UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

CONCEPCIÓN E IMPLEMENTACIÓN DE UN ESQUEMA DE AUTENTICACIÓN BASADO EN CERTIFICADOS DIGITALES PARA EL IMP

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

CARLOS ROBERTO ZAINOS HERNÁNDEZ

Director de Tesis: M. C. Uriel Tirado Ríos
Codirector de Tesis: M. C. Marco Antonio Vigueras Villaseñor

CIUDAD UNIVERSITARIA

Enero de 2004

DEDICATORIA

A mis padres, Rocío y Roberto:
Por haberme regalado el don de la vida, por todo su apoyo y amor incondicional, por haberme indicado el camino a seguir y por ser un ejemplo vivo de lucha y superación. Doy gracias a Dios por haberme dado unos padres como ustedes.

A ti Papá, porque aunque físicamente ya no estás aquí lo has estado y estarás siempre en mi pensamiento y en mi corazón; se muy bien que te sentirías muy orgulloso de mí, y que donde quiera que me estés viendo recibas la dedicatoria de esto que representa la culminación de una etapa de mi vida; Gracias Papá.

A ti Mamá, porque has sido padre, madre y amiga a la vez en todo momento. No tengo palabras para agradecerte todo lo que has hecho por y para mi, a ti te debo todo lo que tengo y todo lo que soy; Gracias Mamá.

A mis Hermanos Dulce(tuti), Paquita y Bebeto:
Porque aunque son pequeños en edad he aprendido muchas cosas de ustedes, los quiero mucho; son muy importantes para mi. Gracias por querer compartir todo esto conmigo.

A mis tíos y padrinos Aurora y Julián:
Porque han sido y son parte importante de mi formación personal y profesional. Gracias por todo el apoyo recibido durante este tiempo. Sé que sin su valiosa ayuda esto hubiese sido más difícil. Los quiero, admiro y respeto como si fuesen mis padres.

A Zandra, Pepe y Manolo:
Porque más que primos han sido como hermanos para mí. Gracias por su apoyo y por soportarme durante todo este tiempo.

A toda mi familia:
Familia Zainos y Familia Hernández, en especial a mi abuelito Cándido, por todo su apoyo y motivación; por estar conmigo en los momentos felices y en los difíciles también.

A mis amigos:
Monique-nike, Lulú, Mariela, Carlina, Esmeralda, Sofía, Rosalía, Andrés OB, Erwin (des enfants), Gerardo Cruz, Ricardo CD, Yunue; todos aquellos que han compartido vivencias y experiencias inolvidables a lo largo de mi vida escolar y cotidiana (CELE, FI, EST4, IMP).

A Myriam:
Gracias por todo tu apoyo sentimental, anímico y moral; gracias por todo tu cariño, atenciones, enseñanzas y vivencias durante este tiempo juntos. Eres muy importante y muy especial para mi; Gracias por coincidir en forma, tiempo y espacio conmigo, JT'ABC Bibi.

Carlos Roberto
Enero de 2004

AGRADECIMIENTOS

A mi Universidad, la UNAM, y particularmente a la Facultad de Ingeniería por haberme brindado la maravillosa oportunidad y experiencia de ser Universitario, y por la formación académica y humana recibida dentro de ella.

A todos mis profesores que, a lo largo de la carrera, han coadyuvado e influido en mi formación profesional.

Al Instituto Mexicano del Petróleo, por haberme dado la oportunidad de complementar e incrementar mi preparación profesional.

Al personal del Laboratorio de Tecnología Informática del IMP, en especial a Gaby Espinosa, Marce, Karla, Hilario García, Ricardo Nicolás, Porfirio Díaz(don porf), a don Robert, a M.A. Martín, José Luis (Luigi), Heriberto (wasawski), Víctor Monreal; por la convivencia, el apoyo, las enseñanzas y los consejos brindados durante mi estancia en el IMP.

Un agradecimiento especial al Ingeniero Felipe Beltrán Trejo, por haberme recibido y orientado dentro de la Gerencia de Tecnología Informática.

A los químicos Noé Rodríguez y Joaquín Eguía, a Maru, Isabel y Aristeo del LHC; de la Gerencia de Productos Químicos por su amistad y apoyo recibido durante mi estancia dentro el IMP.

Un agradecimiento muy especial a mi profesor y codirector de tesis M.C. Marco Antonio Vigueras Villaseñor; gracias por el tiempo invertido y por el apoyo brindado para la realización y culminación de este trabajo de tesis.

Finalmente y sobre todo muchas gracias a mi director de tesis Dr. Uriel Tirado Ríos por haberme aceptado, por haberme soportado y por haberme brindado el apoyo, el tiempo y las facilidades a su alcance para la propuesta, desarrollo e implementación del presente trabajo de tesis.

INDICE

Índice de figuras y Tablas

Capítulo 1

Capítulo 2

Capítulo 3

Capítulo 4

"La seguridad absoluta tendría un costo infinito"
Anónimo

INTRODUCCIÓN

En la actualidad la seguridad informática debe ser una preocupación de todos aquellos que tengan que ver con la concepción y administración de redes y sistemas de cómputo. Existen diversos aspectos a considerar, que tienen que ver con la protección de los recursos informáticos del IMP de ataques desde el exterior o desde el interior de su propia red, con el manejo confidencial de la información sensible de los sistemas del Instituto y con la autenticación de los participantes en transacciones electrónicas.

El presente trabajo de tesis se concentra en el último punto, concerniente a la autenticación de los participantes en comunicaciones y transacciones electrónicas.

Si el añadir una firma electrónica en un documento e-mail tuviera el mismo valor legal o el mismo reconocimiento como el de una firma en mano en un documento, entonces se abrirían muchas posibilidades, tales como la automatización de trámites institucionales dentro del IMP (órdenes de trabajo, solicitudes de vacaciones, oficios, etc.).

Actualmente en el medio comercial así como en el académico y en el de la investigación, encontramos serios esfuerzos para volver seguras las transacciones electrónicas: académicas, comerciales, bancarias y empresariales entre otras. Sin embargo, aún no existe un consenso mundial que permita manejar un solo protocolo de autenticación, por lo que existen varios estándares y esquemas para ello, algunos de los cuales implican la generación, el manejo y almacenamiento de los llamados certificados digitales.

El trabajo de esta tesis consiste en el estudio de estos protocolos, estándares y esquemas existentes, para la generación, el manejo y el almacenamiento de los llamados certificados digitales implicados en el proceso de la autenticación, y seleccionar el más adecuado para llevar a cabo su implementación dentro del Instituto Mexicano del Petróleo.

Introducción

La Seguridad Informática, y en general la seguridad en los sistemas de cómputo, es una preocupación de todos aquellos que tienen que ver con la concepción, construcción, administración y soporte de redes y sistemas de cómputo.

El Instituto Mexicano del Petróleo mantiene un uso intensivo de Internet como una herramienta para llevar a cabo sus actividades cotidianas, así como para comunicarse con otras comunidades de investigadores, necesita por lo tanto contar con un servicio "seguro" de correo, de acceso a directorios, de acceso remoto (redes privadas virtuales) y transferencia de archivos.

Los principales procesos de negocio del IMP se basan en la venta de investigación, productos y/o servicios principalmente orientados al sector petrolero, que en el trasfondo tienen un gran trabajo de estudio, investigación y desarrollo de conocimiento, en pocas palabras: información. La mayor parte de esta información se maneja y administra en los equipos de cómputo con los que cuenta el IMP, adicionalmente estos equipos se encuentran interconectados por medio de la Intranet IMP, si a esto agregamos que la comunidad del IMP es aproximadamente de 5000 usuarios distribuidos geográficamente en lugares distintos que además tienen acceso a la red Internet, entonces vemos que se tienen serias responsabilidades en la protección de dicha información.

Como ya se ha mencionado, existen diversos aspectos que tienen que ver con la protección de los recursos informáticos del IMP. Dicha protección se centra principalmente en ciertos objetivos primordiales, mas no únicos, los cuales son:

- Ataques desde el exterior (Internet, Extranets) o desde el interior de su propia red (Intranet IMP)
- El manejo confidencial de la información sensible de los sistemas del Instituto y,
- La Autenticación de los participantes en transacciones electrónicas.

Lo que a continuación se menciona representa un intento de proveer una respuesta a cuatro preguntas básicas: ¿Qué es la seguridad? ¿Para qué sirve? ¿Cómo se toma una decisión en seguridad?, y la más importante, ¿Cómo se justifica? Estas preguntas no tienen respuesta fácil. La seguridad está relacionada con todos los aspectos de la vida.

La Seguridad es una necesidad básica. Está interesada en la preservación de la vida y las posesiones, es tan antigua como la vida misma. Los primeros conceptos de seguridad se encuentran ya en el inicio de la escritura. La evidencia escrita más temprana de conceptos relacionados con la seguridad se encuentra en códigos legales, tales como el Sumerio (3.000 AC) o el de Hammurabi (2.000 AC). Más tarde, aparece en obras generalmente referidas al arte de la guerra y gobierno. La Biblia, Homero, Sun Tzu, Cicerón, Virgilio, Cesar, Frontino, Suetonio, son ejemplos relevantes de obras de autores donde se encuentran ciertas evidencias de temas y principios de seguridad.

La meta es ambiciosa. La seguridad como ciencia académica no existe, y es considerada por los "estudiosos" como una herramienta dentro del ámbito en que se la estudia: relaciones nacionales-internacionales, estudios de riesgo, prevención de crímenes y pérdidas, etc. Muchos sostienen que es una teoría tan amplia, compleja y abstracta como la pobreza, la belleza o el amor y ni siquiera arriesgan su definición.

El amplio desarrollo de las nuevas tecnologías informáticas actualmente está ofreciendo un nuevo campo de acción a conductas antisociales y delictivas manifestadas en formas antes imposibles de imaginar, ofreciendo la posibilidad de cometer delitos tradicionales en formas no tradicionales.

Seguridad Informática

El término seguridad, dentro del ámbito de la computación y la informática, tiene más de un significado. Incluso los mismos profesionales de la Seguridad Informática no se ponen de acuerdo respecto al significado exacto de dicho término. Por consiguiente es conveniente emplear definiciones *operativas* de seguridad.[1]

DEFINICIÓN[2]- Podemos entender como **Seguridad** una característica de cualquier sistema (informático o no) que nos indica que está libre de todo peligro, daño o riesgo, y que es, en cierta manera, infalible.

Es necesario abordar un concepto muy importante, e íntimamente relacionado con el de Seguridad Informática, que es el de la Seguridad de la Información.

Seguridad de la Información[3]

La información es un activo (un valor o un bien) que, como otros activos importantes del negocio, tiene valor para una organización; por lo tanto necesita ser protegida de manera conveniente.
La *Seguridad de la Información* protege a la información de una amplia gama de amenazas con el objetivo de asegurar la continuidad del negocio, reduce al mínimo el daño en el mismo y maximiza el retorno de inversión y las oportunidades de negocio.

La Seguridad de la Información se caracteriza, o consiste, básicamente en la preservación de:

- **Confidencialidad** - Aseguramiento de que la información es accesible solo por aquellos elementos autorizados para hacerlo.
- **Integridad**- Salvaguarda la exactitud y totalidad de la información y de los métodos de procesamiento.
- **Disponibilidad**- Aseguramiento que solo los usuarios autorizados tengan acceso a la información y a los valores o bienes asociados a ella cuando lo requieran.

La Seguridad de la Información es alcanzada implementando un conjunto de controles convenientes, los cuales pueden ser: políticas, prácticas, procedimientos, estructuras organizacionales y funciones de software. Estos controles necesitan ser establecidos para asegurarse de que los objetivos específicos de la seguridad de la organización sean resueltos.

La seguridad de la Información incluye aspectos específicos muy importantes, los cuales mencionaremos y utilizaremos más adelante.

Por lo anteriormente expuesto, podemos concluir que la Seguridad Informática significa planear, organizar, coordinar, dirigir y controlar las actividades orientadas a mantener y garantizar la integridad física y lógica de los recursos implicados en materia informática y computacional.

La Seguridad es igualmente una cultura de los usuarios de la infraestructura informática.
De manera simple, Seguridad informática es un conjunto de soluciones técnicas a problemas no técnicos.

Áreas que cubre la Seguridad Informática

A grandes rasgos podemos mencionar las siguientes:

---

[1] *Seguridad Práctica en UNIX e Internet;* O'Reilly Cap. 1 p5, Mc Graw Hill
[2] Seguridad en Unix y Redes. Libro electrónico v2.1 Antonio Villalón Huerta; julio 2002
[3] INTERNATIONAL STANDARD ISO/IEC 17799:2000; Introduction pVIII

- Políticas de Seguridad
- Seguridad Física
- Autenticación
- Integridad
- Confidencialidad
- Disponibilidad
- No repudio
- Control de Acceso
- Auditoria

Objetivos de la Seguridad Informática

Los objetivos de la Seguridad Informática implican la protección de redes de datos frente a muchos tipos de amenazas. Las corporaciones se enfrentan a un desafío puesto que el uso de la tecnología para sacar el máximo partido conlleva a proporcionar acceso a sus redes corporativas y a los recursos que estas redes contienen, tanto a los usuarios internos como externos de la corporación. Por tanto, necesitan controlar los tipos de acceso que se otorgan a diferentes usuarios para recursos distintos. Los aspectos a cubrir son muchos y muy variados.

En primer lugar, las empresas necesitan asegurarse de que determinados recursos del sistema, desde el equipo individual que contiene datos y programas hasta toda la red, estén disponibles para los usuarios autorizados a utilizar dichos recursos, pero que no estén disponibles para nadie más. La tecnología ofrece una forma de proporcionar esta protección en las *capacidades de administración de acceso* de los sistemas operativos. Los sistemas operativos utilizan mecanismos de seguridad para administrar de forma activa el acceso a archivos y a otros recursos. Los mecanismos incluyen perfiles de usuario individuales que identifican a los usuarios y a sus privilegios de acceso, igualmente mecanismos que diferencian entre los diferentes entornos de recursos. Estos mecanismos protegen contra amenazas como intrusos que intentan obtener acceso a los datos con el fin de manipularlos o robarlos.

Las empresas también necesitan poner los recursos especificados a disposición de los usuarios como pueden ser los clientes, que están fuera de su propia red interna. Idealmente, estos usuarios deben tener acceso a los recursos en cualquier momento bajo condiciones de rendimiento óptimo: el acceso confiable es una parte crítica del valor que se proporciona. Los servicios bancarios y de compra en línea son ejemplos de servicios que proporcionan las empresas, mediante el uso de sus redes, que implican que los usuarios externos tengan acceso e interactúen con datos corporativos. Sin embargo, la disponibilidad de recursos puede ser el objetivo de diversos ataques. El ataque de Negación de Servicio (DOS) es, por ejemplo, un tipo de ataque bien conocido. Su objetivo es hacer difícil o imposible el acceso a los datos, lo que supone una pérdida de negocios para la empresa.

Además de proteger los datos contra personas que atacan los sistemas con la intención de causar daño, los mecanismos de seguridad también protegen a organizaciones y ayudan a los usuarios autorizados, impidiendo que causen daño a los recursos informáticos de forma accidental o inocente. Esto se consigue al utilizar dispositivos tecnológicos junto con procedimientos de seguridad y directivas de seguridad.

Niveles y costos de la Seguridad Informática

Alcanzar la seguridad absoluta es un mito en el entorno informático, como lo es en la vida cotidiana. Entonces, ¿cuánta seguridad es necesaria? El grado de seguridad que se considera adecuado para una empresa es una función del valor de la información que debe protegerse, las amenazas a las que está sujeta esta información y la exposición a riesgos que la compañía está dispuesta a aceptar.

La siguiente pregunta es: ¿Cuáles son los costos?. Estos incluirán costos de hardware, software, red, mantenimiento, administración y educación adicionales, así como costos de procesamiento menos obvios. Todo servicio basado en la seguridad agrega un cierto nivel de costo de procesamiento al sistema. La consecuencia de este costo de procesamiento puede verse en una reducción del tiempo de acceso, incremento en el tiempo de respuesta u otros síntomas no deseables.

Por tanto, además de entender todos los mecanismos de seguridad disponibles, es importante que un administrador, junto con el dueño de la información, determine el nivel de seguridad necesario para determinados escenarios con el fin de evitar costos innecesarios. El personal encargado de la toma de decisiones debe determinar si merece la pena gastar fondos adicionales en funciones de seguridad para proteger cada recurso. La decisión sobre las medidas de seguridad que deben implementarse implica encontrar el equilibrio óptimo entre rendimiento (productividad), protección de datos y costos implicados.

Si bien es cierto, vale la pena recordarlo y tenerlo muy en cuenta, que en el ámbito de la Seguridad informática es muy difícil justificar o esperar un "retorno de inversión, ROI[4]", sin embargo pudiera compararse con la contratación de un seguro de vida o seguro médico de gastos mayores, los cuales se contratan muchas veces sin esperar un retorno de inversión, esto quiere decir que uno no esperaría sufrir un accidente, padecer una enfermedad grave o perder la vida con el fin de justificar el gasto hecho al adquirirlo.

OBJETIVO DEL TRABAJO

El área de Seguridad Informática del IMP, con la finalidad de reforzar los niveles de seguridad en la red del mismo, se ha propuesto mejorar y robustecer el esquema de autenticación con el que actualmente cuenta (basado en nombre de usuario y password). Dicho esquema se encarga de validar a los participantes en las transacciones electrónicas dentro de la red IMP.
El objetivo principal de esta tesis consiste en concebir e implementar un esquema de autenticación alternativo o adicional basado en certificados digitales. El método a seguir se basará en el estudio de los protocolos, estándares, modelos, esquemas y herramientas existentes para la generación, el manejo y el almacenamiento de los certificados digitales implicados en el proceso de la autenticación.

Un punto importante es especificar y fijar los niveles de seguridad que se quieren alcanzar y los costos que se tiene contemplados para este fin.

Los recursos informáticos de IMP son muy valiosos y requieren tener especial cuidado en ellos. No podemos arriesgar la seguridad de los mismos ni dejarlos vulnerables a ataques, intrusiones o mal uso de los mismos.

[4] ROI (*Return on Investment*) - Retorno sobre la Inversión - Una medida fundamental de evaluación de operaciones de una empresa. Para poder calcularla, es indispensable que la empresa considerada haya generado utilidades en el periodo de que se trate. Cuando una empresa, durante cierto periodo, no ha sido capaz de generar utilidades, se dice de ella que no es rentable. Si el objetivo fundamental de las empresas es generar utilidades, y para lograr ese objetivo es indispensable comprometer recursos monetarios (inversión), entre otros factores, es natural y lógico que se comparen las utilidades del periodo contra la inversión propia necesaria para alcanzar dicha utilidad.
http://dsrefa01.bital.com.mx/aptrix/glosario.nsf/0/fe98079eec4c89f106256a2400821624?OpenDocument

REFERENCIAS

The Handbook of Applied Cryptography
Capítulo 1
Information Security and Cryptography,
A. Menezes, P. Van Oorschot, and S. Vanstone, CRC Press, 1996, en formato PDF

V Taller en Tecnología de Redes e Internet para América Latina y el Caribe
Taller 1
Tecnología de redes, Seguridad Informática
http://www.walc2002.pucmm.edu.do/material/taller1/Seguridad%20Inform%E1tica.ppt

Seguridad Práctica en UNIX e Internet
O'Reilly
2ª Edición
Mc Graw Hill

International Standard ISO/IEC 17799:2000
Information Technology – Code of practice for information security management
First Edition 12/2000

Seguridad en Unix y Redes, Libro Electrónico v2.1
Antonio Villalón Huerta, Universidad Politécnica de Madrid
Julio 2002
http://www.criptored.upm.es/guiateoria/gt_m209a.htm

Capítulo 1

Estudio Bibliográfico

Introducción

Como lo mencionamos en la sección anterior, el área que abarca la Seguridad Informática es muy extensa. El presente trabajo de tesis se centra en 4 puntos principales, mas no únicos, que consideramos son los más importantes para nosotros basándonos en los objetivos que perseguimos, los cuales trataremos de cubrir y estudiar lo más ampliamente posible en el presente capítulo.
Dentro de estos puntos se encuentra el de la Autenticación, el cual es el objetivo principal de este estudio y que da nombre al presente trabajo de tesis. Los 4 puntos a los que hacemos referencia son:

- Autenticación
- Integridad
- Confidencialidad
- No repudio

Además veremos la manera en que dichos puntos son abordados y cubiertos mediante el uso de las llamadas herramientas criptográficas.

1.1 SITUACIÓN ACTUAL

La Autenticación es el proceso de determinar cuando algo o alguien es, en efecto, quien dice ser o lo que dice ser. En redes privadas y públicas (incluyendo Internet), el esquema de autenticación más utilizado se realiza a través del uso de un ***Username*** y de un ***Password***. Con base al conocimiento del password se asume que el usuario es auténtico. Este mecanismo es el que actualmente se maneja en el IMP.

La debilidad de este mecanismo para transacciones electrónicas (como en el caso de intercambio de dinero o de información confidencial) es que los passwords pueden ser robados, accidentalmente revelados u olvidados. Por esta razón, los negocios sobre Internet y muchas otras transacciones electrónicas requieren de procesos de autenticación más robustos, este tipo de procesos se pretenden implementar en las transacciones e intercambios de información que actualmente se llevan a cabo en el IMP.

El uso de certificados digitales proporcionados y verificados por una Autoridad Certificadora (AC) como parte de una infraestructura de llaves se considera llegará a ser la forma estándar de realizar autenticaciones por Internet.

Este tipo de tecnología o esquema de Autenticación comprende el hardware y el software usado por los servicios de Autenticación y aplicaciones clientes que hacen uso de estos servicios para mejorar las facilidades ofrecidas por:

- Correo electrónico,
- Sitios Web,
- Accesos remotos,
- Seguridad de documentos,
- Conferencias Multimedia,
- Directorios, y
- En general servicios de red y comercio electrónico.

Actualmente este esquema se sigue en transacciones seguras sobre Internet, tanto de comercio electrónico como de transferencia de información, por lo que es indispensable contar con facilidades para garantizar la seguridad de los datos, podemos mencionar las siguientes como ejemplo:

- Firmas digitales para garantizar la autenticidad e integridad de los datos
- Encriptado para soportar la confidencialidad de los datos (privacidad), y

- No-repudiación para soportar facturación.

A fin de comprender mejor de lo que estamos hablando y para tener un panorama mas claro del problema al que nos enfrentamos consideremos la siguiente situación:

Primero consideremos que existe la necesidad de comunicación entre dos entes, por ejemplo dos personas que están en dos países diferentes, por lo tanto una de las mejores formas de comunicación es por Internet. Uno de los problemas más sentidos es ¿cómo saber que efectivamente la persona con quien me estoy comunicando es precisamente la que dice ser?. Este problema lo llamaremos el problema de verificación de Identidad o de la AUTENTICACIÓN.

En la práctica este problema se ha resuelto de la siguiente manera. Por ejemplo, si dos personas se ven en la calle y éstas ya se conocen anteriormente, simplemente se saludan con sus nombres, esto es, cada una de ellas verifica la identidad de la otra visualmente y aceptan que es la persona, que ya conocen. La práctica dice que es muy improbable que haya equivocación, salvo casos muy raros, cómo que haya dos personas muy parecidas, que tenga la misma apariencia, en fin, casos que en general no ocurren. Ahora si las dos personas se conocen pero no se ven, por ejemplo, una está del lado de una puerta y la otra persona del otro lado, ¿cómo pueden reconocerse? Podría una de ellas preguntar por el nombre de otra y reconocer su voz, quizá preguntarle por algo de su familia que sólo ellos conocen, una vez que estos quedan satisfechos por las respuestas aceptan que la identidad de la otra persona es quien dice ser.

Un caso más complicado es cuando dos personas tienen que validar su identidad pero no se conocen anteriormente. Por ejemplo si una persona va a recoger un boleto de avión a una agencia de viajes y tiene que acreditar su identidad, digamos que el boleto esta a nombre de Ricardo Nicolás, el empleado de la agencia de viajes acepta la identidad de quien dice ser Ricardo Nicolás si cumple ciertos requisitos, por ejemplo si le muestra una identificación oficialmente válida (el pasaporte para casi todo el mundo, licencia de manejo, credencial de elector etc.), el empleado compara la foto de la identificación con la apariencia del portador y acepta que quien es portador es realmente Ricardo Nicolás si la foto es parecida al mismo.

Los anteriores ejemplos son muy frecuentes en la vida diaria, de esa manera se pueden aceptar o rechazar la identidad de una persona. Este proceso es también requerido en otros casos, por ejemplo, en una escuela al hacer un examen, en el aeropuerto al documentar el viaje, al solicitar un tramite oficial, al cobrar un cheque, al firmar un contrato, en fin.

Existe otra circunstancia donde es necesario verificar la identidad, en ésta, casi en general es necesario quedarse con un comprobante de tal verificación, es el caso, sí quiero cobrar un cheque, el cajero del banco generalmente pide la identificación oficial y posteriormente una vez comprobada la identidad visual, pide que este cheque este "firmado". Este proceso entendido por todos ha sido una forma tradicional de tener un comprobante legalmente aceptado para comprobar a posteriori que yo efectúe esa transacción.

La firma tradicional, firma calígrafa o cualquier otro nombre con el que se le conozca, tiene varias características; la principal de ellas es que es aceptada legalmente, esto quiere decir que si alguna persona firmó un documento adquiere tanto los derechos como las obligaciones que de él deriven, y si estas obligaciones no son acatadas, el portador del documento tiene el derecho de reclamación mediante un litigio. La autoridad competente acepta las responsabilidades adquiridas con sólo calificar a la firma como válida.

Como en los problemas que tratamos de resolver, análogos al último escenario, es necesario verificar la identidad de los participantes; estudiaremos y utilizaremos a la "firma" como elemento que sirve para demostrar la identidad.

Podemos resumir que existen dos procedimientos importantes, el primero el proceso de firma, que es el acto cuando una persona "firma" manualmente un documento. Y el proceso de verificación de la firma, que es el acto que determina si una firma es válida o no. Por otro lado es importante hacer notar que la firma comprueba la identidad de una persona, de tal modo que así se sabe quién es la persona quien firmó, y ésta persona no puede negar las responsabilidades que adquiere en un documento firmado.

Podemos realizar las siguientes definiciones[1] de los procesos implicados:

Proceso de firma: este proceso es muy simple y consiste sólo en tomar un bolígrafo y estampar, dibujar o escribir garabatos en un papel. En general este garabato debe ser el mismo y es elegido a gusto de la persona. Se usa como una marca personal. Es importante mencionar que por una lado lo que identifica a la persona que firma (quien hace el garabato) es la forma misma de la firma, pero también características de escritura, como la velocidad de escritura, la presión que se aplica al bolígrafo, la inclinación de la escritura, etc.

Proceso de verificación: existen en general dos métodos de verificación de la firma, uno es el más usado y simple, que es el visual, esté método lo aplica cualquier cajero al pagar un cheque, o al efectuar un pago con tarjeta de crédito. En muchos casos la firma es rechazada por no pasar este método, sin embargo legalmente no es suficiente el método visual. El método legalmente definitivo es el peritaje de la firma en laboratorio, que consiste en verificar a la firma independientemente de la forma, tomando en cuenta otras características como la presión de escritura, la velocidad de escritura, la inclinación de escritura, las características particulares de alguna letra etc. El conjunto de estas propiedades varía dependiendo de cada país y de sus leyes. Recalcamos que el resultado es tomado como definitivo, legalmente y en teoría.

