UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

INFRAESTRUCTURA DE CLAVE PUBLICA PARA LA UNIVERSIDAD DON BOSCO

TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTADO POR:
EDA VERONICA CASTRO
ADONIS DOMENICO MAJANO
ALEX GIOVANNI HURTADO

CIUDADELA DON BOSCO SEPTIEMBRE 2003

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

AUTORIDADES

RECTOR:

ING. FEDERICO MIGUEL HUGUET RIVERA

SECRETARIO GENERAL:

HNO. MARIO HOLMOS

DECANO DE LA FACULTAD DE INGENIERÍA:

ING. CARLOS BRAN

ASESOR DE TRABAJO DE GRADUACIÓN:

ING. JUAN CARLOS CASTRO

JURADO EVALUADOR:

ING. ANGEL SORIANO

ING. GIOVANNI VASQUEZ

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN

F._____ **F.**_____

ING. ANGEL SORIANO **ING. GIOVANNI VASQUEZ**

F._____

ING. JUAN CARLOS CASTRO

ASESOR

INDICE

1 INTRODUCCION

En la actualidad son muchas las personas en Internet que tienen necesidades básicas de seguridad es decir, evitar el acceso fraudulento a la información almacenada dentro de su computador o red de computadoras, por lo tanto, resulta primordial manejar mecanismos que controlen el acceso de usuarios no autorizados, con lo que se pretende garantizar que la información envíada sólo será conocida por el destinatario.

Estos factores están estrechamente relacionados en muchos aspectos, no dejan de adoptar soluciones y políticas totalmente distintas. Así, mientras el control de acceso es generalmente tarea del sistema operativo o elementos propios de la red (como son los firewalls), existen hoy en día aplicaciones que manejan con privacidad e integridad las transacciones personales en total comodidad.

En nuestro país las instituciones públicas, privadas, centros educativos, organizaciones, ONGs, etc; desconocen los mecanismos primordiales relacionados a la seguridad, esto hace que sus sistemas y medios de comunicación sean vulnerables y es aquí donde se centra el presente trabajo de graduación.

2 OBJETIVOS

2.1 Objetivo General

Implementación de una Infraestructura de Clave Pública (PKI) para la Universidad Don Bosco, que permita cubrir con las necesidades de seguridad en las comunicaciones.

2.2 Objetivos Específicos

- Implementar una infraestructura de clave pública (PKI)
- Proporcionar a la UDB los mecanismos de seguridad que permitan una comunicación segura entre sus diferentes entidades.
- Implementar el envío y recepción de correo electrónico seguros mediante el uso de Certificados Digitales.

3 ALCANCES

- Con este proyecto se pretende que exista una comunicación segura entre los empleados y alumnos de la institución, sin gente no deseable que se de cuenta de los movimientos de información en la institución.

- Dejar un legado de información para futuros proyectos aplicados para la Universidad Don Bosco en esta área.

- Incrementar en cierta medida los conocimientos de nueva tecnología en el área de seguridad por Internet.

4 BREVE HISTORIA DE LA CRIPTOGRAFÍA

Los mensajes cifrados han jugado un papel destacado en la Historia. Arma de militares, diplomáticos y espías, son la mejor defensa de las comunicaciones y datos que viajan por Internet. Esclavos con textos grabados en su cuero cabelludo, alfabetos de extraños símbolos, escritos de tinta simpática, secuencias interminables de números... Desde la Antigüedad, el hombre ha hecho gala de su ingenio para garantizar la confidencialidad de sus comunicaciones.

La criptografía (del griego kryptos, "escondido", y graphein, "escribir"), el arte de enmascarar los mensajes con signos convencionales, que sólo cobran sentido a la luz de una clave secreta, nació con la escritura. Su rastro se encuentra ya en las tablas cuneiformes, y los papiros demuestran que los primeros egipcios, hebreos, babilonios y asirios conocieron y aplicaron sus inescrutables técnicas, que alcanzan hoy su máxima expresión gracias al desarrollo de los sistemas informáticos y de las redes mundiales de comunicación.

Los criptogramas han protagonizado buena parte de los grandes episodios históricos y un sinfín de anécdotas. Existen mensajes cifrados entre los 64 artículos del Kamasutra, el manual erótico hindú del Vatsyayana, abundan en los textos diplomáticos, pueblan las órdenes militares en tiempos de guerra y, por supuesto, son la esencia de la actividad de los espías.

4.1 Los Métodos Clásicos

Los métodos clásicos son aquellos que existieron desde siempre y son métodos desarrollados para cifrar mensajes escritos a mano o en máquinas de impresión. Los métodos clásicos se basan en la sustitución de letras por otras y en la transposición, que juegan con la alteración del orden lógico de los caracteres del mensaje. Así, a los métodos clásicos le han salidos dos formas de cifrado, denominados grupos, que son:

- Métodos por Sustitución
- Métodos por Transposición

Los Métodos por Sustitución son aquellos que cambian palabras por otras. Esta simple forma de cifrar siempre ha obtenido buenos resultados.

Los Métodos por Transposición son aquellos que alteran el orden de las palabras del mismo mensaje. Dentro de los métodos clásicos podemos encontrarnos con varios sistemas como los que siguen a continuación:

- **Cifrado César o Monoalfabético Simple**. Es un método extremadamente simple y fue empleado por los romanos para cifrar sus mensajes, de ahí el nombre de César, ya que fue en su reinado cuando nació. Este sistema de cifrado, consiste en reemplazar cada letra de un texto por otra que se encuentre a una distancia determinada. Se sabe que César empleaba una distancia de 3.

- **Cifrado monoalfabético General.** Es un sistema que se basa en sustituir cada letra por otra de forma aleatoria. Esto supone un grado más de complejidad en el método de cifrado anterior

- **Cifrado por sustitución poli alfabética.** Es un método que emplea más de un alfabeto de sustitución. Esto es, se emplean varias cadenas de palabras aleatorias y diferentes entre sí para después elegir una palabra distinta según una secuencia establecida. Aquí nacen las claves secretas basadas en números. Este sistema es algo más complejo que las anteriores y a veces resulta difícil descifrar mensajes cuando empleamos más de diez columnas de palabras aleatorias.

- **Cifrado inverso.** Es quizás una de las formas más simples de cifrar una imagen y probablemente reconocida por todos nosotros. Es normal escribir del revés cuando estamos aburridos, pero lo cierto es que este es un sistema de cifrado. La forma de hacerlo es simplemente escribiendo el mensaje al revés.

- **Cifrado en figura Geométrica.** Es más complejo que la versión anterior. En esta ocasión el mensaje ya empieza a escribirse siguiendo un patrón preestablecido y es cifrado siguiendo una estructura geométrica basada en otro patrón. Este último patrón puede ser verdaderamente complejo según la extensión del mensaje escrito y la forma de seguimiento de la línea.

- **Cifrado por transposición de filas.** Consiste en escribir el mensaje en columnas y luego utilizar una regla para reordenarlas. Esta regla elegida al azar será la clave para cifrar el mensaje. También aquí es importante saber la clave

secreta para poder descifrar el mensaje. En esta ocasión el mensaje puede estar fuertemente cifrado si se emplean textos relativamente largos.

Todos los métodos criptográficos clásicos emplean la misma clave para cifrar y descifrar un mismo mensaje. Con la llegada de los ordenadores, la resolución de estos sistemas se tornó prácticamente trivial y por eso han surgido nuevos métodos de cifrado más trabajados y seguros. Algunos de ello también basados en claves secretas, cuya computación es prácticamente inalcanzable o bastante compleja.

4.2 Enigma, la Victoria Aliada

El siglo XX ha revolucionado la criptografía a principios de la centuria se diseñaron teletipos equipados con una secuencia de rotores móviles. Estos giraban con cada tecla que se pulsaba. De esta forma, en lugar de la letra elegida, aparecía un signo escogido por la máquina según diferentes reglas en un código poli alfabético complejo. Estos aparatos, se llamaron traductores mecánicos. Una de sus predecesoras fue la Rueda de Jefferson, el aparato mecánico criptográfico más antiguo que se conserva.

La primera patente data de 1919, y es obra del holandés Alexander Koch, que comparte honores con el alemán Arthur Scherbius, el inventor de Enigma[1] una máquina criptográfica que los nazis creyeron inviolable, sin saber que a partir de 1942, propiciaría su derrota. En efecto, en el desenlace de la contienda, hubo un factor decisivo y apenas conocido: los aliados eran capaces de descifrar todos los mensajes secretos alemanes.

Una organización secreta, en la que participó Alan Turín, uno de los padres de la informática y de la inteligencia artificial, había logrado desenmascarar las claves de Enigma, desarrollando más de una docena de artilugios -las bombas- que desvelaban los mensajes cifrados. La máquina alemana se convertía así en el talón de Aquiles del régimen, un topo en el que confiaban y que en definitiva, trabajaba para el enemigo.

Los códigos de la versión japonesa de Enigma (llamados Purple, violeta) se descifraron en el atolón de Midway. Un grupo de analistas, dirigidos por el comandante Joseph J. Rochefort, descubrió que los nipones señalaban con las siglas AF su objetivo. Para comprobarlo, Rochefort les hizo llegar este mensaje: "En Midway se han quedado sin instalaciones de desalinización". Inmediatamente, los japoneses la retransmitieron en

código: "No hay agua potable en AF". De esta forma, el almirante Nimitz consiguió una clamorosa victoria, hundiendo en Midway cuatro portaviones japoneses.

Mientras los nazis diseñaron Enigma para actuar en el campo de batalla, los estadounidenses utilizaron un modelo llamado Sigaba y apodado por los alemanes como "la gran máquina". Este modelo, funcionó en estaciones fijas y fue el único artefacto criptográfico que conservó intactos todos sus secretos durante la guerra.

La existencia de Enigma y el hecho de que los aliados conociesen sus secretos fueron, durante mucho tiempo, dos de los secretos mejor guardados de la II Guerra Mundial. ¿La razón? Querían seguir sacándole partido tras la guerra potenciando su uso en diversos países, que, al instalarla, hacían transparentes sus secretos.

Finalizada la contienda, las nuevas tecnologías electrónicas y digitales se adaptaron a las máquinas criptográficas. Se dieron así los primeros pasos hacia los sistemas criptográficos más modernos, mucho más fiables que la sustitución y transposición clásicas. Hoy por hoy, se utilizan métodos que combinan los dígitos del mensaje con otros, o bien algoritmos de gran complejidad. Un ordenador tardaría 200 millones de años en interpretar las claves más largas, de 128 bits.

5 TIPOS DE SISTEMAS CRIPTOGRAFICOS

La criptografía actual se inicia en la segunda mitad de la década de los años 70. No es hasta la invención del sistema conocido como DES (Data Encryption Standard) [2] en 1976 que se da a conocer mas ampliamente, principalmente en el mundo industrial y comercial. Posteriormente con el sistema RSA (Rivest, Shamir, Adleman) [3] en 1978,

se abre el comienzo de la criptografía en un gran rango de aplicaciones: en transmisiones militares, en transacciones financieras, en comunicación de satélite, en redes de computadoras, en líneas telefónicas, en transmisiones de televisión etcétera.

La criptografía se divide en dos tipos:

- Criptografía de clave privada o simétrica.
  Ejemplo : DES (Data Encryption Standard)
- Criptografía de clave pública o asimétrica.
  Ejemplo: RSA (Rivest, Shamir, Adleman)

5.1 Criptografía de Clave Privada o Simétrica

La criptografía de Clave Privada o simétrica se refiere al conjunto de métodos que permiten tener comunicación segura entre las partes siempre y cuando anteriormente se hayan intercambiado la clave correspondiente que llamaremos clave simétrica. La simetría se refiere a que las partes tienen la misma llave tanto para cifrar como para descifrar.

Este tipo de criptografía se conoce también como criptografía de clave privada o criptografía de llave privada.
Los métodos criptográficos clásicos, como el de Julio César, pueden considerarse simétricos y los principales algoritmos simétricos actuales son DES[2], IDEA[4], RC5[5] y el novedoso AES[6].

Cabe destacar el sistema DES[2] por ser el estándar utilizado por el Gobierno de Estados Unidos en todas sus comunicaciones. En 1977 el Departamento de Comercio y la Oficina Nacional de Estándares de Estados Unidos, en colaboración con IBM, desarrolló el sistema de encriptación simétrico llamado Data Encryption Standard (DES).

Este sistema no es el más seguro, ya que su longitud de clave es de 56 bits. Cuanto mayor sea la clave más seguridad proporcionará. Por este motivo, este sistema se considera actualmente poco práctico.

No obstante, Estados Unidos, en su búsqueda de un algoritmo más seguro y fiable para ser usado como estándar, convocó un concurso a nivel mundial, y hace dos años anunció su intención de adoptar un nuevo estándar. Así nació el nuevo sistema que recibe el nombre de Advanced Encryption Standard (AES). Este algoritmo, también denominado Rijndael, fue desarrollado por los belgas Vicent Rijmen y Joan Daemen. El gran avance de este estándar es que además de trabajar con claves de 128 y 192 bits, acepta longitudes de 256 bits.

Ventajas: Estos sistemas, pese a no ser del todo seguros, cuentan con la ventaja de la simplicidad y la rapidez.

Desventajas: Los sistemas de cifrado simétrico cuentan con desventajas como son la distribución de las claves y la dificultad de almacenar y proteger muchas claves diferentes.

