad de operación de todos los agentes.

5.3. Política de certificados de la autoridad certificadora raíz de la Secretaría de Economía de México.

La Política de Certificados, es un conjunto de reglas que indican la aplicabilidad de un certificado a una comunidad y clase de aplicaciones con requerimientos comunes de seguridad. [37]

En dicho documento se describe la Política de Certificados para la Autoridad Certificadora Raíz de la Secretaría de Economía (ACR-SE). La Política de Certificados se aplica a la solicitud, validación, aceptación, emisión o revocación de los certificados digitales dentro de una Infraestructura de Clave Pública (PKI).

La ACR-SE, a través de la Dirección General de Normatividad Mercantil (DGNM), certificará las claves públicas de las Autoridades Certificadoras que hallan sido acreditados por la DGNM.

La Autoridad Certificadora Raíz de la SE, se establece para crear y desarrollar una PKI a nivel nacional para el desarrollo del comercio electrónico.

5.3.1. Comunidad y aplicabilidad de la ACR-SE

La comunidad y aplicabilidad de la ACR-SE están determinadas en esta Política de Certificados. La ACR-SE emitirá certificados de identidad personal (CD-IPAC) para sus agentes certificadores; certificados digitales de autoridad certificadora (CD-ACPSC) a las autoridades certificadoras de las personas físicas y morales de carácter privado o público que hayan sido acreditados como Prestadores de Servicios de Certificación por la DGNM; a la Autoridad Certificadora (CD-ACSIGER) a del SIGER para el ámbito del Registro Público de Comercio y a las autoridades certificadoras de las áreas que integran a la Secretaría de Economía.

5.3.2. Requerimientos de seguridad para la ACR-SE y sus claves.

- La ACR-SE operará en un servidor de misión crítica redundante desconectado de la red, el intercambio de información con sus entidades subordinadas será mediante dispositivos de almacenamiento removible, únicamente para efectos de certificación de las mismas.

- El intercambio de información entre el servidor Web de la ACR-SE con los Autoridades Certificadoras Subordinadas, será en los términos establecidos en las reglas de la 5 a la 5.3 de las RGPSC.

- La clave privada de la ACR-SE estará en todo momento cifrada, en un dispositivo de alta seguridad que cumpla con la norma FIPS 140-2 nivel 3.

- Tanto el *hardware* como el *software* que opera la ACR-SE se mantendrá en todo momento físicamente seguro.

- El par de claves RSA de la ACR-SE tendrá una longitud de 2048 bits.

- Se establecerá un procedimiento periódico de respaldo de los servidores que opere la ACR-SE. Las copias se guardarán en un lugar seguro, protegido de accesos no autorizados.

- Si la clave privada de la ACR-SE estuviera comprometida, se procedería a la revocación de la misma y del certificado de la ACR-SE, así como todos los certificados emitidos por ella, no importando la fecha de emisión. A partir de ese momento, deberán revocarse todos los certificados emitidos por las Autoridades Certificadoras Subordinadas ala ACR-SE y no deberán emitir certificados válidos hasta que no se restaure la identidad de la ACR-SE y se vuelvan a generar certificados respectivos a las Autoridades Certificadoras Subordinadas.

5.3.3. Requerimientos de seguridad impuestos a las autoridades certificadoras subordinadas y sus claves.

- Las Autoridades Certificadoras operarán en un servidor de misión crítica redundante.

- Éste servidor podrá estar conectado a la red, en tal caso, el intercambio de información se hará entre el servidor y sus usuarios por lo menos vía SSL o la tecnología que ofrezca mayor seguridad, asimismo, deberá deshabilitar todos los servicios de red que no se requieran para el buen funcionamiento del servicio, manteniendo seguros y monitoreados aquellos que sean necesarios.

- La clave privada de la Autoridad Certificadora estará cifrada en un dispositivo que cumpla con el estándar FIPS 140 nivel 3.

- Tanto el hardware como el software del servidor de misión crítica que opera la Autoridad Certificadora se mantendrá en todo momento físicamente seguro.

- El par de claves RSA de una Autoridad Certificadora tendrá como mínimo una longitud de 2048 bits.

- El par de claves RSA de los certificados emitidos por las Autoridades Certificadora Subordinadas tendrá como mínimo una longitud de 1024 bits.

5.3.4. Periodos de validez de los certificados digitales.

El período de validez del Certificado Digital de la ACR-SE no será menor a 10 años a partir de su fecha de emisión.

El período de validez de los Certificados Digitales de Autoridad Certificadora subordinada no será menor de 10 años a partir de su fecha de emisión, igualmente para los certificados de servidor.

Cuando se haya superado cuatro quintos del tiempo de vida de la ACRSE, se generará un nuevo certificado digital y en su caso una nueva identidad. A partir de ese momento, las nuevas inscripciones se harán firmando certificados con esa nueva identidad. De este modo las Autoridades Certificadoras Subordinadas dispondrán de una quinta parte del tiempo para solicitar nuevos certificados a la nueva identidad.

5.3.5. Disposición de certificados.

Cada Autoridad Certificadora debe mantener un repositorio o base de datos con los certificados que emita, de manera que estén disponibles al público a través de un servicio de distribución de certificados.
Así mismo, la ACR-SE mantendrá constancia, en las páginas Web habilitadas para tal fin, de los certificados emitidos o revocados por ésta.

5.3.6. Abreviaturas encontradas en la Política de Certificados para la Autoridad Certificadora Raíz de la Secretaría de Economía (ACR-SE).

AC	Autoridad Certificadora.
ACR-SE	Autoridad Certificadora Raíz de la Secretaría de Economía.
CD-IPAC	Certificados de identidad personal.
CD-ACPSC	Certificados digitales de autoridad certificadora.
CD-ACSIGER	Autoridad Certificadora de la SIGER.
DGNM	Dirección General de Normatividad Mercantil.
PSC	Prestadores de Servicios de Certificación.
RGPSC	Reglas Generales de Prestadores de Servicios de Certificación.

5.3.7. Procedimiento de emisión de certificados digitales por la Autoridad Certificadora.

La emisión de un certificado digital firmado por la ACR-SE se hará bajo el procedimiento descrito a continuación:

1. El PSC, deberá designar al profesional informático y responsable directo de la Autoridad Certificadora; Las Instituciones Públicas Gubernamentales designarán al titular del área responsable que emitirá sus certificados; el cual deberá mantener una relación permanente con la ACR-SE.

2. El PSC deberá presentar su documento mediante el cual fue acreditado por la DGNM de conformidad con lo requerido en el trámite SE-09-026-B. Para la identificación fehaciente del titular del certificado, se requerirá su presencia física y deberá presentar una identificación oficial vigente como el pasaporte, credencial del IFE o cedula profesional.

3. La DGNM remitirá dicha información a la ACR-SE, la cual analizará la información requerida en el punto anterior, para determinar si procede o no la emisión del certificado.

4. Estas solicitudes quedarán en poder del ACR-SE. Es responsabilidad de la ACR-SE comprobar que dichas solicitudes están debidamente requisitadas y que todos los datos que aparecen en las mismas son correctos.

5. Confirmada la autenticidad y validez del o los documentos presentados por el PSC, la ACR-SE verificará la razonable coincidencia entre la fotografía contenida en aquellas y la apariencia física del solicitante.

6. La ACR-SE requerirá a las Autoridades Certificadoras subordinadas que firme original y copia del documento de solicitud para verificar la firma autógrafa del documento de solicitud con la que aparece en las credenciales oficiales presentadas, después de lo cual procederá también a la firma autógrafa de la solicitud, considerada a partir de ese momento como aceptada.

7. La ACR-SE, emitirá el certificado correspondiente con el precertificado que presentarán las Autoridades Certificadoras Subordinadas de las siguientes formas:

   a) Las Autoridades Certificadoras Subordinadas se autocertificarán su AC, en el nivel más seguro de sus instalaciones, dicho certificado será presentado en un medio de almacenamiento removible, el cual será certificado por la ACR-SE.

   b) Las Autoridades Certificadoras Subordinadas emitirán su precertificado en PKCS#10 en su AC, en el nivel más seguro de sus instalaciones, dicho certificado será presentado en un medio de almacenamiento removible, el cual será certificado por la ACR-SE.

5.4. VeriSign.

VeriSign es el proveedor de certificados SSL elegido por el 93% de las empresas integrantes de la lista Fortune 500 y los 40 bancos principales del

mundo, empresas dedicadas al comercio electrónico. Además es la primera compañía en ofrecer certificados digitales en el mundo.

VeriSign es una empresa de seguridad informática famosa por ser una autoridad de certificación reconocida mundialmente. Emite certificados digitales RSA para su uso en las transmisiones seguras por SSL, principalmente para la protección de sitios en internet en su acceso por https. [38]

Sin duda a medida que el mundo digital evoluciona, VeriSign ha tenido que ir a la par, para poder brindar los servicios de protección de voz y datos de todo el mundo, logrando procesar nada menos que 31,000 millones de direcciones Web y correos electrónicos cada día.

5.4.1. Ejemplo del uso de los certificados de VeriSign.

Cuando se conecta mediante un explorador habilitado para SSL con un servidor Web de banca en línea que tiene un certificado de servidor de una entidad emisora de certificados como VeriSign, se producen los siguientes sucesos:

- Tiene acceso a la página Web de inicio de sesión protegida de su banco, utilizando su explorador Web. Si utiliza Internet Explorer, en la esquina inferior derecha de la barra de estado del explorador aparece un icono de candado cerrado, para indicar que el explorador está conectado a un sitio Web seguro. Otros exploradores indican que las conexiones son seguras de otra forma.

- El servidor Web del banco envía automáticamente un certificado de servidor a su explorador Web.

- Para autenticar el servidor Web, su explorador Web comprueba el almacén de certificados de su equipo. Si la entidad emisora de certificados que emitió el certificado a su banco es de confianza, la transacción puede proseguir, y el certificado del banco queda almacenado en su almacén de certificados.

- Para cifrar todas las comunicaciones con el servidor Web del banco, su explorador Web crea una clave de sesión exclusiva. Su explorador Web cifra la clave de sesión con el certificado del servidor Web del banco para que sólo el servidor Web del banco pueda leer los mensajes que envía desde su explorador. (Algunos de estos mensajes contendrán su nombre y contraseña de inicio de sesión, y más información sensible, de modo que este nivel de seguridad es necesario).

- Se establece la sesión segura, y se puede enviar información sensible entre su explorador Web y el servidor Web del banco de forma segura.

5.5. Thawte.

Thawte es una filial de VeriSign. Es la más antigua y reconocida marca en certificados digitales. Thawte es una marca largamente reconocida por usuarios a lo largo de todo el planeta, utilizando certificados Thawte, se tiene la confiabilidad de un sito Web y le permite a los usuarios ofrecer mayores garantías a sus visitantes. Las características del certificado SSL-123 son las siguientes:

- Encriptación de 256 bit.
- El mayor radio de reconocimiento por parte de los navegadores.
- Se expiden en el acto.
- Verificación de todos los dominios.
- Sello de "Thawte Trusted Site" para la página Web.
- Reexpediciones de certificado ilimitadas.
- Soporta IDN. [39]

5.5.1. Certificados SSL de servidor Web.

El Certificado de servidor SSL de Thawte ofrece procedimientos exhaustivos de autenticación (verificación de identidad y nombre de dominio). Ofrece también codificación de 256, 128, 56 o 40 bits según la capacidad del navegador del cliente y el sistema de codificación instalado en el servidor. Esto garantiza que la información se mantenga en privado entre el servidor Web y los navegadores de los clientes.

Capítulo 6 IMPLEMENTACIÓN DE CERTIFICADO EN HTTPS, SSH Y SFTP

6.1. Nociones Preliminares

Como parte de la investigación, se planteo la instalación de un certificado en un servidor de pruebas con sistema operativo *Ubuntu 8.4 Hardy Heron*, el cual será un servidor dedicado para brindar tres tipos de servicios, como son servidor Web, autenticación por medio de SSH y un servidor FTP seguro mediante SFTP. Para el servicio SSH y SFTP se lleva a cabo la utilización de una infraestructura de llaves publicas/privadas.

Para dicho servidor se llevo a cabo la instalación de una tienda de comercio electrónico, la cual permitirá el acceso a los clientes y llevar a cabo sus compras en línea, haciendo uso de tarjetas de crédito, teniendo la seguridad de que sus datos serán enviados de forma segura (cifrados) mediante SSL. Además de que el cliente tenga la certeza de que sus datos viajan seguros entre el navegador y el servidor Web, el modulo de administración de la tienda, también permitirá la autenticación mediante contraseña, logrando con esto agregar un plus al sistema, ya que no solo serán cifrados los datos sensibles del administrador, sino que se utiliza la autenticación por medio de usuario y contraseña, mismos datos que están almacenados en el servidor fuera del alcance de cualquier usuario anónimo.

Cabe mencionar que las herramientas utilizadas en esta implementación son OpenSource, por lo cual no se lleva a cabo ningún gasto extra, ni mucho menos es necesario la utilización de licencias, además de que existen procedimientos bien documentados sobre la instalación de dichos servicios en diferentes distribuciones de sistemas operativos. Aunado a esto existen foros públicos, en los cuales se resuelve en la mayoría de los casos los problemas más frecuentes.

6.2. Requerimientos de software.

Nombre del Servidor:	Servidor (Servidor)
Conexión a Base de Datos:	localhost (127.0.0.1)
Sistema Operativo del Servidor:	Linux 2.6.24-19-generic
Base de Datos:	MySQL 5.0.51a-3ubuntu5.2
Servidor HTTP:	Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.3 con el parche Suhosin-Patch mod_ssl/2.2.8 OpenSSL/0.9.8g
Versión PHP:	5.2.4-2ubuntu5.3 (Zend: 2.2.0)
Sistema de Comercio Electrónico 1:	Oscommerce V2.2 RC2
Sistema de Comercio Electrónico 2:	ZenCart 1.3.8

6.3. Instalación de un certificado autofirmado en servidor Web seguro HTTPS.

Instalar apache2:

```
rovskyhp@Servidor: sudo apt-get install apache2 apache2.2-common
```

Habilitar el modulo para el manejo de SSL:

```
rovskyhp@Servidor: a2enmod ssl
```

Acceder a la carpeta ssl.

```
rovskyhp@Servidor: cd /etc/apache2/ssl
```

Ejecutar un script para crear el certificado de seguridad autofirmado para el servidor esto genera un certificado valido por 365 días.

```
rovskyhp@Servidor:/etc/apache2/ssl$ sudo apache2-ssl-certificate --
force -days 365
```

Lo cual mostrará en pantalla las siguientes líneas y hará unas preguntas necesarias para la creación del certificado.

```
creating selfsigned certificate
replace it with one signed by a certification authority (CA)
enter your ServerName at the Common Name prompt
If you want your certificate to expire after x days call this programm
with -days x
Generating a 1024 bit RSA private key
.....................................................++++++
.........++++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:MX
State or Province Name (full name) [Some-State]:Michoacan
Locality Name (eg, city) []:Morelia
Organization Name (eg, company; recommended) []:UVAQ
Organizational Unit Name (eg, section) []:ISC
server name (eg. ssl.domain.tld; required!!!) []:localhost
Email Address []:rovskyhp@localhost.com
```

Esto crea el archivo apache.pem en el directorio /etc/apache2/ssl
Crear el directorio que será utilizado como DocumentRoot para el servidor web.

```
rovskyhp@Servidor:$ sudo mkdir /var/www/html
```

Ahora se crea la configuración del sitio SSL, tomando como base la que existe por defecto al haber instalado apache2.