Hay que hacer notar que con la firma queda resuelto legalmente el problema de la autenticidad o el de comprobar la identidad de una persona, y de la misma manera el problema que podría aparecer si una persona rechaza ser el autor de una firma es también resuelto con los métodos anteriores, al menos legalmente y en teoría.

Es importante hacer notar también que la firma frecuentemente se encuentra asentada en un documento de identidad oficialmente válido, como el pasaporte, la credencial de elector, la licencia de conducir, y otros.

CRIPTOGRAFÍA

Para poder utilizar lo anteriormente expuesto, como un conjunto de herramientas para resolver nuestros problemas en el campo de la informática, es necesario hacer mención y saber lo básico de un tema muy importante que sirve de base para aplicar una solución adecuada, que es el concepto de **Criptografía**.

DEFINICIÓN[2]: La Criptografía es el estudio de técnicas matemáticas relacionadas con aspectos de la seguridad de la información: como confidencialidad, integridad de los datos, autenticación de la entidad y autenticación del origen de los datos (no repudio).

La Criptografía no es el único mecanismo para proporcionar seguridad de la información, pero si provee ciertas técnicas para ello.
La Criptografía como ciencia, estudia los problemas básicos de la seguridad en la transmisión de la información por un canal inseguro.

[1] *Firma Digital y Certificados Digitales*, José de Jesús Ángel; SeguriData, Documento electrónico.
[2] *Handbook of Applied Criptography, en formato PDF*; A. Menezes, P. Van Oorschot, Capítulo 1,CRC Press, Canadá 1997

OBJETIVOS DE LA CRIPTOGRAFÍA

De todos los objetivos de la seguridad de la información, los siguientes cuatro forman un marco (framework) sobre el cual se pueden derivar los demás:

- Privacidad o Confidencialidad
- Integridad de datos
- Autenticación
- No repudio

A continuación damos su definición[2]:

1- Confidencialidad- Es un servicio utilizado para resguardar el contenido de la información de todos excepto de aquellos autorizados para ello (accesar a la información). Secreta es un término sinónimo de confidencialidad y de privacidad. Hay numerosas maneras de proveer confidencialidad, extendiéndose de la protección física a los algoritmos matemáticos que hacen a la información no comprensible a simple vista.

2- Integridad de los datos- Es un servicio el cual se orienta a prevenir la alteración no autorizada o accidental de los datos. Para asegurar la integridad de datos, uno debe de tener la habilidad de detectar la manipulación de datos por partes o entes no autorizadas. La manipulación de datos incluye cosas tales como la inserción, borrado y la substitución.

3- Autenticación- La autenticación es un servicio relacionado con la identificación. Esta función se aplica a las entidades así como a la información. Dos entidades que entran en comunicación deben identificarse. La información enviada por un canal se debe autenticar en cuanto a origen(fuente), fecha de origen, contenido de la información, tiempo de envío, etc. Por estas razones, este aspecto de la criptografía se subdivide generalmente en dos clases importantes: Autenticación de la entidad y Autenticación del origen de los datos. La autenticación del origen de los datos proporciona de manera implícita la integridad de los mismos (si es que se modifica el mensaje quiere decir que la fuente ha cambiado).

4- No Repudio- Es un servicio el cual previene y evita que una entidad niegue comisiones o acciones realizadas anteriormente. Cuando se presentan conflictos debido a una entidad que niega que ciertas acciones fueron tomadas o hechas, son necesarios entonces los medios para resolver esta situación. Por ejemplo, una entidad puede autorizar la compra de ciertos artículos por otra entidad y luego negar que dicha autorización fue efectuada. Un procedimiento que implique la confianza en terceros es necesario para resolver este conflicto.

Una meta fundamental de la criptografía es tratar adecuadamente estas cuatro áreas en teoría y práctica. La criptografía trata de abordar la prevención y la detección del engaño y de otras actividades malévolas.

ALGORITMOS CRIPTOGRÁFICOS

Un algoritmo es el conjunto de pasos necesarios para resolver un problema, y en particular un problema matemático. En el campo de la ciencia de las computadoras, los algoritmos se implementan como partes de un programa que se conoce como una rutina o una biblioteca. Usualmente el programa principal realiza la operación matemática sobre varios conjuntos de datos llamando repetidas veces a las bibliotecas de algoritmos. Algunos, particularmente complejos, se pueden implementar en hardware especializado: un ejemplo serían los chips de aceleración de video tridimensional que se encuentran incorporados en las tarjetas de video de las Pc's actuales.

Los *algoritmos criptográficos*[3] son algoritmos matemáticos y están diseñados de manera que se puedan llamar con diferentes conjuntos de datos para entrar en funcionamiento. Por ejemplo, un algoritmo de cifrado se puede llamar con los datos de la tarjeta de crédito para codificarlo una vez y para codificar una receta en otro momento.

Una expresión conocida ahora es la de Proveedor de Servicios Criptográficos, CSP (*Criptographyc Service Provider).* En esencia, un CSP es una biblioteca de algoritmos criptográficos (algoritmos de cifrado, algoritmos de firma, y otros), los cuales se pueden llamar a través de una interfaz claramente definida para realizar una función criptográfica en particular. Los algoritmos criptográficos son complejos, y en algunos casos, se benefician de un acelerador de hardware para agilizar algunas de las operaciones matemáticas.

CRIPTOLOGÍA Y CRIPTOANÁLISIS

La criptografía es un área fascinante y en la actualidad desde su objeto de estudio la podemos dividir en dos disciplinas: *Criptología y Criptoanálisis.*

La Criptología se dedica al estudio e invención de nuevos algoritmos criptográficos. Después de años de trabajo, los criptólogos (matemáticos e investigadores dedicados al estudio de la criptología) lanzan sus inventos para que la comunidad criptográfica los revise. Es en esta parte donde aparece el Criptoanálisis.
El Criptoanálisis se dedica al estudio y análisis de las debilidades de los algoritmos criptográficos, forzándolos y atacándolos en funcionamiento y diseño con el objetivo de descifrarlos.

El trabajo de un criptoanalista ve sus frutos cuando logran descifrar dichos algoritmos.
Cuando esto sucede, los criptólogos aprenden algo nuevo sobre cómo hacer mejores algoritmos y regresan a sus laboratorios con esta información para crear algoritmos más seguros.

La criptografía es la única vertiente de la ciencia de la computación de la que podemos pensar que tiene dos ramas paralelas, opuestas y simbióticas.

CLASIFICACIÓN DE LA CRIPTOGRAFÍA

Las Técnicas Criptográficas se encuentran clasificadas, típicamente, en dos tipos genéricos [4] :

- Criptografía simétrica , o de clave privada y
- Criptografía asimétrica, o de clave pública.

La **Criptografía Simétrica** resuelve el problema de la confidencialidad, y usa algoritmos como DES[5] TDES[6] y AES[7] para transmitir información cifrada, ya que solo con una única clave simétrica puede leer el contenido de la información. Esta clave la llamaremos "clave o llave simétrica" y tiene en general una longitud de 128 bits. El problema aquí es que antes de realizar la conexión segura es necesario que ambos lados tengan la misma clave simétrica (ver figura 1.1).

[3] PKI, Infraestructura de claves públicas; Andrw nAsh.William Duane RSA Press
[4] *Handbook of Applied Criptography, en formato PDF*; A. Menezes, P. Van Oorschot, Capítulo 1,CRC Press, Canadá 1997
[5] DES- Data Ecryption Standard – Algoritmo más común de cifrado simétrico
[6] Triple DES – Técnica usada para hacer más fuerte el cifrado DES; un mensaje de datos se cifra tres veces usando múltiples claves DES.
[7] Asimetric Encription System – Algoritmo de cifrado asimétrico

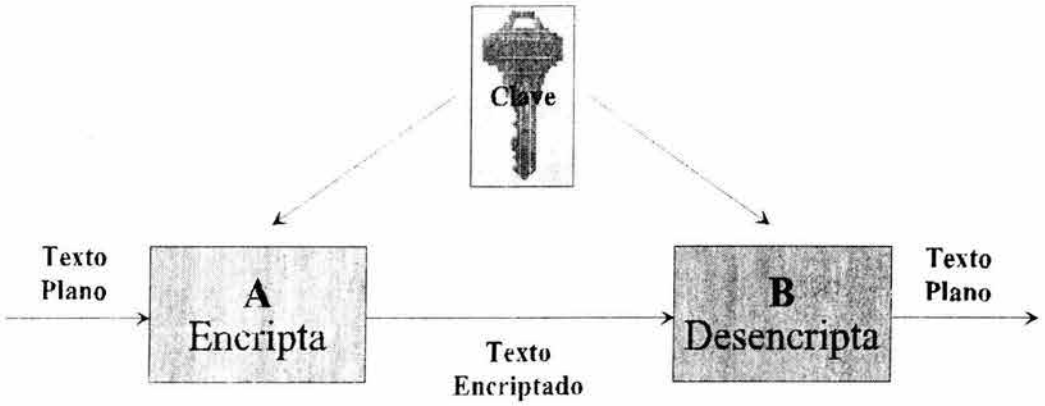


Figura 1.1
Esquema de Criptografía Simétrica

Antes de continuar, conviene tener en cuenta los siguientes puntos importantes en los cuales se resumen las características de la criptografía simétrica:

- Con la criptografía simétrica, se utiliza la misma clave para cifrar y descifrar.
- El cifrado simétrico es rápido
- El cifrado simétrico es seguro
- El texto cifrado que resulta de un cifrado simétrico es compacto.
- Dado que la clave simétrica debe llegar al receptor, el cifrado simétrico está sujeto a la interceptación
- El número de claves en la criptografía simétrica para **n** usuarios es: **n(n-1)/2** llaves, por tanto se convierte en una pesadilla en cuanto al manejo, intercambio y distribución de llaves para poblaciones muy grandes.
- La criptografía simétrica requiere de una administración compleja de claves
- La criptografía simétrica no se ajusta a las firmas digitales o a la aceptación

La **Criptografía Asimétrica** consiste en algoritmos basados en problemas de un solo sentido, es decir, que por un lado sea muy fácil realizarlo, pero a la inversa sea "difícil" de realizarlo, como es el problema de la factorización entera, es fácil realizar el producto de dos números pero es "difícil" factorizar un número producto de dos números primos grandes. La Criptografía de "clave o llave pública" o "asimétrica", fue introducida en 1976 por Whitfield Diffie y Martin Hellman de la Universidad de Stanford. Desde entonces la tecnología ha seguido un interesante desarrollo y puede ser hoy día considerada madura.

En contraste con el sistema simétrico, el sistema de clave pública ó asimétrico utiliza un par complementario de claves para separar las funciones de encripción ó encriptación y desencripción. Una clave, la privada (private key), es mantenida en secreto tal como la clave del sistema simétrico, mientras que la otra, la clave pública (public key), no necesita ser mantenida en secreto. El sistema debe tener la propiedad de que, conocida la clave pública no sea factible determinar la clave privada.

Este enfoque de dos claves puede simplificar la administración de las mismas, dado que minimiza el número de claves que necesitan ser manejadas y almacenadas en la red, que permitirán distribuir las claves a través de sistemas sin protección tales como servicios de directorio público.

Potencialmente, existen dos modos de usar los sistemas de criptografía por clave pública, dependiendo de una forma u otra de como sea utilizada esta, como clave de encripción o como clave de autenticación.

Supongamos que exista un directorio público que contenga las claves públicas para un conjunto de pares de comunicación. Usando las claves públicas como claves de encripción, cualquier parte puede enviar un mensaje a cualquier otra parte. Para ello el remitente simplemente utiliza la clave del receptor para encriptar el mensaje. Sólo el poseedor de la correspondiente clave privada podrá leer el mensaje. Este es el modo de encripción o encriptación(ver figura 1.2).

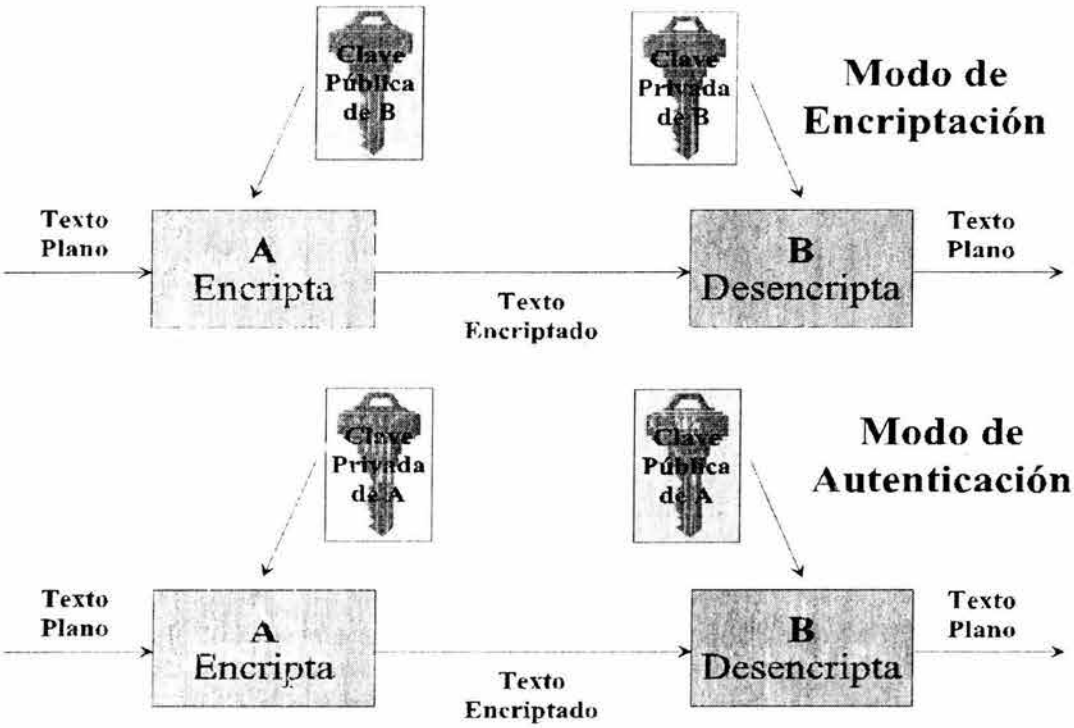


Figura 1.2
Esquema de Criptografía Asimétrica

Usando la clave privada como clave de encripción, el sistema criptográfico de clave pública puede ser utilizado como autenticación del origen de los datos y por consiguiente de la integridad del mensaje. En este caso cualquier persona puede obtener la clave de desencripción del directorio (clave pública) y de esa forma leer el mensaje. El lector también conoce que sólo el poseedor de la correspondiente clave privada pudo haber creado ese mensaje. Este es el modo de autenticación.

Un sistema de clave pública que pueda operar en ambos modos es llamado sistema criptográfico de clave pública reversible. Aquellos que sólo pueden operar en el modo de autenticación se denominan sistemas criptográficos de clave pública irreversible.

Los sistemas de clave pública presentan un desafío mucho mayor en el diseño del algoritmo que los sistemas simétricos, dado que la clave pública representa información adicional que puede ser utilizada para atacar al algoritmo.

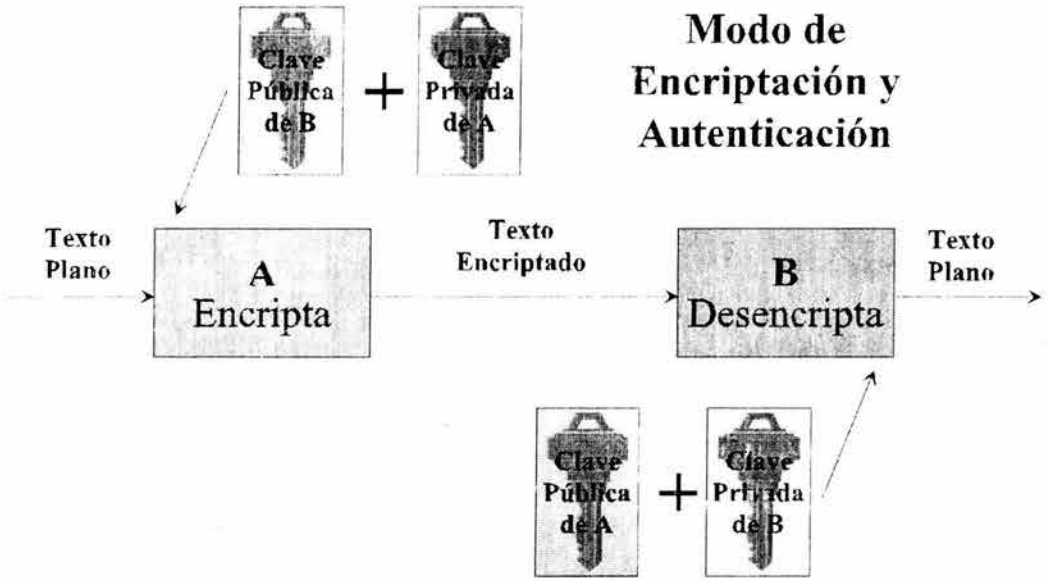


Figura 1.3
Modo de encriptación y autenticación

Existe un método de encriptar y autenticar los mensajes utilizando consecutivamente las clave pública de B - receptor - y la clave privada de A - remitente - para encriptar el mensaje, y la clave pública de A - remitente - y la clave privada de B - receptor - para desencriptarlo (ver figura 1.3). De esta forma, el remitente se asegura que sólo el receptor podrá leer el mensaje, dado que es el único que posee la clave privada de B, y el receptor se asegura que el remitente es el único que pudo haberlo enviado, ya que éste es el único que posee la clave privada de A.

En este caso tenemos dos claves en cada caso que se le asocian a una entidad, un usuario por ejemplo. Una clave pública que sirve para cifrar información y solo quien tiene la clave privada asociada a esta clave pública puede descifrar el mensaje. Esto es usado para intercambiar claves simétricas. Por otra parte con la clave privada se firman documentos y se verifica la firma con la clave pública.

Es claro que la clave pública puede ser conocida por cualquier persona, sin embargo la clave privada es solo conocida por el dueño a quien se le asociaron el par de claves. La clave privada debe de guardarse de manera confidencial, ya sea en su computadora personal, en su PDA[8], en un Smart Card (tarjeta inteligente) o algún dispositivo personal.

Podemos resumir los puntos más importantes de la Criptografía Asimétrica en lo siguiente:

- Con la criptografía asimétrica lo que está cifrado con una clave (pública o privada) sólo se puede descifrar con la otra (pública o privada).
- El cifrado asimétrico es seguro.
- Dado que no se necesita enviar una clave al receptor, la codificación asimétrica no sufre por la intercepción de llaves.
- El número de claves que se necesitan distribuir es el mismo que el número de participantes de ahí que la criptografía asimétrica funciona bien en escalas de poblaciones muy grandes.
- La criptografía asimétrica no tiene los problemas complejos de distribución de claves.

[8] PDA - Personal Digital Assistant, o lo que es lo mismo una PALM ó agenda electrónica

- La criptografía asimétrica no exige una relación previa entre las partes para realizar el intercambio de claves.
- La criptografía asimétrica soporta firmas digitales y aceptación.
- El cifrado asimétrico es relativamente lento.
- El cifrado asimétrico expande el texto cifrado.

En la práctica la criptografía simétrica y asimétrica se usan conjuntamente. La simétrica para intercambiar grandes volúmenes de información por su rapidez. Y la asimétrica para el intercambio de las claves simétricas y la firma digital.

FUNCIONES HASH (Funciones de Resumen o Digestión)

Las funciones hash son algoritmos muy comunes en la computación y quizá son los que tienen un uso más común. A pesar de su amplio uso, la mayoría de las personas no sabe cuál es el sentido de un hash o verificación.

Una *función Hash* es una función computacional eficiente de mapeo de cadenas binarias de longitud arbitraria a cadenas binarias de alguna longitud fija, llamados valores hash[9].

Un algoritmo hash toma un gran bloque de datos y lo comprime en una *huella digital* (fingerprint) o *reseña* (digest) de los datos originales, esto es, con un hash se toma un gran bloque de datos y se calcula una ecuación a través de ellos. El resultado del hash es un valor más pequeño que los datos originales. Una de las propiedades más importantes de estas funciones hash es que si se cambia un solo bit de los datos originales, el valor del hash del resultado será diferente.

Un ejemplo ilustrativo de lo que es un hash es el valor de *verificación de redundancia cíclica* (*Cyclic Redundancy Check, CRC*), que se pone al final de la mayoría de los mensajes de comunicación (ver figura 1.4).

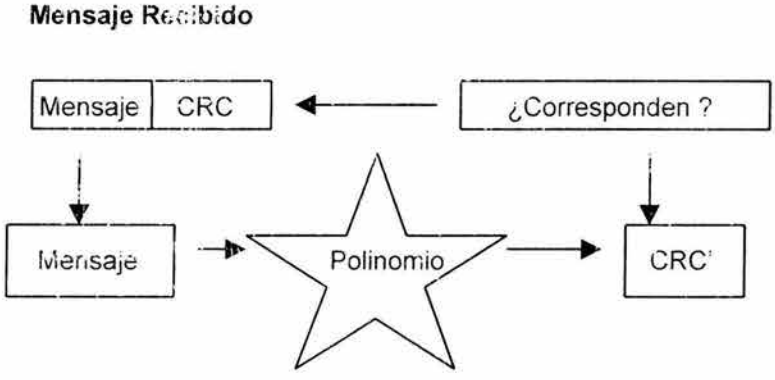


Figura 1.4
Verificaciones de la Redundancia Cíclica

Cuando los mensajes se van a enviar a través de una línea de comunicaciones, es común ejecutar una ecuación polinómica a través de los bytes del mensaje. Este polinomio da un resultado, el código CRC, el cual se une al final del mensaje antes de enviarlo. Cuando el sistema receptor capta el mensaje con el CRC unido a él, en esencia ejecuta el mismo polinomio a través del mensaje (excluyendo el CRC original) y produce una segunda copia del CRC (CRC'). Luego, se comparan

[9] *Handbook of Applied Criptography, en formato PDF*; A. Menezes, P. Van Oorschot, Capitulo 1,CRC Press, Canadá 1997

ambos CRC's, el original CRC con el nuevo CRC', si ambos corresponden, hay un alto grado de confianza de que el mensaje no fue modificado durante su viaje a través de la red.

En un sentido, el CRC actúa como una huella digital o reseña del mensaje, el cual se puede verificar en el receptor del mismo.

Todos los algoritmos hash que se utilizan en criptografía están diseñados con algunas propiedades especiales:

- No se puede poner a funcionar el hash hacia atrás y recuperar algo del texto claro inicial.
- El resumen o reseña resultante no dirá nada sobre el texto claro inicial.
- Desde el punto de vista computacional, no es factible crear/descubrir texto claro que verifique un valor específico. Esto evita que un pirata informático trate de sustituir un documento sin que se presenten fallas en la correspondencia del resumen del mismo.

Existen varios algoritmos hash criptográficos. MD2 (*Message Digest v2*) es un hash de RSA que produce un resumen de 128 bits que se optimiza para procesadores de baja tecnología de 8 bits. El MD5 (*Message Digest v5*) también produce un resumen de 128 bits, pero está optimizado para procesadores de 32 bits. El hash SHA-1 (*Secure Hash-1*)también está optimizado para procesadores de alta tecnología y produce resúmenes de 160 bits.

Con todo lo anterior, podemos definir los conceptos de firma digital y certificado digital[10].

FIRMA DIGITAL

Firma Digital: es un número natural, de mas o menos 300 dígitos si se usa el sistema **RSA**[11], que tiene las mismas propiedades que la firma convencional. Es decir es posible asociar un número único a cada persona o entidad. existe un método de firma y un método de verificación de la firma. El concepto de firma digital fue introducido por Diffie y Hellman en 1976. Básicamente una firma digital es un conjunto de datos asociados a un mensaje que permite asegurar la identidad del firmante y la integridad del mensaje. Esta firma digital resuelve satisfactoriamente el problema de autenticación y no rechazo.

Proceso de Firma Digital

En primer lugar, la creación de una firma digital requiere que se cree un hash criptográfico para el contenido importante de los documentos o para todo el contenido de el o los documentos que se desean firmar, con el fin de garantizar que esa información no se modifique (ver figura 1.5)

Por ejemplo, para un contrato el contenido importante incluirá todo el texto del documento; para un formulario web, el hash incluirá por lo menos los datos de los campos de entrada, con información sobre la descripción del producto, información sobre el usuario y forma de pago (si hablamos de comercio electrónico) y así sucesivamente, mientras que es posible que las etiquetas HTML (*Hiper Text Markup Language*) que se usan para formateado no lo necesiten. A continuación el valor hash (reseña, resumen o digestión) resultante se cifra utilizando la clave privada de quien firma.

[10] *Firma Digital y Certificados Digitales*, José de Jesús Ángel: SeguriData Documento electrónico.
[11] RSA – Rivest, Shamir, Adleman; quienes propusieron dicho protocolo para brindar servicios de seguridad basados en criptografía asimétrica.

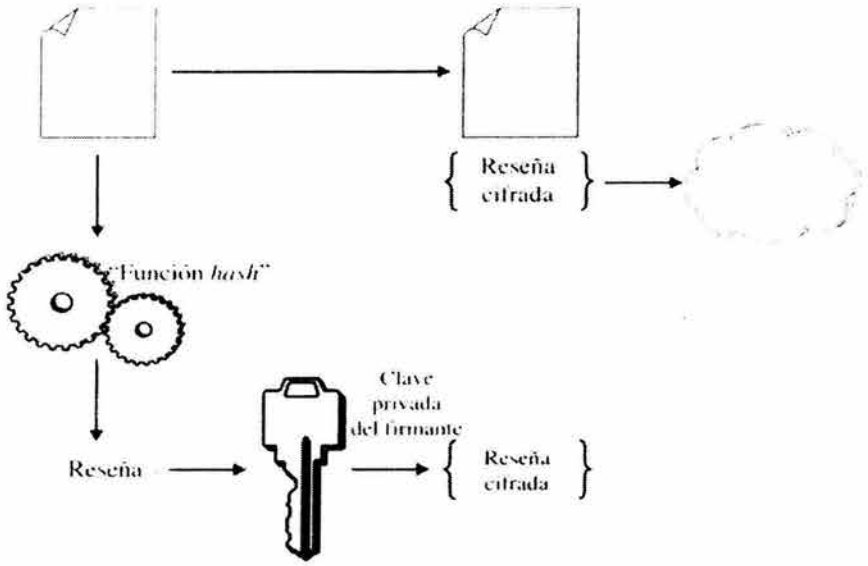


Figura 1.5
Proceso de firma digital.

Proceso de Verificación de la Firma

Para verificación y validación de la firma digital se necesita volver a generar el hash del contenido significativo del documento. El valor hash original que se cifró utilizando la clave privada del firmante se debe descifrar con la clave pública del mismo, y después se deben comparar los dos valores hash (ver figura 1.6). Si dichos valores corresponden y son iguales, la firma se considera verificada y por consiguiente válida.