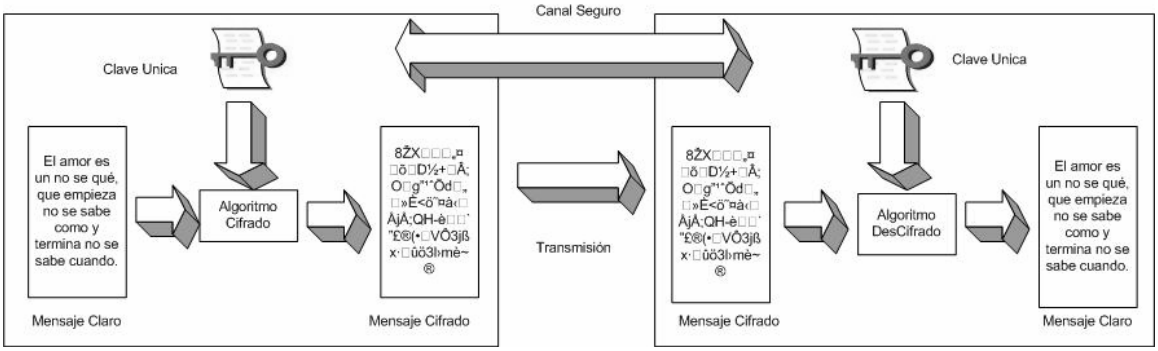

Figura No. 1 Cifrado y Descifrado de Datos Clave Secreta.

5.2 Criptografía de Clave Pública o Asimétrica

La criptografía de Clave Publica o asimétrica es por definición aquella que utiliza dos claves diferentes para cada usuario, una para cifrar que se le llama clave pública y otra para descifrar que es la clave privada. El nacimiento de la criptografía asimétrica se dio al estar buscando un modo más práctico de intercambiar las llaves simétricas. Diffie y Hellman [7], proponen una forma para hacer esto, sin embargo no fue hasta que el

popular método de Rivest Shamir y Adleman RSA[8] publicado en 1978, cuando toma forma la criptografía asimétrica, su funcionamiento esta basado en la imposibilidad computacional de factorizar números enteros grandes.

Los algoritmos de clave pública están basados en el uso de dos claves. Una clave, la pública, que se da a conocer al resto de usuarios, y otra clave, la privada, que se mantiene en secreto. Cada una de las claves realiza la función opuesta a la que realiza la otra. Para que el funcionamiento de un algoritmo de clave pública sea correcto debe ser computacionalmente intratable el problema de obtener la clave privada a partir de la clave pública. Ejemplos de estos problemas son la factorización de números primos grandes o el cálculo del logaritmo discreto. La ventaja de los algoritmos de clave pública es que eliminan la necesidad que tienen los de clave secreta de tener un secreto compartido entre los usuarios que desean usar un sistema criptográfico. Su principal desventaja es que son más costosos temporalmente. Además la generación de claves es más costosa en los algoritmos de clave pública. Mientras que en los algoritmos de clave secreta la generación de la clave puede ser aleatoria, en los de clave pública debe hacerse siguiendo un procedimiento determinado debido a la relación existente entre las dos claves.

Los tres usos principales de los algoritmos de clave pública son:

- Cifrado de mensaje
- Firma Digital
- Intercambio de Claves.

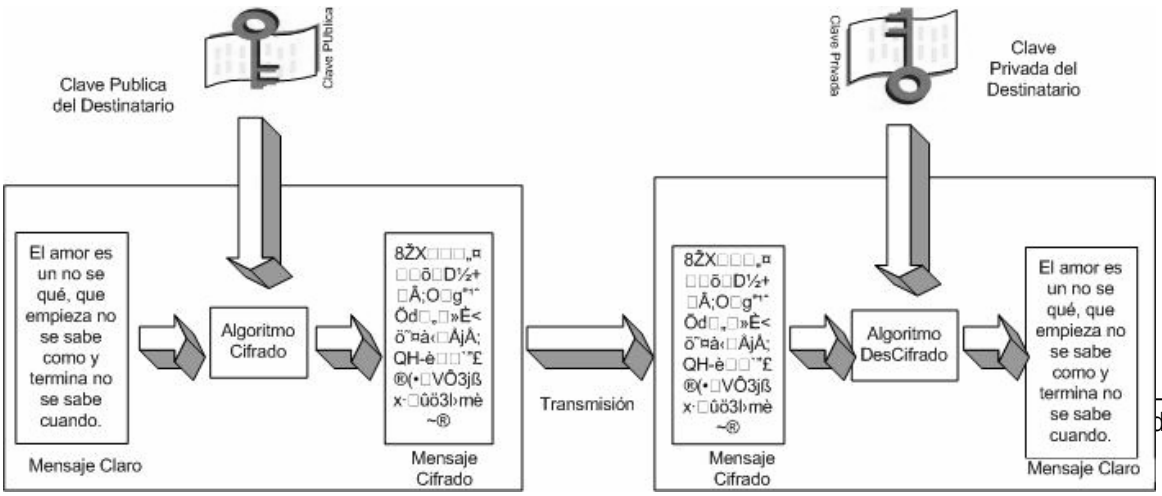

Figura No. 2 Cifrado y Descifrado de Datos Clave Publica.

El cifrado de mensajes como se muestra en la figura No. 2 se realiza cifrando el mensaje con la clave pública y descifrándolo con la clave privada correspondiente. Así, por ejemplo, si un usuario A quiere mandar un mensaje cifrado a otro usuario B puede cifrarlo usando la clave pública de B. Después sólo B podrá descifrarlo ya que sólo él posee la clave privada correspondiente. Otra posible forma de cifrar un mensaje es combinar un algoritmo de clave secreta con uno de clave pública. Se utiliza el algoritmo de clave secreta para cifrar el mensaje y el de clave pública para transmitir la clave secreta utilizada. Así se alivia parte del alto coste computacional provocado por el cifrado mediante clave pública.

La firma de mensajes se realiza cifrando un resumen del mensaje mediante la clave privada del usuario que firma el mensaje, y añadiendo esta firma al mensaje original. Cualquier otro usuario puede comprobar si la firma es correcta descifrándola mediante la clave pública y comparando lo obtenido al mensaje original. Si ambos resultados son iguales la firma es correcta. (Ejemplo de Firma Digital Figura No. 3)

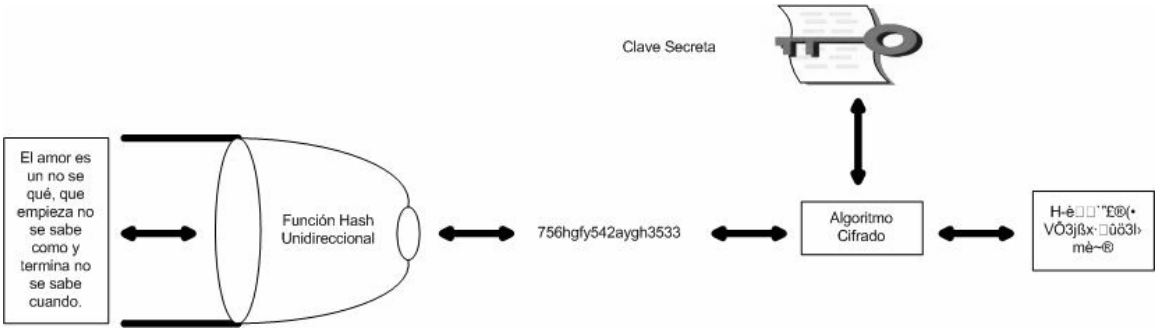

Figura No. 3 Ejemplo de Firma Digital.

Los sistemas de clave secreta son mucho más rápidos que los de clave pública, pero éstos permiten la firma digital. Por ello se suelen utilizar los sistemas simétricos para cifrar información y los asimétricos para firmar y para el intercambio de claves de sesión. Resumiendo:

- Los sistemas simétricos permiten cifrar la información, garantizando la CONFIDENCIALIDAD

- Los sistemas asimétricos permiten realizar firma electrónica, garantizando INTEGRIDAD, AUTENTICACIÓN y NO REPUDIO

Como para el caso del cifrado de mensajes, existe una forma de firmar mensajes menos costosos computacionalmente. Consiste en aplicar una función hash sobre el mensaje original, y cifrar el resultado mediante la clave privada. El resultado de esta operación es la firma. Para comprobarla lo que se hace es aplicar la misma función hash sobre el mensaje y compararla con el resultado de descifrar la firma usando la clave privada.

5.2.1 Función Hash

Generalmente en el intercambio de información lo que se cifra no es el mensaje original, sino un resumen o hash[10]. Del mismo. Por ello se explica a continuación las funciones que permiten la generación de resúmenes hash y las propiedades e importancia de las mismas.

Una de las aplicaciones más interesantes de la criptografía, es la posibilidad real de incluir en un mensaje una firma digital. Con los sistemas de clave simétrica esto era inviable. No obstante, dado que los sistemas de clave pública son muy lentos, en vez de firmar digitalmente el mensaje completo, en un sistema criptográfico se incluirá como firma una operación con clave privada sobre un resumen o hash de sólo una centena de bits.

A partir de un mensaje en texto plano, se obtiene su resumen al aplicar una función hash determinada. Este resumen se firma con la clave privada del emisor y se envía el receptor. Simultáneamente se envía el mensaje original al receptor. Éste descifra el

resumen mediante la clave pública del emisor, y aplica la misma función hash que él al mensaje recibido para obtener un resumen. Compara el resumen recién obtenido y el enviado por el emisor, si son iguales el mensaje es el inicial.

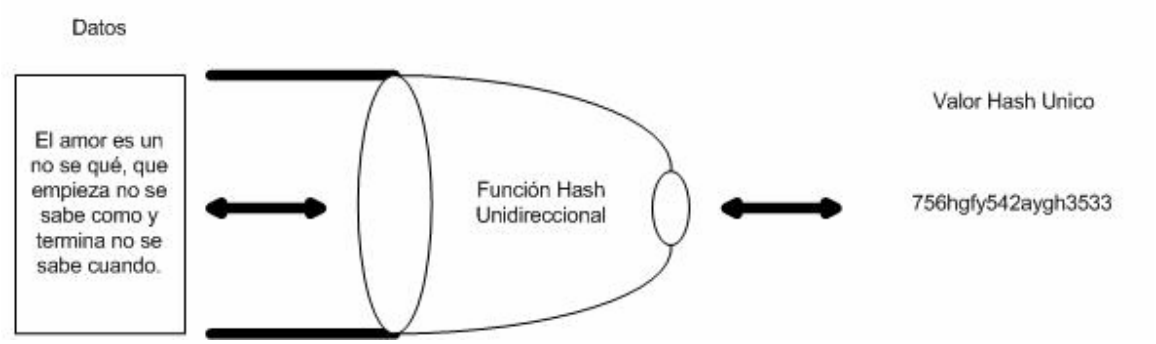

Figura No. 4 Función HASH

Algunas de las características de las funciones hash son las siguientes:

- Unidireccional: Conocido un resumen, es computacionalmente imposible encontrar el mensaje a partir de dicho resumen.
- Compresión: A partir de un mensaje de cualquier longitud, el resumen debe tener una longitud fija. Lo normal es que la longitud del resumen sea menor que la del mensaje original.
- Difusión: El resumen es una función compleja de todos los bits del mensaje.
- Colisión simple. Conocido M, será computacionalmente imposible encontrar otro mensaje tal que su resumen sea igual al resumen de otro mensaje. Se conoce como resistencia débil a las colisiones.
- Colisión fuerte. Será computacionalmente difícil encontrar un par de mensajes cuyos resúmenes sean iguales. Se conoce como resistencia fuerte a las colisiones.
- El intercambio de claves consiste en utilizar algún algoritmo de clave pública para negociar una clave secreta entre dos partes.

6 INFRAESTRUCTURA DE CLAVE PÚBLICA (PKI)

Para aprovechar todas las posibilidades que ofrece la criptografía de clave pública aparece la necesidad de algún mecanismo que sirva para distribuir las claves públicas entre los distintos usuarios. Con este objetivo aparecen las infraestructuras de clave pública (PKI) [11].

Una PKI "*es un sistema que ofrece servicios para el uso de criptografía de clave pública a un conjunto de usuarios*". Entre sus funciones básicas encontramos la gestión de certificados y de claves. Para un buen funcionamiento de una PKI son importantes las siguientes características:

- Transparencia: El usuario no necesita conocer los mecanismos de gestión de claves y certificados que utiliza la PKI para poder utilizar los servicios que ésta ofrece.
- Escalabilidad: La adición de nuevos usuarios a la PKI no supone decrementos importantes en las prestaciones de la misma.
- Compatibilidad: La implementación de la PKI es independiente del software que se utiliza a nivel de usuario (por ejemplo, navegadores o programa de correo).

- Seguridad: La PKI debe implementar mecanismos que permitan a los usuarios confiar en las operaciones realizadas utilizando sus servicios.
- Eficiencia: La interacción de los usuarios con la PKI debe realizarse con unos tiempos de respuesta pequeños.
- Disponibilidad: No se deben producir intervalos de tiempo en los que no se pueda usar la PKI.

En los puntos siguientes se comentan los elementos principales que debe implementar una PKI.