Acceder a la carpeta sites-available en donde se encuentra instalado apache2.

```
rovskyhp@Servidor:$ cd /etc/apache2/sites-available
```

Ahí se tiene que copiar el archivo default que tiene la configuración por default para el servidor web, nombrando al archivo creado ssl, además deberán de crearse los enlaces simbólicos para habilitar los sitios que están disponibles en el directorio sites-available.

```
rovskyhp@Servidor:/etc/apache2/sites-available$ sudo cp -afr default
ssl

rovskyhp@Servidor:/etc/apache2/sites-available$ sudo ln -s
/etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

Para este servidor de pruebas se pretende tener instalado una tienda electrónica, que contará con un certificado SSL en su modulo de administración además del modulo de compras del cliente, para ello habrá que modificar los archivos antes creados en el directorio sites-available.

Modificar el archivo /etc/apache2/sites-available/default, especificar el puerto que atenderá este VirtualHost que por defecto será el 80, además se deberá especificar el documento raíz para el servidor web y crear los directorios que se quieren proteger de usuarios anónimos.

```
rovskyhp@Servidor:/etc/apache2/sites-available$ sudo vim default
```

```
NameVirtualHost *:80
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName localhost
        DocumentRoot /var/www/html/
        DirectoryIndex index.php
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/html/tienda/catalog/admin/>
                deny from all
        </Directory>
        <Directory /var/www/html/tiendazencart/admin/>
                deny from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog /var/log/apache2/error.log
```

```
      # Possible values include: debug, info, notice, warn, error,
crit,
      # alert, emerg.
      LogLevel warn

      CustomLog /var/log/apache2/access.log combined
      ServerSignature On

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

</VirtualHost>
```

Modificar el archivo /etc/apache2/sites-available/ssl, en el cual se deberá especificar el puerto que atenderá este VirtualHost en este caso será el 443, además especificar el documento raíz para el servidor web y crear los directorios permitidos para acceder mediante SSL.

```
      rovskyhp@Servidor:/etc/apache2/sites-available$ sudo vim ssl
```

En este archivo es en donde se tiene que especificar el uso de un certificado, ya sea autofirmado, o firmado por una Autoridad Certificadora. Y se deben agregar las líneas:

```
ServerSignature On
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.pem
```

```
NameVirtualHost *:443
<VirtualHost *:443>
      ServerAdmin webmaster@localhost
      ServerName localhost
      DocumentRoot /var/www/html/
        DirectoryIndex index.php
      <Directory />
            Options FollowSymLinks
            AllowOverride None
      </Directory>
      <Directory /var/www/html/tienda/catalog/admin>
            AllowOverride AuthConfig
            Order deny,allow
      </Directory>

      <Directory /var/www/html/tiendazencart/admin>
            deny from all
            AllowOverride AuthConfig
            Order deny,allow
      </Directory>

      ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

```
      <Directory "/usr/lib/cgi-bin">
            AllowOverride None
            Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
            Order allow,deny
            Allow from all
      </Directory>

      ErrorLog /var/log/apache2/error.log

      # Possible values include: debug, info, notice, warn, error,
crit,
      # alert, emerg.
      LogLevel warn
      CustomLog /var/log/apache2/access.log combined
      ServerSignature On
      SSLEngine on
      SSLCertificateFile /etc/apache2/ssl/apache.pem

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>
</VirtualHost>
```

Ahora se debe configurar los puertos que tiene que atender el servidor web apache, modificando el archivo /etc/apache2/ports.conf. Siempre y cuando no exista ya la línea Listen 443 se deberá agregar al archivo. Para el servidor de pruebas la línea ya existía.

```
      rovskyhp@Servidor:/etc/apache2/ $ sudo vim ports.conf
```

```
Listen 80
<IfModule mod_ssl.c>
    Listen 443
</IfModule>
```

Después de realizada la configuración del servidor web solo basta reiniciar el servicio.

```
      rovskyhp@Servidor:/etc/apache2/ $ sudo service apache2 restart
      *Restarting web server apache                          [ OK ]
```

Oscommerce V2.2 RC2.

Se deberá instalar Oscommerce utilizando algún tutorial en línea que permita tener una instalación funcional, ya que a esta investigación lo que le concierne es la instalación del certificado, por lo que solo se menciona la configuración para que el modulo de administración de OsCommerce funcione sobre el servidor seguro haciendo uso del certificado autofirmado.

Configuración de Oscommerce V2.2 RC2.

Modificar dentro del directorio de instalación, el archivo /directorioDeInstalacion/catalog/includes/configure.php, agregando una s a la variable HTTPS_SERVER y cambiando el campo de ENABLE_SSL por true.

```
define('HTTP_SERVER', 'http://192.168.2.186');
define('HTTPS_SERVER', 'https://192.168.2.186');
define('ENABLE_SSL', true);
```

Modificar dentro del directorio de instalación, el archivo /directorioDeInstalacion/catalog/admin/includes/configure.php, agregando una s a las variables y cambiando el campo de ENABLE_SSL_CATALOG por true.

```
define('HTTP_SERVER', 'https://192.168.2.186');
define('HTTP_CATALOG_SERVER', 'https://192.168.2.186');
define('HTTPS_CATALOG_SERVER', 'https://192.168.2.186');
define('ENABLE_SSL_CATALOG', 'true');
```

Recargar la configuración del servidor web.

```
rovskyhp@Servidor:$ sudo service apache2 force-reload
*Reloading web server config apache2                              [ OK ]
```

Abrir en un navegador el modulo de administración.

Para el caso de este servidor el sistema OsCommerce se encuentra instalado en /var/www/html/tienda/catalog.

```
      https://192.168.2.186/tienda/catalog/admin
```

Se observa que al ser un certificado autofirmado, el navegador nos informa que el sitio no proporciona información de identidad, esto no sucede cuando el certificado es emitido por una Autoridad Certificadora reconocida tal como VeriSign o Thawte.

Como se puede observar en la figura 6.1 y figura 6.2, los navegadores no reconocen el certificado autofirmado, para lo cual el usuario debe confirmar que desea acceder a esa página, de lo contrario el certificado será rechazado y no podrá realizar la conexión a la página solicitada.

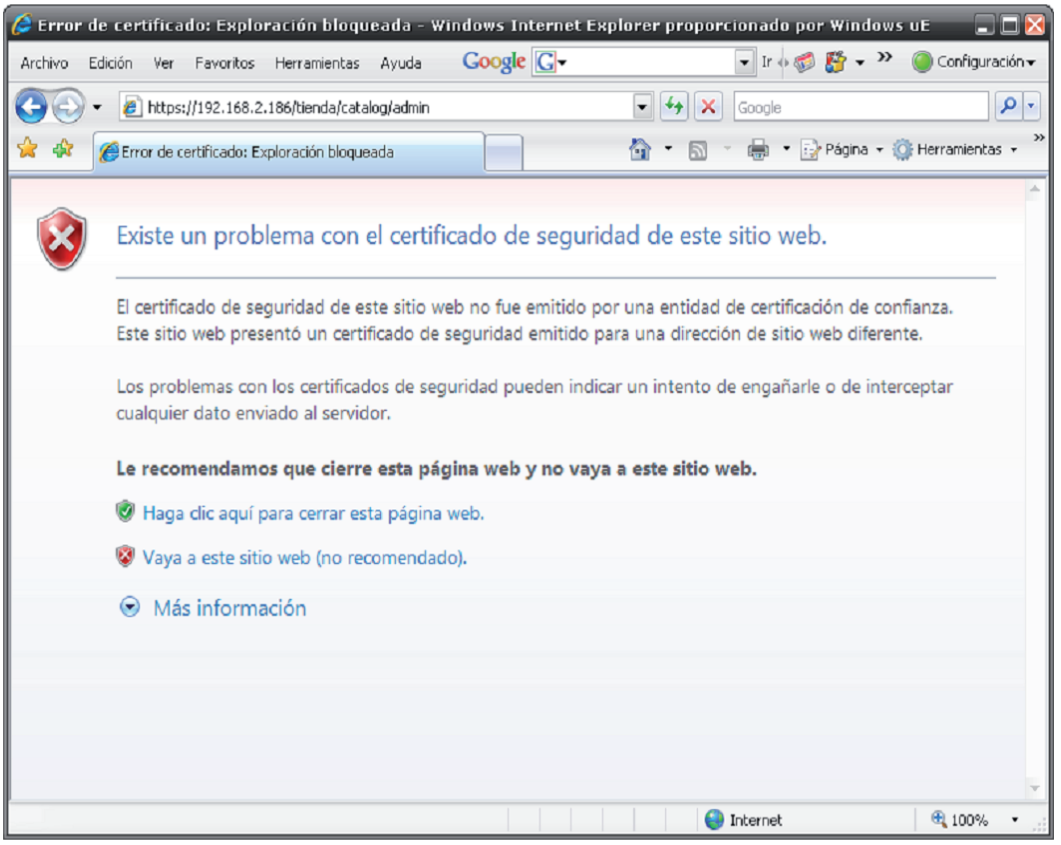

Figura 6.1 Se muestra el error del certificado autofirmado en Internet Explorer.

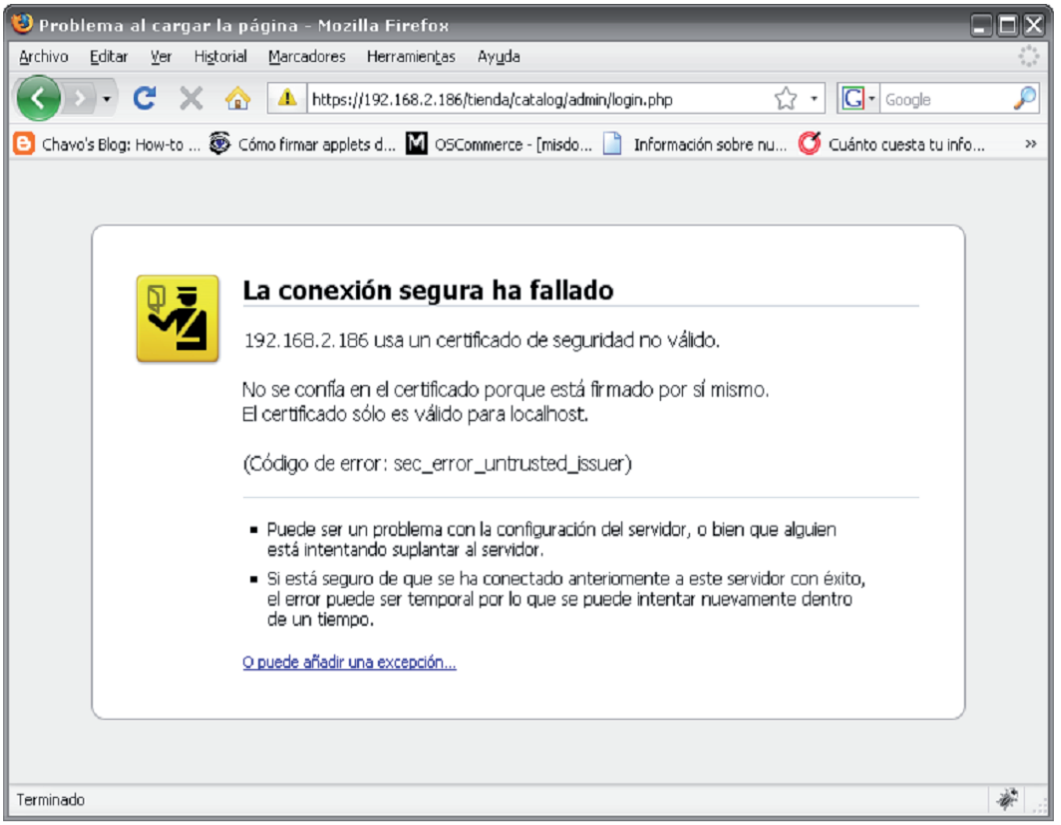

Figura 6.2 Se muestra el error del certificado autofirmado en Mozilla Firefox.

En la siguiente figura 6.3 se puede observar que los navegadores cuentan con herramientas para agregar excepciones en caso de que el usuario está seguro

de la identidad del servidor, aquí es importante mencionar que la excepción puede ser temporal o permanente, de esta forma nunca tendrá que realizar la operación de validar la autenticidad del servidor con el cual pretende establecer la conexión.

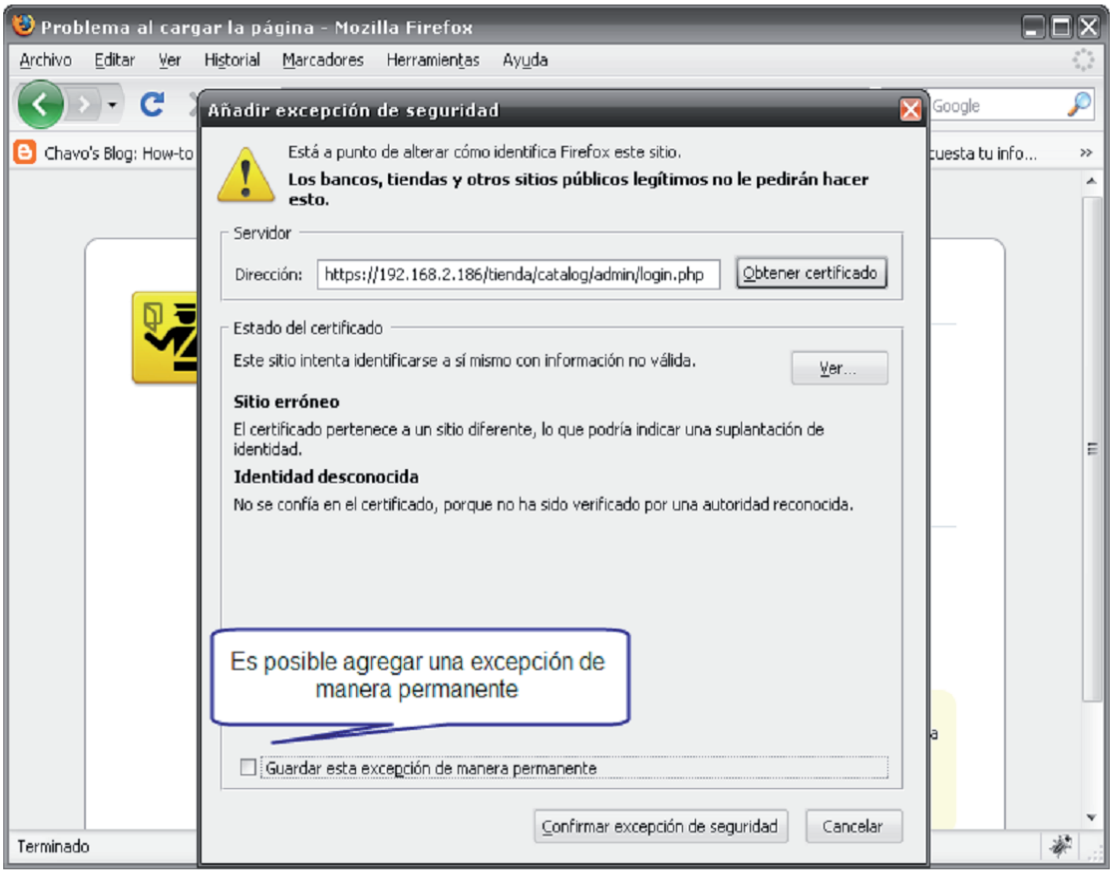

Figura 6.3 Añadir excepciones a certificado autofirmado utilizando Mozilla Firefox.