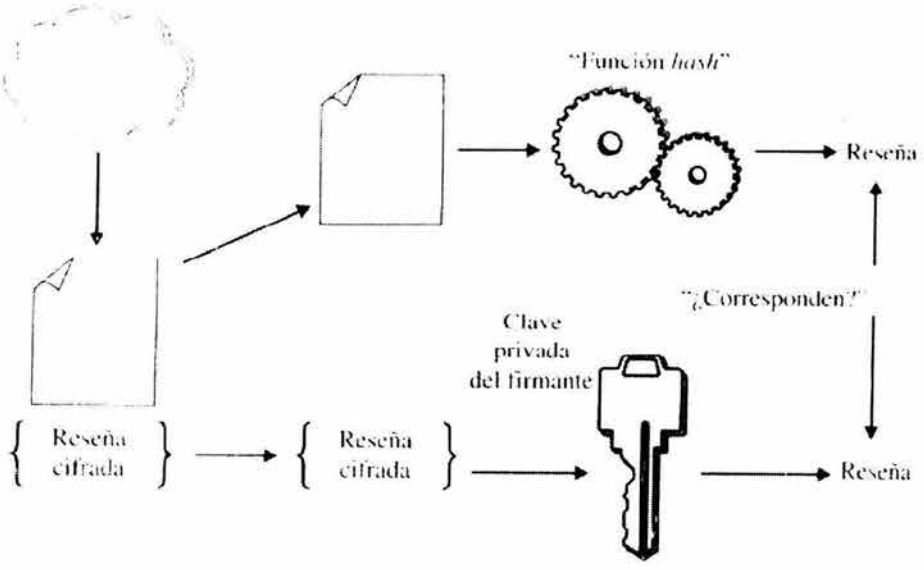


Figura 1.6
Verificación de la firma digital

Tipos de firma digital

1) El método más usado para firmar digitalmente es el conocido como RSA, lo importante de este método es que es el más usado actualmente y por lo tanto es conveniente usarlo para poder ser compatible. Para que sea seguro la longitud de sus claves (una pública y otra privada) debe de ser de 1024 bits, es decir un número de un poco más de 300 dígitos.

2) Otro método reconocido para firma digital es el llamado **DSA**[12], que es oficialmente aceptado para las transacciones oficiales en el gobierno de Estados Unidos. Este método usa también claves del mismo tamaño que RSA, pero esta basado en otra técnica. Aún así, se ha podido mostrar que es casi equivalente en seguridad a RSA.

3) Una tercera opción es el método que usa curvas elípticas, este método tiene la ventaja a los dos anteriores a reducir hasta en 164 bits, es decir como 45 dígitos las claves, manteniendo la misma seguridad. Por lo que es más propio para ser usado donde existen recursos reducidos como en Smart Cards, PDAs, etc. Actualmente este método se ha integrado como el reemplazo oficial de DSA para el gobierno de Estados Unidos.

4) Entre los posibles ataques a los anteriores métodos esta la posible remota construcción de una computadora cuántica, esta podría efectuar una cantidad tan grande de cálculos al mismo tiempo que podría romper los sistemas anteriores, incluso existen ya estos algoritmos que romperían los sistemas. Sin embargo existe otro método de forma que, aún con la computación cuántica, no existe aún algoritmo que pueda romperlos. Este sistema, que esta basado en lattices (retículas), se conoce como NTRU (Number Theory Research Unit) y entre otras cualidades es más eficiente que RSA.

5) Existen aún más métodos para firmar, incluso algunos métodos derivados de las anteriores técnicas, sin embargo no han podido tener el impacto de las anteriores, de hecho puede crearse un método de firma para un caso particular.

Así mismo es importante hacer las siguientes anotaciones:

1) La firma convencional es usada cuando la comunicación es personal, si esta comunicación fuese por ejemplo por teléfono no es posible usar la firma convencional. La firma digital está precisamente diseñada para poder ser usada a grandes distancias, y principalmente cuando esta comunicación esta hecha por dos computadoras e Internet, además puede ser usada por muchos dispositivos electrónicos.

2) Cabe también mencionar, que aunque la firma convencional puede ser enviada vía fax o por un documento que copie el garabato, ésta no es válida legalmente. Esta firma convencional se usa solo por conveniencia de alguna corporación o institución, por ejemplo al usar un sello que estampa la firma de algún ejecutivo, es usada sólo por la rapidez que representa usarla, pero legalmente no es válida. Sólo es válida aquella que es derivada del puño y letra de la persona. Por su parte la firma digital garantiza ser mejor que la convencional y sería de gran beneficio si esta tuviese validez legal.

3) Quizá la mayor diferencia entre la firma convencional y la firma digital es que la primera en su método de verificación existe una gran probabilidad de error, según algunos hasta del 20%, y en el caso de la firma digital, este error es inapreciable. Es una fuerte razón para que la firma digital tenga valor legal.[13]

[12] DSA – Siglas de Digital Signature Algorithm, otro algoritmo usado en firmas digitales.
[13] En nuestro país esto aun esta en proceso, en algunos otros países ya se ha implementado.

CERTIFICADO DIGITAL

Certificado Digital: es un archivo de aproximadamente 1k de tamaño, que contiene, primero los datos del propietario, después su clave pública y la firma digital de una autoridad competente. Cuando una persona solicita un certificado digital, se generan su par de claves, la pública y la privada. La clave pública viene en el certificado digital explícitamente. La clave privada queda en custodia del propietario del certificado. El tercer elemento importante que tiene el certificado digital es la firma digital de una autoridad certificadora, quien esta como aval de que los datos corresponden al propietario. El certificado digital queda muy parecido entonces a un documento oficial de identificación como un pasaporte o una licencia de conducir.

Los Certificados Digitales son documentos electrónicos (digitales) que sirven para asegurar la veracidad de la Clave Pública perteneciente al propietario del certificado ó de la entidad, con la que se firman digitalmente documentos que puedan proporcionar las más absolutas garantías de seguridad respecto a los cuatro elementos fundamentales mencionados anteriormente:

1. La autenticación del usuario/entidad (es quien asegura ser).
2. La confidencialidad del mensaje (que sólo lo podrá leer el destinatario).
3. La integridad del documento (nadie los ha modificado)
4. El No repudio (el mensaje una vez aceptado, no puede ser rechazado por el emisor).

Es, por tanto, muy importante estar realmente seguros de que la Clave Pública que manejamos para verificar una firma o cifrar un texto, pertenece realmente a quien creemos que pertenece.

Sería nefasto cifrar un texto confidencial con una Clave Pública de alguien, que no es nuestro intencionado receptor. Si lo hiciéramos la persona a quién pertenece la clave pública con la que lo hemos cifrado, podría conocer perfectamente el contenido de este, si tuviera acceso al texto cifrado.

De la misma forma, si manejáramos una clave pública de alguien que se hace pasar por otro, sin poderlo detectar, podríamos tomar una firma fraudulenta por válida y creer que ha sido realizada por alguien que realmente no es quien dice ser.

Otro dato a tener en cuenta, es que un certificado no puede falsificarse ya que van firmados por una Autoridad de Certificación o Autoridad Certificadora (AC). Si algún dato se modificase la firma no correspondería con el resumen (Hash[14]) que se obtendría de los datos modificados. Por tanto al utilizarlo, el software que los gestiona daría un mensaje de invalidez.

Un certificado digital contiene una clave pública, y una firma digital.

Para su correcto funcionamiento, los certificados contienen además la siguiente información:

- Un identificador del propietario del certificado, que consta de su nombre, sus apellidos, su dirección e-mail, datos de su empresa como el nombre de la organización, departamento, localidad, provincia y país, etc.
- Otro identificador de quién asegura su validez, que será una Autoridad de Certificación.
- Dos fechas, una de inicio y otra de fin del período de validez del certificado, es decir, cuándo un certificado empieza a ser válido y cuándo deja de serlo, fecha a partir de la cual la clave pública que se incluye en él, no debe utilizarse para cifrar o firmar.
- Un identificador del certificado o número de serie, que será único para cada certificado emitido por una misma Autoridad de Certificación. Esto es, identificará inequívocamente a un certificado frente a todos los certificados de esa Autoridad de Certificación.

[14] Función Hash, Función de resumen o digestión. La fórmula matemática que cambia un bloque de texto en un bloque único de texto cifrado de longitud fija.

- Firma de la Autoridad de Certificación de todos los campos del certificado que asegura la autenticidad del mismo.

Los navegadores actuales gestionan y almacenan las Claves Públicas de los certificados que permiten al emisor de mensajes firmarlos y encriptarlos utilizando las claves públicas de los destinatarios. Para estar completamente seguros en cualquier transacción es necesario utilizar, al menos dos tipos de certificados, uno general para comunicaciones seguras (X.509) y otro especifico para transacciones económicas (SET, Secure Electronic Transaction).

Formato del certificado digital[15]

En la actualidad tenemos un formato (estándar) que se ha extendido casi para todas las aplicaciones, este es el llamado X.509[16]. Este formato contiene los datos del poseedor del certificado, la clave pública del propietario, y la firma de una autoridad certificadora. La mejor propiedad del formato X.509 es que contiene el mínimo necesario de información para poder realizar muchas transacciones, principalmente comerciales y financieras. Sin embargo para otras aplicaciones puede ser un poco robusto.

La Autoridad Certificadora crea un certificado mediante la firma de la información recopilada acerca de la entidad. Esta información incluye la clave pública y el *Nombre Distinguido*[17] de la entidad, y puede incluir un identificador único que contiene información adicional acerca de aquella.
X509 define la forma de un certificado expedido por una Autoridad Certificadora con un nombre de AC, para un sujeto con nombre distinguido CA que utiliza la siguiente forma simbólica:

$$CA\langle\langle A\rangle\rangle = CA\ \{V, SN, AI, CA, UCA, A, UA, Ap, T^A\}$$

En este caso el certificado contiene la siguiente información:

V	El número de la versión del certificado
SN	El número de serie
AI	Identifica el algoritmo de firma que se utilizó para firmar el certificado
CA	El nombre distinguido de la Autoridad Certificadora (AC)
UCA	Un identificador único para la CA expedidora (opcional)
A	El nombre distinguido del sujeto identificado por el certificado
UA	Un identificador único para el sujeto (opcional)
Ap	La clave pública del sujeto A
T^A	El periodo de validez del certificado descrito por una fecha inicial y una fecha de término, durante las cuales el certificado tiene validez.

X509 también describe el formato de un certificado que utiliza una descripción ASN.1[18] que, incluso es menos legible que la forma simbólica que se mencionó previamente.

El formato General de un certificado X509 se presenta en la figura 1.7 en forma gráfica:

Los campos estándar definidos para cada certificado incluyen lo siguiente:

[15] PKI, Infraestructuras de claves Públicas; Andrew Nash,William Duane, RSA PRESS Mc Graw Hill
[16] X509- El estándar que define el certificado digital, emitido por la IETF
[17] Un Nombre Distinguido es una cadena única integrada por múltiples atributos.
[18] ASN.1- Abstract Syntax Notation Number One. Lenguaje de Definición de Notaciones. International standard : ITU-T X.680 to X.683 | ISO/IEC 8824-1 to 4; http://www.asn1.org/paper/index

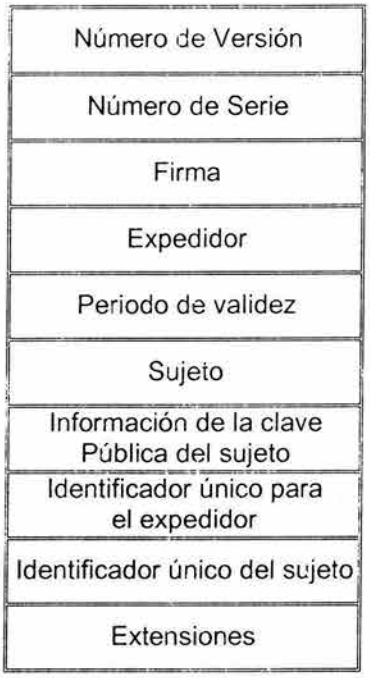


Figura 1.7
Formato general del certificado X509

- **Versión Number** - (Numero de versión)- Indica el formato y el contenido permisible definido dentro de un certificado de una versión en particular. La versión más reciente de los certificados X509 es la número 3.
- **Serial Number** - (Número de serie)- El número de serie para cada certificado expedido por una AC particular es único.
- **Signature** (firma) - Identifica el tipo de función Hash y el algoritmo de firma utilizados para firmar el certificado.
- **Issuer** (Expedidor) – Identifica la autoridad certificadora que expidió y firmó el certificado.
- **Validity** (Validez) – Define las fechas de inicio y término entre las cuales el certificado se puede considerar válido.
- **Subject** – (sujeto) – Nombre del individuo o entidad que se va a identificar con el certificado que corresponde a la clave pública que se encuentra en el certificado.
- **SubjectPublicKeyInfo** (información de la clave pública del sujeto) – Contiene la clave pública que va a certificar al sujeto del certificado. Además de la clave pública, también identifica el algoritmo para el cual se puede usar la clave pública.
- **IssuerUniqueIdentifier** (Identificador único del expedidor) Opcional y solo se puede utilizar en las versiones 2 y 3
- **SubjectUniqueIdentifier** (Identificador único de sujeto) Opcional y solo se puede utilizar en las versiones 2 y 3
- **Extensions** (Extensiones) Permite codificar información adicional en lo certificados sin requerir modificación el formato del mismo. Las extensiones estándar las define X509.

Un certificado digital típico, almacenado en Internet Explorer, y los detalles del mismo se ven mas o menos de la siguiente manera (Figura 1.8 y Figura 1.9):

Figura 1.8
Contenido de un certificado digital

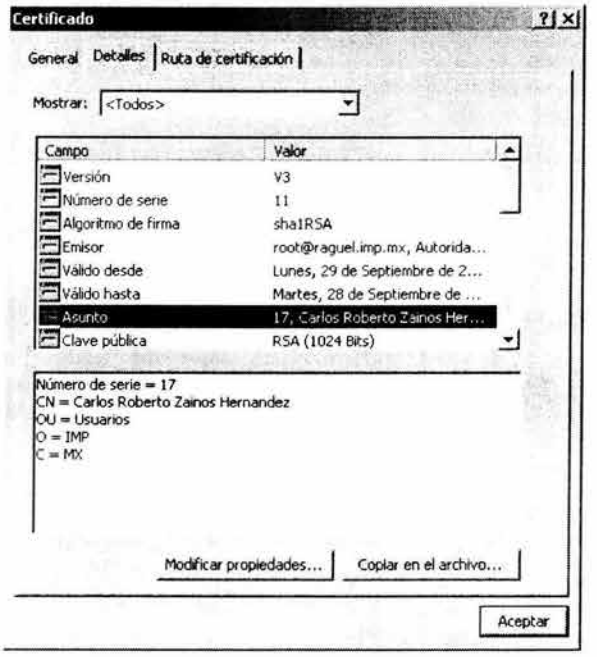


Figura 1.9
Detalles del cerificado digital

Los nombres del sujeto (subject) y del emisor (issuer) están definidos utilizando convenciones que se definen en el estándar X.500[19] [ISO/IEC 9594: *Information Technology -- Open Systems Interconnection --*

[19] X.500 – Estándar de la ITU-T que define el sistema de nombrado en Internet

The Directory).]. Esto permite establecer un nombre único mediante la definición de un concepto conocido como un *nombre distinguido (DN)*. Un nombre distinguido (Distinguished Name, DN) puede incluir el uso de información diferenciadora, tal como la organización para la cual el individuo trabaja, la dirección donde se localiza o la manera de contactarlo a través de Internet.

Terminología

Como un punto importante, definiremos a continuación algunos términos que identifican a los participantes en el uso de certificados.
Una *Entidad Destino* es una persona u objeto que se identifica mediante un certificado, o un usuario de un certificado que identifica otra entidad destino. Una entidad destino incluye personas, nodos de red (tales como servidores web, cortafuegos (Firewalls) o enruteadores), programas ejecutables y casi cualquier objeto que tenga una identidad única y al que se le pueda asignar un certificado.

El *Propietario del Certificado* es la entidad destino que se identifica en el campo del sujeto (subject) del certificado y, en ocasiones, se puede identificar como el *sujeto del certificado*.

El *Usuario del Certificado* es una entidad destino que recibe un certificado y lo utiliza con el fin de establecer la identidad de un propietario de certificado. Como el usuario del certificado se basa en la identidad establecida en el mismo, en ocasiones se le conoce como la *parte confiante*.

Tipos de certificados

Dependiendo del uso que se vaya a dar al certificado y de qué persona o entidad lo solicita, las Autoridades Certificadoras han dividido los certificados en varios tipos. Del tipo de certificado a emitir van a depender las medidas de comprobación de los datos.

Los certificados, según las comprobaciones de los datos que se realizan, se dividen en cuatro clases[20]:

- **Certificados de Clase 1**: Corresponde a los certificados más fáciles de obtener e involucran pocas verificaciones de los datos que figuran en él: sólo el nombre y la dirección de correo electrónico del titular.
- **Certificados de Clase 2**: En los que la Autoridad Certificadora comprueba otros datos que pueden estar contenidos en el certificado como por ejemplo la licencia de conducir, el número de Seguro Social, la fecha de nacimiento etc.
- **Certificados de Clase 3**: En los que se añaden a las comprobaciones de la Clase 2 la verificación de crédito de la persona o empresa.
- **Certificados de Clase 4**: que a todas las comprobaciones anteriores suma la verificación del cargo o la posición de una persona dentro de una organización (todavía no formalizados los requerimientos; está en estudio).

Desde el punto de vista de la finalidad, los certificados electrónicos se dividen en:

- **Certificados SSL para cliente**: usados para identificar y autenticar a clientes ante servidores en comunicaciones mediante el protocolo SSL (Secure Socket Layer, Capa de Conectores Segura), y se expiden normalmente a una persona física, bien un particular, bien un empleado de una empresa.

[20] Esta clasificación algunas veces no es aplicable en todos los entornos en los cuales se utilizan certificados, depende mucho del país, de la legislación, de las políticas de seguridad etc.

- **Certificados SSL para servidor**: usados para identificar a un servidor ante un cliente en comunicaciones mediante el protocolo Secure Socket Layer, y se expiden generalmente a nombre de la empresa propietaria del servidor seguro o del servicio que éste va a ofrecer, vinculando también el dominio por el que se debe acceder al servidor. La presencia de éste certificado es condición imprescindible para establecer comunicaciones seguras SSL.

- **Certificados S/MIME**: usados para servicios de correo electrónico firmado y cifrado, que se expiden generalmente a una persona física. El mensaje lo firma digitalmente el remitente, lo que proporciona Autenticación, Integridad y No Rechazo. También se puede cifrar el mensaje con la llave pública del destinatario, lo que proporciona Confidencialidad al envío.

- **Certificados de firma de objetos**: usados para identificar al autor de archivos o porciones de código en cualquier lenguaje de programación que se deba ejecutar en red (Java, JavaScript, CGI, etc). Cuando un código de éste tipo puede resultar peligroso para el sistema del usuario, el navegador lanza un aviso de alerta, en el que figurará si existe certificado que avale al código, con lo que el usuario puede elegir si confía en el autor, dejando que se ejecute el código, o si por el contrario no confía en él, con lo que el código será rechazado.

- **Certificados para AC**: que identifican a las propias Autoridades Certificadoras, y es usado por el software cliente para determinar si pueden confiar en un certificado cualquiera, accediendo al certificado de la AC y comprobando que ésta es de confianza.

APLICACIONES QUE USAN CERTIFICADOS

En esta sección introduciremos algunas de las aplicaciones más comunes que utilizan certificados digitales. Estas aplicaciones y los servicios que usan los trataremos más ampliamente en secciones más adelante.

Nivel de Socket Seguro

El protocolo de nivel de socket (conector) seguro SSL (*Secure Socket Layer*) , nos es familiar a la mayoría de nosotros quienes hemos tenido acceso a información segura (codificada) en Internet. El SSL se selecciona para garantizar una sección web en donde utilizamos nuestro navegador para acceder a un URL que está precedido por *https:* y no por *http:* . Esta forma de URL dirige al navegador a un puerto diferente en el servidor web, y toda la información que se intercambie durante la sesión establecida utilizará la codificación (viajará encriptado) para garantizar la privacidad.
Además, los certificados se utilizan para establecer la identidad del servidor web y posiblemente la del cliente. En la sección siguiente se aborda esto de manera mas detallada.

Correo Electrónico Seguro

El correo electrónico se asegura utilizando diferentes mecanismo. El emisor puede establecer su identidad firmando digitalmente el mensaje. Dicho emisor crea la firma utilizando su clave privada y enviando después el certificado correspondiente con el correo. Esto permite que el receptor valide la identidad de quien firma y que verifique que el contenido del mensaje no se ha modificado.

Firmar el correo electrónico no ofrece privacidad al contenido del mensaje; los detalles de éste podía leerlos cualquier persona que lo intercepte. Para brindar privacidad, el emisor debe obtener el certificado del receptor y usar la clave pública de este último para cifrar el contenido. La validación del certificado del receptor permite que el emisor tenga confianza en que la información se haya cifrado con la clave pública correcta, y que el receptor sea la única persona que pueda descifrar el contenido.

Redes Privadas Virtuales

Las Redes Privadas Virtuales (*Virtual Private Networks, VPN*) permiten el uso público de Internet, como si se tratase de una red privada. Para hacerlo, la información se debe codificar a medida que pasa entre los usuarios de la red. Sin embargo, como el acceso ala red no está restringido y cualquier nodo de la red se puede enviar a otro, se deben establecer las identidades de los nodos participantes.

En el caso de las VPN's, se establecen las identidades de las máquinas que se están comunicando y no de los usuarios de las máquinas (aunque puede ser necesario establecerlo).
En el caso de una VPN, la identidad que se establece es la del nodo de red y la dirección de red que se está usando. En este caso, los certificados expedidos hacia entidades destino incluyen la dirección de red del sujeto que se está identificando con el certificado. Las VPN's tienen diferentes modos de operación, pero en general, cada entidad destino se identificará presentando sus certificados a las otras y verificando que posean las claves privadas correspondientes.

Además de servir como mecanismo confiable y seguro de identificación en la red, el certificado de identidad digital permite disfrutar de otra serie de beneficios: se puede enviar y recibir información confidencial, asegurándose que sólo el remitente pueda leer el mensaje enviado; se puede acceder a sitios web de manera segura con la identidad digital(certificado), sin tener que usar el peligroso mecanismo de passwords; se pueden firmar digitalmente documentos, garantizando la integridad del contenido y autoría de los mismos; y todas aquellas aplicaciones en que se necesiten mecanismos seguros para garantizar la identidad de las partes y confidencialidad e integridad de la información intercambiada, como comercio electrónico, declaración de impuestos, pagos provisionales, uso en la banca, etc.

Algunas aplicaciones de Internet como navegadores (por ejemplo, Internet Explorer o Netscape Navigator) o programas para correo electrónico, ya traen incorporados los elementos que les permiten utilizar los certificados digitales, por lo que los usuarios no necesitan instalar ningún software adicional.

RECAPITULACIÓN SOBRE CRIPTOGRAFÍA

Antes de poder utilizar lo anteriormente expuesto como una herramienta en la búsqueda de la solución a nuestros problemas, capturemos algunos puntos principales que es necesario recordar.

- Las mejores aplicaciones de criptografía combinan algoritmos simétricos y asimétricos para respaldar las fortalezas de cada uno.
- Con la combinación de criptografía simétrica y asimétrica, las claves asimétricas llegan a ser efímeras: se utilizan una vez y luego se descartan. Las claves simétricas se usan para el cifrado de bloques.
- Las claves asimétricas se suelen usar para envolver las claves simétricas y protegerlas durante su tránsito, lo mismo que para cifrar verificaciones (hashes) de datos para crear firmas digitales.
- Las claves públicas están protegidas del engaño al codificarlas en un certificado digital, junto con la identidad del propietario.
- Las autoridades de confianza firman certificados digitales. La mayor parte del software contiene listas, cargadas previamente, de dichas autoridades.
- Las firmas digitales deben incluir una marca de hora precisa y confiable si van a resistir el rechazo.

TRANSACCIONES SEGURAS EN INTERNET

A menudo cuando navegamos por el web, hay ocasiones en las que se necesita enviar datos sensibles como, por ejemplo, la información personal o los números de la tarjeta de crédito.

En tales casos es importante que se verifique la autenticidad del servidor al que le estamos enviando la información, porque no tiene sentido enviar información delicada si no sabemos quién la va a recibir. Además de esto, es importante que la comunicación entre el navegador y el servidor web estén cifradas, de manera que un pirata informático no pueda tener acceso a la información cuando ésta viaje a través de Internet. Para esto, los servidores web soportan un protocolo denominado nivel de *socket* seguro (Secure Socket Layer, SSL).

A continuación veremos este proceso paso a paso pero de una manera muy general (ver Figura 1.10).

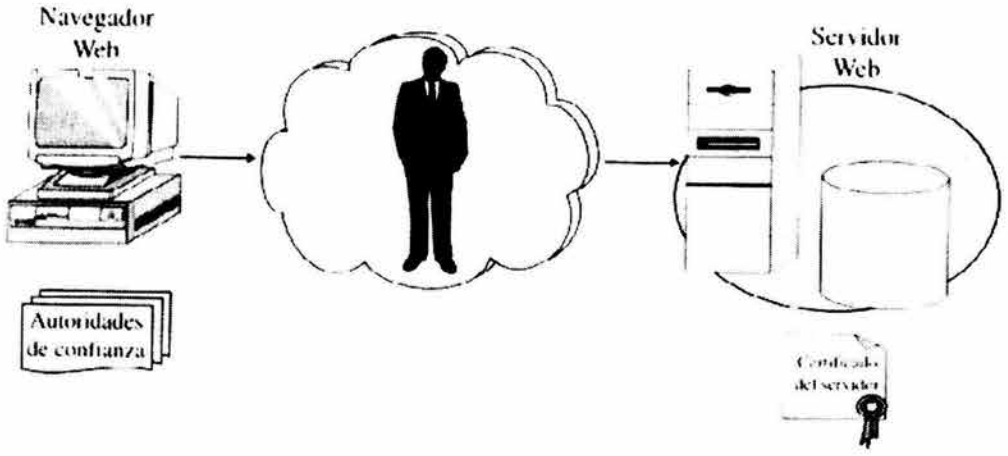


Figura 1.10
Configuración SSL

En la figura anterior vemos que el servidor web presenta un certificado digital que contiene la identidad de dicho servidor web., lo mismo que la clave pública de este. Resulta implícito, aunque no se demuestra, el hecho de que el servidor web también está haciendo la comparación de la clave privada; además, se observa que el navegador tiene una tabla cargada previamente de autoridades de confianza.

En esta sección por simplicidad abordaremos lo que se conoce como "SSL del Servidor", más adelante trataremos este mecanismo de cifrado de manera más detallada. Como podemos ver, cuando utilizamos un SSL del servidor, el navegador autentica al servidor web y se crea un canal cifrado entre uno y otro. Sin embargo, el servidor no autentica al navegador; esto es algo bastante común. Si por ejemplo estamos realizando una compra en amazon.com, lo que deseamos es asegurarnos que realmente estamos conectados con Amazon y deseamos cifrar la información de nuestra tarjeta de crédito. Amazon no necesita autenticar a nuestro navegador, debido a que nos estaremos autenticando directamente, a través de la información de nuestra tarjeta de crédito.