6.1 COMPONENTES DE UNA PKI

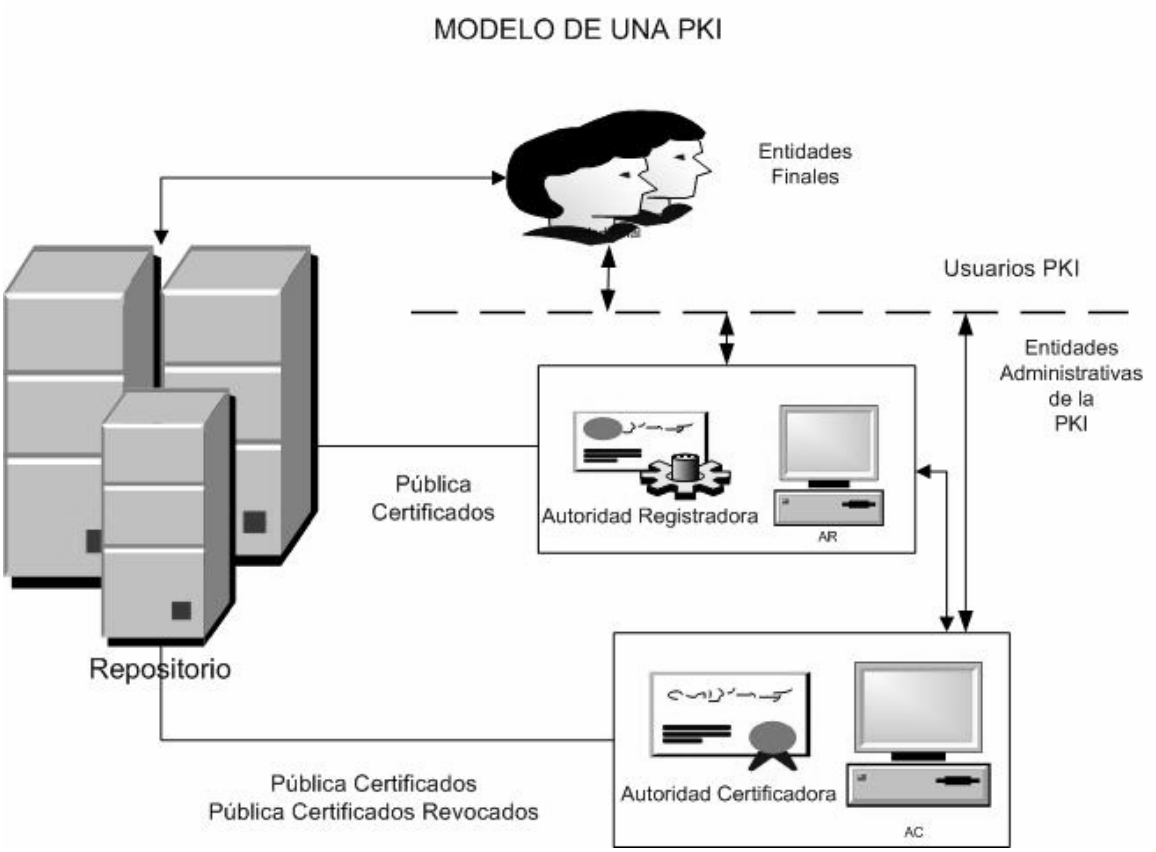

Figura No. 5 Modelo de una PKI.

En una PKI encontramos dos tipos de componentes principales: las autoridades certificadoras (AC) y las entidades finales. En algunos casos puede aparecer un tercer tipo de componentes, que son la autoridades de registro (AR).

6.1.1 CERTIFICADOS DE CLAVE PÚBLICA

Para poder utilizar las posibilidades que ofrece la criptografía de clave pública los usuarios necesitan estar seguros de que la clave pública que tienen asociada a otro usuario es la correcta. La PKI debe ser capaz de generar certificados.

6.1.1.1 Certificados Digitales

Un certificado digital es una estructura de datos que enlaza a una clave pública con la entidad a la que pertenece durante un intervalo de tiempo. Además de la clave pública

pueden aparecer en el certificado otros atributos que se desean enlazar con la entidad. A la entidad poseedora de la clave la llamaremos sujeto del certificado. Para que este enlace sea creíble, el certificado debe ir firmado por alguna entidad especial en la que confíen el resto de entidades. A la entidad que firma un certificado la llamaremos emisor del certificado. Gracias a esta firma se protege al receptor de un certificado frente a la modificación del certificado. Al estar el certificado firmado y gracias a las características de la firma digital, el certificado no ha podido ser modificado, ya que en ese caso la firma sería incorrecta. Aunque se han definido varios formatos de certificados digitales, el X.509, aceptado por la International Telecommunication Union-Telecomunication Standardization Sector (ITU-T) y por ISO/International Electrotechnical Comission (IEC), es el de mayor aceptación actualmente. Otros formatos definidos son los siguientes:

- SPKI (Simple Public Key Infraestructure) [12].
- PGP (Pretty God Privacy) [13].
- Certificados de Atributos

Pero ninguno de estos formatos tiene la aceptación ni ofrece las posibilidades que ofrecen los X.509. [14].

En el contexto de la criptografía de clave pública aparece la necesidad de distribuir la clave pública. Las estructuras de datos utilizadas como base para esta distribución son los certificados digitales.

6.1.1.2 Aplicaciones de los Certificados

Algunas de las aplicaciones más habituales en las que se utilizan certificados digitales son:

- Para la autenticación de Servidores Web---> Certificados de Servidor Web
- Para la autenticación de Clientes Web---> Certificados de Cliente Web
- Para la protección de correos electrónicos ---> Certificado de correo electrónico
- Para el sellado de tiempos ---> Certificado de tiempo

6.1.1.3 Ejemplo de Certificados Digitales.

6.1.1.4 Distribución de certificados

Los certificados se pueden distribuir de varias formas, dependiendo de la estructura del entorno PKI. Se pueden distribuir, por ejemplo, por los propios usuarios o a través de un servicio de directorios. Puede que ya exista un servidor de directorios dentro de una organización, o se pude suministrar uno como parte de la solución PKI. La creación de certificados por parte de una PKI no serviría de nada si estos no pudiesen ser distribuidos a los usuarios de la misma.

6.1.1.5 Copia de seguridad y recuperación de claves

En algunos casos un usuario puede perder su clave privada (por ejemplo, pierde la clave que usaba para protegerla o se produce un fallo de hardware). Si esta clave se utilizaba para cifrar, podría suponer el grave problema de que no se pueda recuperar la información cifrada. Para solucionar este problema la PKI debe ofrecer algún mecanismo de copia de seguridad de estas claves que permita su posterior recuperación en caso de pérdida. Esta copia deberá realizarla algún componente de la PKI en el que se confíe. Típicamente este componente será una AC.
Las únicas claves sobre las que se necesita realizar una copia de seguridad son las claves privadas utilizadas para cifrar. De hecho, puede que sólo el usuario tenga conocimiento de claves que no sean usadas para cifrado (por ejemplo, claves privadas utilizadas para firma digital).

6.1.1.6 Actualización de claves

Para garantizar la seguridad de un par de claves puede ser necesario que su uso se limite a un periodo de tiempo determinado. Finalizado este periodo de tiempo es necesario cambiar el par de claves generando uno nuevo.
Por razones de transparencia, este cambio debe ser automatizado de forma que no necesite intervención por parte del usuario. Además, el usuario no debe experimentar una interrupción en el funcionamiento del sistema a causa de que ha expirado su par de claves.

6.1.1.7 Historial de claves

El par de claves de un usuario puede cambiar con cierta frecuencia. Además, puede haber problemas si se pierde una clave utilizada para cifrar. Por esto aparece la necesidad de mantener un historial de claves.
La PKI deberá mantener un historial con las claves privadas viejas de los usuarios y los certificados asociados a su clave pública correspondiente. Así, cuando se necesite descifrar un documento cifrado con una clave privada distinta a la que utiliza actualmente el usuario, se recurre a este historial de claves para obtenerla. Este proceso debe ser también completamente transparente al usuario. Tanto el

almacenamiento como la utilización de la clave correcta deben ser automáticos y no requerir interacción con el usuario.

6.1.2 AUTORIDAD CERTIFICADORA (AC)

Las AC son entidades capaces de certificar la correspondencia entre una entidad y una clave pública. Para ello deben ofrecer un grado de seguridad que haga que el usuario pueda depositar su confianza en ellas. Cuando un usuario confía en una AC considera que todos los certificados emitidos por la misma son auténticos y correctos. Una AC debe ofrecer una serie de funciones relacionadas con la gestión de certificados. Las más importantes son:

- Registro de usuarios: deben realizar el mantenimiento de los usuarios afiliados a la AC. Estos usuarios deben poder ser identificados y se debe ofrecer algún mecanismo de autenticar para cuando quieran realizar operaciones con la AC.

- Emisión de certificados: deben poder crear certificados que enlacen a un usuario con una clave pública.(Figura No. 4)

- Gestión de certificados: deben incluir operaciones sobre certificados como por ejemplo revocación, renovación y suspensión.

- Distribución de certificados e información asociada a los mismos: deben ofrecer algún mecanismo para que los usuarios puedan acceder a los certificados y a información relacionada con los mismos (por ejemplo, estado de revocación).

- Gestión de las claves: deben ofrecer mecanismos para la creación de claves, su almacenamiento y recuperación en caso de pérdida y su renovación.

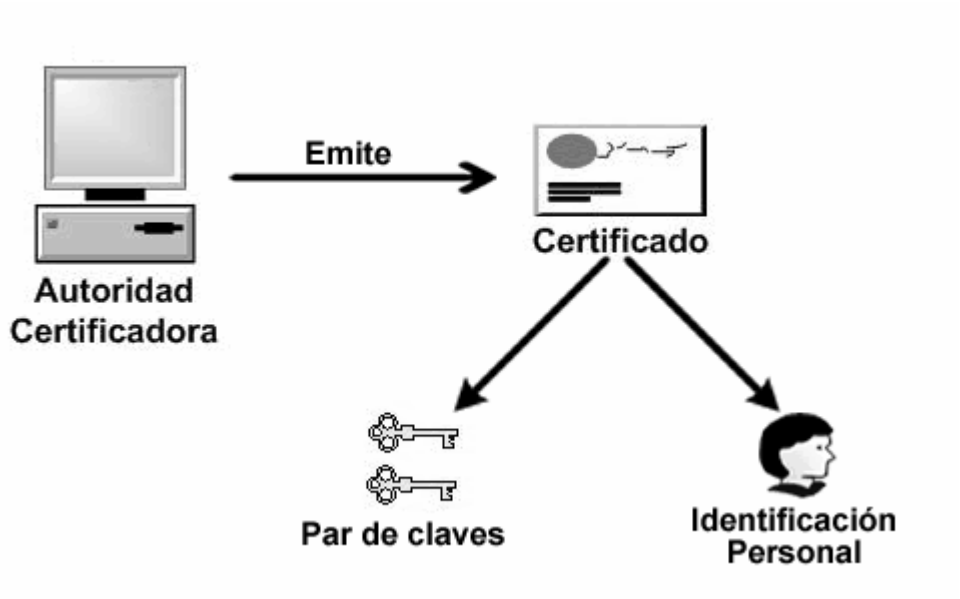

Figura No. 6 Autoridad Certificadora.

6.1.2.1 AUTORIDADES DE REGISTRO (RA)

En algunos casos aparece la necesidad de separar el proceso de autenticar los usuarios y gestión de las claves de los procedimientos relacionados con los procedimientos de gestión de los certificados. Por ejemplo, puede que la organización que quiere implantar la PKI tenga usuarios geográficamente dispersos. Además, para autenticar se exige alta seguridad, con lo cual puede necesitarse que la persona se presente físicamente. En cambio, los procedimientos de gestión de certificados no exigen tanta seguridad, con lo cual no se necesita una presencia física. Por tanto ambos procedimientos pueden ir separados, realizándose la gestión de claves y autenticar en varios nodos, evitado así largos desplazamientos al usuario. En cambio, la gestión de certificados se puede hacer centralizada, disminuyendo así los gastos. En este marco aparecen las autoridades registradoras (AR).

Una AR es una entidad que se comunica tanto con las entidades finales como con la AC, y que realiza funciones de autenticar usuarios y gestión de claves.
La AR aparece como un puente entre la entidad final y la AC. Para todos los procesos relacionados con autenticar usuarios y gestión de claves la entidad final interactuará con la AR, mientras que para los procedimientos relacionados con certificados interactuará directamente con la AC. (Ver Figura No. 5)

6.1.2.1.1 Repositorios de datos

La alternativa más aceptada actualmente para la distribución de la información en una PKI es la publicación de la misma en un repositorio. Un repositorio es una base de datos con una localización determinada y que puede ser accedida fácilmente. La información que se necesita distribuir se introduce en un repositorio, y desde éste puede ser accedida por los usuarios cuando la necesitan. Así se elimina la necesidad de que los usuarios interactúen entre ellos. La gestión de la información está centralizada en las AC.

Existen varias tecnologías para la creación de repositorios. Las principales son las siguientes:
- LDAP (Lightweight Directory Access Protocol) [15].
- Directorios X.500
- Servidores web y transmisión mediante http
- Servidores ftp
- Bases de datos corporativas

De todas estas tecnologías la más utilizada es LDAP[15]..

El uso de tecnologías estándar permite que el acceso a la información pueda realizarse por cualquier entidad que tenga conexión con la localización en que se encuentra el repositorio. Hay que tener en cuenta que en el repositorio no se almacena información que deba ser secreta, sino que el tipo de información que se almacena es toda pública, y no supone un problema que entidades ajenas a la PKI puedan acceder a ella. Sí que se debe proteger al sistema de accesos no autorizados para realizar modificaciones en la información contenida en el repositorio. La localización del repositorio debe ser comunicada al usuario para que éste pueda acceder al mismo. Esta localización se expresará típicamente en forma de una dirección IP o de un nombre DNS. Esta dirección se debe comunicar al dar de alta un usuario en la PKI, y la forma de hacerlo dependerá de la forma en que se realice este proceso.