Es posible ver el certificado en un navegador, esto con la finalidad de que el cliente pueda verificar la autenticidad del servidor al que intenta conectarse, en dicho visor se muestran algunos de los campos del certificado que permite la conexión segura con el servidor mediante HTTPS, tal como se observa en la figura 6.4.

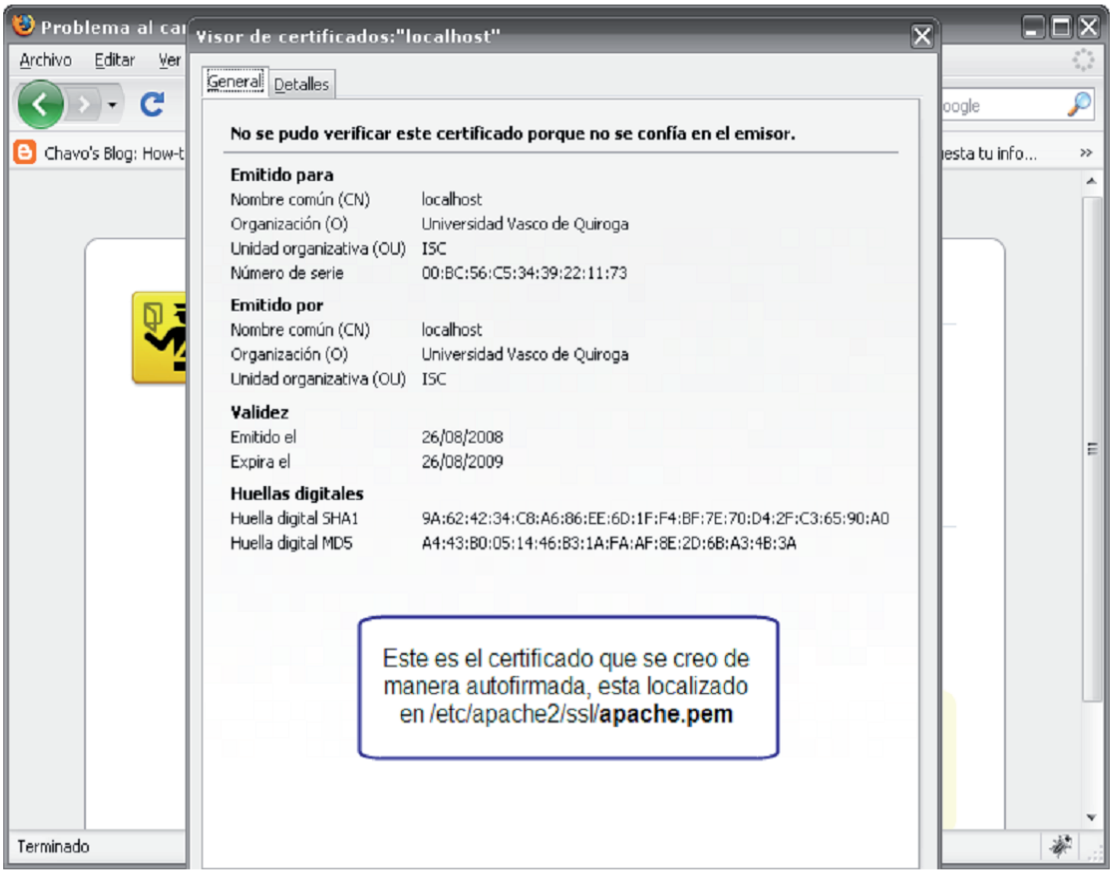

Figura 6.4Visualización del contenido del certificado autofirmado.

Una vez que se ha agregado la excepción y que se ha aceptado el certificado, es decir que se ha importado el certificado del servidor en el navegador del cliente, se permite el acceso al modulo protegido por medio de HTTPS, esto se puede observar en la figura 6.5.

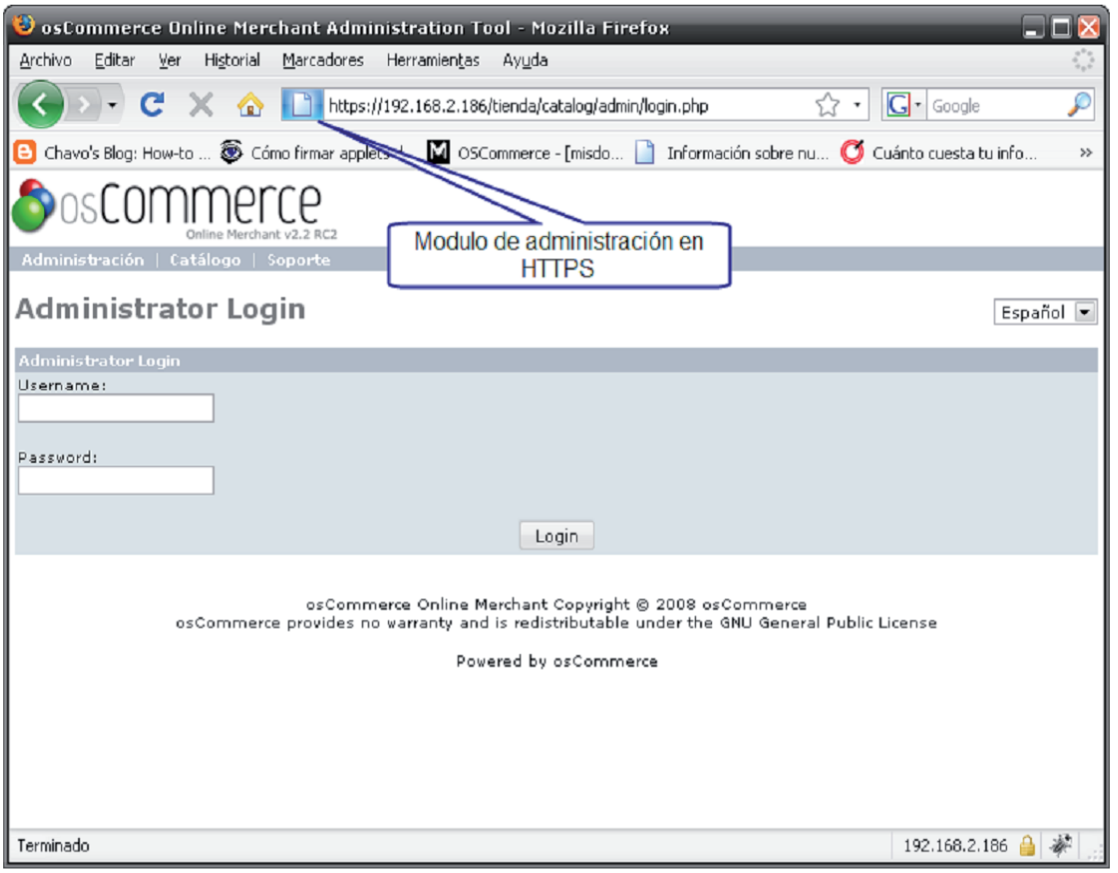

Figura 6.5 Se muestra la navegación segura mediante HTTPS en el servidor remoto.

Para este servidor de pruebas, el modulo del cliente es atendido por el protocolo HTTP, es decir en este modulo no existe una conexión cifrada, como se muestra en la figura 6.6. Dicho modulo también cuenta con un apartado de navegación segura, la sección de compras electrónicas y el pago con tarjeta de crédito son atendidas por el protocolo HTTPS como se observa en la figura 6.7.

Figura 6.6 Pagina del cliente, en ella se observa que se esta utilizando el protocolo HTTP.

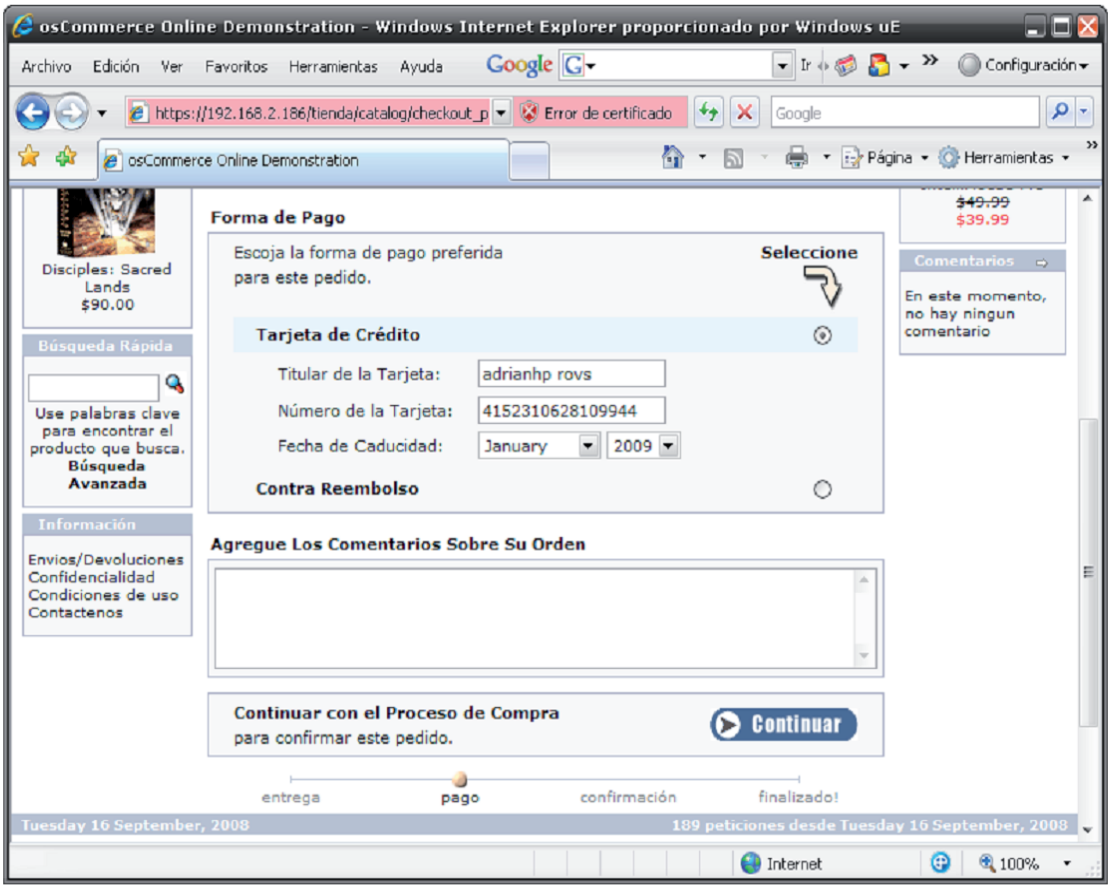

Figura 6.7 Pagina del cliente realizando la compra electrónica, la información sensible viaja cifrada.

ZenCart 1.3.8.

Se deberá instalar ZenCart utilizando algún tutorial en línea que permita tener una instalación funcional, ya que en esta investigación lo que concierne es la instalación del certificado, por lo que solo se menciona la configuración para que el modulo de administración de ZenCart funcione sobre el servidor seguro haciendo uso del certificado autofirmado. Para este caso en particular se pretende proteger el modulo de administración por medio de un certificado autofirmado, pero además brindar de seguridad haciendo uso de una contraseña para el acceso a dicho modulo.

Configuración de ZenCart 1.3.8.

Modificar dentro del directorio de instalación, el archivo /directorioDeInstalacion/includes/configure.php, agregando una s a la variable HTTPS_SERVER y cambiando el campo de ENABLE_SSL por true.

```
define('HTTP_SERVER', 'http://192.168.2.186');
define('HTTPS_SERVER', 'https://192.168.2.186');
define('ENABLE_SSL', true);
```

Modificar dentro del directorio de instalación, el archivo /directorioDeInstalacion/ /admin/includes/configure.php, agregando una s a las variables y cambiando el campo de ENABLE_SSL_CATALOG por true, al igual que el campo ENABLE_SSL_ADMIN.

```
define('HTTP_SERVER', 'https://192.168.2.186');
define('HTTPS_SERVER', 'https://192.168.2.186');
define('HTTP_CATALOG_SERVER', 'https://192.168.2.186');
define('HTTPS_CATALOG_SERVER', 'https://192.168.2.186');
define('ENABLE_SSL_CATALOG', 'true');
define('ENABLE_SSL_ADMIN', 'true');
```

Recargar la configuración del servidor web.

```
rovskyhp@Servidor:$ sudo service apache2 force-reload
*Reloading web server config apache2                    [ OK ]
```

Proteger por contraseña el acceso al modulo de administración.

Además del certificado, se opto por proteger el modulo de administración mediante una contraseña la cual será almacenada en un archivo llamado zencart.passwd, que no podrá ser visible ya que estará almacenado fuera del directorio raíz del servidor web. Para ello se deberán seguir los siguientes pasos.