Hay casos en que para el servidor web es necesario autenticar al navegador, como en el caso en que una compañía quiere poner toda su información corporativa en un servidor web para la misma empresa. En esta situación sería importante garantizar que únicamente los empleados de la empresa tuvieran acceso al servidor; para tal efecto se usa una autenticación SSL del lado del cliente.

Regresando a nuestro caso de ejemplo, el primer paso en el proceso es para que el servidor web envíe su o sus certificados digitales al navegador web (ver Figura 1.11).Recordemos que como toda la información en el certificado digital es pública, no importa que éste viaje en claro entre el servidor web y el navegador.

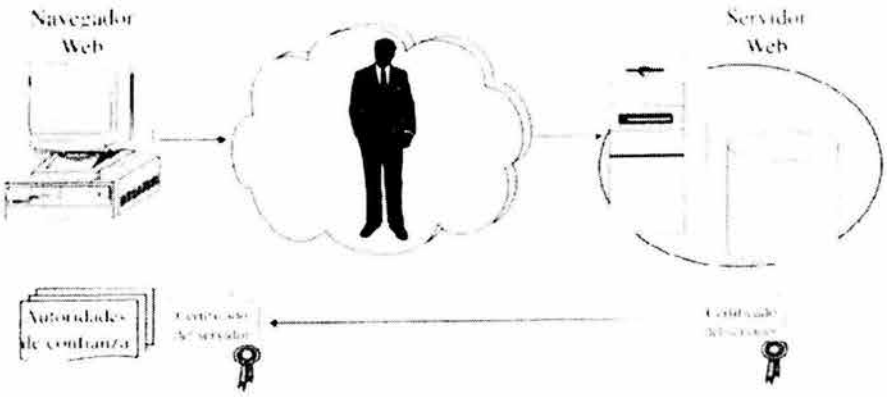


Figura 1.11
Intercambio de certificados

En la figura 1.12 podemos ver al navegador extrayendo la clave pública del certificado digital del servidor web. Antes de que el navegador pueda confiar en la clave pública, debe validar el certificado del servidor, viendo si está firmado por una fuente de la lista de autoridades de confianza. Suponiendo que así sea, el navegador calculará el hash del certificado y lo comparará con el hash que está en el certificado (descifrado utilizando la clave pública de la autoridad de confianza). Si los hashes corresponden, el navegador sabe que el certificado no ha sido alterado. A continuación, verifica las fechas de validez codificadas en el certificado para estar seguro de que éste no ha vencido (expirado). Suponiendo que no ha vencido, hará una verificación más especial asociada con los certificados de servidor web. Parte de la información de identidad en el certificado de servidor es el URL[21] (*Unified Resource Location*) del servidor web. El navegador hará una revisión extra para garantizar que el nodo que envía la información tenga el mismo URL que está codificado en la información de la identidad. Si todas estas verificaciones corresponden, entonces el navegador extraerá la clave pública del certificado del servidor web.

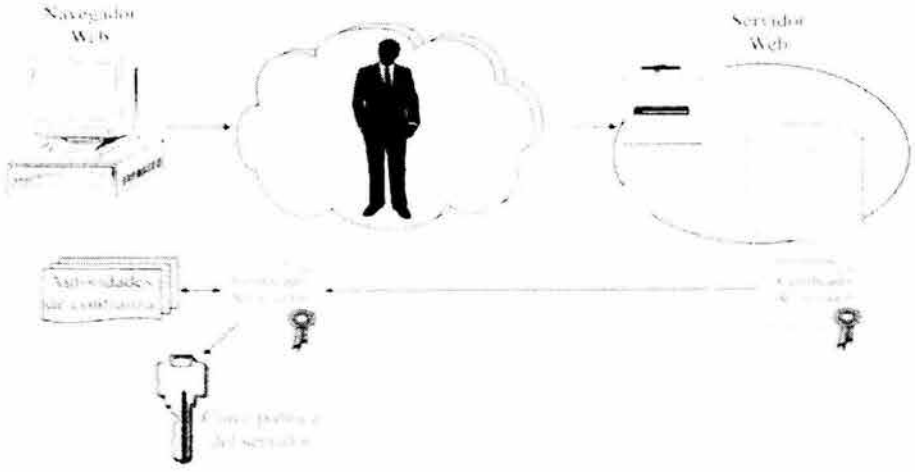


Figura 1.12
Extracción de la clave pública

[21] URL- Uniform Resouce Locator; localizador uniforme de recursos, permite localizar o acceder de forma sencilla cualquier recurso de la red desde el navegador de la WWW.

Una vez que el navegador tiene la clave pública del servidor, entonces se genera una clave aleatoria de cifrado simétrico que se usará para cifrar la conversación entre el navegador y el servidor. Recordemos que un algoritmo de cifrado simétrico se utiliza debido a que los cifrados simétricos son rápidos y no expanden los datos durante la operación de cifrado. Para mover la clave de cifrado simétrico hacia el servidor web, el navegador realiza una operación de empaquetado de la clave (ver Figura 1.13).

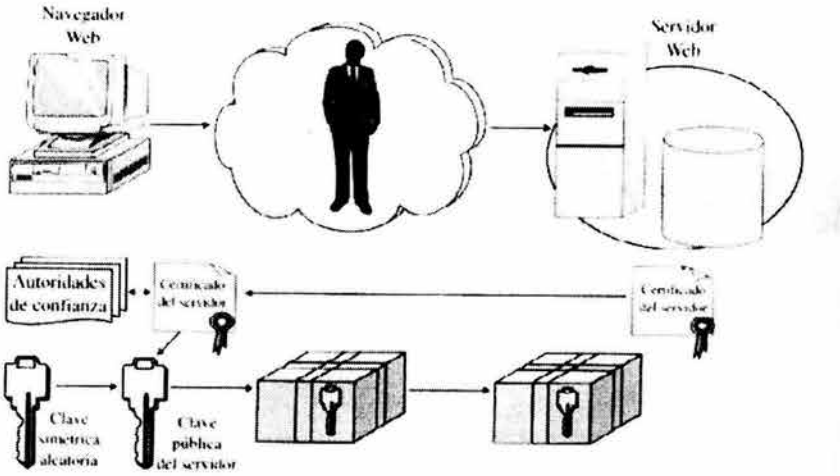


Figura 1.13
Empaquetado de la clave

La clave simétrica es cifrada utilizando la clave pública del servidor web, que se extrajo del certificado digital presentado por el servidor . El navegador envía la clave empaquetada al servidor. Como la clave simétrica esta cifrada usando la clave pública del servidor, y como éste es la única entidad que tiene la clave privada correspondiente, un pirata informático no puede extraer la clave simétrica.
Ahora que el servidor web tiene la clave empaquetada, puede usar la clave privada para descifrar la anterior (ver Figura 1.14). Esto lleva a la clave simétrica original que generó aleatoriamente el navegador web.

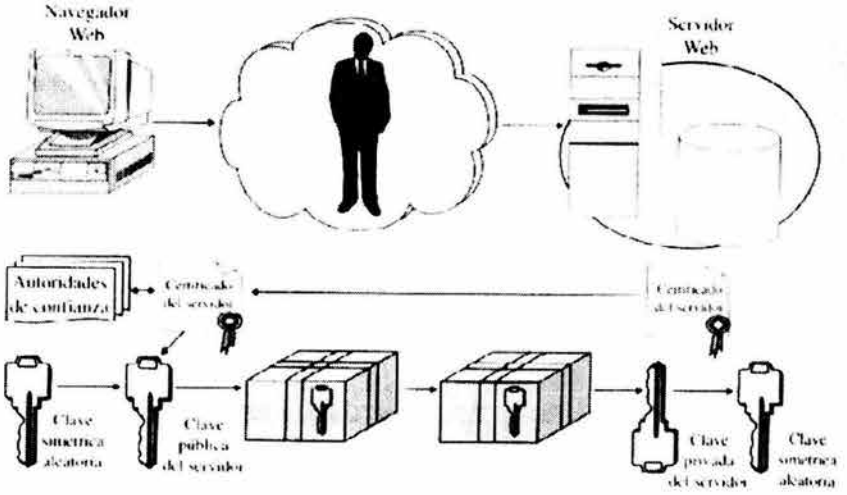


Figura 1.14
Desempaquetado de la clave

En este punto tanto el navegador como el servidor web tienen una copia de la misma clave de cifrado simétrico.
En la figura 1.15 se puede ver que ambos extremos de la conversación tienen la misma clave simétrica. Ahora pueden comenzar una conversación codificada utilizando el intercambio de la clave simétrica para cifrar y descifrar los datos de cada uno.

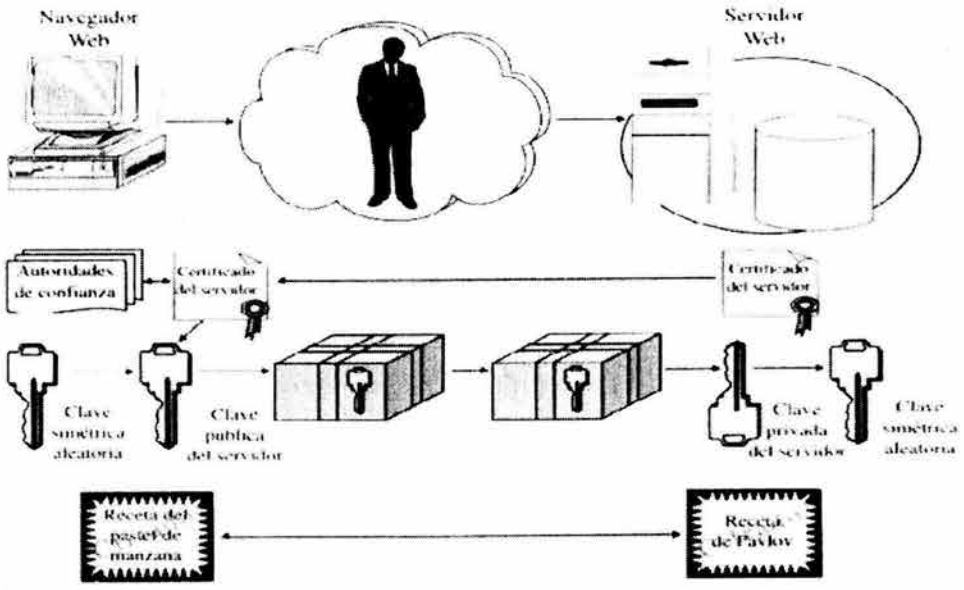


Figura 1.15
Comienzo de la conversación

Algo más sucedió en este intercambio y es ligeramente sutil. Como mencionamos al comienzo, el navegador necesita asegurarse de estar hablando con el servidor web correcto; en otras palabras, debe reconocer la autenticidad del servidor.

La verificación que mencionamos en el procesamiento del certificado donde el navegador verifica la URL del servidor, no es una revisión suficiente. Esto se debe a que un sitio web criminal podría estar fingiendo ser el sitio web real. En tal caso, todo el tráfico del sitio verdadero se dirigiría hacia el sitio de la estafa. Este tipo de ataque es bastante común en Internet y, con frecuencia, se logra cuando el pirata informático compromete a un servidor DNS[22] (*Domain Name Sevice*) y dirige todo el tráfico al sitio ilegal. Por consiguiente la simple verificación del URL no es suficiente. Se necesita algo más fuerte. Explicamos a continuación cómo ocurre la autenticación.

El navegador web generó una clave aleatoria de cifrado simétrico y luego la cifró usando la clave pública del servidor web. El hecho de que el servidor web pudiera participar en una conversación codificada con el navegador web, le dice a éste último que el servidor ha tenido éxito en descifrar la clave privada correspondiente, necesaria para realizar la operación de desempaquetado.

Como se puede ver, hemos simplificado mucho el protocolo SSL para mostrar los conceptos principales. En secciones posteriores se detalla mas a fondo lo que es este protocolo, cómo trabaja, cómo se usa y cuales son sus principales ventajas y desventajas.

[22] DNS- Servicio de resolución de nombres de dominio en Internet.

OTROS PROTOCOLOS SEGUROS

Protocolo TLS - Transport Layer Security.-
Para intentar corregir las deficiencias observadas en SSL v3 se buscó un nuevo protocolo que permitiera transacciones seguras por Internet, sobre todo teniendo en cuenta que SSL es propiedad de la empresa Nestcape. El resultado de esta búsqueda fue el protocolo TLS, que permite una compatibilidad total con SSL siendo un protocolo público, estandarizado por el IETF[23](*Internet Engineering Task Force*).
TLS busca integrar en un esquema tipo SSL al sistema operativo, al nivel de la capa TCP/IP, para que el efecto "túnel" que se implementó con SSL sea realmente transparente a las aplicaciones que se están ejecutando.

Protocolo S-HTTP.-El protocolo Secure HTTP fue desarrollado por Enterprise Integration Technologies, EIT, y al igual que SSL permite tanto el cifrado de documentos como la autenticación mediante firma y certificados digitales, pero se diferencia de SSL en que se implementa a nivel de aplicación. Se puede identificar rápidamente a una página web servida con este protocolo porque la extensión de la misma pasa a ser shtml en vez de html como las páginas normales.
El mecanismo de conexión mediante S-HTTP, que ahora se encuentra en su versión 1.1, comprende una serie de pasos parecidos a los usados en SSL, en los que cliente y servidor se intercambian una serie de datos formateados que incluyen los algoritmos criptográficos, longitudes de clave y algoritmos de compresión a usar durante la comunicación segura.

En cuanto a estos algoritmos, lo usados normalmente son RSA para intercambio de claves simétricas, MD2, MD5 o NIST-SHS como funciones hash de resumen, DES, IDEA, RC4 o CDMF como algoritmos simétricos y PEM o PKCS-7 como algoritmos de encapsulamiento.

A diferencia de SSL, el protocolo S-HTTP está integrado con HTTP, actuando a nivel de aplicación, como ya hemos dicho, negociándose los servicios de seguridad a través de cabeceras y atributos de página, por lo que los servicios S-HTTP están sólo disponibles para el protocolo HTTP. Recordemos que SSL puede ser usado por otros protocolos diferentes de HTTP.

Protocolo SET

Las carencias de SSL a la hora de implementar las cuatro condiciones básicas de una transacción segura, hicieron que diferentes empresas y organismos buscaran un nuevo sistema que permitiera realizar operaciones sensibles por Internet de forma segura, con el objeto de estimular la confianza de los consumidores en el comercio electrónico.

En febrero de 1996 un grupo de empresas del sector financiero, informático y de seguridad (Visa International, MasterCard, Microsoft, Nestcape, IBM, RSA, ect.) anunciaron el desarrollo de una nueva tecnología común destinada a proteger la compras a través de redes abiertas como Internet basadas en el uso de tarjetas de crédito. Esta nueva tecnología se conoce con el nombre de Secure Electronic Transactions (Transacciones Electrónicas Seguras), SET, y ha sido creada exclusivamente para la realización de comercio electrónico usando tarjetas de crédito.

SET se basa en el uso de certificados digitales para asegurar la perfecta identificación de todas aquellas partes que intervienen en una transacción on-line basada en el uso de tarjetas de pago, y en el uso de sistemas criptográficos de clave pública para proteger el envío de los datos sensibles en su viaje entre los diferentes servidores que participan en el proceso. Con ello se persigue mantener el carácter estrictamente confidencial de los datos, garantizar la integridad de los mismos y autenticar la legitimidad de las entidades o personas que participan en la transacción, creando así un protocolo estándar abierto para la industria que sirva de base a la expansión del comercio electrónico por Internet.

[23] IETF- Grupo de trabajo de Ingeniería de Internet; http://www.ietf.org

INFRAESTRUCTURA DE CLAVES O LLAVES PUBLICAS (PKI)

La criptografía de clave pública suministra las herramientas que permiten operaciones de seguridad, como en las firmas digitales y en la distribución de claves. La tecnología básica ha estado disponible en diferentes formas durante los últimos 27 años (teniendo en cuenta la fecha en que se publicó el documento original de Diffie y Hellman)[24]. RSA se desarrolló en 1978 y sigue siendo el algoritmo de clave pública de más amplia difusión en la actualidad.

PKI (Public Key Infrastructure) es una arquitectura de seguridad que ha sido introducida para proporcionar un incremento en el nivel de seguridad y de confianza en el intercambio de información sobre Internet que es insegura y que lo seguirá siendo de manera creciente. La Infraestructura de Claves Públicas constituye el marco de referencia que permite desplegar servicios de seguridad que se basan en el cifrado.

Introducción

La expansión en el uso de la tecnología de clave pública se ha logrado por el conjunto de servicios, interfaces de programación, herramientas administrativas y aplicaciones de usuario que forman una Infraestructura de claves Públicas (Public Key Infrastructure, PKI). Aunque los estándares, las tecnologías y la manera de implementar PKI están en evolución, como es típico de cualquier infraestructura compleja, tiene una estructura bastante estándar que es fácilmente reconocible y generalmente aceptada.

El término PKI puede ser muy confuso, inclusive para un tecnólogo, debido a que es empleado para definir muchas cosas distintas. Por un lado tenemos que PKI puede significar los métodos, tecnologías y técnicas que juntas proporcionan una infraestructura segura.

Por el otro lado, puede significar el uso de un par de llaves publica y privada para autenticarse y realizar una prueba de contenido. Una infraestructura PKI es concebida para ofrecer a sus usuarios básicamente los siguientes beneficios:

- Certeza de la calidad de la información enviada y recibida electrónicamente
- Certeza de la fuente y destino de dicha información
- Aseguramiento del tiempo y de la sincronización de esa información
- Certeza de la privacidad de la información
- Aseguramiento de que la información puede ser utilizada como una evidencia ante la ley en una corte[25].

Esta infraestructura se logra utilizando Criptografía de clave o llave pública, la cual utiliza un par de llaves criptográficas relacionadas para verificar la identidad del emisor (firmando) y/o asegurando la privacidad (encriptando), como ya se ha explicado antes.

La infraestructura PKI ha sido desarrollada principalmente para brindar soporte en el intercambio de información de manera segura sobre redes o canales inseguros –como Internet- donde tales características no pueden ser proporcionadas de otra manera fácilmente. La Infraestructura de PKI puede, sin embargo, ser usada solo como una facilidad para el intercambio de información sobre redes privadas, caso nuestro en el IMP, incluyendo redes internas corporativas. PKI puede también ser utilizada para entregar llaves criptográficas entre usuarios (incluidos dispositivos como los servidores) de manera segura, y para facilitar la entrega de otros servicios de seguridad criptográfica.

[24] Diffie W. and M. Hellman, "*New directions inCriptography*", IEEE Transactions on Information Theory, 1976

[25] En México la legislación actual no lo tiene contemplado y no lo permite, en paises como Argentina, España o Estados Unidos esto si es posible y el uso de esta infraestructura trae consigo implicaciones y responsabilidades legales.

En resumen, de manera muy simple y a grandes rasgos podemos decir que PKI es una tecnología de autenticación. Utilizando una combinación de llaves criptográficas públicas y secretas, PKI permite otros servicios de seguridad que incluyen la confidencialidad de datos, la integridad de los mismos y administración de las llaves. PKI permite crear la entidades y la confianza que se necesita para los procesos de identificación y autenticación, y para administrar el cifrado de clave pública que ofrece soluciones escalables de cifrado y seguridad.

La criptografía de claves públicas no es suficiente

En las secciones anteriores hemos mencionado un conjunto de métodos y técnicas de codificación que permiten ofrecer cualquiera de los servicios de seguridad que nosotros probablemente necesitamos para los objetivos que perseguimos.

Las claves públicas/privadas las hemos presentado como componentes significativos de los servicios que son utilizados por estas soluciones. De modo que podemos preguntarnos ¿por qué se requiere una infraestructura para las claves pública/privada?. Trataremos de responder a esta cuestión con un ejemplo.

Uno de los mecanismos más interesantes que se basan en la criptografía de clave pública es la generación de firmas digitales. Los documentos electrónicos han carecido de una forma amplia de aceptación que le permita al autor certificar el contenido de un documento o verificar que dicho contenido no haya sido modificado con respecto al original. Las firmas digitales soportan estos atributos. Una forma digital podría serle necesaria simplemente para efectos de verificación, para permitir que una persona se cerciore de haber revisado o firmado una forma de gastos. Alternativamente, para darle a la forma electrónica de un contrato un estatus legal, las partes involucradas deben suministrar su aprobación de una manera que sea reconocible.

La legislación reciente en Estados Unidos y algunos países de Europa ha suministrado la base para el reconocimiento legal de diferentes formas de firmas electrónicas. Éstas incluyen firmas digitales generadas con sistemas que se basan en criptografía de clave pública.

Recordando un poco el proceso de creación y verificación de una firma digital, vemos que ambos procedimientos se basan en la generación y comparación del hash del documento o documentos que se firman, así como en la utilización del par de claves que se utilizan para cifrar/descifrar dicho hash.

El principal problema al que nos enfrentamos es: ¿Cómo sabemos que la pareja de claves que se usa para cifrar y descifrar el valor hash realmente pertenece al individuo que creemos que firmó el documento?

La creación de una pareja de claves pública/privada es una operación simple para un usuario; se puede usar cualquier navegador web. Un intruso inteligente puede haber sustituido su clave pública por la clave pública que ya se utilizó para identificar a otra persona – después de todo la clave pública es pública y debe estar localizada en un sitio de fácil acceso – o puede convencernos de tomar la clave, identificándose falsamente con la identidad de otra persona. Una vez que se ha conocido la clave pública, el intruso utiliza la clave privada que posee para generar una firma. La secuencia de pasos descrita anteriormente para la verificación de la firma dará un resultado correcto, llevando al usuario a validar la firma al confiar en ella, pero incorrectamente.

De manera que llegamos a un punto en el cual utilizar únicamente las técnicas criptográficas no es suficiente para satisfacer las metas de un servicio de seguridad, tal como la generación de una firma digital verificable. La simple posesión de un apareja de claves pública/privada no es suficiente para permitir el establecimiento de una identidad confiable.

Necesidad de identidades de confianza

El papel primario de PKI es establecer identidades digitales en las que se pueda confiar. Éstas se pueden usar junto con mecanismos criptográficos para prestar un servicio de seguridad como autenticación, autorización, o validación de una firma digital, para que los usuarios del servicio puedan tener una confianza razonable de que no se les va a engañar.

El problema con la creación de una identidad en la que se pueda confiar es encontrar una persona o institución que esté preparada para dar fe suficiente de la identidad. Entre las personas dignas de confianza se podría incluir a un amigo de mucho tiempo, el médico de la familia, el juez de un tribunal, una institución de confianza o un notario .

Establecer identidades de confianza es un evento que tiene lugar en muchas formas cada día. Estas van desde identificarse uno mismo ante otra persona, autoridad o institución pública o privada, hasta la verificación de dichas identidades, validez, estado y utilidad dependiendo del ámbito en que se les utilice.

Algunos ejemplos de identidades de confianza en nuestra vida cotidiana pueden ser : la credencial de elector, la licencia de conducir, el pasaporte, la tarjeta de afiliación al seguro social, tarjetas bancarias entre otras. Es preciso notar que algunos documentos son más confiables que otros. En México por ejemplo, la licencia de conducir no siempre es aceptada ya que no existe un solo organismo que se encargue de la emisión y control de las mismas.

Componentes de una PKI

El modelo básico de una arquitectura PKI se ha mantenido por largo tiempo sin grandes cambios desde que fue por primera vez publicada[RFC2459 Internet X509 Public Key Infrastructure]. El último modelo esta reflejado en la más reciente versión de la certificación en Internet y el perfil de CRL [RFC3280]. Una PKI consiste, básicamente, de cuatro tipos de componentes:

- **Autoridades de certificación AC(ó CA, Certification Authority):** que emiten y revocan certificados PKCS (Public Key Criptography Standards)[26] y usualmente Listas de Revocación de Certificados (CRL's). Puede también realizar una variedad de funciones administrativas, aunque estas son delegadas a menudo a una o más Autoridades Registradoras.
- **Autoridades Registradoras AR (ó RA, Registration Authority):** que atestiguan o verifican el enlace entre claves publicas y los certificados que las contienen y otros atributos. Son un componente opcional que pueden asumir un numero de funciones administrativas de la AC.
- **Clientes o Entidades Finales.** Una entidad final es un término utilizado para denotar a los usuarios finales, dispositivos (como por ejemplo servidores, routers, etc.) o alguna otra entidad que pueda ser identificada dentro del campo "sujeto" de un certificado de clave pública. Las entidades finales o clientes típicamente consumen y/o soportan servicios relacionados con PKI.
- **Repositorios de Certificados**. Un repositorio es un término genérico utilizado para denotar algún método para almacenar certificados y las listas de revocación de certificados (CRL's) de tal manera que puedan ser recuperados por los clientes o entidades finales.

Estos componentes y su interoperación se representan en la siguiente figura (Figura 1.16)

[26] Estándares criptográficos de clave pública. Una serie de especificaciones desarrolladas por RSA Laboratories que definen elementos y estructuras de datos criptográficos comunes. En el Anexo A-1 se encuentran listados los PKCS's existentes así como una explicación más amplia de ellos.

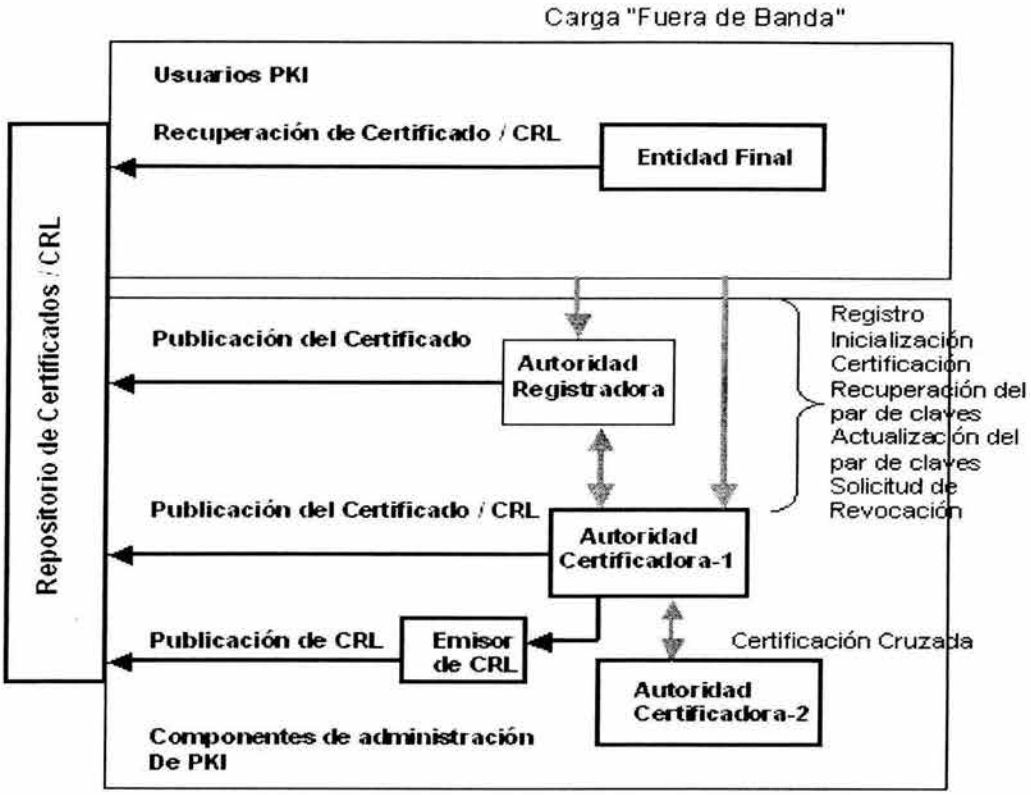


Figura 1.16
Modelo básico de la Arquitectura PKI[27]

Si bien es cierto que el anterior esquema es el que se propone inicialmente, dicho esquema puede modificarse para adaptarse a las necesidades de cada organización, así como a sus políticas de seguridad y certificación que cada una pueda implementar.