Los principales problemas que pueden aparecer con el uso de repositorios están relacionados con el rendimiento. Al estar la información centralizada se puede sobrecargar el servidor donde se encuentra el repositorio. La técnica habitual en sistemas distribuidos para solucionar estos problemas es la replicación. Pero la

replicación de datos provoca la aparición de problemas adicionales de sincronización y actualización de las distintas réplicas. No se comentarán aquí las distintas técnicas existentes para solucionar estos problemas. Además se puede incurrir en sobrecarga de la red al transmitirse la información por la misma. Para minimizar en lo posible esta sobrecarga, será necesario conseguir que la información enviada tenga un tamaño tan pequeño como sea posible. Hay que tener en cuenta que la obtención de información será uno de los procesos más utilizados en una PKI, y por tanto debe evitarse, en lo posible, que suponga un cuello de botella del sistema.

6.1.2.1.2 Privacidad de la información

Se ha comentado anteriormente que la información contenida en el repositorio no tiene porqué ser protegida contra lectura. Esto no siempre es cierto, ya que, en algunos casos, el que esta información sea pública puede provocar que se pueda extraer información relativa a la organización que ésta no desea que sea pública. Por ejemplo, supongamos que en los certificados utilizados por una organización se almacena información acerca del cargo que desempeña el empleado que es sujeto del mismo, así como su departamento. Utilizando los certificados emitidos por esta empresa se podría deducir la infraestructura en que está distribuida la misma.

Una posible forma de solucionar este problema sería usar identificadores de usuario que tuviesen sentido sólo dentro de la organización. Por ejemplo podrían ser identificadores numéricos. Así, una entidad externa a la organización no podría deducir nada de estos identificadores. Pero también se reduce la interoperabilidad con entidades externas, ya que no les resultaría posible identificar al sujeto de un certificado. Otra posible forma sería proteger los accesos al repositorio mediante algún método de autenticar. Pero aparece el mismo problema de interoperabilidad, ya que, o bien las entidades externas no pueden acceder, o bien pueden obtener la información, no solucionándose el problema. Debido a los problemas de estos dos métodos será necesario buscar algún método para proteger esta información. Una buena solución sería incluir en los certificados la mínima información necesaria para la identificación del sujeto. Así se reduciría la posibilidad de extraer información de los mismos, aunque no se eliminaría.

6.1.2.1.3 Acceso Externo a un Repositorio

Como ya se ha comentado, será necesario que entidades externas a la PKI puedan acceder al repositorio mediante el cual ésta distribuye la información. Esto supondrá en algunos casos el acceso a la red propia de la organización propietaria de la PKI. A continuación se comentan los principales modelos para realizar este acceso.

6.1.2.1.3.1 Acceso Directo

En este modelo el acceso de una entidad externa a un repositorio se realiza directamente al mismo. Será adecuado cuando exista confianza entre ambas PKI o el repositorio esté protegido mediante algún mecanismo de autenticar. Este método podría sobrecargar al servidor que contiene el repositorio de la PKI, ya que además de los accesos internos deberá soportar los accesos externos. Supone además un aumento en el consumo del ancho de banda de la red de la organización.

6.1.2.1.3.2 Repositorio Compartido

La existencia de un repositorio compartido posibilitaría que todas las PKI que necesiten interactuar publicasen su información en un único repositorio externo a todas ellas. La gestión de este repositorio podría ser compartida por las PKI que publican en él o ser realizada por una entidad externa. Los métodos de acceso y de actualización de la información contenida en el repositorio no necesitan ser iguales para todas las PKI participantes. La implementación del repositorio deberá permitir la actualización por medio de los distintos protocolos que utilicen las PKI.

Este método podría presentar problemas de rendimiento en la lectura cuando sean muchas las PKI que compartan un mismo repositorio. Una posible solución sería la replicación del repositorio. Así se aumentarían las prestaciones frente al acceso. Pero esta replicación reduciría el rendimiento al realizar actualizaciones, por la necesidad de mantener la sincronización entre todas las copias. Según qué operación sea la que predomine interesará un grado mayor o menor de replicación.

6.1.2.1.3.3 Replicación Ínter dominio

La replicación ínter dominio consiste en que cada PKI mantenga en su repositorio la información de los repositorios de las otras PKI. Cada PKI debe informar a las demás de los cambios en su repositorio y enviarles la nueva información.

En este caso será interesante el uso de un protocolo común para las actualizaciones, ya que de otra forma se incurre en una complicación elevada del proceso de automatización de las actualizaciones. En este método serán muy eficaces las lecturas, ya que siempre se realizarán del repositorio de la propia PKI. En cambio, el costo de las escrituras será elevado, ya que se tiene que escribir en todas las PKI que puedan necesitar la información que se va a publicar.

6.1.2.1.3.4 Repositorio borde

Esta técnica consiste en que cada PKI mantiene un repositorio externo a la red interna de la empresa. A este repositorio se le llama repositorio borde. De esta forma cada PKI mantiene dos repositorios, uno externo y uno interno. Las consultas de componentes de la PKI se realizan al repositorio interno y las de elementos ajenos a las PKI al repositorio borde.

Mediante este método se evita la necesidad de soportar varias formas de actualizar la información del repositorio, ya que las actualizaciones en cada repositorio las realiza siempre la misma PKI. Además se elimina la necesidad de que entidades externas accedan al interior de la red de la empresa. Las escrituras presentan una complicación mínima añadida al mantenimiento del repositorio interno, ya que lo único que se tiene que mantener es otra réplica de la misma información. En el caso de las lecturas, la separación de las internas y las externas evita la sobrecarga añadida por las consultas externas.

6.1.2.1.3.5 Protocolos de intercambio en línea.

Otra posibilidad existente para la distribución de la información es su intercambio a través de la misma red. Por ejemplo, el uso de S/MIME[16], para correo electrónico permite el intercambio de certificados y LRC (Listas de Certificados Revocados) [17]. Este método se utiliza principalmente cuando no existe un mecanismo de repositorios

en la red sobre la que se implementa la PKI o como apoyo a algún método basado en repositorios.

6.1.2.2 Modelos de confianza

El gran número de usuarios que puede tener una ICP hacen que sea necesaria la existencia de varias AC en la misma, ya que el trabajo a realizar puede ser excesivo para una única AC. Estas AC deberán estar distribuidas de alguna forma que permita determinar a las entidades finales en cuales de los certificados que les llegan pueden confiar. A la forma de realizar esta distribución se le llama modelo de confianza. Diremos que una entidad confía en un certificado cuando considera que el enlace especificado en el mismo entre una entidad y una clave pública es correcto. Asimismo, una entidad confía en una AC cuando confía en todos los certificados emitidos por la misma. La emisión de certificados por parte de una AC que enlacen a otra AC con una clave pública permite que entidades que sólo confiaban en la primera AC pasen a confiar en la segunda.

En los apartados siguientes comentamos diversas estructuras posibles para la distribución de AC. Se introduce también la certificación cruzada, elemento básico de alguno de los modelos.

6.1.2.2.1 Herencia Estricta

En la herencia estricta, los componentes de la PKI se distribuyen en forma de árbol. En la raíz tendremos una AC a la que llamaremos AC raíz. En las hojas están las entidades finales. Entre ambos extremos tendremos una serie de niveles de nodos, que serán las AC intermedias. En este esquema todos los componentes confían en la AC raíz. Después cada AC emite certificados para los componentes situados en nodos hijos de la misma. Así, partiendo de un certificado emitido por la AC raíz se puede obtener un camino que lleve hasta un certificado emitido para una entidad final. Para que se pueda comprobar la validez de un certificado, es necesario que cada componente de la estructura posea una copia de la clave pública de la AC. Esta clave debe ser distribuida por algún método seguro, de forma que no pueda ser modificada.
Habitualmente este proceso necesitará el apoyo de algún mecanismo que funcione al margen de la red, como por ejemplo el correo convencional o el teléfono.

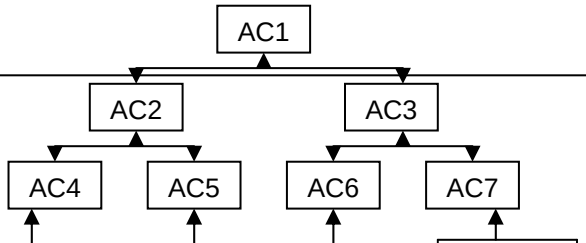

Figura No. 7 Herencia Estricta.

6.1.2.2.2 Arquitectura Distribuida

En este modelo no se confía en una única AC raíz, sino que distintas entidades finales pueden tener su confianza depositada en distintas AC. Así se obtienen una serie de subestructuras independientes, que típicamente tendrán estructura de herencia estricta con una AC que funcionará como raíz en esa subestructura. En el caso de que las subestructuras estén compuestas por una única AC y entidades finales, la configuración se llamará completamente igualada. En esta estructura todas las AC son independientes unas de otras. Si en cada subestructura hay más de una AC (hay AC subordinadas a otras) tendremos una configuración completamente arbolada. Se pueden tener también configuraciones híbridas.

Para que una entidad final pueda confiar en certificados emitidos en otra subestructura se usará la certificación cruzada entre las AC que sirven como raíz en ambas subestructuras. A continuación se comentan dos posibles configuraciones para realizar la certificación cruzada.

6.1.2.2.3 Configuración en Red

En esta configuración se puede establecer certificación cruzada de cada AC raíz con todas las demás. En el caso de que se realice la certificación cruzada de todas con todas hablaremos de configuración en red completa. Si no hay certificación cruzada entre todas las AC raíz tendremos configuración en red parcial. Este tipo de configuraciones exigen un gran número de certificaciones cruzadas. En configuración en red completa se necesitan del orden de n^2 certificaciones cruzadas, siendo n el número de AC raíz.

6.1.2.2.4 Configuración centralizada

En esta configuración tendremos una AC central con la que cada una de las AC raíz establecerá certificación cruzada. A la AC central se le suele llamar eje, y su función principal es la de interconectar las distintas infraestructuras. La ventaja de esta configuración frente a la anterior es que necesita menos certificaciones cruzadas. En el caso de conexión completa se necesitan del orden de n conexiones (siendo n el número de AC raíz) frente a las n^2 del modelo anterior. Podría parecer que esta configuración es igual a la de herencia estricta, pero esto no es cierto. La diferencia estriba en la clave pública por la que cada entidad final comienza las comprobaciones. Cada entidad final tiene la clave pública de la AC raíz de la subestructura a la que pertenece, no la de la AC central. Así, la validación del camino de certificación parte de la AC raíz, y pasará en caso de ser necesario por la AC central mediante certificación cruzada, pero en ningún caso comenzará por esta AC central.

6.1.2.2.5 Modelo Web

Este modelo se diferencia de los anteriores en que la confianza no se deposita en una única AC, sino que se deposita en varias. En este caso también hay varias subestructuras, pero, a diferencia de la arquitectura distribuida, la entidad final deposita su confianza en todas las AC raíz. Podemos considerar este modelo como una extensión del de herencia estricta, en el que cada entidad final puede formar parte de varias estructuras. El nombre del modelo viene de su nacimiento en el entorno de la World Wide Web. Está soportado por la mayoría de los navegadores, como por ejemplo Netscape Navigator y Microsoft Explorer.

Este modelo presenta más problemas de seguridad que los anteriores por culpa de que se deposita la confianza en varias AC. En caso de mal funcionamiento (voluntario o involuntario) de alguna de estas AC resulta difícil para el usuario encontrar el responsable del error. En cambio, en los otros modelos al estar depositada la confianza en una única AC los errores siempre partirán de ésta.

6.1.2.2.6 Certificación cruzada

La certificación cruzada es un mecanismo utilizado para establecer una relación de confianza entre dos AC. Mediante la misma se puede conseguir que entidades que

confían en certificados emitidos por una AC pasen a confiar también en los certificados emitidos por la otra AC. Diremos que una AC A certifica cruzadamente a otra AC B cuando firma un certificado que enlace a B con la clave pública de B. En la certificación cruzada se usan certificados. En estos certificados tanto el sujeto como el emisor serán AC. Además, el uso de la extensiones de los certificados puede permitir incluir restricciones a esta relación, obteniendo así una relación de confianza con respecto únicamente a algunos de los certificados emitidos por la otra entidad.

La certificación cruzada puede ser unilateral o bilateral. La certificación unilateral ocurre en un único sentido, es decir, una AC A certifica cruzadamente a una AC B, mientras que B no certifica cruzadamente a A. Este método se utiliza para establecer la herencia entre AC en las estructuras de árbol. En éstas la AC padre certifica cruzadamente a todas las AC que son hijas suyas.

La certificación bilateral se realiza en ambos sentidos. En este caso la AC A certifica cruzadamente a la AC B y al mismo tiempo B certifica cruzadamente a A. Este método se utiliza principalmente en la arquitectura distribuida para establecer la confianza entre dos subestructuras. Supongamos que se desea establecer una certificación cruzada bilateral entre la AC A y la AC B. Para ello se emiten dos certificados. Un certificado lo emite A para la clave de B poniendo a B como sujeto. Este certificado será el certificado cruzado directo para B y el certificado cruzado inverso para A. En el otro se certifica la clave de A con B como emisor y A como sujeto. Éste será el directo para A y el inverso para B. (Figura No. 8)

En una certificación unilateral sólo se emitirá uno de los certificados. Si A certifica cruzadamente a B se emitirá el inverso de A, que es directo para B.