Creación del archivo que contendrá la ubicación del password para el modulo de administración, en dicho archivo se agregan algunas preferencias para la autenticación.

```
rovskyhp@Servidor:$ sudo vim directorioDeInstalacion/admin/.htaccess
```

Agregar las siguientes líneas al archivo y guardarlo al finalizar:

```
AuthType Basic
AuthUserFile /var/www/conf/zencart.passwd
AuthName "Panel de control"
require valid-user
satisfy any
```

Antes de la creación del archivo de contraseñas, se debe crear la carpeta en donde se almacenara el archivo de contraseñas, este directorio se creara en un directorio fuera del directorio raíz del servidor web, con la finalidad de que no pueda ser accedido por ningún usuario, para ello se debe seguir el siguiente paso:

Creación de la carpeta que contendrá el archivo de contraseñas:

```
rovskyhp@Servidor:$ sudo mkdir /var/www/conf/
```

Creación del archivo de contraseñas, haciendo uso de una utilería del servidor web apache:

```
rovskyhp@Servidor:$ htpasswd -c /var/www/conf/zencart.passwd admin
New password:ESCRIBIR EL PASSWORD
Re-type new password:CONFIRMAR EL PASSWORD
Adding password for user admin
```

Recargar la configuración del servidor web.

```
rovskyhp@Servidor:$ sudo service apache2 force-reload
*Reloading web server config apache2                              [ OK ]
```

Abrir en un navegador el modulo de administración.

Para el caso de este servidor el sistema ZenCart se encuentra instalado en /var/www/html/tiendazencart/

Para poder acceder es necesario ingresar la dirección:

```
http://192.168.2.186/tiendazencart/admin
```

Como se puede observar al intentar ingresar al modulo de administración se muestra un formulario en el cual se deberá ingresar el usuario y el password, como se muestra en la figura 6.8.

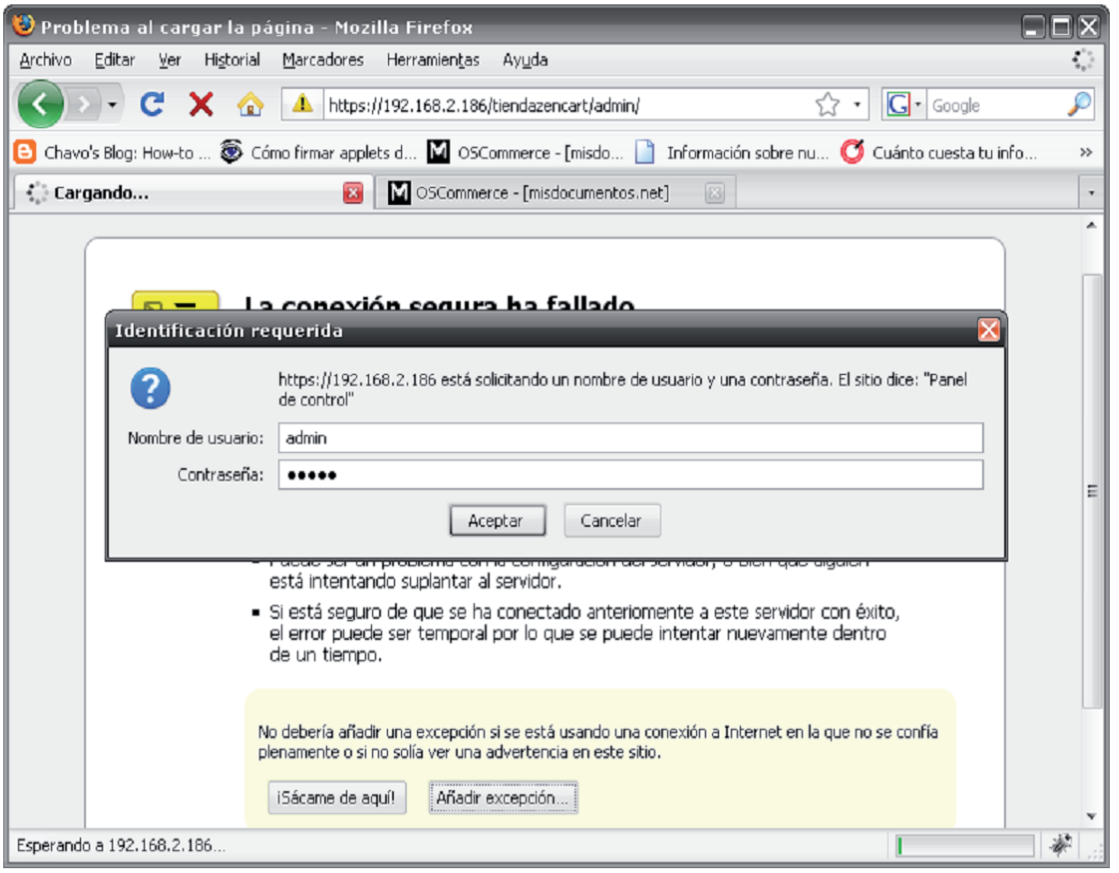

Figura 6.8 Autenticación al modulo de administración de la tienda electrónica mediante usuario/password.

Como se puede observar al intentar ingresar al modulo de administración se muestra un formulario en el cual se deberá ingresar el usuario y el password, como se muestra en la figura 6.9.

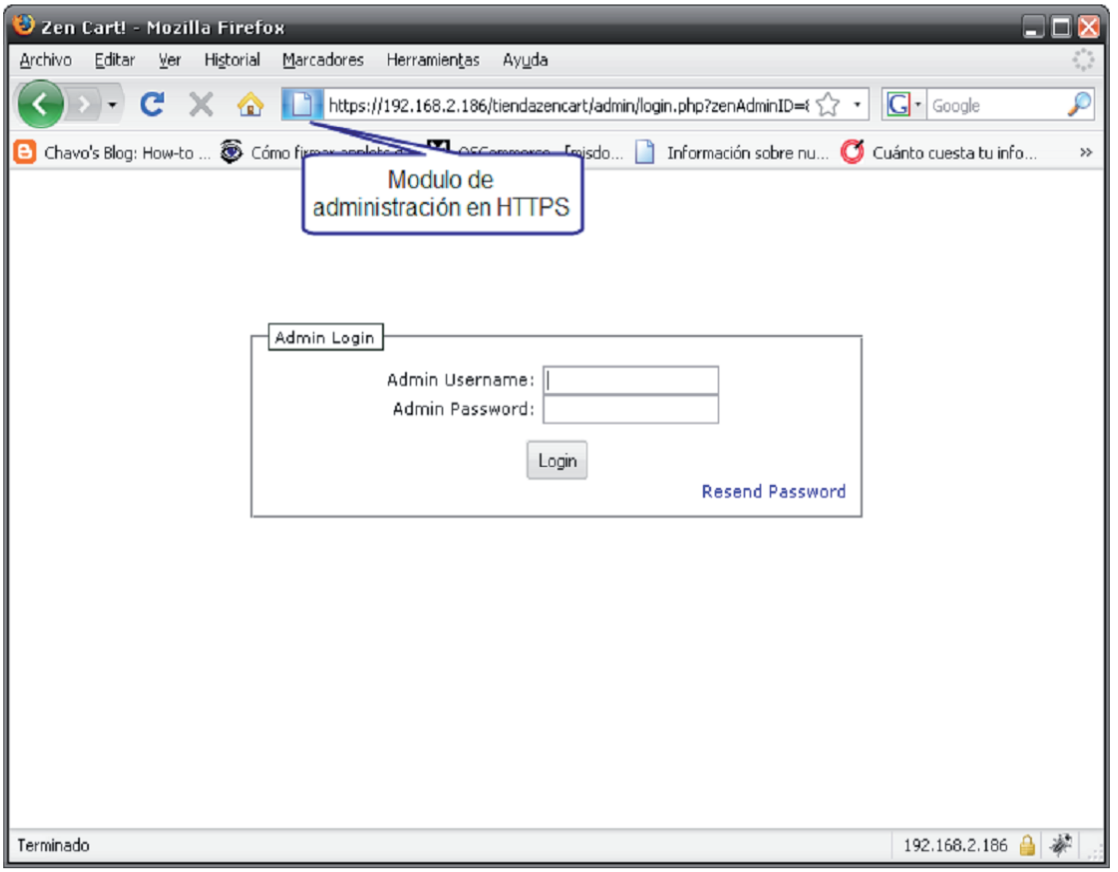

Figura 6.9 Se muestra la navegación segura mediante HTTPS en el servidor remoto.

Debido a que lo que se intenta proteger son los datos sensibles tal como usuario y contraseña mediante el uso de un certificado, se muestra en la figura 6.10 el filtrado de paquetes de una conexión insegura, es decir bajo el protocolo HTTP, en una página que recibe los parámetros de usuario (*rovskyhp*) y contraseña (*pita*) mediante el método POST. Mientras que en la figura 6.11 se analiza en la trama el uso del protocolo HTTPS, protegido por un certificado, en el cual el usuario (*admin*) y contraseña (*admin*) viajan cifrados en todo momento.

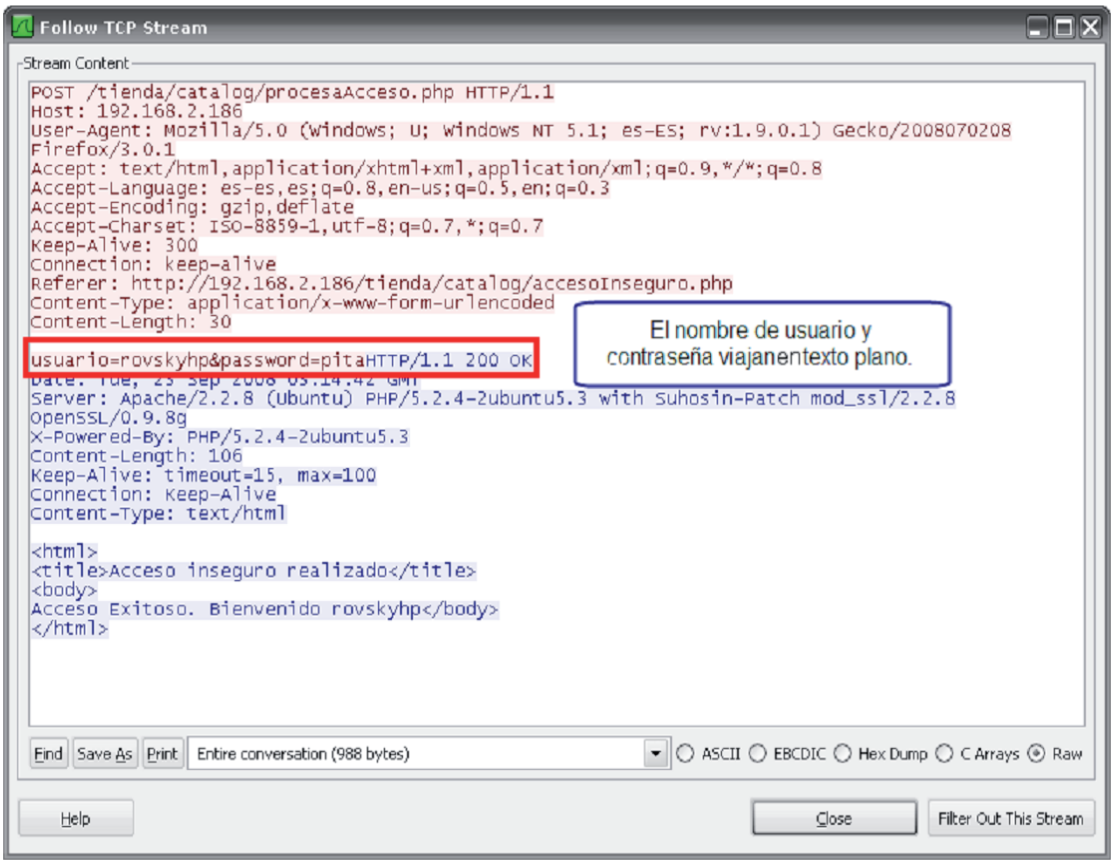

Figura 6.10 Se muestra la trama del protocolo HTTP, el usuario y contraseña viajan en texto plano

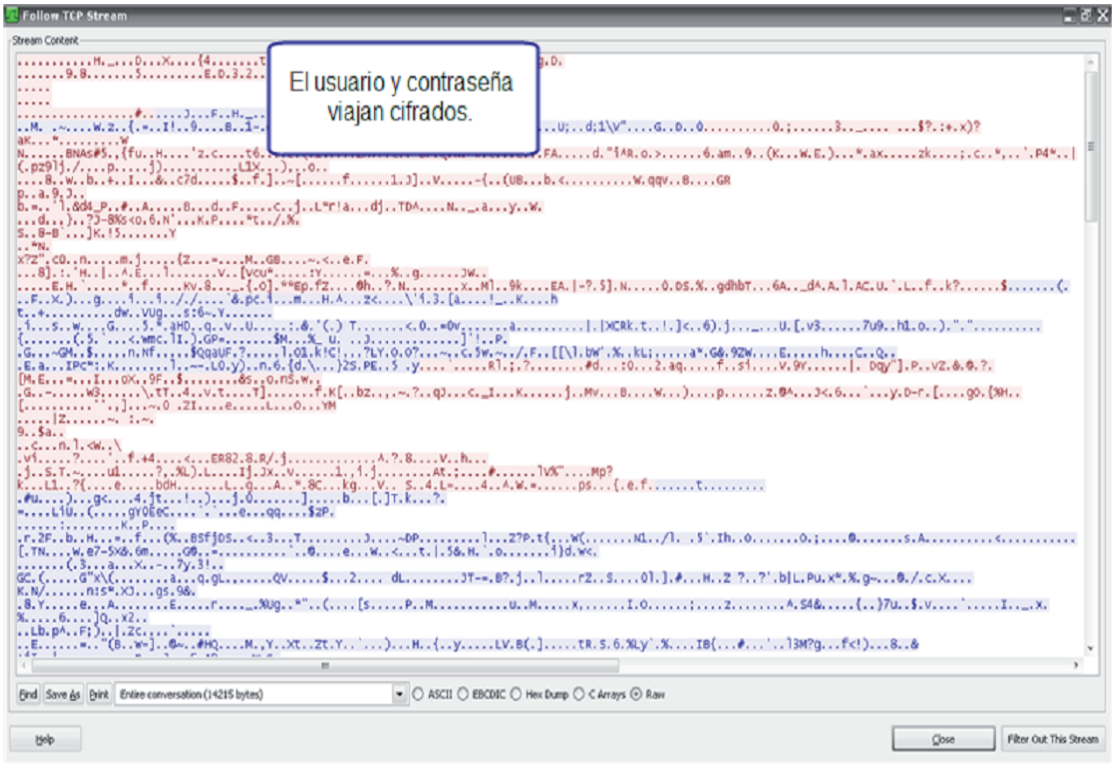

Figura 6.11 En la trama del protocolo HTTPS, se observa que los datos viajan cifrados.

6.4. Autenticación en un servidor mediante SSH con llaves pública y privada.

Esta modalidad de autenticación en SSH, permite acceder al servidor remoto haciendo uso de llaves pública y privada por cada usuario que intente conectarse, lo cual conllevara a no utilizar la contraseña del usuario en el servidor remoto como se observa en la figura 6.12. Además de que se reducirá el riesgo de que un usuario no deseado obtenga la contraseña por medio de ataques por fuerza bruta.