FUNCIONES ADMINISTRATIVAS DE UNA PKI[28]

La infraestructura de PKI asocia un número de funcionas administrativas que "potencialmente necesitan ser soportadas por los protocolos de administración" [RFC 3280]. La figura 1.16 ilustra la interacción entre los diversos componentes de una PKI y resume los tipos de funciones administrativas que pueden existir entre dichos componentes. Estas funciones administrativas, las más comunes, se mencionan a continuación:

Registro:

Este es el proceso por el cual un sujeto o entidad primero se presenta ante una AC (directamente o por medio de una AR), antes de que la AC emita un certificado o certificados para el sujeto en cuestión. EL registro involucra que el sujeto aporte su nombre, nombre de dominio, organización etc., así como algunos otros atributos para ser colocados en el certificado, seguido de la verificación por parte de la AC (posiblemente con la ayuda de la AR) de acuerdo a los Enunciados de Prácticas de Certificación (CPS[29]) de la validez del nombre y de otros atributos, verifica que sean correctos.

[27] PKI Basics, A Technical Perspective; PKI Forum's Business Working Group; Formato PDF; Noviembre 2002
[28] Son las más importantes, las que además forman parte de la administración de la PKI, la cual incluye algunos otros aspectos y funciones contempladas en la fase de diseño la dicha infraestructura
[29] CPS- También conocidos como Declaraciones Prácticas de Certificación (*Cetification Practices Statement, CPS*). Se refieren a las reglas que describen la manera como las diferentes facetas de una AC están limitadas y operan.

Inicialización:

Al proceso de registro sigue el de inicialización. La inicialización es cuando el sujeto obtiene los valores necesarios para comenzar la comunicación con la PKI. Por ejemplo, la inicialización puede involucrar que este le provea al sistema cliente la llave publica o el certificado de la AC, ó generando él mismo su propio par de claves pública/privada.

Generación de claves:

Dependiendo de las políticas de la AC, el par de claves pública/privada puede cada una ser generada por el usuario en su ambiente local, o generada por la AC o la AR. En el caso último, las llaves materiales pueden ser distribuidas al usuario en un archivo encriptado, en un dispositivo de prenda (token) físico, una tarjeta inteligente o una PC card

Certificación:

Este es el proceso en el cual la AC emite un certificado para la clave publica del sujeto, y regresa el certificado al sujeto, directamente o vía una AR, o lo almacena en un repositorio.

Actualización de claves:

Todos los pares de claves necesitan ser actualizadas regularmente (por ejemplo reemplazadas con un nuevo par) y nuevos certificados deben ser emitidos. Esto sucedería en dos casos:
Normalmente, cuando una clave ha sobrepasado su periodo máximo de vida útil; y excepcionalmente, cuando una clave ha sido comprometida y debe ser reemplazada.

Claves expiradas:

En un caso normal, una PKI necesita proveer las facilidades para permitir la transición de un certificado con una clave existente a un nuevo certificado con una nueva clave Esto es particularmente cierto cuando la clave a ser actualizada es la de la AC. Los usuarios conocerían de antemano que la clave expirara en un cierto tiempo; la PKI, trabajando junto con los certificados utilizados por aplicaciones. Deben permitir el trabajo antes y después de la transición.

Solicitudes de Revocación:

Como lo hemos mencionado anteriormente, los certificados de clave pública son emitidos con un cierto periodo de validez. Sin embargo, las circunstancias que existían cuando el certificado fue emitido pueden cambiar antes de que el certificado expire de manera "natural". Las razones para la revocación de un certificado incluyen el que la seguridad de la clave privada se haya comprometido, un cambio de asignación de claves, cambio de nombre, etc.

Por lo tanto, es algunas veces necesario revocar un certificado antes de la fecha de expiración. La solicitud de revocación permite a una entidad final solicitar la revocación del certificado que se le ha emitido. Por supuesto, la entidad final no puede por ningún motivo estar involucrada de manera directa durante el proceso de revocación. Ella solo se limita a realizar la solicitud y a confirmar dicha solicitud en su momento.

Certificación Cruzada:

Como lo muestra la figura 1.16, existe una *Certificación Cruzada* entre AC's. Un certificado cruzado es un certificado de clave pública que es emitido por una AC para otra AC. En otras palabras, una certificación cruzada es un certificado de clave pública que contiene la clave pública de una AC que ha sido firmada digitalmente por otra AC.

Muchos interpretan la certificación cruzada como una referencia de certificación intra dominio (o dentro de la misma organización sobre la que se esté trabajando). En esta parte nos limitaremos a mencionar que este tipo de certificación existe y más adelante detallaremos esto si es que así lo consideramos necesario.

Tal vez el elemento más importante dentro de la infraestructura PKI es la Autoridad Certificadora, la cual se detalla a continuación:

La **Autoridad de Certificación ó Autoridad Certificadora AC (CA en inglés)**, es quien firma digitalmente los certificados, asegurando su integridad y certificando la relación existente entre la Clave Pública contenida y la identidad del propietario. La firma de la AC es la que garantiza la validez de los certificados.

La confianza de los usuarios la Autoridad de Certificación es fundamental para el buen funcionamiento del servicio. El entorno de seguridad (control de acceso, cifrado, etc.) de la AC ha de ser muy fuerte, en particular en lo que respecta a la protección de la Clave Privada que utiliza para firmar sus emisiones. Si este secreto se viera comprometido, toda la infraestructura de Clave Pública (PKI) se vendría abajo.

Las Autoridades Certificadoras pueden realizan las siguientes tareas:

- Emisión de los certificados de usuarios registrados y validados por la Autoridad de Registro (AR).
- Revocación de los certificados que ya no sean válidos Un certificado puede ser revocado por que los datos han dejado de ser válidos, la clave privada ha sido comprometida o el certificado ha dejado de tener validez dentro del contexto para el que había sido emitido.
- Emisión y publicación de CRL's (CRL - lista de certificados revocados).
- Renovación de certificados.
- Publicar certificados en el directorio repositorio de certificados.

La emisión de certificados y la creación de claves privadas para firmas digitales acostumbra a depender de una pluralidad de entidades que están jerarquizadas de una manera que las de nivel inferior obtienen su capacidad de certificación de otras entidades de nivel superior. Finalmente, en la cúspide de la pirámide suele hallarse una autoridad certificadora, que puede pertenecer al Estado.

Los certificados indican la autoridad certificadora que lo ha emitido, identifican al firmante del mensaje o transacción, contienen la clave pública del firmante, y contienen a su vez la firma digital de la autoridad certificadora que lo ha emitido. De esta manera, las partes que intervienen en una transacción aportan como credencial los certificados de su correspondiente entidad certificadora. Para llegar a ser una entidad certificadora deberá mediar una solicitud a una autoridad certificadora de nivel superior, que podrá denegar la licencia si el solicitante no ofrece la fiabilidad o los conocimientos necesarios, ni cumple los requisitos establecidos en la ley.

USO DE PKI EN APLICACIONES

Las aplicaciones pueden incorporar PKI de múltiples formas, donde lo más difícil es reconstruir la aplicación desde el comienzo y usar funciones criptográficas de bajo nivel para implantar PKI. Por fortuna, la implantación de PKI no tiene que ser tan difícil.

SERVICIOS BASADOS EN PKI

Los servicios basados en PKI son funciones reutilizables que suministran funciones PKI de uso común. Los servicios que aquí se mencionan incluyen firma digital, autenticación, registros de hora seguros, servicio notarial seguro y un servicio de aceptación.

FIRMA DIGITAL

Hemos tratado ampliamente lo referente a las firmas digitales en apartados anteriores. En esta sección lo que haremos será simplemente una mención a cómo robustecer y utilizar esta herramienta de una manera más amplia.

Basándonos en las características que distinguen a la firma digital, podemos imaginar un esquema en el cual se le de el mismo valor e importancia la firma digital como lo tiene ahora la firma autógrafa.
Un servicio PKI de firma digital se conforma de dos partes: Un servicio de generación de firmas y Un servicio de validación de firmas.

El primero de ellos requiere acceso a la clave privada del firmante; como dicha clave lo identifica, ella es sensible y debe estar protegida. Si alguien la roba puede firmar y hacerse pasar por el verdadero propietario. Por consiguiente, un esquema de firmas suele ser parte de una aplicación segura que cuenta con acceso protegido a la clave de la firma. En contraste, un servicio de verificación puede ser más abierto. Por lo general, las claves públicas, una vez que las firma una entidad firmante de confianza, se consideran de conocimiento público. El servicio de verificación recibe los datos firmados, la firma y la clave pública, o el certificado de clave pública y después verifica si la firma da validez a los datos suministrados. A cambio, devuelve una indicación de éxito o fracaso de la verificación.

AUTENTICACIÓN

Un servicio de autenticación PKI usa firmas digitales para establecer la identidad. En la mayor parte de los servicios de autenticación de PKI, el proceso básico es presentar la entidad que se va a autenticar con una parte de los datos de preguntas aleatorias. La entidad debe entonces, firmar o cifrar la pregunta con su clave privada, dependiendo del tipo (o tipos) de uso de su clave. Si el interrogador puede verificar la firma o descifrar los datos con la clave pública en el certificado de la entidad, esta última queda autenticada.

REGISTRO DE HORA

Un servicio de registro de hora ofrece pruebas de que un conjunto de datos existió en una hora especificada. Los registros de hora seguros se pueden usar para establecer cuándo ocurrió una acción electrónica, como en el caso de una transacción o la firma de un documento. Esto tiene particular utilidad si la acción tiene consecuencias legales o financieras.

En general un servicio de registro de hora sigue un modelo de solicitud y respuesta rápida. La entidad que quiere un registro de hora seguro envía una solicitud al servicio correspondiente. La solicitud contiene un hash de los datos que deben llevar el registro de hora; después, el servicio respectivo obtiene una lectura de la hora de su reloj y firma la concatenación del hash de datos y la hora, con la clave privada del servicio de tiempo. Para que el servicio tenga algún valor, su reloj debe de ser muy preciso, como un reloj atómico. Dado que el servicio de registro de hora solamente requiere un hash de los datos y no los datos mismos, el servicio puede ser totalmente anónimo.

SERVICIO NOTARIAL SEGURO

Un servicio notarial seguro se modela de acuerdo con el de una notaría física, la firma del notario constituye una declaración de que un testigo imparcial observó el acto de firma de un documento. Por entrenamiento, un notario humano cumple tres funciones básicas:

- Identificación positiva del firmante
- Determinación de la voluntad del firmante para firmar (es decir, que no está siendo obligado).
- Una evaluación de la conciencia del firmante de las consecuencias de firmar.

Históricamente, para esto se ha exigido que las personas que deben cumplir con una acción notarial deben estar físicamente presentes, junto con sus documentos, ante el notario.

En algunos países (España por ejemplo y algunas regiones de Estado Unidos), las firmas digitales se pueden usar como el equivalente de la firma física de un notario.

En la práctica, las notarías electrónicas se están interpretando en diferentes formas. Una de ellas es muy similar a un servicio seguro de hora, excepto en que el notario mantiene un control de la acción registrada por la hora, incluyendo el hash presentado, el registro seguro de hora resultante y la información acerca del solicitante.

ACEPTACIÓN

Un servicio de aceptación ofrece evidencia innegable de una interacción entre dos partes. En oposición a la autenticación, el enfoque de la aceptación está en una acción y en verificar lo que las dos partes intentaron e hicieron de hecho para participar en la acción. A un usuario de la banca por Internet, por ejemplo, le agradaría tener un servicio de aceptación cuando se transfieren fondos de su cuenta corriente a la cuenta de un comerciante para el pago de una factura. Al usuario le gustaría asegurarse de que el banco no puede negar que la transferencia ocurrió, en caso de que el comerciante reclame más tarde que el pago no se hizo. De manera similar, al banco de Internet le gustaría asegurarse que el usuario no pueda negar que se hizo la transferencia. La aceptación es un término que, con frecuencia, se une al mundo de PKI, desafortunadamente sin un entendimiento completo de sus implicaciones. Por ejemplo, una firma digital por si sola no suele suministrar suficiente evidencia para soportar aceptación.

Los principales estándares que abordan la aceptación proceden de la Organización Internacional de Estandarización, ISO (*International Organization of Standarization*). Los estándares ISO que cubren el tema de la aceptación son la serie X-400 y la serio X-800.

PROTOCOLOS BASADOS EN PKI

El nivel de socket (conector) seguro es, a simple vista, el protocolo más ampliamente conocido y adoptado que se basa en PKI. Sin embargo, antes de que existiera SSL, existía Diffie Hellman. Recientemente han ganado popularidad los protocolos IPSec y S/MIME. En la siguiente sección nos dedicaremos a estudiar más a fondo el protocolo SSL y su evolución a TLS, sin dejar de hacer mención a IPSec, S/MIME y al precursor Diffie Hellman.

Intercambio de Claves Diffie-Hellman

En 1976 Whitfield Diffie y Martín Hellman publicaron el algoritmo Diffie-Hellman, el primero de clave pública, el cual permitía que dos partes computaran un secreto compartido. Su uso es muy amplio, debido a su característica de venta principal: no necesita cifrado, por lo que sus costos de implementación son bajos. Uno de los usos más comunes de un secreto compartido Diffie-Hellman es como base de claves de cifrado adicionales que las dos partes usarán para proteger la información que intercambian.

El concepto fundamental subyacente del algoritmo Diffie-Hellman es la dificultad matemática de calcular logaritmos discretos en un campo finito. Una deficiencia del algoritmo de Diffie-Hellman es que es vulnerable a los ataques de hombre en medio, esto es, que un tercero puede interceptar la comunicación entre los dos extremos del canal de comunicación y manipular los intercambios, de modo que los participantes originales aún creen que se están comunicando directamente.

Secure Socket Layer (SSL)

Como ya hemos vistos, toda transacción segura por la red debe contemplar los aspectos de Autenticación, Integridad, Confidencialidad y No Repudio. Son varios los sistemas y tecnologías que se han desarrollado para intentar implementar estos aspectos en las transacciones electrónicas,

siendo sin duda SSL el más conocido y usado en la actualidad. SSL permite la Confidencialidad y la Autenticación en las transacciones por Internet, siendo usado principalmente en aquellas transacciones en la que se intercambian datos sensibles, como números de tarjetas de crédito o contraseñas de acceso a sistemas privados. SSL es una de las formas base para la implementación de soluciones PKI (Infraestructuras de Clave Pública).

Secure Socket Layer es un sistema de protocolos de carácter general diseñado en 1994 por la empresa Nestcape Communcations Corporation, y está basado en la aplicación conjunta de Criptografía Simétrica, Criptografía Asimétrica (de llave pública), certificados digitales y firmas digitales para conseguir un canal o medio seguro de comunicación a través de Internet. De los sistemas criptográficos simétricos, motor principal de la encripción de datos transferidos en la comunicación, se aprovecha la rapidez de operación, mientras que los sistemas asimétricos se usan para el intercambio seguro de las claves simétricas, consiguiendo con ello resolver el problema de la Confidencialidad en la transmisión de datos. Así mismo se ofrece privacidad en las comunicaciones por medio de los *códigos de autenticación de mensajes (Message Autentication Codes,* MAC's). SSL implementa un protocolo de negociación para establecer una comunicación segura a nivel de socket (nombre de máquina más puerto), de forma transparente al usuario y a las aplicaciones que lo usan.

Actualmente es el estándar de comunicación segura en los navegadores web más importantes (protocolo HTTP), como Nestcape Navigator e Internet Explorer, y se espera que pronto se saquen versiones para otras otros protocolos de la capa de Aplicación (correo, FTP, etc.).
La identidad del servidor web seguro (y a veces también del usuario cliente) se consigue mediante el Certificado Digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles (Autenticación), mientras que de la seguridad de Integridad de los datos intercambiados se encarga la Firma Digital mediante funciones hash y la comprobación de resúmenes de todos los datos enviados y recibidos.

Desde el punto de vista de su implementación en los modelos de referencia OSI (*Open System Interconection*) y TCP/IP (*Transmision Control Protocol / Internet Protocol*), SSL es un protocolo de dos capas que opera sobre un protocolo de transporte confiable, usualmente TCP., y la capa de protocolos de aplicación, por ejemplo http, (ver Figura 1.17); sustituyendo los sockets del sistema operativo, lo que hace que sea independiente de la aplicación que lo utilice, y se implementa generalmente en el puerto 443.

Figura 1.17
Pila de Protocolo SSL

SSL proporciona servicios de seguridad a la pila de protocolos, encriptando los datos salientes de la capa de aplicación antes de que estos sean segmentados en la capa de Transporte y encapsulados y enviados por las capas inferiores. Es más, también puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 214 bytes, volviéndolos a reensamblar en el receptor.

Resumen del Protocolo

Una vez que una sesión SSL ha sido establecida esta puede ser reutilizada, con esto se evitan problemas en el rendimiento al seguir nuevamente los pasos en el establecimiento de una nueva sesión.

Establecimiento de una Sesión

La sesión SSL se establece siguiendo una secuencia de negociación de acuerdos (handshake) entre el cliente y el servidor. Esta secuencia puede variar, dependiendo si el servidor está configurado para proporcionar un certificado de servidor o administración de la información de cifrado.
Aquí resumiremos un escenario común, si se desean conocer otras posibilidades que se ofrecen, se anexa en la parte final de este trabajo la especificación completa del protocolo SSL de la IETF.
La secuencia de negociación de acuerdo (Handshake) es utilizada por el cliente y el servidor para:

- Negociar el conjunto de algoritmos de cifrado que se utilizarán durante la transferencia de los datos.
- Establecer y compartir una llave de sesión entre el cliente y el servidor.
- Opcionalmente autenticar al servidor ante el cliente.
- Opcionalmente autenticar al cliente ante el servidor.

Este procedimiento se ilustra de manera simplificada en la figura 1.18

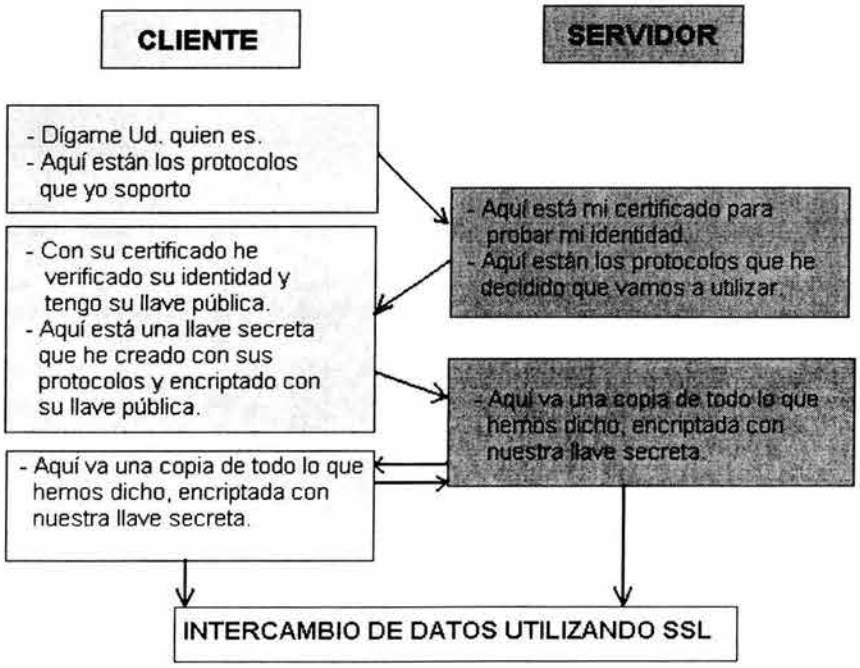


Figura 1.18
Secuencia de Handshake simplificada

La fase de negociación del conjunto de algoritmos de cifrado permite al cliente y al servidor seleccionar el conjunto de protocolos que ambos puedan soportar. El protocolo SSL V3 define 31 conjuntos de algoritmos de cifrado. Un conjunto de algoritmos de cifrado (también conocido como asociación de seguridad o "security association")consiste de:

1- Un método de intercambio de llave
2- Cifrado que se utilizará para la transferencia de datos

3- Mensajes de resumen o digestión para la creación del códigos de autenticación de mensaje (*Message Autentication Code, MAC*)

El método de intercambio de llave define cómo se acordará la llave criptográfica simétrica que compartirán, que será utilizada para la transferencia de datos entre el cliente y el servidor. SSL V2 utiliza intercambio de llaves RSA, SSL V3 soporta una selección del algoritmo de intercambio de llave, incluyendo el intercambio de llave RSA cuando se utilizan certificados; e intercambio de llave Diffie-Hellman cuando no se usan certificados y cuando no existe prioridad de comunicación entre el servidor y el cliente.

La selección del método de intercambio de llave incluye el si usarán o no firmas digitales con el intercambio de llave, así como el tipo de firmas a utilizar. Firmar con una llave privada proporciona seguridad contra un ataque de hombre en medio durante el intercambio de información utilizada en la generación de la llave compartida.

La versión más reciente de SSL es la 3.0. que usa los algoritmos simétricos de encripción DES, TRIPLE DES, RC2, RC4[30] e IDEA[31], el asimétrico RSA, la función hash MD5 y el algoritmo de firma SHA-1 (Secure Hash Algorithm v1). SHA fue diseñado para su uso en los estándares de firma digital (Digital Signature Standar, DSS).

Los algoritmos, longitudes de clave y funciones hash de resumen usados en SSL dependen del nivel de seguridad que se busque o se permita, siendo los más habituales los siguientes:

RSA + Triple DES de 168 bits + SHA-1: soportado por las versiones 2.0 y 3.0 de SSL, es uno de los conjuntos más fuertes en cuanto a seguridad, ya que son posibles 3.7E50 claves simétricas diferentes, por lo que es muy difícil de romper. Por ahora sólo está permitido su uso en Estados Unidos, aplicándose sobre todo en transacciones bancarias.
RSA + RC4 de 128 bits + MD5: soportado por las versiones 2.0 y 3.0 de SSL, permite 3.4E38 claves simétricas diferentes que, aunque es un número inferior que el del caso anterior, da la misma fortaleza al sistema. Análogamente, en teoría sólo se permite su uso comercial en Estados Unidos, aunque actualmente ya es posible su implementación en los navegadores más comunes, siendo usado por organismos gubernamentales, grandes empresas y entidades bancarias.
RSA + RC2 de 128 bits + MD5: soportado sólo por SSL 2.0, permite 3.4E38 claves simétricas diferentes, y es de fortaleza similar a los anteriores, aunque es más lento a la hora de operar. Sólo se permite su uso comercial en Estados Unidos, aunque actualmente ya es posible su implementación en los navegadores más comunes.
RSA + DES de 56 bits + SHA-1: soportado por las versiones 2.0 y 3.0 de SSL, aunque es el caso de la versión 2.0 se suele usar MD5 en vez de SHA-1. Es un sistema menos seguro que los anteriores, permitiendo 7.2E16 claves simétricas diferentes, y es el que suelen traer por defecto los navegadores web en la actualidad (en realidad son 48 bits para clave y 8 para comprobación de errores).
RSA + RC4 de 40 bits + MD5: soportado por las versiones 2.0 y 3.0 de SSL, ha sido el sistema más común permitido para exportaciones fuera de Estados Unidos. Permite aproximadamente 1.1E12 claves simétricas diferentes, y una velocidad de proceso muy elevada, aunque su seguridad es ya cuestionable con las técnicas de criptoanálisis actuales.
RSA + RC2 de 40 bits + MD5: en todo análogo al sistema anterior, aunque de velocidad de proceso es bastante inferior.
Sólo MD5: usado solamente para autenticar mensajes y descubrir ataques a la integridad de los mismos. Se usa cuando el navegador cliente y el servidor no tienen ningún sistema SSL común, lo que hace imposible el establecimiento de una comunicación cifrada. No es soportado por SSL 2.0, pero sí por la versión 3.0.

[30] RC2 y RC4 son algoritmos de llave simétrica propiedad de RSA
[31] IDEA es un o de los mejores algoritmos disponibles y criptográficamente es el más fuerte

La clave de encripción simétrica es única y diferente para cada sesión, por lo que si la comunicación falla se debe establecer una nueva sesión SSL, y la contraseña simétrica se generará de nuevo.

La secuencia de Handshake utiliza tres protocolos , el Protocolo SSL de Acuerdo (SSL Handshake Protocol) para establecer la sesión SSL entre el cliente y el servidor, el Protocolo SSL de Especificación de cambio de cifrado (SSL Change Cipher Spec Protocol), para indicar un cambio en las cifras usadas; consta de un mensaje único que está cifrado con la especificación de cifrado corriente o actual; y el Protocolo SSL de Alerta (SSL Alert Protocol), este protocolo transfiere mensajes acerca de un evento, incluidas la severidad y una descripción del mismo. Estos eventos son, básicamente, condiciones de error, como un MAC no válido, o un certificado que ha expirado o algún parámetro ilegal. El protocolo de Alerta también se usa para compartir información acerca de una terminación de conexión planeada.

Estos protocolos se encuentran encapsulados en el Protocolo SSL de Registro (SSL Record Protocol), al igual que la información del protocolo de aplicación. Ver figura 1.17

Transferencia de Información

El protocolo SSL de registro (SSL Record Protocol) es utilizado para transferir información de aplicación así como de control de SSL entre el cliente y el servidor, posiblemente fragmentando esta información en unidades más pequeñas, como se mencionó anteriormente.
Esto se ilustra en la siguiente figura (Figura 1.19):

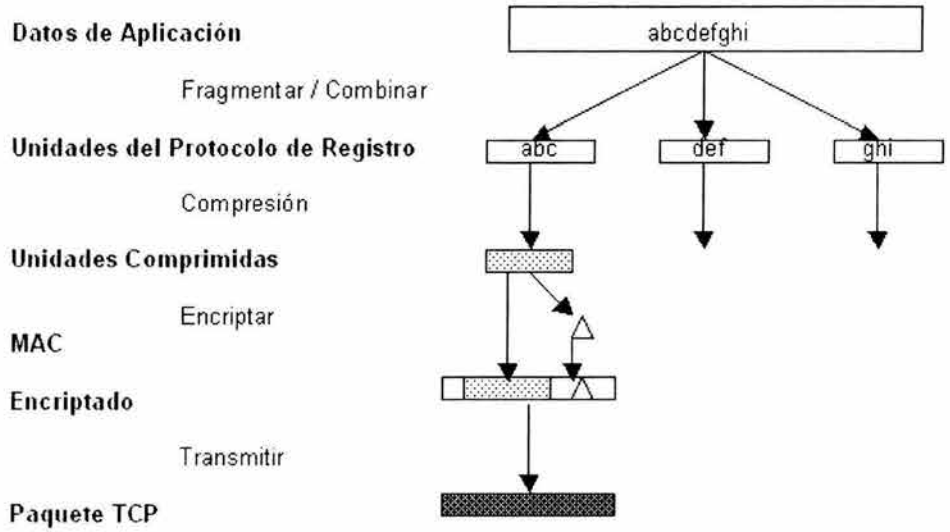


Figura 1.19
Protocolo SSL de Registro

SSL proporciona cifrado de alto nivel de los datos intercambiados (se cifran incluso las cabeceras HTTP), autenticación del servidor (y si es necesario también del cliente) e integridad de los datos recibidos.