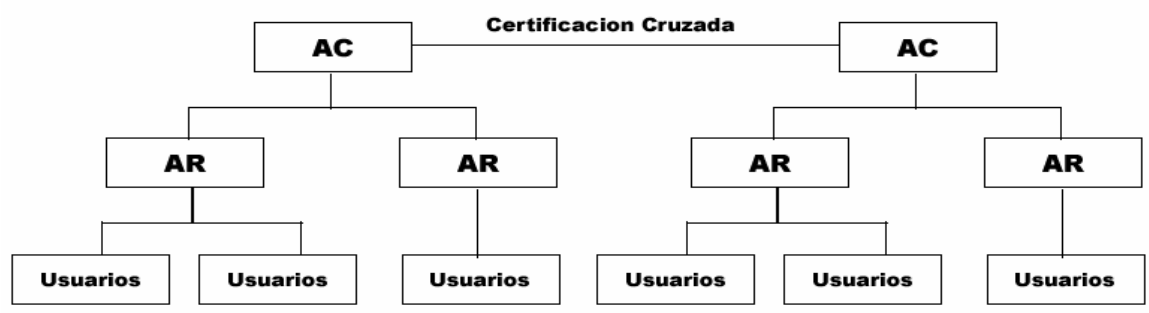

Figura No. 8 Certificación Cruzada.

6.2 DISTRIBUCIÓN DE LA INFORMACIÓN EN UNA PKI

Dado que uno de los principales objetivos de una PKI es ofrecer información a los usuarios sobre las claves públicas de otros usuarios, es un factor de vital importancia la forma en que esta información se distribuye.

Los certificados ofrecen una buena forma de garantizar la correspondencia entre una clave pública y un usuario. Pero para que sean útiles los usuarios deben poder acceder a ellos con comodidad y rapidez. Es importante además que puedan acceder a otra información relacionada con los certificados como es su estado de revocación, las políticas de certificación y certificados cruzados.

Las características más importantes de esta distribución son:

- Escalabilidad: la adición de nuevos usuarios al sistema no supone un gran decremento en las prestaciones del mismo.
- Eficiencia: la información se obtiene en poco tiempo.
- Disponibilidad: la información está accesible en todo momento de tiempo.
- Integridad: los datos distribuidos son correctos y no pueden ser modificados malintencionadamente.

A continuación se comentan los métodos más importantes para permitir el acceso de los usuarios a esta información.

6.2.1 Entrega privada

La forma más sencilla de distribuir la información sería que cada usuario fuese el encargado de entregar el certificado que le corresponde a los usuarios que pueden necesitarlo.

Una posible forma de realizar esta entrega sería mediante disquete. El usuario copia el certificado en un disquete y se lo entrega al usuario que puede necesitarlo. Otra posible forma sería añadir el certificado a un mensaje de correo electrónico. Este método podría resultar eficaz en comunidades pequeñas de usuarios, en las que la mayoría de los mismos se conociesen y pudiesen tener un contacto directo. Pero no en una comunidad

de usuarios grande, ya que sería muy complicada la distribución de la información directamente entre sus usuarios.

Además resulta muy difícil mediante este método el mantener la información actualizada. Por ejemplo, cuando un usuario necesitase comprobar el estado de revocación de un certificado debería usar algún método especial, como una llamada telefónica directa al sujeto del certificado, para comprobar este estado. Así, el usuario debe estar muy implicado en todo el proceso de certificación, reduciendo mucho la transparencia.

6.2.2 Entidad final

Las entidades finales representan a los usuarios de la PKI. Utilizan los servicios ofrecidos por las AC para poder obtener las posibilidades que ofrece la criptografía de clave pública. Entre estas posibilidades tenemos:

- Cifrado de mensajes.
- Firma digital de mensajes.
- No repudiación.
- Descarga de software de confianza.
- Seguridad en servidores Web.

En general, las entidades finales no deben realizar ninguna función de gestión de los certificados. Son simples usuarios, que obtienen los certificados desde una AC y los utilizan. Sí que deben tener la posibilidad de almacenar los certificados, para no tener que obtenerlos cada vez que los necesiten. En cuanto a las claves sí que deben ofrecer varias posibilidades. Por una parte, en algunos casos es más interesante la generación de la clave en la entidad final que en la AC. Por otra parte es necesario que las claves se puedan almacenar de forma segura, para así mantener la seguridad de la PKI.

7 PROTOCOLOS DE SEGURIDAD

Un protocolo de seguridad es la parte visible de una aplicación, es el conjunto de programas y actividades programadas que cumplen con un objetivo específico y que usan esquemas de seguridad criptográfica.

El ejemplo más común es SSL[19]. (**S**ecure **S**ockets **L**ayer) (que vemos integrado en el Browser de Netscape y hace su aparición cuando el candado de la barra de herramientas se cierra y también sí la dirección de Internet cambia de http a https), otro ejemplo es PGP[13]. que es un protocolo libre ampliamente usado de intercambio de correo electrónico seguro, uno más es el conocido y muy publicitado SET[20]. que es un protocolo que permite dar seguridad en las transacciones por Internet usando tarjeta de crédito, IPsec que proporciona seguridad en la conexión de Internet a un nivel más bajo.

Estos y cualquier protocolo de seguridad procura resolver algunos de los problemas de la seguridad como la integridad, la confidencialidad, la autenticación y el no rechazo, mediante sus diferentes características

Las características de los protocolos se derivan de las múltiples posibilidades con que se puede romper un sistema, es decir, robar información, cambiar información, leer información no autorizada, y todo lo que se considere no autorizado por los usuarios de una comunicación por red.

7.1 PROTOCOLO SSL

Secure Sockets Layer (SSL) es un protocolo diseñado por Netscape Communications, dispone de un nivel seguro de transporte entre el servicio de transporte en Internet (TCP) y las aplicaciones que se comunican a través de él.

7.1.1 Características

Proporciona conexiones seguras sobre una red insegura como es Internet, asegurando las siguientes características:

1. Conexión privada: la información se cifra utilizando criptografía de clave simétrica (IDEA, DES, etc).

2. Autenticación: del servidor por medio de certificados digitales, y del cliente utilizando criptografía de clave asimétrica (RSA,EL GAMAL, etc).

3. Integridad: la integridad de los mensajes se asegura usando funciones hash seguras (MD5, SHA-1, etc).

Además, proporciona características adicionales:

1. Extensibilidad: es capaz de soportar nuevos protocolos, métodos de cifrado, etc.

2. Eficiencia: al utilizar compresión, minimiza el tiempo necesario para establecer la conexión.

3. Compatibilidad: productos con diferentes versiones de SSL pueden ínter operar entre sí.

SSL se compone de dos partes diferenciadas:

1. Handshake Protocol: se encarga de establecer la conexión y determinar los parámetros que se van a utilizar posteriormente (fundamentalmente se trata de establecer cual va a ser la clave simétrica que se utilizará para transmitir los datos durante esa conexión).

2. Record Protocol: comprime, cifra, descifra y verifica la información que se transmite.

Este sistema es transparente para las aplicaciones finales, es totalmente independiente del protocolo de aplicación usado. Por tanto, podemos situar protocolos como HTTP, FTP, o Telnet.

7.1.2 Funcionamiento

El denominado Handshake Protocol se compone de dos fases, autenticación de servidor y autenticación de cliente, no siendo obligatoria esta última. En primer lugar, el servidor, respondiendo a una petición del cliente, le envía su certificado y las preferencias en cuanto a algoritmos de cifrado se refiere. En ese momento, el cliente genera una clave maestra, la cifra con la clave pública del servidor y la transmite al servidor. El servidor recobra la clave maestra y se autentica respecto al cliente devolviendo un mensaje cifrado con la clave maestra. Los datos siguientes son cifrados con claves derivadas de esta clave maestra.

En la segunda fase opcional, el servidor envía un reto al cliente. Éste se autentica respecto al servidor retornándole el reto firmado digitalmente por el cliente, así como su certificado (el cual incluye su clave pública).

A partir de aquí, lo demás consiste en cifrar y descifrar la información que se transmitió, en el protocolo de aplicación utilizado.

7.1.3 Algoritmos utilizados

SSL soporta gran variedad de algoritmos criptográficos. Durante la fase de acuerdo o "handshaking", se utiliza RSA (clave pública). Después del intercambio de claves, se usan unos cuantos algoritmos, entre los que se incluyen RC2, RC4, IDEA, DES y Triple-DES. Como función resumen se usa MD5 o SHA-1. Los certificados siguen el formato X.509.

7.1.4 Implementación

Los diferentes protocolos que utilizan los servicios de SSL usan puertos diferentes a los que les correspondería si no fuesen sobre SSL. La IANA ha reservado los siguientes puertos para su uso por SSL:

· 433: HTTP sobre SSL (https)

· 465: SMTP (correo electrónico) sobre SSL (ssmtp), no confirmado.

· 563: NNTP (servicio de noticias, News) sobre SSL (snntp), no confirmado.

El protocolo SSL está ampliamente extendido. La presencia de https:// en el URL de un servidor indica que se trata de un servidor "seguro" y que debe utilizarse SSL en la comunicación entre dicho servidor y cliente (navegador). Esto queda indicado (en el caso de Netscape Navigator) de la siguiente forma:

- La llave de la parte inferior izquierda del navegador aparece completa, no partida como habitualmente (en los casos del MS Internet Explorer y de Netscape Communicator es un candado cerrado el que aparece en la esquina inferior izquierda).
- Aparece una línea azul en el límite superior de la línea de visualización de la pantalla del navegador.

- La información del documento alojado en el servidor seguro incluye los datos del certificado que avala al servidor seguro.

A diferencia de S-HTTP, que es un protocolo substitutivo de HTTP, SSL extiende su soporte a otros protocolos habituales en Internet. Esta es una de las principales ventajas que aporta este último. Mientras que S-HTTP proporciona cifrado en el nivel de aplicación (en este caso WWW), SSL lo hace en el nivel de conexión, proporcionando un canal seguro en el nivel de red. Por lo demás, S-HTTP y SSL pueden convivir, utilizándose uno u otro en diferentes instantes de una transacción comercial, o incluso utilizándose simultáneamente.

El sistema es tan robusto como lo sea el menos seguro de los algoritmos que utilice. Claves públicas cortas o claves DES o RC4 de 40 bits deben utilizarse con precaución. Estos son los problemas que plantean las leyes de EE.UU.

La principal desventaja de SSL no estriba en sus fundamentos teóricos o implementación, sino, fundamentalmente, la menor protección que proporcionan las versiones exportables de los productos basados en este protocolo.

8 LDAP (PROTOCOLO DE ACCESO A DIRECTORIOS SIMPLES)

LDAP (``Lightweight Directory Access Protocol'', «Protocolo Ligero de Acceso a Directorios») es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio. Se usó inicialmente como un front-end o interfaz final para X.500, pero también puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio.

Un directorio es como una base de datos, pero en general contiene información más descriptiva y más basada en atributos. La información contenida en un directorio normalmente se lee mucho más de lo que se escribe. Como consecuencia los directorios no implementan normalmente los complicados esquemas para transacciones o esquemas de reducción (rollback) que las bases de datos utilizan para llevar a cabo actualizaciones complejas de grandes volúmenes de datos. Por contra, las actualizaciones en un directorio son usualmente cambios sencillos de «todo o nada», si es que se permiten en algo.

Los directorios están afinados para proporcionar una repuesta rápida a operaciones de búsqueda o consulta. Pueden tener la capacidad de replicar información de forma amplia, con el fin de aumentar la disponibilidad y la fiabilidad, y a la vez reducir el tiempo de respuesta. Cuando se duplica (o se replica) la información del directorio, pueden aceptarse inconsistencias temporales entre la información que hay en las réplicas, siempre que finalmente exista una sincronización.

Existen muchas maneras distintas de proporcionar un servicio de directorio. Los diferentes métodos permiten almacenar en el directorio diferentes tipos de información, establecer requisitos diferentes para hacer referencias a la información, consultarla y actualizarla, la forma en que protege al directorio de accesos no autorizados, etc. Algunos servicios de directorio son locales, proporcionando servicios a un contexto restringido (por ejemplo, el servicio de finger en una única máquina). Otros servicios son globales, proporcionando servicio en un contexto mucho más amplio.

8.1 Funcionamiento

El servicio de directorio LDAP se basa en un modelo cliente-servidor. Uno o más servidores LDAP contienen los datos que conforman el árbol del directorio LDAP o base de datos troncal. El cliente LDAP se conecta con el servidor LDAP y le hace una consulta. El servidor contesta con la respuesta correspondiente, o bien con una indicación de dónde puede el cliente hallar más información (normalmente otro servidor LDAP). No importa con qué servidor LDAP se conecte el cliente: siempre observará la misma vista del directorio; el nombre que se le presenta a un servidor LDAP hace referencia a la misma entrada a la que haría referencia en otro servidor LDAP. Es ésta una característica importante de un servicio de directorios universal como LDAP.

8.2 Características de LDAP

- Corre sobre TCP/IP, OSI
- Simple

- Omite duplicados
- Usa cadenas que representan datos de complicada estructura de sintaxis ANS.1(Abstract Sintax Notation One)

8.3 Entradas , Objetos y atributos en LDAP

Un servicio de directorio nos da el medio de organizar y simplificar un acceso a los recursos en cualquier sistema basado en red. Usuarios y administradores, pueden no conocer el nombre exacto de los recursos que necesitan. Pero seguro que conocen uno o más atributos de los objetos que desean encontrar. Con un servicio de directorio se puede preguntar que me muestre una lista de objetos que coincidan con atributos que se conozcan. Un servicio de directorio hace posible encontrar un objeto basándose en uno o más de sus atributos.