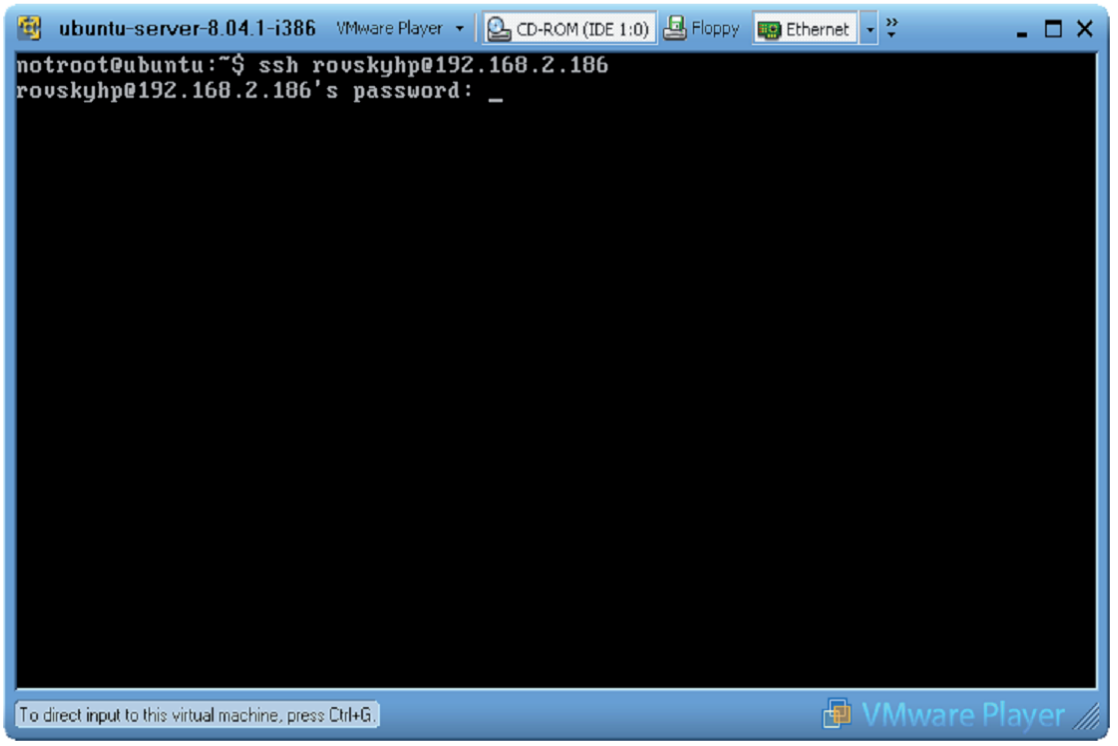

Figura 6.12 Se muéstrala autenticación en un servidor remoto haciendo uso del método usuario-contraseña.

Instalar en el servidor OpenSSH-Server.

```
rovskyhp@Servidor:~$ sudo apt-get install openssh-server
```

Instalar en la maquina cliente OpenSSH-Client.

```
notroot@ubuntu:~$ sudo apt-get install openssh-client
```

Una vez instalado OpenSSH en ambos equipos se crearán el par de llaves en la maquina cliente, haciendo uso de la herramienta ssh-keygen, tal como se muestra en la figura 6.13, se debe ingresar al directorio en donde se desean crear las llaves, si se deja en blanco se almacenarán por default en /home/notroot/.ssh/id_rsa. Además se preguntará una frase secreta, la cual deberá ser mayor a 4 bits, esta frase será la que se utilizará para descifrar la llave privada y permitirá iniciar sesión en el servidor, no es recomendable dejar en blanco este campo ya que cualquier persona que pueda obtener la clave privada podrá iniciar sesión en el servidor.

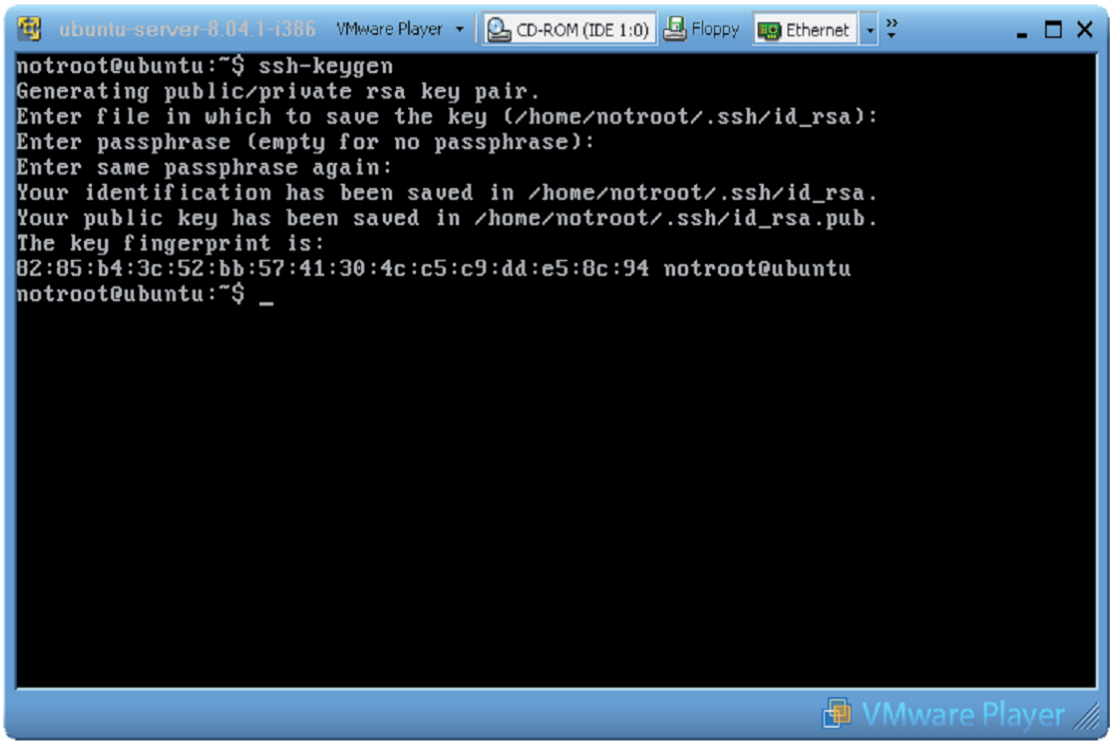

Figura 6.13 Creación de llaves en maquina cliente.

Hasta este momento se han creado las llaves y han sido almacenadas en el directorio /home/notroot/.ssh/id_rsa. El archivo id_rsa es la llave privada, mientras que el archivo id_rsa.pub la llave pública, esta última es la que se tiene que almacenar en el servidor.

En este momento el siguiente paso será copiar el archivo id_rsa.pub en el servidor para lo cual utilizamos el comando scp, como se sigue utilizando el método usuario-contraseña se solicitará la contraseña del usuario del servidor remoto, esto se observa en la figura 6.14.

```
notroot@ubuntu:~$ scp ~/.ssh/id_rsa.pub rovskyhp@192.168.2.186:~/
```

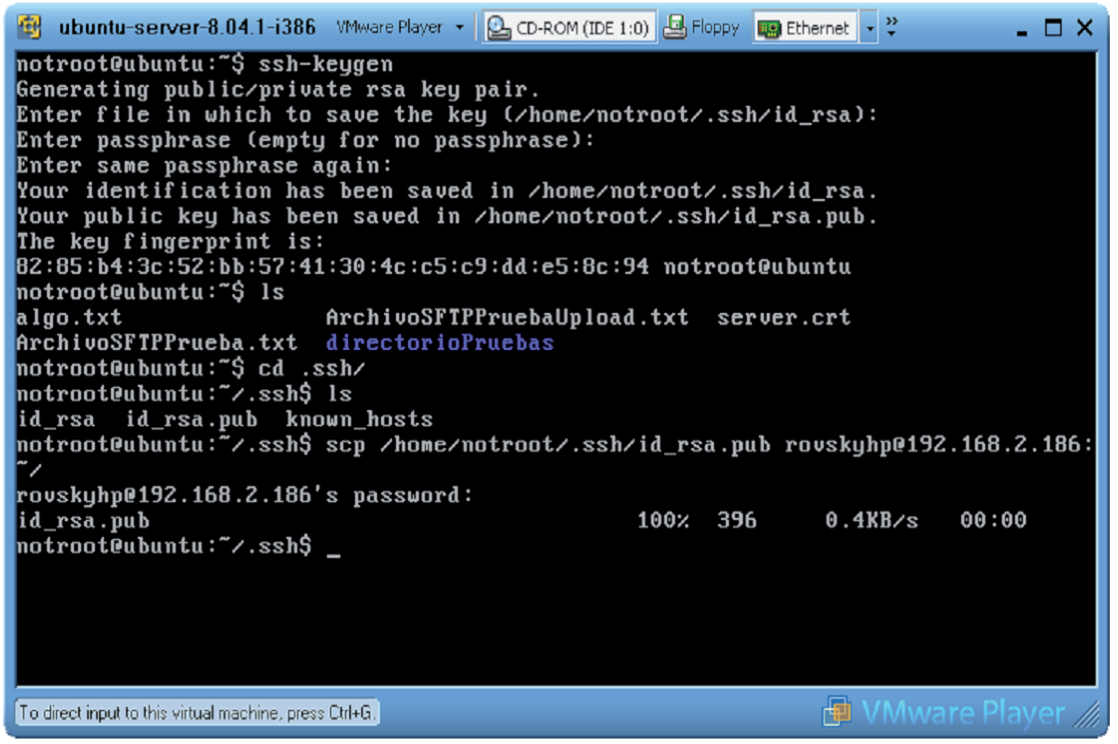

Figura 6.14 Copiando mediante scp la llave pública en el servidor.

Se debe iniciar sesión en el servidor remoto para que a continuación se agregue la llave pública al archivo que contiene las llaves autorizadas para el servidor remoto, una vez que se haya agregado la llave pública en authorized_keys se debe borrar, con la finalidad de evitar duplicidad de información.

```
rovskyhp@192.168.2.186:~$ cat id_rsa.pub >> ~/.ssh/authorized_keys
rovskyhp@192.168.2.186:~$ rm id_rsa.pub
```

Ahora se debe de configurar el SSH en el servidor remoto para aceptar la autenticación por medio de las llaves, se tiene que modificar el archivo /etc/ssh/sshd_config poniendo las variables siguientes con el valor yes. Y por ultimo recargar la configuración del servicio.

```
rovskyhp@192.168.2.186:~$ sudo vim /etc/ssh/sshd_config
RSAAuthentication yes
PubkeyAuthentication yes
rovskyhp@192.168.2.186:~$ sudo sevice ssh reload
```

Una vez hecho esto ahora se deberá realizar la conexión mediante el uso de la llave privada como se muestra en la figura 6.15, ingresando la frase con la que se creó la llave anteriormente.

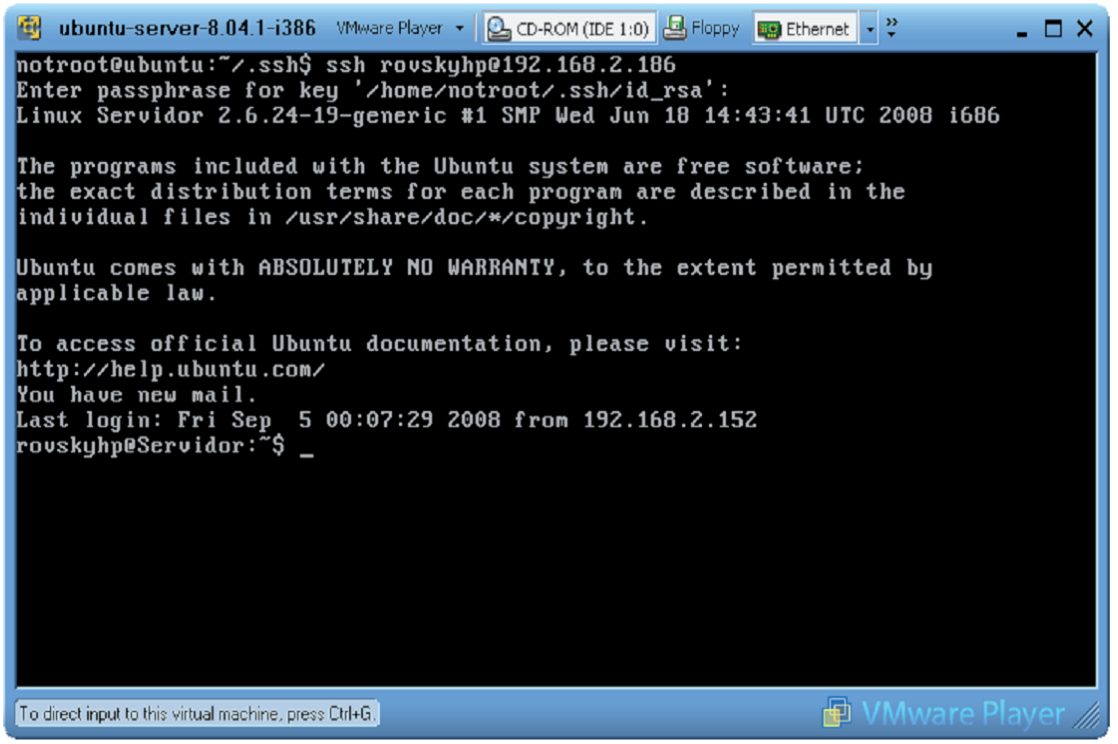

Figura 6.15 Conexión a un servidor remoto mediante el uso del método de llaves pública y privada.

Ahora se debe de configurar el SSH en el servidor remoto para no permitir la autenticación por medio de usuario y contraseña, se tiene que modificar el archivo /etc/ssh/sshd_config. Y por ultimo recargar la configuración del servicio.

```
rovskyhp@192.168.2.186:~$ sudo vim /etc/ssh/sshd_config
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no

rovskyhp@192.168.2.186:~$ sudo sevice ssh reload
```

De esta forma si el usuario no cuenta con la llave privada en su equipo no podrá iniciar sesión en el servidor remoto.

De la misma manera que en el apartado anterior, se muestran las tramas de los servicios SSH, con la finalidad de demostrar que en una conexión mediante Telnet con usuario (rovskyhpTelnet) y contraseña (pita), los datos son enviados en texto plano como se observa en la figura 6.16. Mientras que haciendo uso de SSH los datos permanecen cifrados (figura 6.17) al momento de realizar la conexión con el servidor.