Durante el proceso de comunicación segura SSL existen dos estados fundamentales, el estado de sesión y el estado de conexión. A cada sesión se le asigna un número identificador arbitrario, elegido por el servidor, un método de compresión de datos, una serie de algoritmos de encripción y funciones hash, una clave secreta maestra de 48 bytes y una bandera (flag) de nuevas conexiones, que indica si

desde la sesión actual se pueden establecer nuevas conexiones. Cada conexión incluye un número secreto para el cliente y otro para el servidor, usados para calcular los MAC's de sus mensajes, una clave secreta de encripción particular para el cliente y otra para el servidor, unos vectores iniciales en el caso de cifrado de datos en bloque y unos números de secuencia asociados a cada mensaje.

Un esquema ilustrativo del procedimiento completo de una conexión utilizando SSL se muestra en la siguiente figura (figura 1.20):

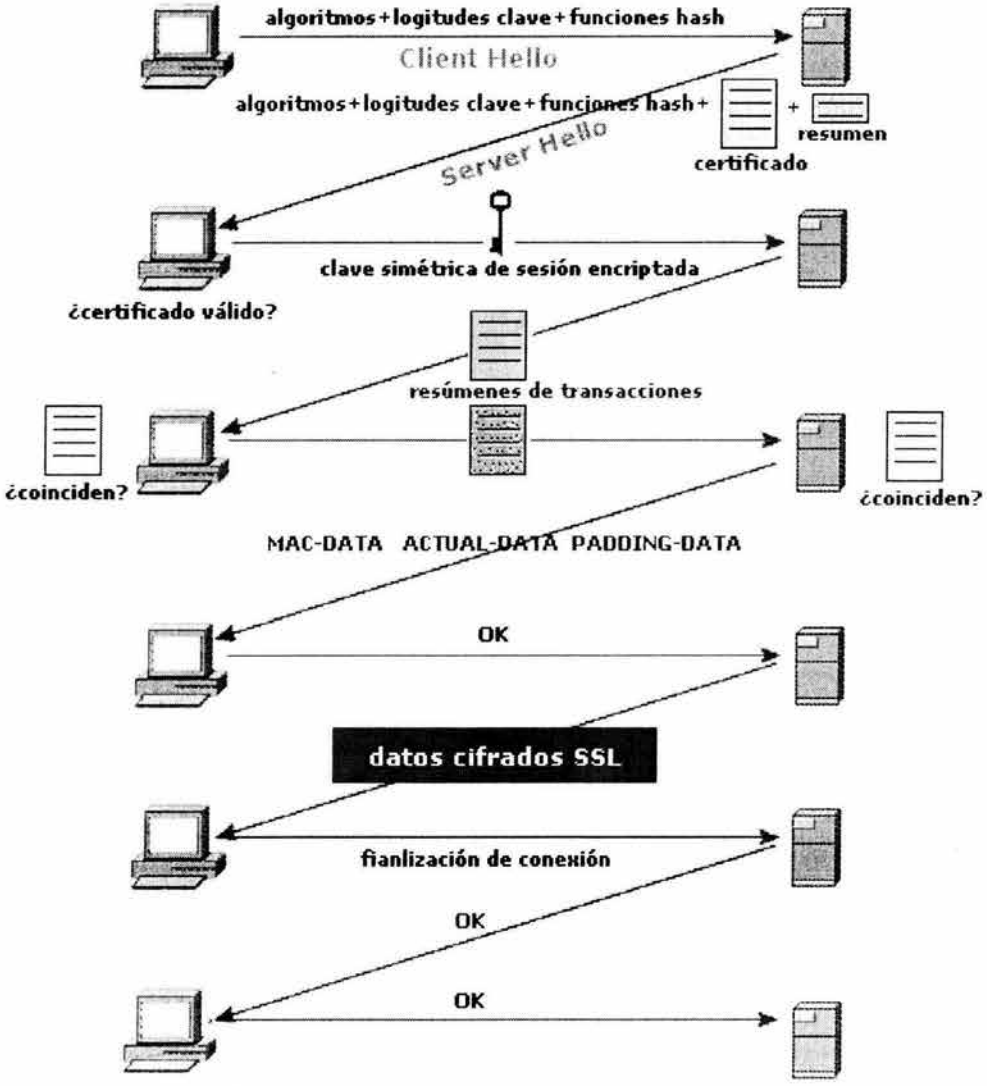


Figura 1.20
Procedimiento completo de una conexión SSL

¿Cómo podemos saber si una conexión se está realizando mediante SSL?. Generalmente los navegadores disponen de un icono que lo indica, generalmente un candado en la parte inferior de la ventana. Si el candado está abierto se trata de una conexión normal, y si está cerrado de una conexión segura. (ver Figura 1.21)

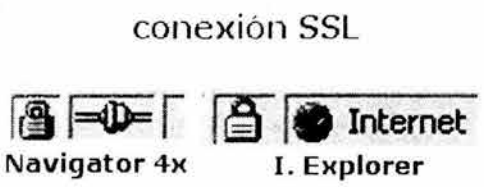


Figura 1.21
Indicadores de conexión utilizando SSL

Si hacemos doble click sobre el candado cerrado nos aparecerá el Certificado Digital del servidor web seguro.

Además, las páginas que proceden de un servidor SSL vienen implementadas mediante protocolo HTTP seguro, por lo que su dirección, que veremos en la barra de direcciones del navegador, empezará siempre por https, como por ejemplo:

https://www.htmlweb.net ó https://raguel..imp.mx

Implementación de SSL

Por la parte del cliente, SSL viene implementado por defecto en los navegadores Internet Explorer y Nestcape Navigator, lo que permite a cualquier usuario con uno de estos navegadores poder realizar transacciones por Internet de forma segura sin tener que conocer el sistema a fondo ni preocuparse de instalar programas adicionales (por lo menos autenticando al servidor web y con confidencialidad e integridad asegurada en la transacción).

La implementación en la parte servidora es un poco más compleja. En primer lugar, es obligatoria la obtención de un Certificado Digital para el servidor seguro, solicitándolo a una Autoridad Certificadora de prestigio reconocido o a la propia. Ya con el servidor certificado, el usuario podrá realizar transacciones con él.

Existen en la actualidad diferentes versiones del conjunto de protocolos SSL que se pueden implementar en los distintos servidores y que corren bajo los sistemas operativos más comunes (IIS en Windows NT-2000-XP, Apache en Unix/Linux, etc.).

Ventajas e inconvenientes de SSL

La tecnología basada en los protocolos Secure Socket Layer proporcionó grandes avances en la implantación de sistemas de comunicación seguros, que han hecho posible un crecimiento importante en las transacciones por Internet. Si estudiamos SSL desde el punto de vista de las bases necesarias para considerar una comunicación segura podemos sacar las siguientes conclusiones:

1. Autenticidad: SSL requiere para su funcionamiento la identificación del servidor web ante el cliente y la realiza adecuadamente, pero normalmente no se produce una identificación en sentido contrario. Es decir, no es obligada en la mayoría de los casos la presencia del certificado del usuario que se está conectando al servidor.

2. Confidencialidad: SSL proporciona una buena seguridad de que los datos no van a ser capturados por extraños de forma útil en el proceso de transferencia de los mismos, pero no proporciona ninguna seguridad después de finalizar la conexión.

3. Integridad: ocurre algo parecido a lo anterior. En el corto proceso que dura el envío de datos si podemos estar seguros de que éstos no van a ser modificados, puesto que SSL lo impide. Pero una vez que finaliza la conexión segura no podemos estar tranquilos.

4. No Repudio: en este aspecto SSL por si solo falla al máximo, ya que no hay por defecto establecido ningún método para dejar constancia de cuándo se ha realizado una operación, cuál ha sido y quiénes han intervenido en ella. SSL no proporciona formas de emitir recibos válidos que identifiquen una transacción.

Vemos pues que SSL, por si solo, carece de muchos de los elementos necesarios para construir un sistema de transacciones seguras usando Internet. Para tratar de cubrir estos fallos se han intentado sacar al mercado y estandarizar otros sistemas diferentes, como SET, pero el caso es que hasta ahora ninguno de ellos ha conseguido desplazar a SSL. ¿Por qué? Tal vez sea porque, a pesar de sus fallos, SSL es una tecnología rápida, fácil de implementar, barata y cómoda para el usuario, que no tiene que conocer cómo funciona, tan sólo usarla. Y desde el punto de vista del servidor o de la empresa que le facilita el hospedaje (hosting), SSL es igualmente sencillo de implementar, no precisando de servidores con características especiales.

Transport Layer Security (TLS)

El protocolo de seguridad en la capa de transporte TLS es descendiente directo de SSL v3. Este protocolo en su primera versión se definió por primera vez por la IETF en el RFC 2246 en 1999. Actualmente la última revisión se encuentra en RFC 3546 y en el draft 2246 de IETF publicados en el 2003. A continuación citamos a grandes rasgos sus principales características y modo de operación:

Objetivos Generales:

El protocolo TLS fue diseñado para evitar que las aplicaciones cliente/servidor:

- Sean espiadas.
- Sufran Intromisiones.
- Sean Falsificadas

Fases de TLS:

Al igual que SSL, TLS se compone de las siguientes fases:

- Handshake.
- Transmisión de datos de la aplicación.

Handshake negocia:

- Algoritmos de cifrado.
- Claves.
- Autenticación del servidor y cliente.

Transmisión de datos :

- Una vez completado el handshake, se inicia la transmisión de los datos de la aplicación.
- Se cifran todos los datos utilizando las claves de la sesión negociadas durante el handshake.

Restricciones de exportación:

Para versiones fuera de Estados Unidos se exporta una versión de SSL/TLS que utiliza claves de 40 bits, en Estados Unidos se comercializa una versión de 128 bits.

Modelo de capas de TLS:

El protocolo está compuesto de 2 capas:

- TLS record protocol
- TLS handshake protocol

En su nivel más bajo, TLS Record Protocol se soporta sobre TCP.

Figura 1.22
Capas de TLS

Privacidad:

- Se usa criptografía simétrica para cifrado de datos (p. ejemplo.: DES, RC4, etc.)
- Las claves para este cifrado simétrico se generan de forma única para cada conexión.
- Se basa en un secreto negociado por otro protocolo (e.g.: TLS Handshake Protocol)
- Se puede utilizar sin cifrado.

Integridad:

- El transporte del mensaje incluye verificación de integridad del mensaje utilizando código de autenticación del mensaje con clave.
- El cálculo del código de autenticación del mensaje MAC se realiza mediante funciones de hash (e.g.: SHA, MD5, etc.)
- Puede operar sin un MAC, pero solo se usa en esta modalidad mientras otro protocolo esté utilizando el protocolo de registro como protocolo de transporte para negociar parámetros de seguridad.

Se puede utilizar TLS Record Protocol para encapsular varios protocolos de nivel más alto.
TLS Handshake Protocol:

- Permite autenticar el servidor y el cliente.

- Negocia el algoritmo cifrador y claves criptográficas antes que el protocolo de aplicación transmita o reciba su primer byte de datos.

Propiedades del Protocolo de Handshake:

- La identidad del punto se puede autenticar usando criptografía asimétrica o clave pública (e.g.: RSA, DSS, etc.) Se requiere para al menos uno de los extremos.
- La negociación del secreto compartido es segura: el secreto negociado no esta disponible para espías, y tampoco se puede obtener el secreto con cualquier conexión auntenticada, aún por un atacante que pueda ubicarse en el medio de la conexión.
- La negociación es confiable: ningún atacante puede modificar la negociación de la comunicación sin ser detectado por las partes comunicantes.

Ventajas:

- TLS es independiente del protocolo de aplicación.
- Los protocolos de alto nivel pueden colocarse transparentemente sobre la capa TLS.
- Las decisiones de iniciar TLS handshaking y como interpretar los certificados de autenticidad se dejan a juicio de los diseñadores e instaladores de los protocolos que funcionan sobre TLS.

Diferencias con SSL versión 3

- TLS se basa en el desarrollo previo de SSL versión 3.0.
- Son muy similares pero no interoperan.
- TLS se identifica como una nueva versión de SSL 3.1.

HMAC y PRF

HMAC se denomina a la operación de construcción del código de autenticación del mensaje MAC, en el handshake con dos algoritmos diferentes: MD5 y SHA-1.
PRF (Pseudo Random Function) toma como entrada un secreto, una semilla, y una etiqueta de identificación y produce una salida de longitud arbitraria. La función PRF de TLS se crea dividiendo el secreto en dos mitades, una mitad para generar datos con P_MD5 y la otra mitad para generar datos con P_SHA-1, luego realiza la operación XOR con las salidas de estas dos operaciones de expansión juntas.

Protocolo de Registro de TLS (TLS Record Protocol)

Es un protocolo de capas. En cada capa los mensajes incluyen los campos:

- Longitud.
- Descripción.
- Contenido.

Funciones de la capa de registro:

- Toma los mensajes a ser transmitidos.
- Fragmenta los datos en bloques manejables.
- Opcionalmente comprime los datos.
- Aplica un MAC.
- Cifra.
- Transmite el resultado.

Los datos recibidos son:

- Descifrados.
- Verificados.
- Descomprimidos.
- Reensamblados y
- Entregados a aplicaciones clientes de alto nivel.

Clientes del protocolo de registro (4):

- El protocolo de handshake.
- El protocolo de alerta.
- El protocolo de cambio de especificación de cifrador.
- y el protocolo de datos de la aplicación.

Propiedades de la capa de registro:

- Soporta tipos adicionales.
- Ignora los tipos no soportados.
- Requiere diseño cuidadoso contra ataques para cualquier protocolo a ser utilizado sobre TLS.

Estados de Conexión de la capa de registro

El estado de conexión TLS es el ambiente operativo del protocolo de registro de TLS.
Especifica un algoritmo de compresión, algoritmo cifrador, y algoritmo de código de autenticación del mensaje.

Los parámetros de estos algoritmos son:

- El secreto MAC.
- Las claves de cifrado de bloques
- Los valores de identificación de la conexión en las direcciones de lectura y escritura.

Existen siempre 4 estados lógicos de conexión importantes:

- Los estados activos de lectura y escritura.
- Los estados pendientes de lectura y escritura.

El protocolo de handshake puede activar selectivamente cualquiera de los estados pendientes, en este caso, el estado activo se reemplaza por el estado pendiente, y el estado pendiente se libera y reinicializa como vacío.

No es lícito crear un estado activo que no haya sido inicializado con los parámetros de seguridad correspondientes.

El estado inicial activo especifica que no utilizará cifrado, compresión o MAC.

Los parámetros de seguridad de cualquier estado son:

- Conexión terminal.
- La entidad puede considerarse parte "cliente" o "servidora" de la conexión.
- Algoritmo cifrador de bloque.

En esta especificación se incluye el algoritmo utilizado para realizar el cifrado de bloque:

- El tamaño de la clave del algoritmo.
- Parte de la clave es secreta si se utiliza cifrado de bloque o de trama.
- El tamaño del bloque de cifrado (opcional) y si se trata de un cifrado exportable.

Capa de registro, Componentes:

- Algoritmo de código de autenticación de mensajes MAC.
- Algoritmo de compresión.
- Secreto maestro.(48bytes compartido).
- Número aleatorio del cliente (32bytes).
- Número aleatorio del servidor (32bytes)

La capa de registro utilizará los parámetros de seguridad para generar los siguientes 6 pasos:

- Secreto MAC de escritura del cliente.
- Secreto MAC de escritura del servidor.
- Clave de escritura del cliente.
- Clave de escritura del servidor.
- Vector de inicialización IV de escritura del cliente (para cifradores de bloque solamente).
- Vector de inicialización IV de escritura del servidor (para cifradores de bloque solamente).

Protocolo de cambio de la especificación del cifrador

- Sirve para señalizar las transiciones entre estrategias de cifrado.
- El protocolo consiste de un solo mensaje, el que es cifrado y comprimido bajo el estado de conexión activo (no el pendiente) El mensaje consiste de un solo byte de valor 1.
- Se envía tanto al cliente como al servidor para notificar a la parte receptora que los registros subsiguientes serán protegidos bajo la nueva especificación y las nuevas claves de cifrado.

Protocolo de Alerta

Los mensajes de alerta expresan la severidad del mensaje y una descripción de la alerta.

Niveles de alerta:

- Fatal: inmediata finalización de la conexión
- A discreción: a discreción del implementador
- Advertencia: puede continuar pero se invalidad el identificador

Alerta de cierre

Cada parte debe intercambiar de mensajes notificación de cierre para evitar ataque por truncamiento.

Alertas de Error

Cuando se detecta un error, la parte que lo detecta envía un mensaje a la otra parte.

Hasta la transmisión o recibo de un mensaje de alerta fatal, ambas partes cierran inmediatamente la conexión.

Se requiere que Servidores y clientes olviden cualquier identificador de sesión, claves, y secretos asociados con una conexión fallida.

Protocolo de Handshake

Diálogos entre las partes (Figura 1.23)

Figura 1.23 - Flujo del mensaje para un handshake completo.

*Indica mensajes opcionales o dependientes de la situación y que por tanto no siempre se envian.

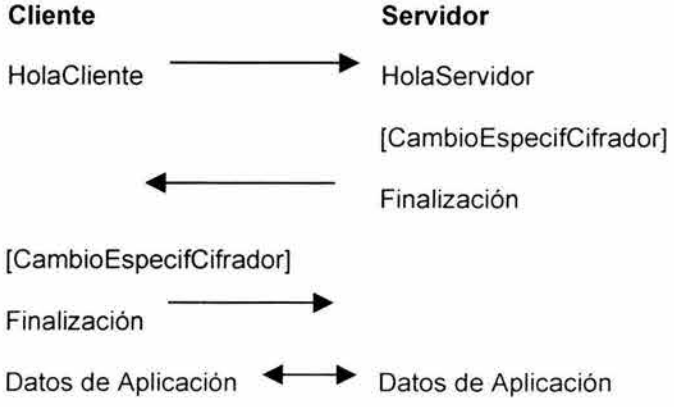


Figura 1.24 - Flujo de mensajes para un handshake abreviado.

Cálculos Criptográficos

TLS requiere que se especifiquen:

- Un conjunto de algoritmos.
- Un secreto maestro.
- Los valores aleatorios del cliente y del servidor.

Se determinan por el conjunto de cifradores seleccionado por el servidor y revelado en el mensaje "Hola" del servidor:

- La autenticación.
- Cifrado.
- y algoritmos MAC.

El algoritmo de compresión se negocia en los mensajes "Hola".

Los valores aleatorios se intercambian en los mensajes "Hola".

Todo lo que falta es calcular el secreto maestro.

Cálculo del secreto maestro

Se utiliza para convertir el secreto pre-maestro en secreto maestro.

El secreto pre-maestro se debe eliminar de memoria una vez el secreto maestro se haya calculado.

De 48 bytes del longitud. La longitud de el secreto pre-maestro variará dependiendo en el método de intercambio de claves.

RSA

Un secreto pre-maestro de 48 bytes lo genera el cliente, cifrado bajo la clave pública del servidor, y enviado al servidor. El servidor utiliza su clave privada para descifrar el secreto pre-maestro. Ambas partes luego convierten el secreto pre-maestro en el secreto maestro, como se especifica arriba.

Las firmas digitales RSA se realizan utilizando PKCS #1 bloque tipo 1. El cifrado de clave pública RSA se realiza utilizando PKCS#1 bloque tipo 2.

Diffie-Hellman

Se realiza un cálculo Diffie-Hellman convencional. La clave negociada (Z) se utiliza como secreto pre-maestro, y es convertida en el secreto maestro.

Conjuntos Cifradores Obligatorios

En la ausencia de un perfil de aplicación estándar, una aplicación que cumpla con TLS debe implementar el conjunto cifrador TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

Protocolo de Aplicación de Datos

Los mensajes de aplicación de datos son llevados por la capa de registro y son fragmentados, comprimidos y cifrados basados en el estado de la conexión actual. Los mensajes se tratan como datos transparentes a la capa de registro.

Con lo visto hasta este momento, consideramos que hemos adquirido un buen nivel de conocimientos acerca del tema. Para saber más consulte la bibliografía y referencias que se citan al final de esta sección.

Nuestro siguiente trabajo es realizar la propuesta de solución del problema, la cual tratamos en la siguiente sección basándonos en lo estudiado hasta el momento.

BIBLIOGRAFÍA Y REFERENCIAS

The Handbook of Applied Cryptography
Capítulo 1
Information Security and Cryptography,
A. Menezes, P. Van Oorschot, and S. Vanstone, CRC Press, 1996, en formato PDF

PKI, Infraestructura de claves públicas
Andrew Nash, William Duane, Celia Joseph y Derek Brink
RSA PRESS, 2002
Osborne Mc Graw Hill

Firma Digital y Certificados Digitales,
José de Jesús Ángel
SeguriData Documento electrónico.
http://www.htmlweb.net/seguridad/varios/firma_certificados.html

Transacciones Seguras
Luciano Moreno,departamento de diseño web de BJS Software.
http://www.htmlweb.net/seguridad/ssl/ssl_1.html

Digital Signature Guidelines
Legal Infrastructure for Certification Authorities and Secure Electronic Commerce
American Bar Association
Documento electrónico en formato PDF

Infraestructura de Llaves Públicas, PKI
Thierry de Saint Pierre
Director North Supply Chile
Documento Electrónico en formato PDF

PKI Basics – A Technical Perspective
Shashi Kiran –Nortel
Patricia Lareau –PKI Forum
The PKI Forum's Business Working Group
Noviembre 2002
http://www.pkiforum.org

The TLS Protocol Version 1.0,
Request for Comments: RFC 2246,
IETF, Internet Engineering Task Force
http://www.ietf.org

The TLS Protocol Version 1.1
IETF,Internet Draft
RFC 2246 draft
http://www.ietf.org/internet-drafts/draft-ietf-tls-rfc2246-bis-05.txt

The SSL Protocol Version 3
Transport Layer Security Working Group
INTERNET-DRAFT
Netscape Communications
http://wp.netscape.com/eng/ssl3/draft302.txt

Introducing SSL and Certificates
Documento electrónico
http://www.pseudonym.org/ssl/ssl_intro.html

Artículos sobre el protocolo SSL
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia, España
http://www.iti.upv.es/seguridad/ssl.html

Public-Key Infrastructure (X.509) (pkix)
Active IETF Working Groups
Security Area
http://www.ietf.org/html.charters/pkix-charter.html

The Internet Engineering Task Force
Active IETF Working Groups
Security Area
http://www.ietf.org/html.charters/wg-dir.html

Introducción

En la sección 1-1 nos concentramos en el estudio de algunas herramientas que actualmente se tienen en el campo de la criptografía, de las facilidades que se obtienen al utilizar este tipo de recursos, y de cómo ella nos ayuda en el campo de la seguridad informática. De igual manera vimos las vertientes que se siguen así como los caminos por los que nos lleva la tecnología actual y más o menos cual o cuales son las tecnologías que en un futuro se espera puedan desarrollarse y expandirse con mayor fuerza.

En esta sección nos centraremos en conformar la propuesta de solución considerando lo que hemos investigado y estudiado.

En una buena solución:

Sólo las personas adecuadas obtienen acceso en cualquier momento a la información correcta con el mejor rendimiento posible al menor costo posible.

La anterior es una premisa que trataremos de seguir a fin de encontrar, proponer, diseñar e implementar la mejor solución a los problemas que enfrentamos.

1.2 PROPUESTA

Apoyándonos en lo expuesto en la sección anterior y con base en lo planteado en la sección de INTRODUCCIÓN al trabajo de tesis, podemos ver que el sistema a desarrollar dentro del IMP para satisfacer las demandas y alcanzar las metas que se ha propuesto, deberá integrar varios módulos, dentro de los cuales a simple vista y como una primera aproximación a una solución podemos considerar los siguientes:

- Entidad Certificadora
  Este módulo sería el encargado de generar, almacenar y revocar los certificados digitales.
- Módulo de autenticación maestro de IMP
  Este módulo permitirá que los servidores del IMP pudieran añadir las facilidades de autenticación a las facilidades que ofrecen.
- Servicio de directorio
  Este módulo permitiría el almacenamiento de todos aquellos datos que concierne a los usuarios y recursos que utilizan. Entre los datos de los usuarios distinguiríamos notablemente el nombre de usuario, su certificado digital, entidad que lo certifica, etc.

Al igual que para el servicio de resolución de nombres de dominio en Internet (DNS), podríamos pensar en un servicio de autenticación mundial estructurado de manera jerárquica, de tal manera que en el nivel más alto de esta estructura encontráramos servidores mundiales de autenticación, los cuales autenticarían servidores de autenticación de menor nivel (nacionales por ejemplo) y así sucesivamente descender sobre esta estructura de servidores hasta tener los servidores de autenticación empresariales o locales donde estarían registrados los datos y firmas digitales de su personal, como se pretende implementar en el IMP.

Con base en lo estudiado hasta este momento, tenemos bases suficientes para vislumbrar cual sería una buena solución.

Un esquema de Autenticación basado en certificados digitales por si mismo, trabajando de manera independiente y aislado de otras aplicaciones, no representa una solución óptima e integral a los problemas que queremos resolver. Podríamos diseñar e implementar el sistema como tal, sin embargo el trabajo requerido tendría que incluir el estudio minucioso de su impacto, interacción,

adaptabilidad, funcionalidad, escalabilidad y rendimiento de dicho esquema con los sistemas, aplicaciones y herramientas que actualmente se utilizan y se aplican tanto a nivel interno como a nivel externo de la red del IMP. Este trabajo podría ser más productivo y útil si ampliáramos nuestro campo de estudio y aplicación a una solución, o soluciones, que además de cubrir este punto nos ofrecieran facilidades y beneficios extras sin perder de vista ni restarle importancia al objetivo inicial de este trabajo.

De igual manera el resultado de todo este trabajo podría verse muy limitado o escueto en cuanto a alcances, funcionalidad, rendimiento, pero sobre todo en cuanto al futuro que pudiese tener si es que solo nos concentramos en un esquema de Autenticación basado en certificados.