Otros servicios suministrados por un directorio son:

- Refuerza la seguridad para proteger los objetos de intrusos o de usuarios internos que no tienen permiso para acceder a estos objetos.
- Realiza una copia de si mismo (replicación) en otros ordenadores para estar siempre disponible en caso de fallo o caída en el ordenador en donde reside.
- Dividir un directorio en múltiples almacenamientos que están localizados en diferentes maquinas a lo largo de la red.

Un servicio de directorio son ambas cosas: una herramienta de administración y una herramienta de usuario final.

Una entrada es una unidad en un directorio LDAP. Una entrada es identificada por su único Distinguished Name (DN). Cada entrada tiene atributos, los atributos son fragmentos de información directamente asociados con la entrada. Por ejemplo, la Universidad Don Bosco podría ser una entrada LDAP. Los atributos asociados a la universidad podrían ser el número de fax, la dirección, etc. Las personas podrían ser otra entrada

del directorio LDAP. Atributos comunes a las personas son el número de teléfono y sus direcciones de e-mail. Ciertos atributos son necesarios, mientras que otros son opcionales. Un objectclass (Clases de Objetos) discrimina los atributos necesarios de los que no lo son.

Para importar y exportar información de directorio entre servidores de directorios basados en LDAP, o para describir una serie de cambios que han de aplicarse al directorio, se usa en general del fichero de formato conocido como LDIF (siglas de "LDAP interchange format", «formato de intercambio de LDAP»). Un fichero LDIF almacena información en jerarquías de entradas orientadas a objeto. Un fichero LDIF corriente tiene este aspecto:

```
dn: o=Insflug, c=ES
o: Insflug
objectclass: organization
dn: cn=Luiz Malere, o=Insflug, c=ES
cn: Luiz Malere
sn: Malere
mail: malere@yahoo.com
objectclass: person
```

Como se puede comprobar, cada entrada está identificada unívocamente por un nombre distintivo (DN, "distinguished name"). El DN (nombre distintivo) está compuesto por el nombre de la entrada en cuestión, más la ruta de nombres que permiten rastrear la entrada hacia atrás hasta la parte superior de la jerarquía del directorio.
En LDAP, una clase de objetos define la colección de atributos que pueden usarse para definir una entrada.

El estándar LDAP proporciona estos tipos básicos para las clases de objetos:

- Grupos en el directorio, entre ellos listas no ordenadas de objetos individuales o de grupos de objetos.
- Emplazamientos, como por ejemplo el nombre del país y su descripción.
- Organizaciones que están en el directorio.
- Personas que están en el directorio.

Una entrada determinada puede pertenecer a más de una clase de objetos. Por ejemplo, la entrada para personas se define mediante la clase de objetos person, pero también puede definirse mediante atributos en las clases de objetos inetOrgPerson, groupOfNames y organization. La estructura de clases de objetos del servidor (su esquema) determina la lista total de atributos requeridos y permitidos para una entrada concreta.

Los datos del directorio se representan mediante pares de atributo y su valor. Cualquier pieza de información específica se asocia con un atributo descriptivo.
Por ejemplo el atributo commonName, o cn («nombre de pila»), se usa para almacenar el nombre de una persona. Puede representarse en eldirectorio a una persona llamada Jonás Saqueiro mediante

cn: Jonás Saqueiro

Cada persona que se introduzca en el directorio se define mediante la colección de atributos que hay en la clase de objetos person. Otros atributos que se usan para definir esta entrada serán:

givenname: Jonás
surname: Saqueiro
mail: jonass@midominio.com

Los atributos requeridos son aquellos que deben estar presentes en las entradas que utilicen la clase de objetos. Todas las entradas precisan del atributo objectClass, que lista las clases de objeto a las que pertenece una entrada.

Los atributos permitidos son aquellos que pueden estar presentes en las entradas que utilicen la clase de objetos. Por ejemplo, en la clase de objetos person, se requieren los atributos cn y sn. Los atributos description («descripción»), telephoneNumber («número de teléfono»), seeAlso («véase también»), y userpassword («contraseña del usuario») se permiten pero no se requieren.

Cada atributo tiene la definición de sintaxis que le corresponde. La definición de sintaxis describe el tipo de información que proporciona ese atributo:

- bin binario
- ces cadena con mayúsculas y minúsculas exactas (las mayúsculas y minúsculas son significativas durante las comparaciones)

- cis cadena con mayúsculas y minúsculas ignoradas (las mayúsculas y minúsculas no son significativas durante las comparaciones)
- tel cadena de número de teléfono (como cis, pero durante las comparaciones se ignoran los espacios en blanco y los guiones "-")
- dn "distinguished name" («nombre distintivo»)

9 PROTOCOLO X.509

El protocolo **X.509** es el sistema de certificados de clave pública más utilizado. Su origen es el directorio X.500, inventado por la UIT para dar servicio al correo electrónico X.400. Actualmente se utiliza en los protocolos seguros y en los sistemas de correo Internet más conocidos, excepto el PGP. Permite trabajar con CA y anidar certificados para crear estructuras jerárquicas.

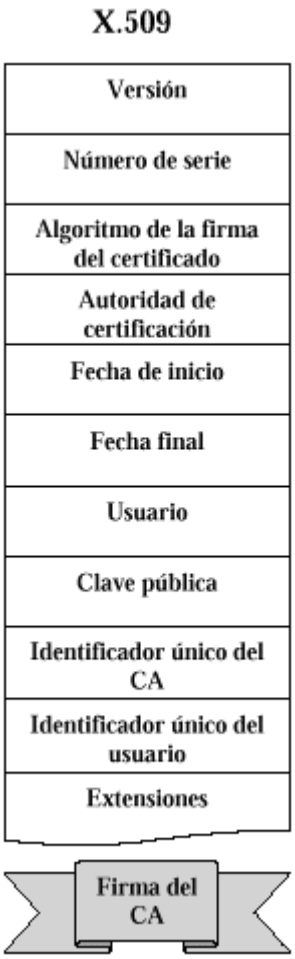

Figura No. 9 Estructura protocolo X.509.

Todos los campos del certificado están escritos en formato ANS.1. Sus contenidos:

- **Versión.** La versión de protocolo X.509.
- **Número de serie (*SerialNumber*).** identificador único del certificado, asignado por el CA.
- **Algoritmo de la firma del certificado (*Signature*).** X.509 permite utilizar diferentes algoritmos para firmar el certificado, este campo lleva el identificador del algoritmo.
- **Autoridad de certificación (*Issuer*).** Nombre de la CA.

- **Fechas de inicio y final (*Validity*).** El certificado solo tiene validez entre estas dos fechas. Es conveniente no permitir un periodo de validez largo y así obligar a renovar claves y certificados con asiduidad.

- **Usuario (*Subject*).** Nombre del usuario.

El estándar, internacionalmente aceptado, para Certificados Digitales, es el denominado X.509, en su versión 3.

Contiene datos del sujeto, como su nombre, dirección, correo electrónico, etc..

Con la versión 3 de X.509, sucesora de la versión 2, no hace falta aplicar restricciones sobre la estructura de las CAs gracias a la definicion de las extensiones de certificados. Se permite que una organización pueda definir sus propias extensiones para contener información específica dentro de su entorno de operación. Este tipo de certificados es el que usa el protocolo de comercio electrónico SET.

9.1 X.509 Versión 3

X.509 y X.500 fueron originalmente diseñados a mediados de los años 80, antes del enorme crecimiento de usuarios en Internet. Es por esto por lo que se diseñaron para operar en un ambiente donde sólo los computadores se interconectaban intermitentemente entre ellos. Por eso en las versiones 1 y 2 de X.509 se utilizan CRLs muy simples que no solucionan el problema de la granularidad de tiempo.

La versión 3 introduce cambios significativos en el estándar. El cambio fundamental es el hacer el formato de los certificados y los CRLs extensible. Ahora los que implementen X.509 pueden definir el contenido de los certificados como crean conveniente. Además se han definido extensiones estándares para proveer una funcionalidad mejorada.

9.1.1 Campos del X.509v3

X.509 da tres procedimientos alternativos para la autenticación en peticiones de servicio, mensajes o envío de información.

- Autenticación a una vía (una transmisión)
- Autenticación a dos vías (una transmisión + respuesta)
- Autenticación a tres vías (una transmisión + respuesta + acuse de recepción)

Procedimientos de autenticación en una red

- KUB Clave Pública de B
- KRB Clave Privada de B
- KAB Clave simétrica de sesión entre A y B
- EKXX[...] Encriptación con la clave EKXX
- DKXX[...] Encriptación con la clave DKXX
- A{ X } Firma por A de X
- tA Marca de tiempo
- rA Testigo (número aleatorio único que A no repetirá durante la vida del mensaje).

En todos estos procedimientos, se supone que las dos partes conocen la clave pública de la otra, bien porque la han obtenido de un directorio, o bien porque en el mensaje inicial va incluida.

Autentificación en una vía

El mensaje mínimo está formado por el testigo y la marca de tiempo. Puede además contener una clave de sesión temporal entre A y B.

El envío de información de A a B, define:

La identidad de A y que el mensaje fue generado por A

Que el mensaje estaba dirigido a B

La integridad y unicidad del mensaje.

Autentificación en dos vías

Consiste en el envío de información de A a B, y a continuación de B a A.

Define además de los anteriores:

La identidad de B, y que el mensaje fue generado por B

Que el mensaje estaba dirigido a A

La integridad y unicidad del segundo mensaje

Autentificación en tres vías

La autentificación de tres vías se emplea cuando el destino y el iniciador no tienen relojes sincronizados o no desean confiar en los relojes. Además de pasar por la autentificación de dos vías, el iniciador envía entonces una respuesta a la respuesta del destino incluyendo el nuevo testigo contenido en la respuesta original. Después de verificar que los valores del testigo son idénticos, ya no hay necesidad de verificar las marcas de tiempo.

10 PROYECTO INFRAESTRUCTURA DE CLAVE PUBLICA PARA UNIVERSIDAD DON BOSCO PKI – UDB

El Proyecto PKI - UDB surge de la necesidad de ampliar los conocimientos de los alumnos a tecnología mas avanzada orientada a Internet. En este caso el presente trabajo de graduación se oriento a realizar una Infraestructura de Clave Publica para la Universidad Don Bosco, consiste en la creación de una serie de servicios para el envió y recibo de información por correo electrónico de una manera segura.

Todos los detalles del proyecto se describen a continuación:

Árbol de Certificación:

El Árbol de Certificación se compone de los siguientes elementos:

1.- Autoridad de Certificación: Que se auto certifica y firmara los certificados de los miembros de la comunidad Universitaria.

2.- Autoridad Registradora: Que serán los encargados de la autenticación e identificación de los usuarios.

3.- Certificados Digitales de Identidad Personal.

10.1 AUTORIDAD CERTIFICADORA

Una Autoridad Certificadora (CA,por sus siglas en inglés)como dijimos anteriormente es la encargada de confirmar que el dueño de un certificado es realmente la persona que dice ser. Una Autoridad Certificadora define las políticas especificando cuáles campos del Nombre Distintivo son opcionales y cuáles requeridos. También puede especificar requerimientos en el contenido de los campos.

Existen varias Autoridades Certificadoras, puede que una autoridad certificadora certifique o verifique la identidad de otra Autoridad Certificadora y así sucesivamente; pero habrá un punto en que una Autoridad no tendrá quién la certifique, en este caso, el certi-

ficado es firmado por uno mismo (``self-signed''), por lo tanto, la Autoridad Certificadora es verificada o confiada por ella misma.

Las Autoridades Certificadoras (o notarios electrónicos) deben ser entes fiables y ampliamente reconocidos que firman las claves públicas de las personas, certificando con su propia firma la identidad del usuario.

10.1.1 Características

La Autoridad Certificadora provee los servicios de:

1. Verificación de solicitud de Certificados.
2. Procesamiento de solicitud de Certificados.
3. Firma, asignación y manejo de Certificados.

Los certificados se ofrecen por parte de una Autoridad Certificadora a la solicitud de una persona, entidad u organización que así lo requiera.
A continuación se presenta un ejemplo de cómo le enviaría información cifrada usando la verificación de certificados representado en la fig. 10:

1. Se envía un mensaje pidiendo su certificado.
2. Usted regresa su certificado.
3. Se verifica con la Autoridad Certificadora que su certificado sea válido. Especialmente, que dicha Autoridad Certificadora fue quien le dio el certificado y que su llave pública es la misma que la del certificado.
4. Se recibe la confirmación de la Autoridad Certificadora que el certificado es válido.
5. La información se cifra usando su llave pública y luego es enviada.

Usted recibe la información y la descifrada usando su llave privada.