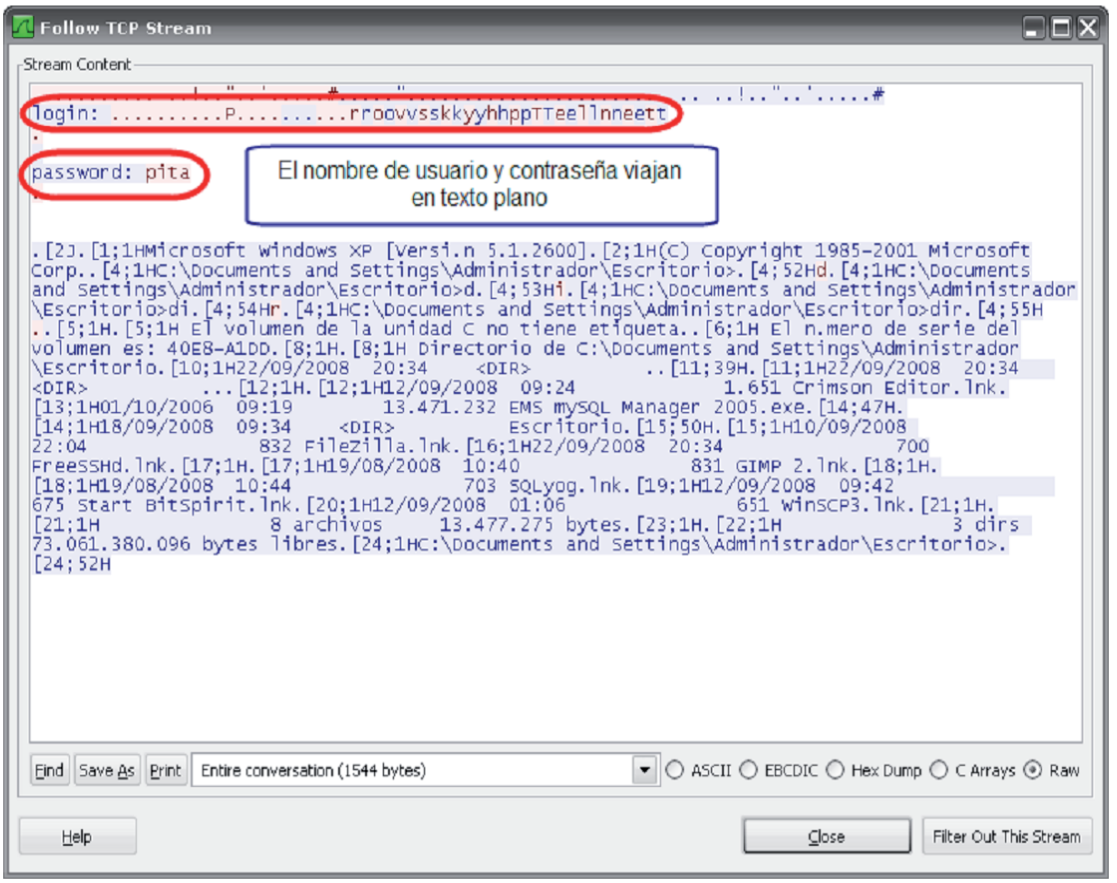

Figura 6.16 Conexión a un servidor remoto mediante Telnet, los datos de usuario y contraseña viajan en texto plano.

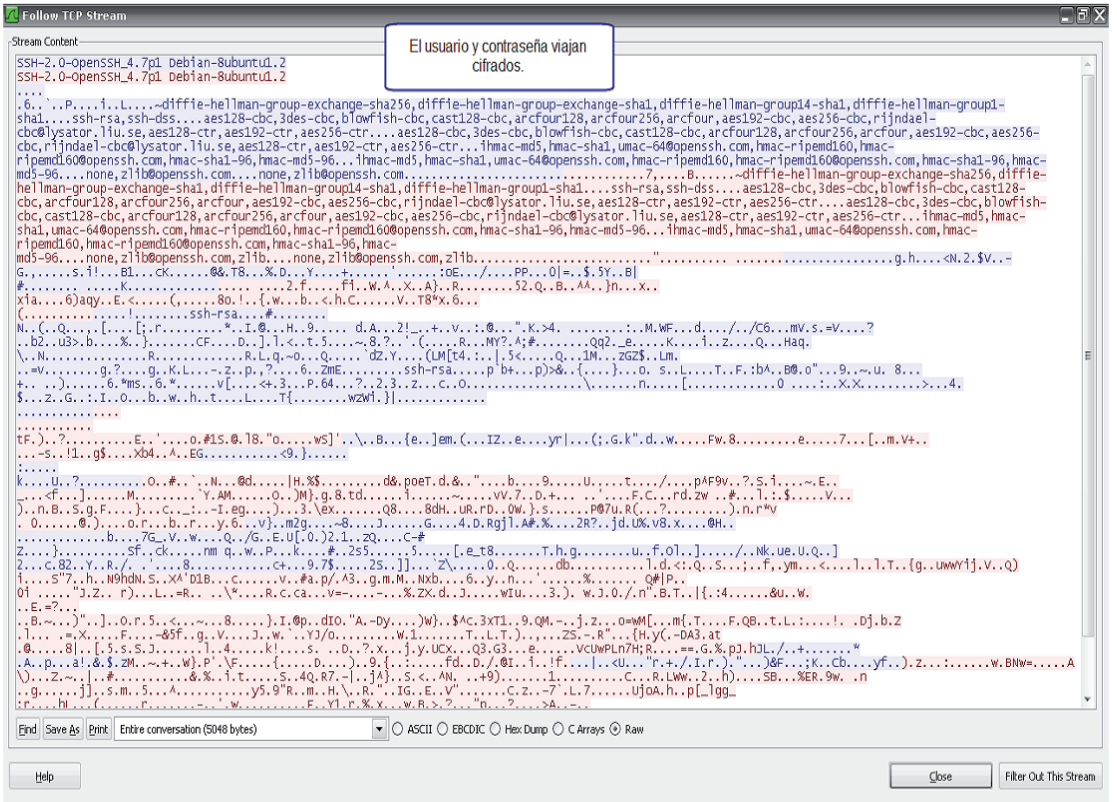

Figura 6.17 En una conexión a un servidor remoto mediante SSH, los datos de viajan cifrados.

6.5. Autenticación en SFTP mediante llaves pública y privada para transferencias de archivos.

SFTP es un subsistema de ssh, es decir corre en el mismo puerto el cual para el caso del servidor de pruebas es el 22, de esta forma la configuración realizada en el apartado anterior, la autenticación se llevará a cabo mediante llaves.

Para llevar a cabo la conexión utilizando un cliente para Windows, se pueden utilizar diversos clientes, el único procedimiento extra que se tiene que realizar seria convertir la llave privada del usuario al formato utilizado por los clientes, el cual tiene por extensión .ppk. Para realizar esta conversión es necesario instalar la herramienta Putty-Gen, la cual es gratuita.

Convertir la llave privada generada en servidor de pruebas al formato de cliente SFTP para Windows.

Ejecutar la herramienta Putty-Gen como se muestra en la figura 6.18.

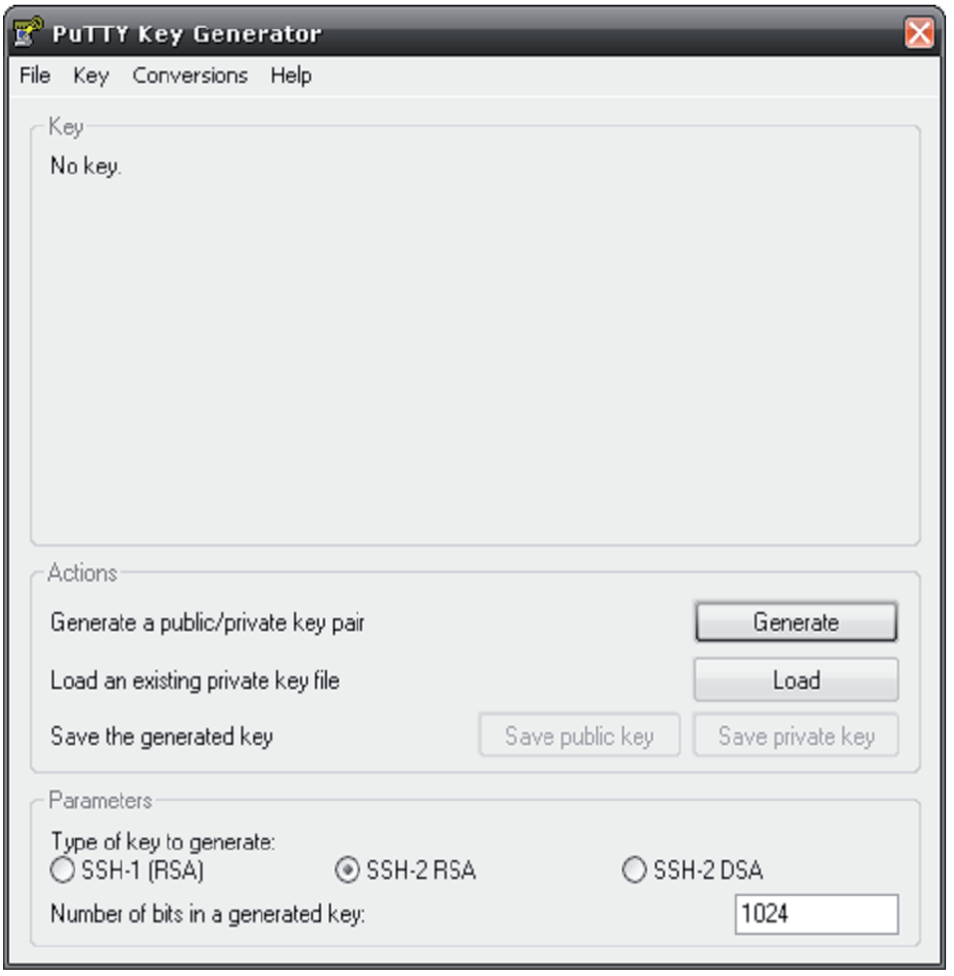

Figura 6.18 Uso de la herramienta Putty-Gen para convertir la llave privada en formato para Windows.

Cargar la llave privada en la herramienta, como se muestra en la figura 6.19, en este paso se debió seleccionar la llave, lo cual muestra un cuadro de texto que permitirá ingresar la frase con la que fue protegida en el servidor de pruebas.

Figura 6.19 Introducir la frase con la que se protegió la llave privada en el servidor.

Al agregar la frase se informa que la llave fue convertida al formato correspondiente, lo siguiente será guardar el archivo generado en el directorio que corresponda en la maquina cliente.

Una vez realizado este procedimiento, se debe ejecutar el cliente para SFTP, para esta implementación se utilizo el cliente FileZilla- Client, que además de ser gratuito, permite la autenticación mediante llaves a servidores remotos, la pantalla de inicio del cliente se muestra en la figura 6.20.

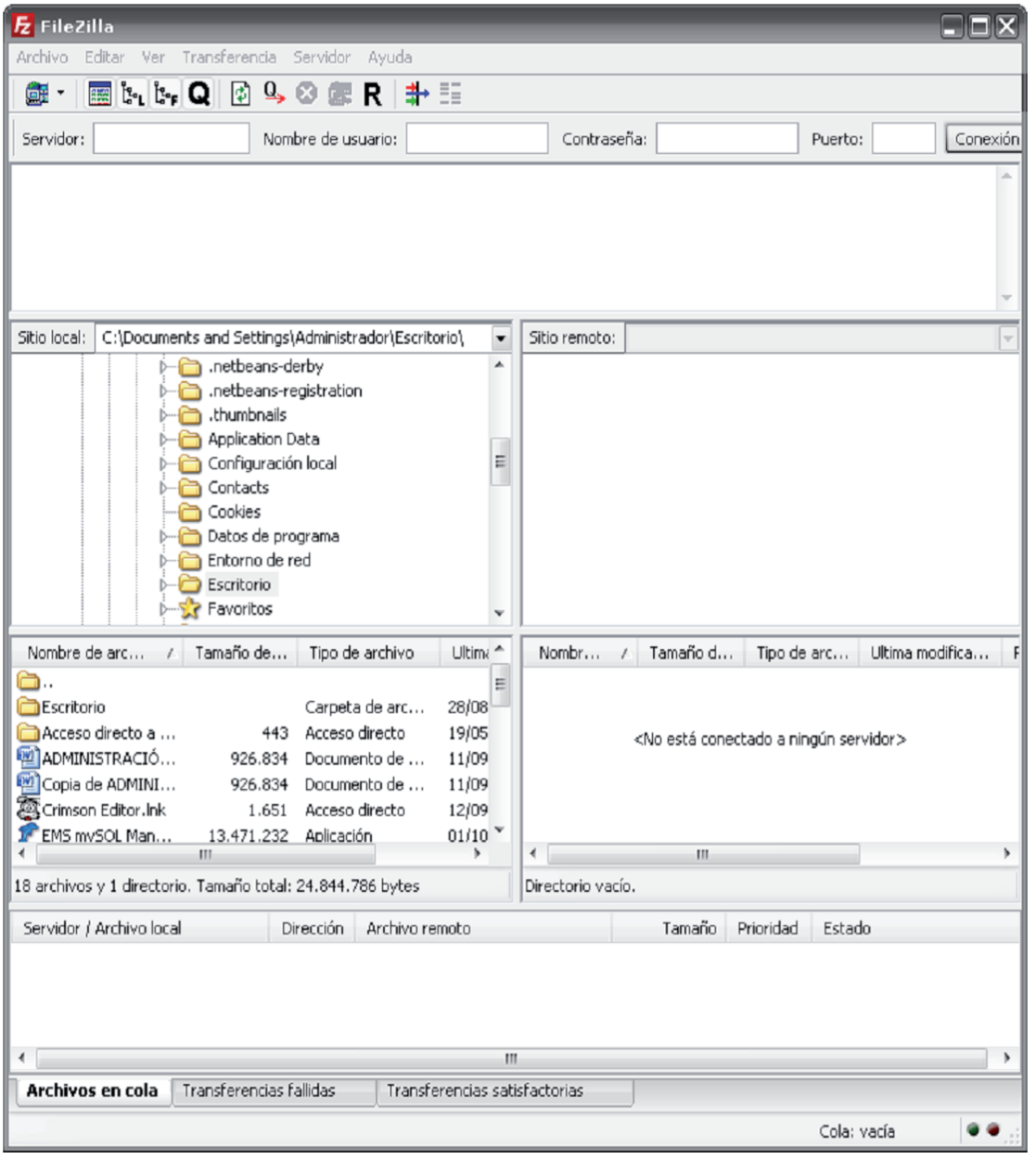

Figura 6.20 Interfaz grafica del cliente FileZilla para SFTP.

Para llevar a cabo la conexión con el servidor de pruebas es necesario agregar el servidor al cliente FileZilla, para ello se lleva a cabo el siguiente procedimiento.