Es por esta razón, principalmente, por la que proponemos ampliar nuestras expectativas y ambiciones en cuanto a la solución que pensamos concebir, lo cual no quiere decir que el objetivo inicial de este trabajo así como la solución que en un principio visualizamos no sean útiles o no se puedan implementar, sino que es más conveniente en todos los aspectos ampliar nuestra visión de los problemas que enfrentamos así como de las soluciones que para ellos pudiésemos proponer.

Hemos explicado ampliamente el modelo de PKI ya que puede ser esta infraestructura, con los ajustes y adecuaciones necesarias, una candidata ideal para cubrir y superar los objetivos que inicialmente hemos propuesto. Recordemos que una infraestructura de llaves públicas es un diseño estructural para establecer comunicación, mensajería y transacciones seguras sobre las redes, apoyándose para esto principalmente en la criptografía de llaves públicas y los certificados.

Es importante mencionar que, en principio, proponemos la construcción de dicha infraestructura a pequeña escala (solo para el Laboratorio de Tecnologías de la Información, LTI, para atender a una comunidad aproximadamente de 150 usuarios), esto es, un prototipo o "maqueta" de lo que en un futuro consideramos se pueda implementar en todo el IMP.

Para esto es importante hacer las siguientes menciones y anotaciones:

Una PKI incluirá una o varias autoridades de registro para certificar la identidad de los usuarios; una o varias autoridades de certificación que emitan los certificados de clave pública; un repositorio de certificados, accesible vía web u otro medio, donde se almacenen los certificados; las listas de revocación de certificados (CRL), donde se listan los certificados suspendidos o revocados; y, por supuesto, los propios certificados.

Los mayores obstáculos a los que se han enfrentado las empresas pioneras en la implantación de soluciones PKI para sus necesidades de negocio electrónico (e-Business) han sido tradicionalmente:

- La falta de interoperabilidad, ya que el mero hecho de ceñirse al estándar X.509.v3 no garantiza en absoluto que dos certificados generados por dos sistemas desarrollados por casas distintas sean mutuamente compatibles. Además, existen problemas de confianza entre AC's[1] de distintas organizaciones, que puede imposibilitar la verificación con éxito de cadenas de certificación cuya AC raíz sea desconocida o no confiable, invalidándose todo el esquema de PKI.

- El costo ha sido un problema desde el principio. Al no existir un mercado suficientemente maduro en PKI, cada empresa que ofrece soluciones de clave pública ofrece tarifas en función de criterios diversos (por certificado, por uso de certificado, por servidores instalados, etc.) y cobra honorarios también dispares, de manera que la inversión en PKI como respuesta a las necesidades de seguridad y accesibilidad a los activos informáticos de la empresa puede resultar cuando menos inesperadamente elevada.

[1] CA o AC el término aplica al mismo sujeto que en este caso es la Autoridad Certificadora o de Certificación

- PKI termina presentando problemas de escalabilidad, cuando el número de certificados emitidos a los usuarios va creciendo, debido a que las listas de revocación deben ser consultadas en cada operación que involucre certificados y firmas digitales, si se desea una implantación seria y robusta de PKI. Bien es cierto que el esquema de confianza vertical, promulgado por las estructuras de certificación en árbol, resulta más escalable que los modelos de confianza horizontal, como el adoptado por PGP, cuya problemática es tan seria que no se prevé solución satisfactoria.

- Finalmente, la tecnología PKI se le antoja un tanto esotérica al usuario final, que no terminan de entender del todo la jerga relacionada. Acostumbrado a autenticarse sin más que introducir su nombre y contraseña, puede sentirse fácilmente rebasado por la complejidad tecnológica de las firmas digitales y demás funciones criptográficas. En la medida en que no se adopten las tarjetas inteligentes, controles biométricos u otros dispositivos similares criptográficamente robustos, el problema de los usuarios anotando su contraseña (en este caso para acceder a su clave privada) en un post-it pegado en el monitor persistirá por mucho tiempo.

La PKI resulta ideal en una Intranet, en la que se comparten documentos (trabajo en grupo), se accede a recursos de red (cálculo, servidores de archivos, bases de datos, etc.), se intercambia correo certificado entre los empleados, etc. PKI resulta mucho más ágil que los sistemas tradicionales de control basados en nombre de usuario y contraseña, que es el que actualmente se maneja dentro del IMP, y listas de control de acceso.

En el caso de extranets o de Internet, PKI es de uso obligado. De hecho, es la única forma conocida actualmente de prestar confianza a los actores de las transacciones o relaciones electrónicas que no se conocen entre ellos, tanto en el business-to-business entre empresas, como en el comercio al por menor, entre vendedores y compradores particulares por Internet.

La confianza en un grupo de AC mundialmente reconocidas (como Verisign, Entrust) o localmente aceptadas, como sería nuestro caso, permite que las entidades involucradas puedan fiarse unas de otras, a pesar de no existir contacto físico ni vínculo previo entre las partes. SSL, su sucesor TLS, y SET se están convirtiendo en estándares básicos que atestiguan el éxito de las tecnologías de clave pública en escenarios de seguridad descentralizados como Internet.

Las últimas iniciativas de las Administraciones Públicas para descargar procedimientos administrativos, realizados en papel y sometidos a la vanalidad burocrática, hacia procesos digitales interactivos, hacen uso también de tecnología PKI. En las secretarías de estado de nuestro país existen varios ejemplos de ello, aunque tal vez no con los resultados esperados debido principalmente a la poca difusión de la información necesaria hacia los usuarios para poder hacer uso de dicha tecnología.

Ventajas e inconvenientes del control de acceso por certificados

Ventajas

- Permiten autenticarse en muchos servidores distintos, sin necesidad de recordar multitud de contraseñas, ni, lo que es peor, utilizar la misma en todos los servidores.

- Son fáciles de escalar cuando crece el número de usuarios. Tomemos en cuenta que no es necesario mantener bases de datos descomunales con los nombres, contraseñas y privilegios de cada usuario. Gracias a los certificados, utilizando correspondencias entre campos de los certificados y las cuentas del servidor, se pueden crear políticas de acceso muy sofisticadas sin prácticamente necesitar ningún mantenimiento.

- Permiten descentralizar la verificación de permisos de acceso, basándose en la información contenida en el propio certificado.

Inconvenientes

- Cuando un certificado pierde validez por el motivo que sea, bien porque su clave privada ha sido comprometida, porque ha expirado, porque se ha comprometido la clave de la autoridad que lo certificó, etc., debe añadirse a una lista que contiene todos los certificados que han sido inhabilitados o revocados. Estas listas se conocen como Listas de Revocación de Certificados, y deberían ser consultadas por el servidor cada vez que se le presenta un certificado para ser verificado. Se hace evidente la dificultad de mantener las listas y sincronizar su información, especialmente cuando se trata con un número muy elevado de usuarios. Además la necesidad de consultarlas en cada autenticación impone una importante sobrecarga de procesamiento que puede llegar a degradar notablemente el rendimiento del servidor.

- Como suele ser habitual, los usuarios suelen ser el mayor obstáculo para que el sistema funcione correctamente. A menudo cometen fallos que vuelven este método vulnerable ó difícil de gestionar: olvidan la clave que protege su certificado, por lo que no pueden acceder al mismo y deben solicitar uno nuevo. Las consecuencias son que no pueden descifrar correo o archivos que hayan sido cifrados con su clave pública, por lo que quedan irremisiblemente perdidos, y además se debe añadir a la lista de revocación de certificados.

- Pueden borrar inadvertidamente su certificado y/o su llave privada cuando se dedican a hacer limpieza en el disco duro o cuando desinstalan algún programa, e inclusive pierden la computadora que lo almacenaba (lo cual no es descabellado en el caso de computadoras portátiles robadas u olvidadas en el aeropuerto o en otros lugares públicos).

- Dado que el certificado no es más que un archivo protegido por una contraseña, nada impide que lo compartan con otros usuarios, junto con su clave secreta. De ahí la conveniencia de almacenarlos en tarjetas inteligentes, que vuelven más difícil su uso compartido. Almacenar los certificados en el disco duro no es una buena idea, o al menos no es lo que se recomienda.

En la arquitectura general inicial del sistema se propone la utilización de una o varias ARs subsidiarias o subordinadas de una AC central, y un sistema de directorio encargado del almacenamiento de las CRLs y certificados expedidos.

Para descentralizar la tarea de registro (y favorecer y abaratar la comprobación de la identidad del usuario durante la misma) se van a articular una o varias ARs en torno a una AC encargada únicamente de certificar automáticamente las solicitudes que le van a enviar a modo de agentes proxy las ARs. En la AC, por tanto, únicamente se comprobará que la solicitud recibida ha sido generada y enviada por una AR de confianza del sistema (delegando en ella la responsabilidad de comprobar la identidad del usuario).

De este modo, el papel de la AC se limitaría a asegurar la integridad de su clave privada, sobre la que se sustenta todo el sistema y ejecutar las operaciones finales de expedición de certificados, aceptando automáticamente las peticiones de la AR.

Las ARs serían las encargadas de recibir las solicitudes de los usuarios, constituyendo por tanto la interfaz exterior del sistema. Además, sería en las ARs donde se efectúe el procedimiento de registro: comprobación inicial de la identidad y posterior solicitud de certificación.

La IETF cuenta con un grupo de trabajo activo el cual se centra en el estudio y evaluación de la infraestructura PKI [RFC 2459]. Este grupo llamado PKIX Working Group, se estableció a finales de 1995 con la intención de desarrollar estándares de Internet necesarios para soportar una PKI basada en el estándar X.509, (de ahí el nombre de PKIX). Los alcances del trabajo de PKIX se han expandido

más allá de su objetivo inicial. PKIX no solo perfila estándares ITU de PKI, sino que también desarrolla nuevos estándares relacionados con el uso de PKI basadas en X.509 en Internet.[2]

Basándonos en estos estándares y recomendaciones, a continuación presentamos lo que podría ser la estructura de lo que de ahora en adelante llamaremos la Infraestructura PKI-IMP (Figura 1.25).

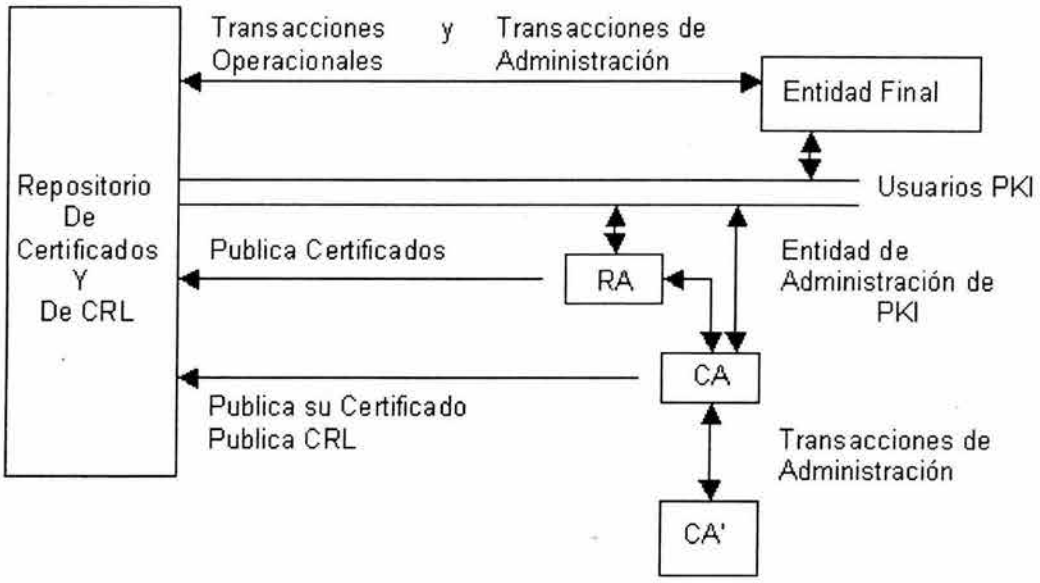


Figura 1.25
Entidades de PKI-IMP[3]

En el siguiente capítulo haremos un estudio más profundo y refinado de este esquema hasta llegar al modelo final que será en el cual nos basaremos para diseñar esta infraestructura.

No hay que perder de vista que existen varios caminos, métodos y herramientas para realizar esto, los cuales van desde contactar a una entidad externa a la organización para que sea ella quien realice el estudio y la implementación de un PKI, con las soluciones y herramientas que ella propone y con los respectivos costos implicados, hasta realizar uno mismo todo el trabajo implicado, proponiendo las herramientas, modelos e inclusive el costo que deseamos pagar.

En la medida de lo posible trataremos de considerar las opciones más importante que actualmente tenemos en cuanto a soluciones de diseño e implementación de PKI.
En cuanto a la toma de decisión de la solución que finalmente seguiremos hasta llegar a ver resultados, será en base al análisis de costo/beneficio que ella nos ofrezca, así como en el cumplimiento de los objetivos que perseguimos.

Software y Herramientas disponibles

A continuación citaremos un conjunto de herramientas y soluciones existentes, las cuales por sus características son candidatas idóneas para la implementación de PKI. De igual manera daremos una breve referencia de lo que son y en que se basan.

[2] Public-Key Infraestructure (X.509) (PKIX); http://www.ietf.org/html.charters/pkix-charter.html
[3] RFC 2459 IETF http://www.ietf.org

Soluciones Comerciales

Esta es quizá la opción más cómoda y práctica en cuanto a la solución de problemas se refiere. Sin embargo hay que considerar que dichas soluciones implican una inversión que en algunos casos no se tiene contemplada o que rebasa los presupuestos disponibles.

Estas soluciones en la actualidad son muchas y muy variadas, las cuales van desde soluciones ofrecidas por grandes corporativos mundiales de la seguridad en Internet (Verisign, Entrust, Sun Mycrosistems, etc) hasta soluciones ofrecidas por empresas a nivel nacional; que por ello no implica que no sean atractivas o viables, pero tampoco que sean soluciones muy accesibles en cuanto a costo se refiere.
Tal vez una de sus principales ventajas es el soporte que ellas brindan, ya que en caso de contingencia extrema, una simple llamada telefónica podría ser la solución al problema.
Existen adicionalmente un conjunto de beneficios que se tiene al adquirir un producto de software, esto representa el valor agregado que cada compañía puede ofrecer dentro de sus productos, sin embargo caemos nuevamente en el alto costo de la inversión que se debe realizar.

Proyecto OpenSSL[4]

El proyecto OpenSSL es un esfuerzo conjunto para desarrollar un kit de herramientas robustas, de grado comercial, de características completas, y de código abierto (Open Source) implementando los protocolos de Secure Sockets Layer (SSL v2/v3) y Transport Layer Security (TLS v1) así como una librería criptográfica de propósito general fuerte y completa.
OpenSSL es una implementación libre no comercial de SSL. Además de la implementación SSL, incluye utilidades para la administración de certificados. También incluye una implementación de PKI, la cual puede ser utilizada fuera de los Estados Unidos.

OpenSSL puede configurarse para actuar como una autoridad certificadora, emitiendo certificados tanto de clientes como de servidor.
OpenSSL se debe configurar para trabajar de manera conjunta con el servidor HTTP de Apache e instalar el módulo mod_ssl el cual permite la interacción de los dos paquetes.
Las principales ventajas de esta herramienta es que es adaptable y configurable a las necesidades particulares y que es de libre distribución.

Proyecto OpenLDAP[5]

El proyecto OpenLDAP es un esfuerzo conjunto para desarrollar una suite de aplicaciones y de herramientas de desarrollo robusta, de grado comercial, completa y de código abierto de LDAP. El proyecto es administrado por una comunidad mundial de voluntarios que utilizan Internet para comunicar, planear y desarrollar la suite OpenLDAP, así como la documentación relacionada.
Esta herramienta permite el manejo y administración de los llamados "directorios" así como de implementar servicios para la consulta y modificación de los mismos.
Al igual que la herramienta anteriormente mencionada, OpenLDAP es adaptable y configurable a lo que se requiera, e igualmente es de libre distribución.

APACHE Software Foundation[6]

La fundación Apache Software proporciona soporte para la comunidad Apache en proyectos de software de código abierto. Los proyectos de Apache están caracterizados por una licencia de software colaborativa, pragmática y abierta (libre), cuyos procesos de desarrollo están basados en consensos, y por el deseo de crear software de alta calidad que lidere el camino en este rubro.

[4] Proyecto OpenSSL http://www.openssl.org
[5] Proyecto OpenLDAP, http://www.openldap.org
[6] Apache Software Foundation, http://www.apache.org

Proyecto APACHE –HTTP Server

El proyecto de Apache HTTP-Server es un esfuerzo por desarrollar y mantener un servidor HTTP de software libre para los modernos sistemas operativos incluyendo UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor de HTTP seguro, eficiente y extensible que proporcione servicios de HTTP en sincronía con los actuales estándares de HTTP.

Apache SSL/TLS Encryption

El módulo del servidor HTTP de apache mod_ssl, proporciona una interfaz hacia las librerías de OpenSSL, la cual proporciona encripción fuerte utilizando los protocolos SSL y TLS.

Proyecto OpenCA[7]

OpenCA Labs es una organización abierta orientada a proporcionar un marco para el estudio de las PKI's y el desarrollo de proyectos similares. Así como los estándares de PKI, tanto los intereses como los proyectos están creciendo rápido; se ha dividido el proyecto original en proyectos más pequeños para acelerar y reorganizar los esfuerzos.

El proyecto de desarrollo PKI de OpenCA, es un esfuerzo conjunto para desarrollar una Autoridad de Certificación robusta, de características completas y de código abierto (Open Source) implementando los protocolos más utilizados con criptografía completamente sólida a nivel mundial. OpenCA está basada en muchos proyectos de código abierto. Entre el software de apoyo está OpenLDAP, OpenSSL, el proyecto Apache, y Apache mod_ssl. El desarrollo del proyecto esta dividido en dos tareas principales: Estudiar y refinar el esquema de seguridad que garantice el mejor modelo para ser usado en una AC y desarrollar software para instalar y administrar fácilmente una Autoridad de Certificación.

Es principalmente en estas opciones a nuestro alcance, en las cuales nos apoyaremos para la implementación de PKI-IMP. En la sección de JUSTIFICACIÓN mencionamos que es importante especificar y fijar los niveles de seguridad y los costos que se tienen contemplados. En principio diremos que en cuanto a niveles de seguridad se refiere, consideraremos satisfactorio un resultado que brinde un nivel mayor de seguridad en el proceso de autenticación que el que nos brinda el esquema existente. Las facilidades extras de integridad y confidencialidad que se puedan obtener serán consideradas como un valor agregado del trabajo, ya que no se tenían contempladas al inicio.

En cuanto a costo se refiere, diremos que el presente trabajo no cuenta con una partida presupuestal por parte del IMP, por tanto, en principio y para fines de construcción de un prototipo, será necesario considerar que no se contarán con recursos financieros para la adquisición de herramientas y/o equipos adicionales a los que actualmente cuenta el área de Seguridad Informática.

Para la fase de diseño nos basaremos en los estándares anteriormente mencionados, así como en las guías y recomendaciones existentes para llevar a cabo este trabajo.

En el siguiente capítulo nos dedicaremos al estudio de estas y otras cuestiones referentes al diseño e implementación de PKI-IMP.

[7] Proyecto OpenCA, http://www.openca.org

REFERENCIAS :

PKI o los cimientos de una criptografía de claves públicas;
Documento electrónico
http://www.iec.csic.es/criptonomicon/susurros/susurros11.html

Public-Key Infraestructure (X.509) (PKIX);
Estatuto del Internet Engineering Task Force, IETF
http://www.ietf.org/html.charters/pkix-charter.html

PKIX Working Group;
RFC 2459 IETF
http://www.ietf.org

Proyecto OpenSSL;
http://www.openssl.org

Proyecto OpenLDAP;
http://www.openldap.org

Apache Software Foundation;
http://www.apache.org

Proyecto OpenCA;
http://www.openca.org

Capítulo 2

Diseño del Sistema

Introducción

Hasta el momento hemos concluido que una solución basada en PKI es lo que más nos conviene dadas las situaciones problema que enfrentamos. Igualmente mencionamos que una infraestructura de llaves o clave pública no es mas que un diseño estructural para establecer comunicación, mensajería y transacciones seguras sobre las redes, y que para esto se apoya principalmente en la criptografía de llave pública y en los certificados digitales.

Con lo recabado hasta este momento, considerando ventajas y desventajas de los posibles métodos y/o procedimientos de diseño de PKI, pero principalmente en la viabilidad, utilidad y conveniencia que para nosotros representen, estableceremos una metodología para concebir y diseñar la PKI-IMP. De igual manera trataremos de hacer una perspectiva de los diferentes modelos, esquemas o caminos que podamos seguir en la construcción de la arquitectura de PKI-IMP, con la finalidad de poder distinguir y adoptar la mejor opción.

2.1 ESPECIFICACIONES FUNCIONALES

Como en todo proyecto, es necesario establecer una serie de pasos a seguir para llevar a cabo nuestro diseño. Estos pasos, así como el diseño mismo, se basan en los estándares, estatutos, guías y recomendaciones que existen en la actualidad. Las referencias y fuentes de estos documentos se citan en la sección de REFERENCIAS al final de esta sección.

Para llevar a cabo el diseño de nuestra PKI seguiremos los siguientes pasos[1]:

1. Análisis de los requerimientos iniciales.
2. Descripción de los servicios que brindará.
3. Diseño de herramientas y funciones para el desarrollo de los servicios de PKI.
4. Protocolos de comunicación.
5. Descripción de las entidades.
6. Interrelación con otras infraestructuras de llaves públicas externas.

1. Análisis de los Requerimientos Iniciales

1.1 Necesidades de diseño

Inicialmente se tenía pensado estudiar e implementar solo un esquema de autenticación más fuerte y robusto basado en certificados digitales que el que actualmente existe, sin embargo, como ya hemos visto, esto por si solo no es suficiente ni garantiza ser una solución óptima y tal vez tampoco garantice el incremento en los niveles de seguridad deseados dentro de la red IMP.
Como consecuencia, y a fin de proponer un esquema más robusto y confiable de seguridad, se propuso adicionar los puntos de integridad, no repudio y confidencialidad; en conjunto una PKI.

PKI-IMP se concibe como un mecanismo para reforzar tanto la autenticación como la seguridad en el intercambio de información sobre la red IMP; se tiene contemplado inicialmente, y como parte de las extensiones de la solución, contar con servicios de correo electrónico seguro, cifrado y descifrado de información, firma electrónica y verificación de la misma, así como autenticación única vía certificado digital, sin embargo en un futuro se tiene pensado que todas las aplicaciones que hagan uso de la red IMP se rijan bajo el esquema PKI.

[1] Esta selección se hizo luego de haber considerado los estándares y las recomendaciones de las fuentes a las que se hace referencia al final de la sección.

De igual manera, se tiene pensado que todo usuario dentro de la red IMP cuente con un certificado y un par de claves pública/privada, de la misma manera como actualmente todos cuentan con una credencial de identificación de trabajadores del IMP o una cuenta de correo electrónico.
El contar con una infraestructura de PKI bien diseñada e implementada, resuelve de manera óptima los objetivos iniciales e igualmente los puntos extras que deseamos cubrir.

1.2 Requerimientos para el diseño de la PKI

Una condición que nos hemos impuesto nosotros mismos es que tanto el diseño como la implementación de PKI-IMP no impliquen modificaciones a la estructura física de la red, ni a la estructura lógica y administrativa de la misma.

No existe un diseño genérico de infraestructura de PKI, esto debido a que cada red y cada lugar que la necesite tiene sus propios requerimientos, restricciones y características. La PKI-IMP se concibe inicialmente a nivel interno, únicamente para trabajar dentro de la red IMP, por lo tanto los niveles de seguridad en cuanto a longitud de claves no deben ser demasiado elevados ya que esto traería consigo un incremento en el tiempo de procesamiento en la parte de cifrado y descifrado así como el incremento en el tráfico dentro de la red, con la consecuencia de hacer más lentas las comunicaciones, aplicaciones y los procesos que se ejecutan sobre ella. En lugar de eso, es más importante pensar en una estructura administrativa muy bien controlada y funcional de dichas claves

Antes de continuar es muy importante tener en cuenta lo siguiente:

Dado el entorno en el cual PKI-IMP será diseñada e implementada, el modelo de confianza que se aplicará es el de AUTOCONFIANZA. Esto es, los usuarios de esta infraestructura confían en PKI-IMP ya que ella misma vigilará y administrará la seguridad que ofrece.
De manera simplificada PKI-IMP confiará solo en PKI-IMP y en ninguna otra infraestructura más.

Se tiene pensado que cada usuario maneje un par de claves (pública/privada) y un certificado digital, siendo estos elementos los únicos que se manejen a nivel usuario final dentro de PKI-IMP. Se propone de entrada un esquema centralizado de solicitud y entrega de dichos elementos, de manera muy parecida al procedimiento que actualmente se sigue en el proceso de obtención de una cuenta de correo Institucional; pudiéndose modificar esto en un futuro a un sistema descentralizado y personalizado, como el que actualmente se sigue cuando uno solicita una cuenta de e-mail en yahoo o hotmail, por ejemplo.

Se propone inicialmente una longitud de claves de 1024 bits, quedando sujeta a las pruebas posteriores tanto de seguridad como de rapidez en el manejo de las mismas pudiéndose incrementar o decrementar la longitud de ellas. OpenSSL ofrece estas y otras facilidades para el manejo e implementación de herramientas criptográficas.

El tiempo de vida de las claves será variable por el momento debido a que se deben utilizar dentro de la fase de pruebas y validación, para lo cual requerimos tanto claves activas como claves expiradas. El tiempo de vida final será fijado de acuerdo a las políticas de seguridad que queden establecidas al final de la fase de implementación de la PKI-IMP.

Se propone el uso de algoritmos criptográficos como RSA, DES, TDES, SHA1 y MD5 para implementar los servicios de PKI-IMP, que son estándares a nivel internacional los cuales son de uso abierto y no restringido para los fines que deseamos alcanzar, considerando que utilizaremos la herramienta OpenSSL y algunas otras de libre distribución.

Es importante mencionar que en principio no se tiene contemplado manejar niveles de usuarios, esto es, dentro de usuarios finales todos tienen los mismos privilegios y responsabilidades. Sin embargo existen entidades, organizaciones y políticas de seguridad que así lo recomiendan. En el transcurso

del desarrollo de la arquitectura y de la administración de PKI-IMP podremos darnos cuenta de qué es lo que más nos conviene.

Tenemos contemplado un repositorio de certificados el cual sería LDAP, basados en las facilidades que brinda para el manejo de certificados x509 los cuales forman parte de la base de nuestra PKI.

Como ya lo mencionamos anteriormente la PKI-IMP en principio trabajará sola, sin interactuar con otras PKI's, sin embargo esto no quiere decir que esto en un futuro sea imposible, ya que el diseño y la construcción se hará con base a estándares internacionales lo cual, consideramos, facilitará su inclusión en esquemas de PKI mas grandes, a nivel nacional por ejemplo. Sin embargo este es un punto en el cual no pretendemos profundizar y que propondremos como un trabajo a futuro.

Es deseable que PKI-IMP se diseñe lo más abierta posible para que pueda ser utilizada por diversas aplicaciones y que tenga la capacidad de crecer gradualmente conforme las necesidades así lo demanden. En el diseño y en la implementación se contemplará esta necesidad.