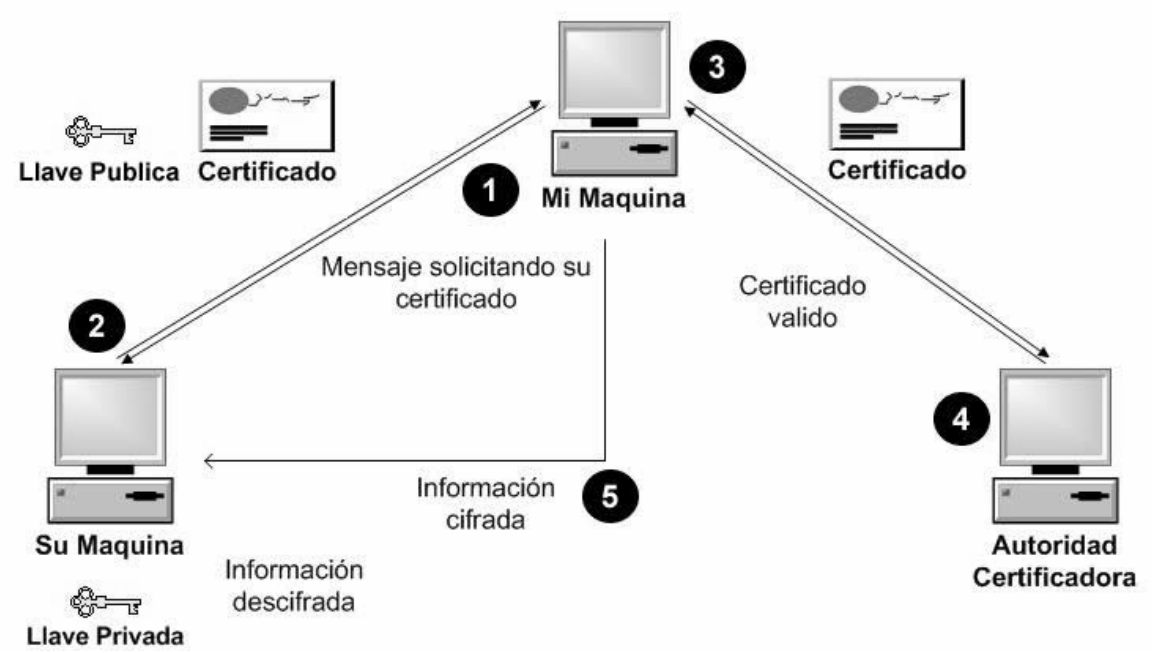

Figura No. 10 Proceso de la Autoridad Certificadora.

10.1.2 Software y Hardware

Software

- Sistema Operativo
  Distribución de Linux Mandrake 9.1
- Servidor Web
  Apache 2.0.44
- Software de seguridad
  OpenSSL 0.9.7b
- Repositorio de Datos
  OpenLDAP 2.0.27
- Software de CA
  OpenCA 0.9.1-1
- Servidor de correo
  Postfix 2.0.6
- Navegador Web
  Netscape 7.1
- Base de Datos
  MySQL 4.0.11a

Para accesar al OpenSSL a través de Perl se utilizan los módulos siguientes del OpenCA:

OpenCA::OpenSSL y OpenCA::X509.

OpenCA::OpenSSL contiene funciones que crean y revocan los certificados

OpenCA::X509 valida el certificado de un cliente.

Hardware

Pentium III 350 Hz

256 MB en Ram

8 GB Disco Duro

Tarjeta de Red 100 Mbps

10.1.3 Configuración Utilizada

En el Anexo, se muestran los pasos básicos para implementar una autoridad certificadora.

10.2 AUTORIDAD REGISTRADORA

La Autoridad de Registro (AR) es la encargada de hacer las veces de interfaz hacia el exterior. Su labor fundamental, es la de llevar a cabo los procedimientos de registro para la emisión de nuevos certificados.

10.2.1 Características

Tras haber comprobado la identidad del sujeto para el que se va a emitir el certificado, la AR envía a la AC una solicitud para firmar un nuevo certificado. Una vez firmado el nuevo certificado se publicará en el directorio y ya podrá ser utilizado.

Otra de las labores de la autoridad de registro es recibir las solicitudes de revocación de un certificado. De acuerdo con los procedimientos adecuados, cuando éstas solicitudes sean verificadas se enviarán también a la AC, quien las firmará. A continuación se publicarán en el directorio y desde ese momento cualquiera que consulte la *lista de certificados revocados* (CRL), verá que ese certificado ya no es válido.

10.2.2 Software y Hardware Utilizado

Hardware

Pentium III 350 Hz
256 MB en Ram
8 GB Disco Duro
Tarjeta de Red 100 Mbps

Software

- Sistema Operativo
  Distribución de Linux Mandrake 9.1
- Servidor Web
  Apache 2.0.44
- Software de seguridad
  OpenSSL 0.9.7b
- Repositorio de Datos
  OpenLDAP 2.0.27
- Software de CA
  OpenCA 0.9.1-1
- Servidor de correo
  Postfix 2.0.6
- Navegador Web
  Netscape 7.1
- Base de Datos
  MySQL 4.0.11a

10.2.3 Configuración Utilizada

En el Anexo, se muestran los pasos que se siguieron para implementar una Autoridad Registradora.

10.3 CONFIGURACIÓN DEL CORREO ELECTRÓNICO

Configuración del Servidor de Correo con Soporte TLS en la Infraestructura De Clave Pública de la Universidad Don Bosco Postfix+TLS+Certificados+PKIUDB

En 1998 comenzó a difundirse el uso de un nuevo sistema de gestión de correo electrónico bajo la denominación de IBM Secure Mailer, aunque posteriormente pasaría a denominarse Postfix. Este producto se desarrolló en el centro de investigación Thomas J. Watson Research Center, de IBM.

El autor, *Wietse Zweitze Venema*, conocido por sus desarrollos software para la protección contra intrusiones de sistemas informáticos, había elaborado un sistema de gestión de correo electrónico cuyas características básicas eran rapidez, facilidad de configuración y, sobre todo, seguridad.

Postfix es un servidor de correo MTA (Mail Transport Agent), muy potente, programado por Wietse Venema, cuya página web es http://www.postfix.org que comenzó siendo una alternativa a Sendmail que controla cerca del 70% el movimiento de correo electrónico en Internet. En este documento se explicar como instalar el MTA Postfix en un Linux Mandrake 9.1 con kernel 2.4.21, pero es totalmente válido para otras versiones o distribuciones de Linux.

10.3.1 Comandos Básicos de POSTFIX

Existen varios comandos que nos pueden ser útiles mientras usemos Postfix. Una breve lista será:

- postfix stop. Este comando para parar el servidor.
- postfix start. Este comando arranca el servidor.
- postfix reload. Este comando hace que el servidor relea la configuración sin parar el servicio.
- mailq. Para ver la cola de mensajes.
- postfix flush. Fuerza el envío de mensajes de la cola de espera.
- postmap. Este comando sirve para construir los ficheros auxiliares de Postfix.
- postconf. Muestra toda la configuración de Postfix.
- newaliases. Este comando reconstruye la base de datos de alias.

10.3.2 Configuración de POSTFIX

En **/etc/postfix** se encuentran los archivos de configuración. Estos son dos: **main.cf** y **master.cf**. **main.cf** es el archivo principal, donde reside el corazón del funcionamiento de Postfix. El archivo **master.cf** es un tipo de archivo tipo **inetd.conf**, donde los distintos programas de Postfix ven su forma de funcionar.

Advertencia: el archivo **master.cf** es el mas delicado y complicado. No modificar si no se sabe que se esta haciendo.

10.3.2.1 Modos de ejecución del servidor

Existen 2 modos de ejecución, por así decirlo. El modo **Internet Site** y el modo Internet **Site With Smarthost**.

Internet Site

El modo Internet site se caracteriza porque el propio servidor se encarga de repartir los mensajes a sus destinatarios directamente, sin pasar por otro servidor predefinido.

Para usar este modo, en el fichero de configuración **/etc/postfix/main.cf** NO debe estar definida la opción

relayhost relayhost =

Esta configuración es útil para ordenadores individuales que no están en una red local o tienen conexión permanente a Internet (como ADSL, cable).

Internet Site With Smarthost

El modo internet site with smarthost se caracteriza porque el servidor no envía los mensajes directamente a sus destinatarios, sino que los envía a otro servidor de correo, y aquel ya se encargara de enviarlo. Para usar este modo, hay que definir la opción **relayhost** y ponerle como argumento la dirección IP o el nombre de host del servidor SMTP que queramos.

relayhost = ca.pkiudb.edu.sv

Esta configuración se suele dar en redes locales que ya tienen un servidor SMTP o en conexiones esporádicas a Internet con módem, por ejemplo (el servidor definido seria el de tu proveedor).

10.3.2.2 Primeras Configuraciones: HOSTNAME, DOMAIN, NETWORKS

Lo primero es ingresar a main.cf el nombre del host (**$myhostname**), el dominio (**$mydomain**). Networks se usa para indicarle a Postfix que maquinas, distinguidas por IP o direccion son consideradas locales y pueden usar al servidor de correo (**MAILSVR** en adelante) para envios. Networks puede ser una colección de Ips o una clase completa.

Por ejemplo, nuestro MAILSVR se llama **ca.pkiudb.edu.sv** y el dominio se llama **pkiudb.edu.sv**, modificamos las siguientes lineas en **main.cf:**

myhostname = ca.pkiudb.edu.sv
mydomain=pkiudb.edu.sv

Lo siguiente es indicar si se quiere enmascarar las direcciones de correo. Esto es para que usuarios que pertenezcan a distintos subdominios aparezcan que son enviados desde un mismo dominio (bar.com). Esto es solamente usado si se tiene un dominio con distintas maquinas. El valor por defecto es **$mydomain**

myorigin = $myhostname
myorigin = $mydomain

El parámetro **mydestination** especifica que dominios entregar localmente, en vez de enviarlo a otras maquinas. El valor por defecto es entregarlo al mismo MAILSVR. Puede especificarse ninguno o varios dominios y tablas de lookup con separaciones por espacios o comas.

mydestination = $myhostname localhost.$mydomain
mydestination = $myhostname $mydomain
mydestination = $myhostname www.$mydomain ftp.$mydomain

Precaución: siempre agregar **$myhostname y localhost.$mydomain** para evitar cadenas repetitivas de entregas de correo.

Generalmente los correos son rechazados. Algunas veces, algunos correos que no son rechazados simplemente no son entregados. Para ello, existe un usuario, postmaster que a quien llegan los correos no entregados. Generalmente llegan aquellos correos con un gran conjunto de errores.

Para saber por que no se entrego correo, la directiva **notify_classes** indica el nivel de error a notificar. Los valores que puede tener son:

- **bounce** : envía a postmaster copias de los correos no entregados, pero estas copias son modificadas para proteger la privacidad del mensaje.
- **2bounce** : envía dos copias del mail que rebota.
- **policy**: informa a postmaster las peticiones rechazadas por políticas UCE de otros servidores. Llega una copia de la transacción.
- **protocol** : informa a postmaster cualquier error de protocolos, cliente o servidor, o intentos de algún cliente de ejecutar comandos no implementados. Se recibe una copia de la transaccion completa.
- **resource** : informa a postmaster de los mail no entregados por algún problema de recursos (errores read/write, queue, etc)
- **software**: informa a postmaster de problemas de software.

Cualquiera de estas opciones pueden ser combinadas.

notify_clases = resource, software

La directiva **mynetworks** permite que una red se considere local para Postfix. Esto es para distinguir entre maquinas conocidas de las extrañas (fuera de la red). Las maquinas consideradas como locales pueden usar a MAILSVR como un open relay incluso.

Puede configurarse una clase A, B o C, dependiendo de la cantidad de maquinas.

mynetworks = 192.168.10.0/24, 127.0.0.0/8

El parametro **inet_interfaces** indica que interfaces de red debe escuchar **MAILSVR**. Los correos enviados a user@direccion_de_red serán entregados localmente, y direccionados a un dominio que este listado en **$mydestination**.

El valor por defecto es **all** (todas las interfaces). Si se tienen interfaces virtuales, se debe indicar cuales de las interfaces escuchar.

inet_interfaces = all
 # dominio virtual
inet_interfaces = virtual.host.name
 # mailer no virtual
inet_interfaces = $myhostname localhost.$mydomain

La opcion **relay_domains** restringe los dominios donde los clientes usan a MAILSVR para enviar correo (relay) o que destinos va a servir MAILSVR.

Por defecto, Postfix relega (relay) correo a: clientes confiables que su dirección esta en **$mynetworks** clientes confiables que estén en **$relay_domains** o algún subdominio clientes no confiables los cuales el destino sea **$relay_domains** o algún subdominio de el.
Postfix además acepta correo para:

Los destinos que estén en $inet_interfaces
Los destinos que estén en $mydestination
Los destinos que estén en $virtual_maps
relay_domains = $mydestination

10.3.2.3 Control de Envíos

El control de envíos significa que se pueden definir qué direcciones de correo pueden enviar correo a través de nuestro servidor, y qué direcciones de correo no pueden enviar correo a nuestro servidor.

Por host o redes

Mediante la directiva mynetworks definimos, se ha definido qué redes o hosts pueden enviar correo a través de nuestro servidor Postfix. Un ejemplo servía

mynetworks = 127.0.0.0/8, 192.168.10.0/24

Con esta configuración estamos definiendo:

- La red 127.0.0.0 puede enviar. Esta red siempre seria nuestra propia máquina (localhost).
- Los 254 hosts de la red 192.168.10.0 pueden usar nuestro servidor.

Opciones Adicionales

La opción **queue_directory** especifica el lugar de la **cola** de Postfix. Es también el directorio raíz de los demonios de Postfix (que corren chrooted).

queue_directory = /var/spool/postfix

command_directory y **daemon_directory** contienen la ruta donde están los comandos de Postfix y los demonios, respectivamente

command_directory=/usr/sbin
daemon_directory=/usr/libexec/postfix

mail_owner indica el usuario que es propietario de la cola de Postfix. Especificar un usuario que no comparta un grupo con otras cuentas y que no posea otros archivos o procesos en la misma maquina. O sea, ni nobody ni daemon. Se debe usar un usuario dedicado.