Seleccionar el menú **Archivo**, después el submenú **Gestión de Sitios**, se muestra una ventana en la se tiene que dar clic en el botón **Nuevo Sitio,** esto muestra los campos necesarios para la conexión, que para el caso del servidor de pruebas se utilizaron los datos siguientes.

- **Servidor:** 192.168.2.186
- **Puerto:** 22
- **Tipo de Servidor:** SFTP- SSH File transfer Protocol

- **Modo de acceso:** Normal
- **Usuario:** rovskyhp

Una vez ingresados los datos, dar clic en el botón Aceptar, esto guardará la conexión a dicho servidor.

Es momento ahora de importar la llave privada que fue generada con el cliente Putty-Gen, lo cual se lleva a cabo de la siguiente manera.

Seleccionar **Editar**, localizado en la barra de menús, ahí dar clic en **Opciones**, se muestra una ventana con las opciones del cliente FileZilla, como se muestra en la figura 6.21, se debe seleccionar la opción SFTP que se muestra en el árbol de opciones de lado izquierdo.

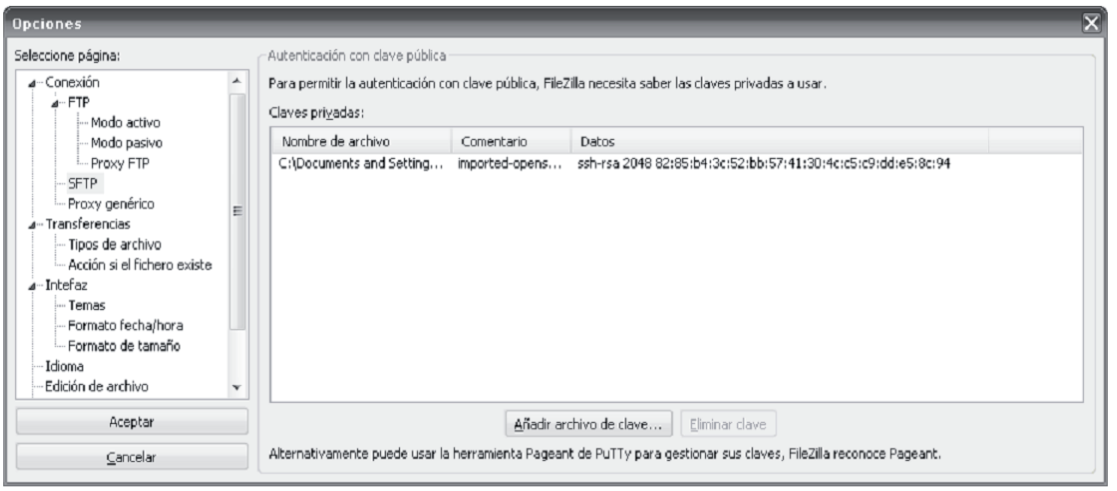

Figura 6.21 Importar la llave en el cliente FileZilla.

En esta ventana se tiene que importar la clave privada, para ello dar clic en el botón Añadir archivo clave. Seleccionar la clave que fue guardada anteriormente con extensión .ppk y para finalizar dar en el botón Aceptar.

Con esto ahora se podrá realizar la autenticación en el servidor de pruebas y llevar a cabo la transferencia de archivos de manera segura haciendo uso de SFTP, para realizar la conexión se deberá seleccionar el servidor al cual se desea conectar, como se muestra en la figura 6.22.

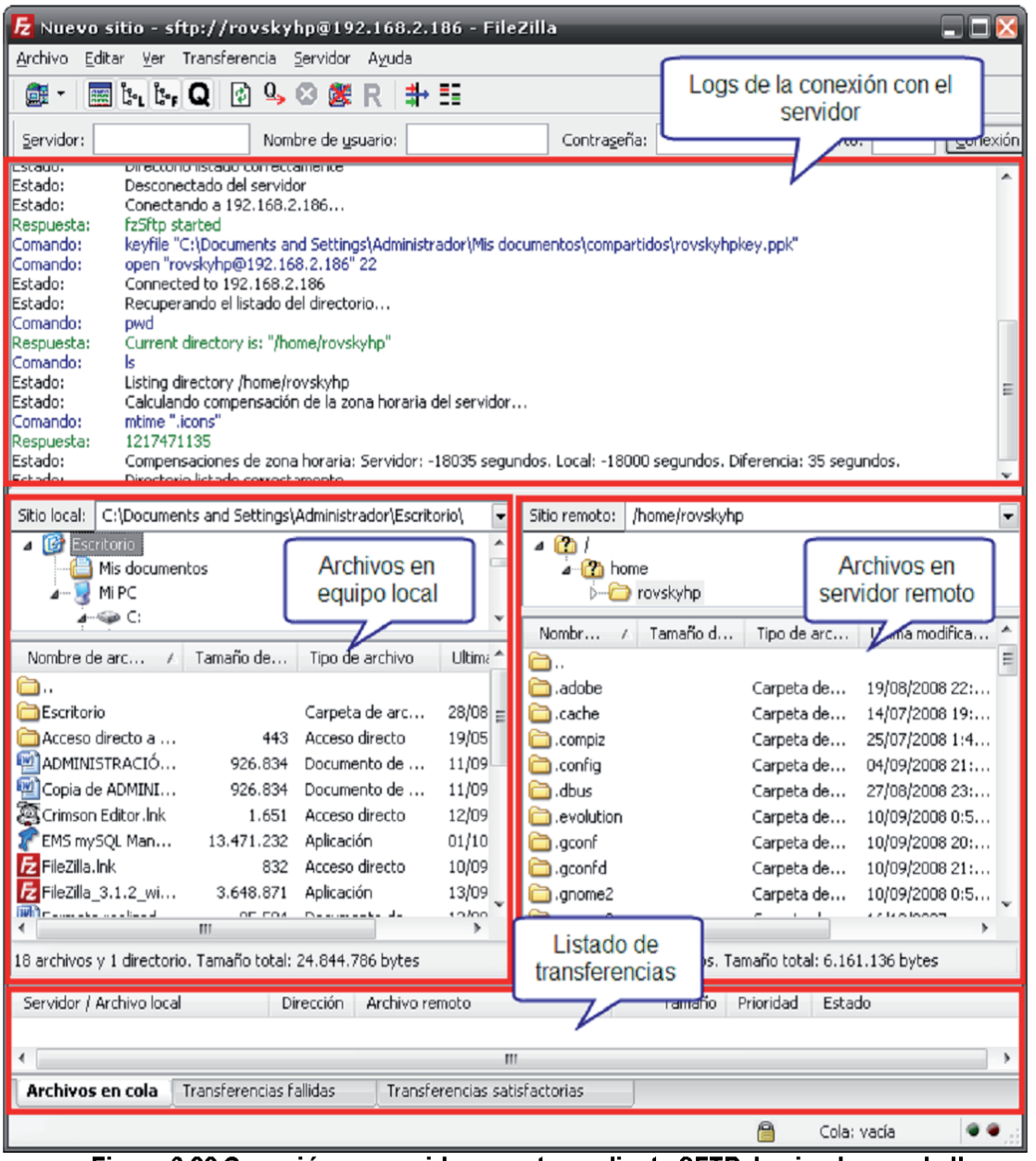

Figura 6.22 Conexión en servidor remoto mediante SFTP, haciendo uso de llaves pública y privada.

Para demostrar que en la utilización de un servidor ftp con sistema de autenticación por contraseñas, los datos viajan en texto plano, se realizo el análisis de paquetes en el puerto 21 (default para FTP). Y en contraparte se analizaron los paquetes en una conexión segura mediante el uso de SFTP, como se observa en las figuras 6.23 y 6.24 respectivamente.

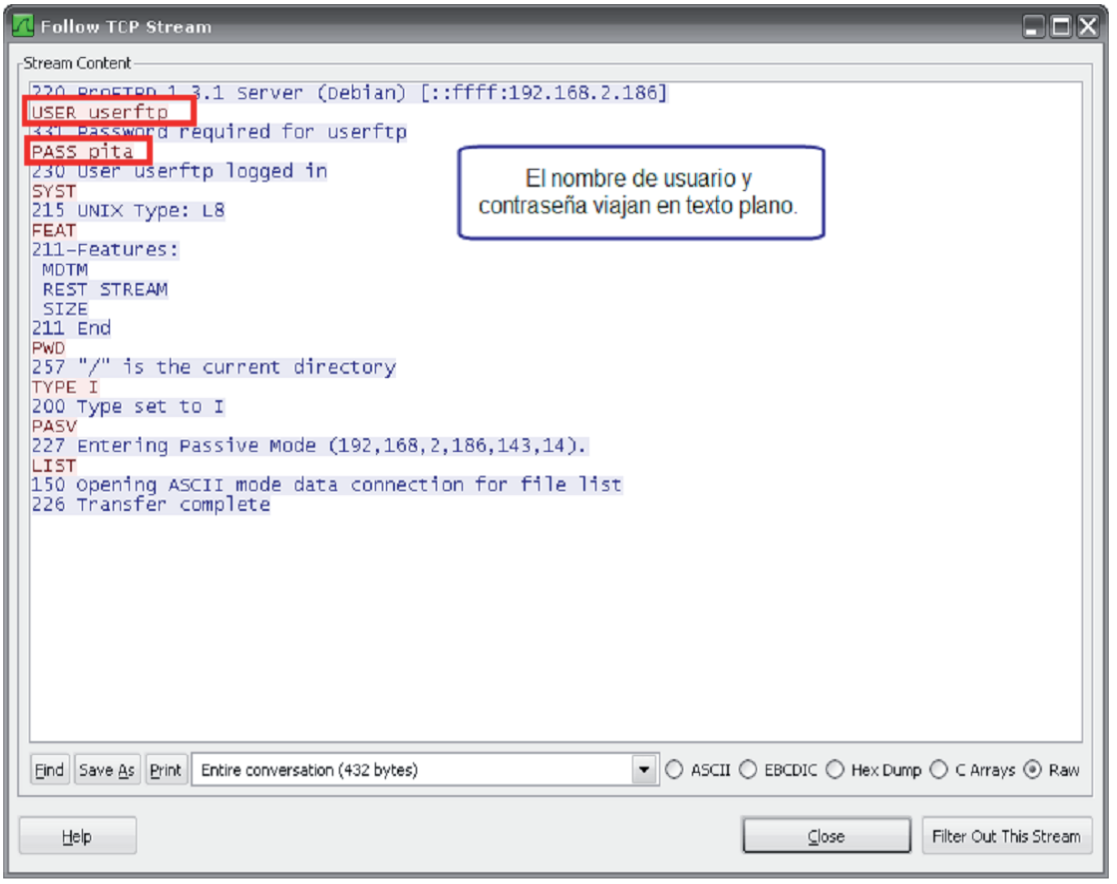

Figura 6.23 La utilización del método usuario/contraseña como medio de autenticación en FTP, es inseguro, debido a que los datos viajan en texto plano.

Figura 6.24 Al realizar la conexión mediante SFTP, los datos viajan cifrados, debido al método de autenticación por llaves pública y privada.

CONCLUSIONES

Estamos viviendo una época de cambio, dicho cambio involucra el uso de la tecnología en nuestras actividades diarias, dentro de las que destacan el uso del correo electrónico, la utilización del Internet para llevar a cabo el pago de servicios en línea, las compras electrónicas, transacciones bancarias, entretenimiento, etc., sin embargo estos canales de comunicación son inseguros por si solos, ya que la información de cualquier persona o empresa es vulnerable a individuos u organismos malintencionados y puede ser alterada o robada con fines de lucro.

Con el desarrollo de esta tesina, se presentaron los certificados digitales, como una herramienta para brindar confiabilidad, disponibilidad, integridad y no repudio, durante el manejo de datos sensibles de un usuario o una organización, para evitar situaciones de fraude, extorsión entre muchas otras que nos acechan en la actualidad.

Esto se llevo a cabo protegiendo una aplicación de comercio electrónico, la cual fue montada en un servidor Web. En dicha aplicación se implemento el uso de certificados digitales, que por conveniencia fueron autofirmados, permitiendo al usuario comprobar la identidad del servidor y evitar, de esa manera, la suplantación de identidad (*spoofing*). Aunado a esto se llevo a cabo la configuración del sitio para que el envió de información sensible tanto del cliente como del administrador de la tienda electrónica viajara de manera segura, haciendo uso del protocolo HTTPS, quedando demostrado la seguridad mediante la captura de imágenes de los protocolos HTTP y HTTPS al momento de enviar información como puede ser el nombre de usuario, contraseña de acceso, número de tarjeta de crédito, NIP, teléfonos, domicilio, entre otros, siendo que, usando el protocolo HTTP los datos viajan en texto plano, es decir, no seguros, por el contrario con HTTPS los datos viajan cifrados.

Es importante mencionar que la utilización de certificados digitales autofirmados no es de lo más recomendable, ya que existen organizaciones reconocidas a nivel mundial tal como VeriSign o Thawte conocidas como autoridades certificadoras, las cuales ofrecen distintos tipos de certificados, dependiendo de la inversión y de la seguridad que uno desee brindar a sus usuarios finales.

Por otro lado, se llevo a cabo la protección de los servicios SSH y SFTP, mediante el método de autenticación por llaves pública y privada. Al realizar esta implementación lo que se pretendía era utilizar el mismo certificado generado para proteger HTTPS, solo que el servicio SSH y SFTP no soportan este método, puesto que se tendrían que generar un certificado por usuario.

Por lo tanto se opto por la implementación del método de autenticación por llaves. De igual manera quedo comprobado que los datos de usuario y contraseña en estos servicios viajan cifrados, brindando a los administradores de red la posibilidad de acceder a un servidor remoto mediante SSH, a usuarios con los privilegios correspondientes para la transferencia de archivos mediante SFTP, dicho de otra manera, esto permitirá contar con la plena

confianza de que su información confidencial permanecerá cifrada mientras tengan una sesión activa con un proveedor de dicho servicio.

Se recomienda ampliamente la utilización de mecanismos de seguridad, tal como los certificados digitales y la autenticación por llaves pública y privada, en los servicios que nos hacen la vida más simple y segura, además de que están disponibles para todo el mundo, desde herramientas gratuitas, hasta comerciales.

BIBLIOGRAFÍA

Índice	Referencia	Fecha de última consulta
[1]	Galán, Verónica. "Cuánto cuesta tu información en Internet". CNNExpansión.com. Sección Tecnología. Publicado el 8 de mayo de 2008. Ciudad de México. http://www.cnnexpansion.com/tecnologia/2008/05/08/cuanto-cuesta-tu-informacion-en-internet	1 de octubre 2008
[2]	Seguridad: una introducción.", DR MANUNTA, Giovanni. Consultor y profesor de Seguridad de Cranfield University. Revista Seguridad Corporativa. http://www.belt.es/bibliografia/HOME2_articulo.asp?id=130	30 de junio 2008
[3]	Canal Audio Visual http://www2.canalaudiovisual.com/ezine/books/acjirlNFORMATICA/1info01.html	30 de junio 2008
[4]	Real Academia Española http://www.rae.es/	30 de junio 2008
[5]	SANCHEZ PALOMARES, Zoila. "Monografía la Junta Interamericana de Defensa y su rol ante la organización de los Estados Americanos, como Agencia Especializada." FORTLESLEY J. Mcnair, WASHINGTON, DC. Abril, 20, 2004. http://library.jid.org/en/mono43/Sanchez%20Soila.doc	30 de junio 2008
[6]	Fundación Alfonso Lara Ramos para la Ciencia y la Tecnología en Internet Seguridad Informática http://www.oferta-informática.es/seguridad-informática.html	30 de junio 2008
[7]	Red Iris http://www.rediris.es/cert/doc/unixsec/node5.html	30 de junio 2008
[8]	Enciclopedia Virus http://www.enciclopediavirus.com/	30 de junio 2008
[9]	Amenazas humanas y métodos más comunes para acceder a los recursos computacionales. 01 de julio de 2005, Instituto Mexicano de Contadores Públicos. http://portal.imcp.org.mx/content/view/639/199/	30 de junio 2008
[10]	Wiki para Ubuntu. http://www.guia-ubuntu.org/index.php?title=Servidor_ssh	30 de junio 2008

[11]	Red Iris. http://www.rediris.es/cert/doc/docu_rediris/ssh.es.html	30 de junio 2008
[12]	Pagina oficial en español de Open SSH. http://www.openssh.com/es/faq.html#1.1	11 de septiembre 2008
[13]	Wikipedia la enciclopedia libre. http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure	11 de septiembre 2008
[14]	Wikipedia la enciclopedia libre. SSL. http://es.wikipedia.org/wiki/Ssl	11 de septiembre 2008
[15]	Wikipedia la enciclopedia libre. SHTTP. http://es.wikipedia.org/wiki/SHTTP	11 de septiembre 2008
[16]	Granados Paredes, Gibrán. Introducción a la criptografía. Revista Digital Universitaria. Vol. 7, No.7, pp 13-17. ISSN 1067-6079. Julio 2006 www.revista.unam.mx/vol.7/num7/art55/jul_art55.pdf	11 de septiembre 2008
[17]	Banco de México. Dirección General de Operaciones de Banca Central. Dirección de Sistemas Operativos y de Pagos. http://www.banxico.org.mx/sistemasdepago/ies/iespub/PDF/IES-DOCTOBANXICO.PDF	11 de septiembre 2008
[18]	Conde González, Miguel A., Mecanismos de seguridad de la información en aplicaciones Web. Facultad de Ciencias, Universidad de Salamanca. Departamento de Informática y Automática. Mayo 2006. http://zarza.usal.es/~fgarcia/doctorado/iWeb/05-07/Trabajos/SeguridadAppWeb.pdf	11 de septiembre 2008
[19]	Conde González, Miguel A., Mecanismos de seguridad de la información en aplicaciones Web. Facultad de Ciencias, Universidad de Salamanca. Departamento de Informática y Automática. Mayo 2006. http://zarza.usal.es/~fgarcia/doctorado/iWeb/05-07/Trabajos/SeguridadAppWeb.pdf	11 de septiembre 2008
[20]	Conde González, Miguel A., Mecanismos de seguridad de la información en aplicaciones Web. Facultad de Ciencias, Universidad de Salamanca. Departamento de Informática y Automática. Mayo 2006. http://zarza.usal.es/~fgarcia/doctorado/iWeb/05-07/Trabajos/SeguridadAppWeb.pdf	11 de septiembre 2008
[21]	Wikipedia la enciclopedia libre. Certificado Digital. http://es.wikipedia.org/wiki/Certificado_digital	11 de septiembre 2008
[22]	ISO/IEC 9594-8:2001 “Information Technology. Open Systems Interconnection. The Directory: Public-Key And Attribute Certificate Frameworks”.	11 de septiembre 2008

ITU-T RECOMMENDATION X.509
http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=34551
http://www.itu.int/rec/T-REC-X.509-199708-S/e).

[23] Wikipedia la enciclopedia libre. Autoridad de Certificación
http://es.wikipedia.org/wiki/Autoridad_de_Certificación
11 de septiembre 2008

[24] Villa Hermosa, Alfonso. "Algunos apuntes técnicos y jurídicos sobre los principales protocolos de seguridad utilizados en Internet". Revista de Derecho Informático: Alfa Redi, No. 68 Marzo 2004. ISSN 1681-5726
http://www.alfa-redi.org/rdi-articulo.shtml?x=1225
11 de septiembre 2008

[25] Soriano, Miquel. "¿Cómo funciona la Seguridad en Internet?". Centre Tecnológic de Telecomunicacions de Cataluya.
http://www.cttc.es/resources/doc/080122-como-funciona-internet-48232.pdf
11 de septiembre 2008

[26] Mendívil, Ignacio. "El abc de los documentos electrónicos seguros". SeguriData. Abril de 2007.
http://www.tierradelazaro.com/public/criptologia/abc.pdf
11 de septiembre 2008

[27] ISO/IEC 9594-8:2001 "Information Technology. Open Systems Interconnection. The Directory: Public-Key And Attribute Certificate Frameworks". ITU-T RECOMMENDATION X.509
http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=34551
http://www.itu.int/rec/T-REC-X.509-199708-S/e).
11 de septiembre 2008

[28] ISO/IEC 9594-8:2001 "Information Technology. Open Systems Interconnection. The Directory: Public-Key And Attribute Certificate Frameworks". ITU-T RECOMMENDATION X.509
http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=34551
http://www.itu.int/rec/T-REC-X.509-199708-S/e).
11 de septiembre 2008

[29] Wikipedia la enciclopedia libre. X.509
http://es.wikipedia.org/wiki/X.509
15 de septiembre 2008

[30] Talens-Oliag, Sergio. "Seguridad en Java". Universidad de Valencia. Instituto de Informática. Diciembre 1999.
http://www.uv.es/sto/cursos/seguridad.java/html/sjava.html#toc4
15 de septiembre 2008

[31] López Peza, Carlos Eduardo. "Sistema de seguridad para intercambio de datos en dispositivos móviles". Centro de investigación y de estudios avanzados del IPN. Departamento de Ingeniería
15 de septiembre 2008

Eléctrica sección Computación. México D.F. 2005.

[32] Wikipedia la enciclopedia libre. Infraestructura de Clave Pública. http://es.wikipedia.org/wiki/Certificaci%C3%B3n_Electr%C3%B3nica — 15 de septiembre 2008

[33] EuPKI, European Libre Software Public Key Infrastructure. http://www.europepki.org/php/home.php — 15 de septiembre 2008

[34] Zimmermann, Philip. "Guía del usuario de PGP". 1994. http://www.rediris.es/pgp/doc/pgp231.html — 15 de septiembre 2008

[35] López, Lourdes. Seguridad en redes telemáticas Parte II: Entornos seguros. http://www.rediris.es/rediris/boletin/32/enfoque1.html — 15 de septiembre 2008

[36] Fundación Dike-Justicia. http://www.fundaciondike.org.ar/seguridad/firmadigital-autoridades2.html — 15 de septiembre 2008

[37] Política de certificados de la autoridad certificadora raíz de la secretaría de economía. 2005. www.firmadigital.gob.mx/politica/CP-SE-V1_m.pdf — 15 de septiembre 2008

[38] VeriSign Latinoamérica. "Seguridad en la Any Era: Equilibrio de riesgo, costo y experiencia del usuario". http://www.verisign.com/latinamerica/esp/index.html — 15 de septiembre 2008

[39] Thawte Español. "Securing your Online Data Transfer with SSL". http://www.thawte.com/es/ — 15 de septiembre 2008

[40] Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes. SSH, The Secure Shell: The Definitive Guide, Segunda Edición. Estados Unidos. Editorial O´Reilly. Mayo 2005. http://oreilly.com/catalog/9780596008956/toc.html — 25 de octubre 2008

[41] Cyrus Peikari, Anton Chuvakin. Security Warrior. Estados Unidos. Editorial O´Reilly. Enero 2004. http://oreilly.com/catalog/9780596005450/toc.html — 25 de octubre 2008

[42] Andrew S. Tanenbaum. Redes de Computadoras. Cuarta Edición. Estados Unidos. Editorial Pearson Educación Latinoamérica. 2003. — 30 de junio 2008

[43] John Viega, Matt Messier, Pravir Chandra. Network Security with OpenSSL. Estados Unidos. Editorial O´Reilly. Junio 2002 http://oreilly.com/catalog/9780596002701/toc.html — 25 de octubre 2008

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ANEXOS

GLOSARIO DE TÉRMINOS.

Acceso remoto.

Conexión de dos equipos ubicados en diferentes lugares físicos por medio de líneas de comunicación ya sean telefónicas o por medio de redes que permiten el acceso de aplicaciones e información de la red. Este tipo de acceso normalmente viene acompañado de un sistema robusto de autenticación.

Algoritmo de cifrado.

Conjunto de reglas o procedimientos por medio de los cuales un texto en claro es codificado, con la finalidad de ser protegido de accesos no deseados.

Ataque.

Acto por medio del cual se intenta sobrepasar las medidas de seguridad de un sistema en particular. Un ataque puede ser activo, resultando en la alteración de la información, o puede ser pasivo, en este caso el efecto es la pérdida de la misma.

Autenticación.

Es la identificación positiva de una entidad de red, tal como un servidor, un cliente, o un usuario.

Autoridad certificadora.

Una entidad externa de confianza cuyo fin es firmar certificados para las entidades de red que ha autentificado usando medios seguros.

Certificado.

Una información que se almacena para autentificar entidades de red tales como un servidor o un cliente. Un certificado contiene piezas de información X.509 sobre su poseedor (llamado sujeto) y sobre la Autoridad Certificadora

(llamada el expendedor) que lo firma, más la clave pública del propietario y la firma de la AC. Las entidades de red verifican las firmas usando certificados de las AC.

Clave privada.

La clave secreta de un sistema criptográfico de Clave Pública, usada para descifrar los mensajes entrantes y firmar los salientes.

Clave pública.

La clave disponible públicamente en un sistema criptográfico de Clave Pública, usado para cifrar mensajes destinados a su propietario y para descifrar firmas hechas por su propietario.

Comercio electrónico.

Cualquier forma de transacción comercial, en la que las partes involucradas interactúan electrónicamente, en lugar de existir un intercambio de persona a persona.

Protocolo de transferencia de archivos (FTP).

Es un protocolo estándar de comunicación, que proporciona un camino simple para extraer y colocar archivos compartidos entre computadoras sobre un ambiente de red.

Firma digital.

La firma digital, consiste en la transformación de algún mensaje utilizando un sistema de cifrado asimétrico de manera que la persona que posee el mensaje original y la clave pública del firmante, pueda establecer de forma segura, que dicha transformación se efectuó utilizando la clave privada correspondiente a la pública del firmante, y si el mensaje es el original o fue alterado desde su elaboración.

Información.

Conjunto de símbolos que representan hechos, objetos o ideas que nos aporta algún conocimiento.

Ingeniería social.

Se refiere a la capacidad que tiene una persona de utilizar otra personalidad y hace uso de sus conocimientos y habilidades sociales para robar la información que está relacionada con los recursos computacionales tal como; llaves físicas o electrónicas, códigos, tarjetas de acceso, llamadas o dar contraseñas a otras personas.

IP-SPOOFING.

Hace referencia al uso de técnicas de suplantación de identidad generalmente con usos maliciosos o de investigación.

No repudio.

El no repudio es un servicio de seguridad que permite probar la participación de las partes en una comunicación.

Robo de identidad.

El robo de identidad, ocurre cuando alguien obtiene la información personal de alguien más, sin que esta tenga conocimiento de lo ocurrido, esto para realizar fraude o robo. El robo de identidad es un vehiculo para realizar cualquier tipo de fraude. Típicamente, la victima cree que esta brindando su información a algún agente que representa a una empresa, o simplemente responde un email que tiene como asunto la actualización de su cuenta de correo, o como ya se menciono también, la victima podría estar llenando una solicitud para ser contratado por alguna empresa fraudulenta.

Seguridad.

Dicho de un mecanismo significa: Que asegura algún buen funcionamiento, precaviendo que este falle, se frustre o se violente.

Seguridad de la Información.

La información permite tomar decisiones importantes en los procesos cognoscitivos y sociales del ser humano. Dicha información podría ser sensible, al mismo tiempo, debe estar al alcance de quien le pueda dar un buen uso y fuera del alcance de quien pueda darle un mal uso; de ahí que surja la necesidad de proteger la información.

Texto cifrado.

Es el resultado de haber aplicado a un texto sin cifrar un algoritmo de cifrado.

X.509

Un esquema de certificado de autentificación recomendado por la International Telecommunication Union (ITU-T) que se usa en la autentificación SSL/TLS.

ABREVIATURAS.

CA	**C**ertificate **A**uthority	Autoridad Certificadora
CRL	**C**ertificate **R**evocation **L**ist	Lista de revocación de certificados
DES	**D**ata **E**ncription **S**tardard	Estándar de encriptación de datos
DoS	**D**enial **o**f **S**ervice	Negación de servicios
FTP	**F**ile **T**ransfer **P**rotocol	Protocolo de transferencia de archivos
HTTP	**H**yper**T**ext **T**ransfer **P**rotocol	Protocolo para transferencia de hipertexto
HTTPS	**S**ecure **H**yper**T**ext **T**ransfer **P**rotocol	Protocolo seguro de Transferencia de hipertexto
ITU-T	**I**nternational **T**elecommunication **U**nion **T**elecommunication Standardization Sector	Sección de Estandarización de las Telecomunicaciones de ITU
PEM	**P**rivacy **E**nhancement for Internet Electronic **M**ail	Correo Privado Mejorado
PGP	**P**retty **G**ood **P**rivacy	Privacidad Bastante Buena
PKCS	**P**ublic **K**ey **C**riptography **S**tandards	Estándares de criptografía de clave pública
PKI	**P**ublic **K**ey **I**nfrastructure	Infraestructura de clave publica
RSA	**R**ivest **S**hamir **A**delman	Iníciales de los creadores de RSA
SCP	**S**ecure **C**opy	Copia segura
SFTP	**S**ecure **T**ransfer **P**rotocol	Protocolo seguro de transferencia de archivos
SHTTP	**S**ecure **HTTP**	HTTP seguro
SSH	**S**ecure **S**hell	Consola segura

SSL	**S**ecure **S**ockets **L**ayer	Capa de conector seguro
TCP	**T**ransfer **C**ontrol **P**rotocol	Protocolo de Control de Transmisión