La instalación de Postfix crea el usuario y el grupo postfix. Seria lógico usarlo para mail_owner.

mail_owner = postfix

10.3.2.4 Soporte de Transportation Layer Security (TLS)

Habitualmente las comunicaciones mediante protocolo *SMTP (Simple Mail Transport Protocol)* se realizan sin utilizar mecanismos de cifrado, por lo que toda la información viaja en claro por Internet, lo que para cierto tipo de usuarios implica
Invalidar el uso de este medio.

En 1999, con aplicación no sólo a SMTP, se definió el protocolo *TLS (Transportation Layer Security)* basado en *SSL (Secure Socket Layers)*, y cuya definición formal puede encontrarse en el *RFC-2246*. TLS básicamente proporciona cifrado en las comunicaciones y autentificación entre ambos corresponsales mediante el uso de *certificados X.509*.

La integración del protocolo **TLS y SMTP** se define en el *RFC-2487*, y se implementa en *ESMTP (Extended Simple Mail Transport Protocol)*, en concreto; en la negociación inicial (EHLO). El servidor ofrece la prestación de TLS mediante la opción **STARTTLS**, invitando al cliente a enviar la directiva STARTTLS y pasar a un estado de comunicaciones cifradas.

Hasta hace poco tiempo, las versiones estables de Postfix no proporcionaban directamente soporte de TLS, aunque en la versión 2.0.6 que se esta utilizando ya esta disponible, gracias a ***Lutz Jänicke*** que las ha creado y mantiene de manera impecable. En la página oficial de Postfix, en la sección *Add-on Software* se encuentra el enlace a su página.
Para saber si un servidor esta ofreciendo servicio TLS se puede establecer una conexión al puerto SMTP (25) e iniciar el dialogo:

$ telnet ca.pkiudb.edu.sv smtp
Trying 138.100.8.30...
Connected to ca.pkiudb.edu.sv
Escape character is '^]'.
220 ca.pkiudb.edu.sv ESMTP Postfix-TLS-SASL/FI-0602 (1.1.11-20020613)
ehlo ca.pkiudb.edu.sv
250-ca.pkiudb.edu.sv
250-PIPELINING
250-SIZE 10240000

250-ETRN

250-STARTTLS

250-AUTH LOGIN PLAIN DIGEST-MD5 CRAM-MD5

250-AUTH=LOGIN PLAIN DIGEST-MD5 CRAM-MD5

250-XVERP

250 8BITMIME

Como puede apreciarse, una vez realizada la identificación inicial mediante la directiva EHLO, el servidor proporciona la relación de funciones que soporta, entre las que aparece el protocolo TLS mediante la opción STARTTLS.

10.3.2.4.1 Configuración.

Se recomienda antes de proceder a configurar y activar el soporte de TLS, se verifique que Postfix esta funcionando sin problema alguno. No es recomendable incrementar los niveles de complejidad innecesariamente.

Como anteriormente se ha comentado, TLS se basa en el uso de certificados X.509, por lo que será necesario disponer de una clave privada y de un certificado firmado por alguna CA (Autoridad de Certificación):

- Certificado de la CA.
- Certificado del Servidor.
- Clave Privada del Servidor.

Normalmente, y evidentemente por razones de seguridad, la clave privada siempre se almacena protegida por una frase de acceso. Esto implica que cada vez que se accede a dicha clave privada será necesario proporcionar la frase.

En un servidor de correo esta circunstancia puede ser un inconveniente, por lo que será necesario disponer la clave privada sin protección de frase de acceso. Como contrapartida, es necesario tener un cuidado exquisito a la hora de configurar las protecciones de directorio y ficheros relacionados.

Suponiendo que la clave privada se encuentra en formato PEM en el fichero **cakey.pem**, con la directiva:

#/usr/local/ssl/bin/openssl rsa -inform pem -in cakey.pem -text -out keysin.pem

se obtendrá en el fichero **keysin.pem** se almacene la clave privada sin frase de protección. Será este fichero el que se proporcionará a Postfix cuando se indique la clave privada del servidor.

Por otra parte, Postfix diferencia la faceta cliente y servidor, por lo que es necesario activar el soporte de TLS en la configuración para la parte cliente y para la parte servidor de manera separada. Las directivas que comienzan por "**smtp**_" corresponden a la faceta cliente, mientras que los que empiezan por "**smtpd**_" son los relativos a la faceta servidor.

En base a lo visto, una configuración básica de Postfix para soporte TLS sería:

 # CONFIGURACION TLS
 # Cliente
smtp_use_tls = yes
smtp_tls_session_cache_database = sdbm:/etc/postfix/smtp_scache
smtp_tls_key_file = /etc/postfix/cert/keysin.pem
smtp_tls_cert_file = /etc/postfix/cert/cert.pem
smtp_tls_CAfile = /etc/postfix/cert/CertCA.pem

 # CONFIGURACION TLS
 # Servidor
smtpd_use_tls = yes
smtpd_tls_session_cache_database = sdbm:/etc/postfix/smtpd_scache
smtpd_tls_key_file = /etc/postfix/cert/keysin.pem
smtpd_tls_cert_file = /etc/postfix/cert/cert.pem
smtpd_tls_CAfile = /etc/postfix/cert/CertCA.pem

De esta manera, Postfix ofertará como servidor la posibilidad de establecer comunicación cifrada mediante la opción STARTTLS. Al mismo tiempo, y como cliente, intentará establecer comunicación cifrada con todo servidor que le ofrezca dicha posibilidad.

Como puede apreciarse, los certificados y clave privada se han ubicado en el directorio **/etc/postfix/cert**:

drwxr-x--- 2 root root 4096 jun 29 20:46 /etc/postfix/cert

Las protecciones aplicadas a los ficheros son:
-rw------- 1 root root 3877 jun 29 20:44 CertCA.pem
-rw------- 1 root root 4200 jun 26 13:32 cert.pem
-rw------- 1 root root 3640 jun 26 13:32 keysin.pem

Aunque el único con el que hay que tener especial cuidado es **keysin.pem**, pues los otros son públicos: Certificado de CA y Certificado de Servidor.

En el caso de que se establezcan conexiones con servidores cuyos certificados estén firmados por otras CA, sólo tendremos que concatenar en el fichero CertCA.pem los certificados de las respectivas CA.

11 CONCLUSIONES

Con el presente proyecto concluimos que en nuestro país debemos tener muy presente la seguridad en Internet, ya que es un tema que todavía no es muy fuerte, ni ha cobrado mucho auge en nuestro medio.

El proyecto PKI UDB es un indicio de que es necesario fomentar mas la seguridad de nuestros datos que a diario viajan por Internet y que son expuestos a gente mal intencionada que degrada o altera la información, no llegando a su destinatario en su forma original y que a veces es usada para otros fines. Además se pretende dar un paso mas en el desarrollo de tecnologías que nos ayuden como alumnos de la Universidad Don Bosco a implantar proyectos de esta categoría para poder estar al tanto de todas las innovaciones, que requieran un mayor crecimiento para nuestra Universidad.

Hemos pretendido dejar un avance para futuros proyectos que deseen realizar, en base al trabajo realizado.

La Infraestructura de Clave Publica nos enseña que hay diversa maneras de que un mensaje enviado por correo, donde es fundamental que nadie lo vea ni lo altere; pueda ser protegido, en este caso será de mucha utilidad para el envío de correo seguro entre diversos entes de la universidad que pretendan no ser victimas de intersección en el envío o recibo de la información.

En el área informática es necesario mantenernos a la vanguardia de la tecnología, por esa razón este proyecto pretende dar un paso de lo que es seguridad en Internet para que la Universidad Don Bosco por medio de sus estudiantes sea un centro de enseñanza actualizado y le proporcione mejores herramientas a la hora de trabajar en el área de Internet en el cual nadie esta libre de gente inescrupulosa que gusta aprovecharse de la información ajena.

12 REFERENCIAS

[1] D. Kahn, The Codebreakers*, the Story of Secret Writing, Macmillan Publishing Co. NY* 1967

[2] D.R. Stinson, Cryptography Theory and Practice, CRC Press Inc. 1995

[3] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography, CRC Press* 1996

[4] X. Lai, J.R. Massey, *A proposal for a new block encryption standard, Advances in Cryptology EUROCRYPT'90, LNCS 473, pp 389-404*, 1991

[5] R.L. Rivest, *The RC5 encryption algorithm, Fast software Encryption LNCS 1008, pp 86-96*, 1995

[6] http://csrc.nist.gov/encryption/aes/rijndael/
http://www.esat.kuleuven.ac.be/rijmen/rijndael/

[7] W. Diffie, M.E. Hellman, *New Directions in Cryptography, Transactions on Information Theory Vol IT22 No 6,* pp 644-654 1976

[8] R.L. Rivest, A. Shamir, L. Adleman, *A Method for Obtaining Digital Signature and Public-Key Cryptosystems, Communication of the ACM Vol 21 No 2 pp 120-126*, 1978

[9] FIPS 186, “Digital signature standards” ,1994

[10] ISO 10118-1,2,3,4 “*Information technology- security techniques- hash functions*” 1994, 1996

[11] ANSI X9.57 *“Public key cryptography for financial services industry” Certificate management”* 1995

[12] C. Ellison – *“SPKI Requirements”* 1999

[13] P. Zimmermam. “*The official PGP user's guide”*, 1995.

[14] M. Myers and R. Ankney and A. Malpani and S. Galperin and C. Adams. “*X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP”* 1999

[15] Good, G., "The LDAP Data Interchange Format (LDIF)", RFC 2849, June 2000.

[16] N. Freed, Innosoft, N. Borenstein, “*Multipurpose Internet Mail Extensions (MIME)”* 1996

[17]] A. Arsenault and S. Turner. “*Internet X.509 Public Key Infrastructure PKIX Road-map*”, 1999. IETF Internet Draft. Draft-ietf-pkix-roadmap-06.txt

[18] G. N. Drew, *Using Set for Secure Electronic Commerce, , Prentice Hall,* NJ 1999

[19] L. Loeb, *Secure Electronic Transactions, Introduction and Technical Reference, Artech House*, 1998

13 GLOSARIO

AUTENTIFICACIÓN DE MENSAJES

Proceso de autentificación que incluye la identificación de la fuente del mensaje y la verificación de que no ha sido modificado o reemplazado en el tránsito del mismo.

AUTENTIFICACIÓN

Mecanismos del sistema de información para poder identificar a los usuarios que acceden a sus recursos, y asegurar la integridad y autenticidad de los datos

CERTIFICADO DIGITAL

La certificación es el proceso de ligar una clave pública a los datos de su propietario.

CIFRADO

Proceso utilizado para transformar un texto a una forma ininteligible de manera que los datos originales no puedan ser recuperados (cifrado de una vía) o sólo puedan ser recuperados usando un proceso inverso de descifrado (cifrado de dos vías).

CLAVE PRIVADA

Código digital de uso privado, usado conjuntamente con la clave pública, para cifrar y descifrar información.

CLAVE PÚBLICA

Clave de usuario que es conocida por el resto de los usuarios y que es utilizada para verificar firmas creadas con su correspondiente clave pública. Dependiendo del algoritmo, se usa para cifrar mensajes que pueden ser descifrados con su correspondiente clave privada.

CRL

Acrónimo de Certificate Revocation List. Lista emitida por entidades de certificación en la que se publican todos aquellos certificados que han dejado de tener validez.

DAP

Directory Access Protocol. Protocolo de acceso al directorio X509.

DES

Data Encryption Standard. Algoritmo de cifrado.

DIB

Directory Information Base.

DIT

Directory Information Tree. Arbol de Directorio

DPC

Declaración de Prácticas de Certificación.

DSA

Directory System Agent definido en la norma X.509.

FIRMA ELECTRÓNICA

Conjunto de datos, que se añaden al mensaje, para que lo protejan contra cualquier falsificación. Permitiendo al receptor comprobar el origen y la integridad de los datos.

LDAP

Lightweight Directory Access Protocol. Protocolo de servicio de directorio que utiliza un subconjunto del estándar X.500 de directorio para proveer una forma común de identificar al usuario y la información de grupo.

MTA

Agente de transferencia de mensajes (MTA, message transfer agent): Es un componente del sistema de transferencia de mensajes electrónicos, se encarga del encaminamiento y almacenamiento de los mensajes de correo hasta su destino final.

OPENLDAP

LDAP es el estándar para los servicios de directorios. OpenLDAP es el mejor servidor libre que implementa esta norma.

PKI

Estructura en la que los clientes o usuarios y servidores disponen de un par de claves asimétricas, guardando la privada preferiblemente en una tarjeta inteligente y distribuyendo la pública en un certificado emitido por un centro certificador.

RA

Autoridad de Registro. Es la encargada de recibir las solicitudes de certificación provenientes de las entidades destinatarias y decidir su validación o deniego. Entidad intermedia entre el usuario y la Autoridad de certificación (CA), que descarga a la CA de las tareas de identificación y validación de los solicitantes del certificado.

S/MIME

Secure Multipurpose Internet Mail Extensions): Especificación de métodos para dotar de seguridad al correo electrónico.

SMTP (Simple Mail Transfer Protocol)

Protocolo Simple de Transferencia de Correo. Protocolo definido en STD 10, RFC 821, que se usa para transferir correo electrónico entre ordenadores. Es un protocolo de servidor a servidor, de tal manera que para acceder a los mensajes es preciso utilizar

SSL

Acrónimo de Secure Socket Layer. Protocolo creado por Netscape para establecer comunicaciones seguras. Una sesión SSL esta securizada gracias al uso de técnicas de criptografía basadas en clave pública.

X509v3

Protocolo para la generación de certificados digitales.

14 ANEXOS

14.1 Guía de Instalación.

14.2 Manual del Usuario