1.3 Requerimientos de los usuarios

A simple vista resalta que la mayoría de los usuarios serán usuarios no experimentados ni conocedores del tema ni de los servicios que se ofrecen y tal vez no lleguen a comprender su existencia y utilidad.

Hay que hacer mención especial que un "usuario" además de ser un empleado del IMP puede ser también un servidor de alguna aplicación o un servicio dentro de la red IMP, por lo que hay que tenerlos muy en cuenta a la hora de concebir, diseñar e implementar, principalmente las políticas bajo las cuales se regirá la PKI-IMP.

La infraestructura, por lo tanto, debe de ser lo más transparente posible para los usuarios finales, sin embargo debe ponerse especial atención de los riesgos y amenazas "internas", esto es, en la que ellos representan de manera intencionada como de manera no intencionada.

1.4 Requerimientos de Seguridad de la PKI

Existen algunos requerimientos de seguridad que debe cumplir cualquier PKI, a lo largo del diseño se introducirán aspectos que asegurarán estos requerimientos, estos son los que la PKI-IMP debe cumplir en principio:

- Confiabilidad de la información que la PKI maneja: claves, certificados y mensajes.
- Integridad de dicha información dentro de la PKI
- Disponibilidad continua de servicios que la PKI ofrece así como el tiempo de recuperación con respecto a compromisos, robos o ataques, el cual debe de ser corto.
- Seguridad: Debe tener un nivel aceptable de seguridad y no afectar la vulnerabilidad de sistemas individuales cuando se enlacen con la PKI. Los riesgos no pueden ser mayores que un sistema similar en papel.

Estos requerimientos se irán logrando a lo largo del diseño.

2. Servicios.

En esta parte queremos mencionar los servicios que inicialmente brindará la PKI-IMP a sus usuarios, mas adelante se detallará el modelo que se piensa implementar (entiéndase usuarios de PKI tanto a las personas, como aplicaciones y equipos).

1) Servicios de manejo de claves para firmas digitales y para confidencialidad
2) Servicio de manejo de certificados
3) Servicios de publicación y almacenamiento de claves, certificados y CRL,
4) Servicios de interfaz con el cliente.

Estos servicios se describen a continuación:

1) Servicios de manejo de claves para firmas digitales y para confidencialidad

Estos servicios son útiles para los usuarios finales, así como para el funcionamiento interno de la PKI, para firmar los certificados. Es aquí donde aparece la figura de la Autoridad Certificadora que es quien se encarga de firmar con la clave secreta los certificados digitales que emite y verificar la firma con su certificado. Esta entidad estará presente por lo tanto en los usuarios y en los servidores de PKI.

Una vez que los usuarios cuenten con su par de claves y su certificado, entonces podrán hacer uso de los servicios de firma digital, utilizando su clave privada, y de cifrado, utilizando su clave privada así como el certificado de él o los destinatarios.

Adicionalmente, tanto para el proceso de firma como de cifrado, tendremos en cuenta y procuraremos lo siguiente:

1) Asegurar la integridad entre el que firma con la clave privada y el que verifica con la pública (teniendo en cuenta que una persona puede tener más de un certificado) por medio de un mecanismo administrativo de PKI que nos permita validar la firma así como el certificado digital correspondiente en tiempo real. Para esto los actuales sistemas operativos cuentan con un conjunto de librerías criptográficas que fácilmente se pueden utilizar, basta con configurarlos adecuadamente para poder explotar las facilidades que ofrecen.

2) Un mecanismo de almacenamiento de claves privadas y de certificados.
Al igual que para el punto anterior, los sistemas operativos actuales cuentan con repositorios de certificados y de claves privadas que facilitan su uso y administración. En este caso será suficiente con distribuir información precisa del uso de este tipo de herramientas a los usuarios finales en el caso de empleados del IMP, y de configurar dichos repositorios para interactuar con aplicaciones, y viceversa, en el caso de usuarios que sean aplicaciones o servicios dentro de la red IMP.
Actualmente existen mecanismos que permiten almacenar tanto las claves como el certificado en las llamadas tarjetas inteligentes o tokens criptográficos, sin embargo no nos adentraremos mucho en eso ya que dicha tecnología no es muy accesible para nosotros; además de que estos implican una inversión considerable en cuanto a recursos económicos se refiere. Los mencionaremos en la sección de trabajos futuros.

3) Proveer de un medio para recuperación de certificados en caso de pérdida o de no haberlo recibido antes. Para el caso de la clave privada esto no aplicará como un servicio de PKI-IMP, en su lugar se implementará un mecanismo que obligue a los usuarios a proteger y a respaldar ellos mismos su clave privada de manera que durante su ciclo de vida o validez exista dicha clave y al menos un respaldo en resguardo del propietario.

4) Proveer un medio de revocación de certificado que garantice que clave privada con la que se firma no está comprometida, cuando se está verificando la firma con el certificado correspondiente. Esto pensamos lograrlo mediante un manejo eficiente de las solicitudes de revocación y de las listas de revocación de certificados CRL's e igualmente ideando mecanismo que obliguen a los usuarios a mantener actualizadas dichas listas de manera local, en sus equipos, así como de notificar lo más pronto posible a la AC cuando se sospeche que tanto su certificado y/o su clave privada han sido comprometidos .

5) Crear un mecanismo de logs de las acciones que se ejecutan en todas las entidades. Esto pensamos lograrlo desde la administración de la PKI por medio de logs de acceso a los servidores de certificados y de CRL's; así como de la correcta configuración de los clientes de manera que nos permitan monitorizar sus acciones.

2) Servicio de manejo de certificados

Los certificados son los documentos que distribuyen las claves públicas, nosotros proponemos el uso de un solo certificado útil para firmar/verificar la firma, así como para cifrar; esto es posible si en el certificado incluimos las atribuciones que se permiten con dicho certificado, los cuales pueden ser: autenticación del cliente, firma y cifrado digital, no repudio, etc.

Procuraremos contar en este rubro con los siguientes servicios:

1) Generación de certificados digitales que se enlacen correctamente con la Autoridad Certificadora, de manera simple, contemplamos el uso de cadenas de certificación.

2) Generación de certificados para Autoridades Registradoras locales de manera que el agregar una nueva entidad que facilite la administración de PKI-IMP sea un proceso sencillo, rápido y eficiente.

3) Generación y publicación de la lista de certificados revocados, CRL, de manera eficiente; procurando que la seguridad y la confianza de la PKI no se vea afectada.

4) En la medida de lo posible, proveer un mecanismo cómodo para verificar certificados.

3) Servicios de publicación y almacenamiento de claves, certificados y CRL

Es el servicio que nos permite la distribución de certificados, así como otros datos de las personas u otras entidades funcionales que estén dentro de la PKI-IMP. Este servicio debe permitir entre otras cosas que:

- Se publiquen las CRL's actualizadas y los certificados digitales de los usuarios de PKI-IMP así como los de la AC y la AR, de manera que éstos sean fácilmente accesibles por todos los usuarios.
- Existan entradas en las BD para obtener los diferentes certificados con sus datos, excepto para las claves privadas, para las cuales no existirá base de datos alguna por cuestiones de seguridad; al menos del lado de la administración de PKI-IMP.
- Asegurarse que existe un solo nombre para cada objeto en la PKI, al momento de emitir o revocar un certificado y/o sus claves.
- Brindar servicio de directorio confidencial con entradas autorizadas (que no tiene relación con la CRL) pero sí con alguna información personal, y
- Además debe mantener un control estricto sobre la modificación de la información, garantizando siempre que solo le sea permitido al personal autorizado.

4) Servicios de interfaz con el cliente.

Este servicio permitirá que los usuarios de PKI-IMP puedan tener contacto con dicha infraestructura de manera que puedan realizar todo lo que se ha mencionado hasta este momento de una manera sencilla, rápida y cómoda tanto para ellos como para la administración de la PKI. Igualmente es deseable que dicha interfaz esté disponible en todo momento y que a la vez permita tener un control y garantice la seguridad de las transacciones que a través de ella se lleven a cabo. Creemos que una interfaz web, por las facilidades que ofrece, sería lo ideal para esta tarea.

Igualmente se contempla el uso y explotación de los mecanismos y herramientas existentes en los sistemas operativos de los clientes de manera que no sea una interfaz del todo desconocida para ellos.

3. Herramientas y funciones para el desarrollo de los servicios de PKI.

En esta parte toca describir las funciones y herramientas necesarias para cada servicio. Describimos dos de los servicios más importantes: manejo de certificados y manejo de claves; los dos restantes, por su sencillez, los describiremos en la sección de la Arquitectura del Sistema.

1) Manejo de claves para firmas digitales y para confidencialidad.

1.1 Generación del par de claves pública/privada

Se tiene contemplado que sea de manera centralizada, esto es, las crea la AC y las entrega junto con el certificado, el usuario final proporcionará únicamente un password de protección de uso para su clave privada, esto por medidas de seguridad para cuando se desea cifrar/descifrar con la misma.

Sería ideal poder contar con un mecanismo que permita que dicho par de claves sea generado de manera conjunta, tanto por el usuario como por la AC, de manera que ésta última pueda verificar la seguridad de las mismas. Esta variante garantizaría la confidencialidad de las claves para el usuario y la seguridad para la AC. En la medida de lo posible trataremos de orientar este proceso de generación de claves al último escenario descrito.

El motor criptográfico que se empleará principalmente es OpenSSL del lado del servidor y el conjunto de herramientas criptográficas disponibles del lado del cliente.

Ventajas: certeza de la seguridad de la clave, permite brindar servicios de recuperación de clave en caso de pérdidas y se tiene un mejor control en la generación de las mismas ya que el proceso se hará en base a una solicitud, tal y como actualmente se solicita una cuenta de correo institucional.

Desventajas: Tal vez la principal es que el usuario pueda pensar que puede estar siendo engañado o que de alguna manera alguien más, del lado del servidor, pudiese tener una copia de su clave privada, y con esto comprometer la seguridad de su información, esto basándose en que dicha clave privada, la más importante, no fue él mismo quien la generó, sino que le fue asignada mediante algún mecanismo por la autoridad central de la PKI-IMP. Pero al final de cuentas en alguien tenemos que confiar.

1.2 Entrega del par de claves pública/privada

Se tiene contemplado que sea vía una AR y de manera personal, por lo menos la primera vez, con la finalidad de asegurarnos que la entrega sea personal y que no haya errores o se comprometa dicha información. En el caso de que dicha entrega sea consecutiva, por motivos de renovación o de reasignación de claves, el correo electrónico podría ser de mucha utilidad.

Existe también la posibilidad de que dicha entrega se haga en línea, vía un navegador, con el cual pueda descargar el par de llaves en cuestión, ya sea que se le entreguen en algún sitio o que se le puedan enviar en un correo electrónico.

1.3 Actualización del par de llaves pública/ privada

Se hará en base a las políticas bajo las cuales quede regida la PKI, igualmente de manera centralizada y tomando en cuenta el ciclo de vida de las llaves. El procedimiento final se detallará mas adelante.

2) Manejo de Certificados

El formato de diseño que manejaremos es el X509, de él seleccionaremos los campos que creamos más convenientes para incluirlos en los nuestros, tales como: nombre, correo electrónico, área a la cual pertenecen etc; considerando el uso que tendrán así como las facilidades que deben brindar. La administración de dichos certificados, en principio, se tiene contemplada que sea centralizada directamente por la AC, la cual tendrá un responsable quien es el que lleva a cabo el resguardo de la misma y quien esta al pendiente de la AC tanto en materia de seguridad como de administración. Esta responsabilidad recaería forzosamente en el encargado del área de seguridad informática del IMP.

A manera de facilitar tanto la administración de los certificados así como la administración de la AC, creemos conveniente contar con al menos una AR la cual haría el trabajo "pesado", como es la entrega y publicación de los certificados digitales y atender a los usuarios de manera directa.
Con esto logramos que la carga de trabajo para la AC en cuanto al manejo de certificados disminuya y pueda concentrarse en otra labores primordiales para la infraestructura.

El repositorio de certificados proponemos que sea OpenLDAP, el cual tiene como función primordial servir como un directorio de certificados existentes y el cual haría las veces de un "llavero" el cual contiene la llave pública y el certificado de algún otro usuario final con el cual buscamos comunicarnos de manera segura.

Al igual que la AC, tanto la AR como el servidor de directorio OpenLDAP, estarían administrados y vigilados por el área de seguridad informática del IMP, pudiéndose delegar la administración de éstos últimos a personal de confianza de la misma área.

Cadenas de certificados[2]

Para la construcción del modelo de la infraestructura PKI-IMP se tiene contemplado que solo exista una sola AC central en la cual todos los demás usuarios deben confiar, cuyas funciones principales se detallarán mas adelante.

Así mismo se tiene prevista la existencia de al menos una AR la cual estaría encargada de comunicarse con la AC para solicitar los certificados firmados, es esta tal vez la única cadena que exista hasta el momento y en la cual debemos poner especial cuidado, los usuarios finales confían en el certificado de la AR ya que esta a su vez esta certificada por la AC central del IMP.
Para tener un panorama más amplio respecto a esto, veamos y analicemos lo siguiente:

Existen dos filosofías de diseño y operación a nivel internacional en competencia:

- La recomendación x.509 de la ITU-T y
- La infraestructura del grupo de trabajo PKIX [RFC 2510, 2459, 2527] del IETF.

La recomendación x.509 que no asume la existencias de un espacio global de nombre sino de una multitud de espacios locales enlazados donde el concepto de jerarquía desaparece, por ejemplo PGP,

[2] Cadena de Certificado: Es todo el camino que se debe recorrer para verificar un certificado, esto es, cuando un certificado llega a nosotros se debe verificar si se conoce a la AC que lo firmó, de no conocerla, se debe buscar quién firmó a esa AC, si la conocemos terminamos, si no debemos seguir buscando hasta llegar a una AC conocida o una AC autofirmada no conocida por nosotros.

SPKI (Simple Public Key Infraestructure), SDSI (Simple Distributed Security Infraestructure) y DNSSEC (Domain Name System Security extensions).

PKIX de IETF asume un espacio global de nombre y por lo tanto la configuración que se maneja es una AC centralizada o una jerarquía, podemos poner de ejemplos a PEM (Privacy Enhanced Mail) que fue el primero en usarlo, SET (Secure Electronic Transaction), S/MIME entre otros.

Nosotros nos apegaremos a la segunda filosofía, ya que consideramos es la mejor dadas nuestras circunstancias y condiciones de trabajo.

A continuación comentamos sobre algunas de las variantes utilizadas y algunas de sus problemáticas a la hora de diseñarlas e implementarlas.

Variante con AC centralizada.

Consiste en una AC que brinda todos los servicios a todos los usuarios, por lo tanto, todos los usuarios de PKI confían solamente en la firma de ella. La verificación de certificado es muy simple A confía en B, si el certificado de B está firmado por AC. Esta variante puede ser usada para un centro muy pequeño, donde el acceso a la AC por todos los usuarios sea muy sencillo. Cuando la cantidad de usuarios finales aumenta lo natural es crear varias AC y conectarlas.

Jerarquía de AC.

Relación entre las AC con subordinación a superior (ver figura 2.1), como se puede ver en la figura AC-11 está subordinada a AC Raíz y AC- 21 está subordinada a AC-11.Todos los caminos de certificado comienzan por AC Raíz. Generalmente la AC Raíz no emite certificados a entidades finales sino que certificados a agencias certificadoras del próximo nivel. La relación de confianza es en una dirección, la AC subordinada confía en sus superiores

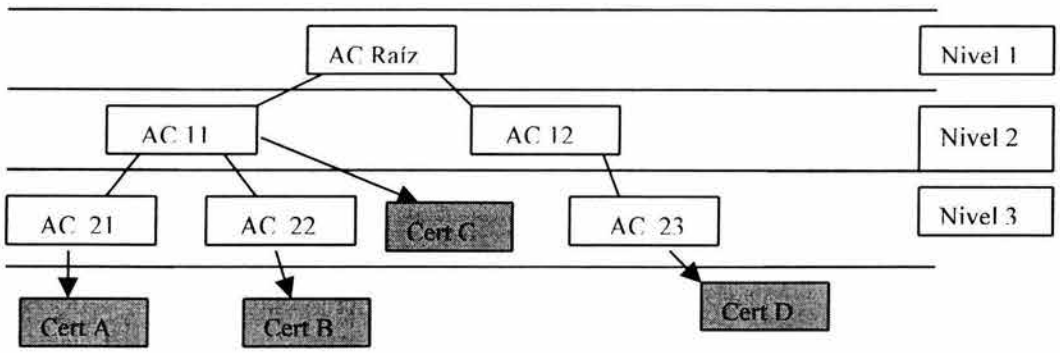


Figura 2.1 Jerarquía de AC's

Debido a la simple estructura y la relación de confianza en una sola dirección de la jerarquía de AC, se presentan las siguientes *ventajas*:

1) Es muy fácil incorporar una nueva comunidad de usuarios, estableciendo una relación entre la AC de la comunidad y la AC Raíz o cualquier otra. La posición de la nueva AC puede estar determinada por las políticas de la organización.
2) La búsqueda de la cadena de certificados es muy simple de buscar porque existe una sola dirección de confianza.
3) El camino de certificado es relativamente corto, nunca será mayor que la profundidad del árbol.

4) Los usuarios no tienen que influir en la búsqueda del camino de certificado.

Desventajas:
1) Si la AC Raíz es comprometida se compromete toda la PKI y no existe una forma directa de recuperarse.
2) La idea de que todos los usuarios confíen en una AC raíz puede ir en contra de las políticas de la organización para la cual diseñamos la PKI.

Para eliminar estas desventajas se diseño una variante de malla (ver más adelante) llamada también punto a punto.

Los niveles de jerarquía pueden definirse de diversas maneras:
- Por zonas geográficas; se define AC por zonas y las personas se certifican en las AC más cercanas, por organización del lugar donde se va a diseñar la PKI (por ejemplo departamentos, gerencias, facultades, etc.),
- Por niveles de política o
- Puede ser una mezcla de las anteriores.

Variante de malla de AC

Esta variante ya no supone nombres globales. Construye una relación punto a punto entre las AC que es conocida como malla o red de confianza. Todas las AC en la malla tienen puntos de confianza y cada usuario en general confían en la AC que le emitió el certificado (Ver figura 2.2). Cada AC enlazada se dan certificado las unas a las otras, por lo tanto un par de certificados describen una relación de confianza bidireccional.

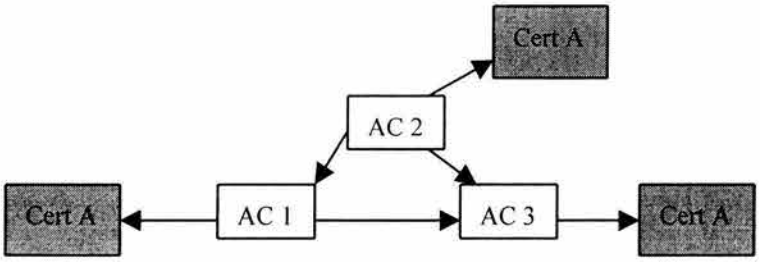


Figura 2.2: Malla de AC's

Ventajas del diseño:

1) Es muy simple incorporar una nueva comunidad de usuarios, pues la nueva AC se incorpora y establece la relación de confianza adecuada con cada AC.
2) Este diseño lo hace muy flexible a cualquier tipo de relación de confianza.
3) El compromiso de una AC no hace comprometer a la PKI completa, para recuperarse solo es necesario revocar los certificados de la AC comprometida y removerla de la PKI. Si algún usuario de esta AC comprometida está asociado con otra AC, aun tiene un punto de confianza y puede continuar comunicándose en la red.

Esta variante es muy buena cuando una organización ha diseñado PKI separadas y desea mezclarlas o comunicarlas.

Desventajas:

1) La construcción de un camino de certificados es más compleja que en un modelo jerárquico y a diferencia de este el método, no es determinista. Al tratar de descubrir el camino puede

presentarse diversas opciones, algunas que lleven a la solución y otras que sean caminos muertos e incluso caer en ciclos infinitos.
2) El máximo tamaño de caminos de certificados es la cantidad de AC que hay en la malla.
3) Es tarea del usuario ayudar en la construcción del camino de certificado.

La elección entre un modelo jerárquico y uno de malla, viene dado por las características de la comunidad a la que se le va a diseñar la PKI.
En nuestro caso el modelo a seguir, jerárquico o de malla, se seleccionará en base a lo que ya hemos comentado ampliamente en las secciones anteriores y se indicará en la sección de la Arquitectura del Sistema.

4. Protocolos de comunicación

Debido a que todo el tráfico de información de la PKI-IMP será sobre la red IMP, y tomando en cuenta las aplicaciones y herramientas que se pretenden utilizar, los protocolos de comunicación utilizados dentro de PKI-IMP son muchos y muy variados. Destacan los que corren sobre TCP/IP, nosotros utilizaremos algunos de ellos en sus versiones de libre distribución, como son: OpenSSL, OpenLDAP, al igual que al conjunto de herramientas soportadas DES, TDES, SSL, TLS,MD5, SHA1 etc, y algunos otros presentes en los sistemas actuales como son TLS, HTTP, HTTPS, S/MIME, y otros de uso cotidiano.

5. Descripción de las entidades

Es el momento de describir cada una de las entidades que conforman y participan en PKI-IMP y su función. No hay que olvidar que esto lo hacemos con base en lo que marca el estándar de PKIX al cual nos hemos apegado.

Autoridad Registradora (AR)

La Autoridad Registradora dentro de PKI-IMP es la responsable de las tareas administrativas asociadas con el registro de la entidad destino (entidad final o clientes), que es el sujeto del certificado expedido por la AC.
Las funciones de la AR con base en las necesidades de instalación de PKI-IMP entre otras incluyen:

- Recibir solicitudes de los individuos o usuarios finales que desean obtener un certificado digital.
- Verificar la identidad de los individuos y entidades finales que desean obtener certificados digitales(Autenticación Personal).
- Verificar la validez de la información suministrada por el individuo
- Validar el derecho del sujeto a los atributos del certificado solicitado.
- Generación, junto con el cliente, del par de claves público/privado.
- Iniciar el proceso de registro con la Autoridad Certificadora en nombre de la entidad destino o cliente (sujeto)
- Emitir precertificados y solicitar los respectivos certificados digitales a la AC.
- Administrar el repositorio de certificados digitales registrados, tanto actuales como históricos.
- Proporcionar los certificados digitales finales firmados por la AC que le fueron solicitados por los usuarios o entidades finales.
- Informar los casos en los cuales se puede suspender o revocar un certificado así como de las situaciones en las que se considere comprometida la seguridad de su clave privada y de su renovación.
- Recibir solicitudes de revocación, verificar y validar la solicitud y enviar esta a la AC para su aprobación.
- Notificar y distribuir cada nueva CRL emitida por la AC.

En general, la Autoridad Registradora maneja los intercambios (que con frecuencia involucran interacciones del usuario) entre la entidad destino sujeto y la PKI para el registro, la entrega del certificado y la clave. Sin embargo bajo ninguna circunstancia la AR en realidad origina o produce declaraciones de confianza sobre el sujeto (diferentes a las de certificarlos como parte del proceso de identidad de la AC). Como resultado, solamente la Autoridad Certificadora puede expedir certificados firmados u originar información del estado de revocación del certificado, como en el caso de una CRL.

Autoridad Certificadora

La autoridad Certificadora, AC, es responsable de firmar y expedir certificados de la entidad destino. Éstos asocian la identidad de la entidad destino del sujeto, según se expresó por medio del nombre del sujeto que se registró con la clave pública correspondiente a la clave privada que poseía ese sujeto.
En nuestro caso la AC de PKI-IMP entre otras cosas tiene las siguientes tareas:

- Normar la PKI-IMP de acuerdo con las políticas que establezca el área de Seguridad Informática del IMP.
- Emitir y firmar certificados digitales.
- Garantizar la unicidad de las claves públicas del sistema.
- Administrar el repositorio de certificados digitales LDAP
- Crear su propio certificado digital y certificar a la AR .
- Difundir ampliamente su clave pública y la clave pública de la AR.
- Firmar los pre-certificados enviados por la AR para así crear el certificado final y devolverlo a la misma para su entrega al usuario final.
- Registrar los certificados digitales en el repositorio siempre y cuando confirme la unicidad de las llaves públicas.
- Revocar certificados y emitir listas de revocación CRL
- Establecer, administrar y mantener las medidas que garanticen la seguridad del sistema.

Además, la AC es la responsable de administrar todos los aspectos del ciclo de vida del certificado después de su expedición. Esto incluye seguir el estado de un certificado y las noticias de revocación del mismo, así como llevar un control tanto de las solicitudes y de los certificados emitidos como de los certificados revocados.

Repositorio

El repositorio se utiliza para el almacenamiento público de certificados y listas de revocación de certificados. Originalmente era un directorio X.500. Para soporte de PKI-IMP, el repositorio será un directorio LDAP, el cual se lista como uno de los protocolos operativos que se soportan específicamente en PKIX.
Además de almacenar certificados y CRL, otras entrada en el directorio, tales como los objetos de directorio de AC, se pueden utilizar para almacenar información adicional, tales como identificar relaciones como la certificación cruzada entre AC's, fotografías de los usuarios etc.

Clientes ó Usuarios

Los clientes o usuarios son todas aquellas entidades que hacen uso de PKI-IMP de manera directa o indirecta. Algunas de las principales tareas realizadas por los clientes son las siguientes:

- Solicitar su certificado digital a la AC a través de la AR
- Autenticarse ante la AR
- Proporcionar una frase de seguridad para su clave privada.
- Recibir su certificado digital ya registrado así como su clave privada.
- Mantener en un lugar seguro su clave privada y respaldar esta.

- No olvidar la frase de seguridad y mantenerla en secreto.
- Solicitar a la AR a través de medios electrónicos, los certificados digitales de aquellos usuarios con los que tiene una relación operativa.
- Descargar periódicamente las CRL actualizadas.
- Verificar el estado del certificado de él o los clientes con los que se quiera comunicar.
- Firmar, verificar firmas, cifrar, descifrar mensajes, autenticarse con otros usuarios y otras acciones que implican el uso de su par de claves y/o su certificado digital.

Los clientes son la parte más activa dentro de la infraestructura de PKI, por lo tanto la administración de PKI-IMP se orientará a mantener siempre disponibles los servicios que los usuarios deseen utilizar en todo momento de manera eficiente.

6.- Interrelación con otras infraestructuras de llaves públicas externas

No se tiene contemplado que PKI-IMP interactúe con otras infraestructuras PKI

En la siguiente sección, referente a la arquitectura del sistema, abordaremos nuevamente las tareas aquí mencionadas y la manera en que interactúan las entidades anteriores.

REFERENCIAS

Metodología para el diseño de PKI y un ejemplo de PKI para la UCLV.
Lic. Mildrey Carbonell Castro. Lic. José Raúl Barreras Milanés.
Universidad central "Marta Abreu" de las Villas.
http://espejos.unesco.org.uy/simplac2002/Ponencias/Segurm%E1tica/VIR007.doc

X.500 Information technology - Open Systems Interconnection - The Directory: Overview of concepts, models and services
International Telecommunications Union ITU-T
http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.500-200102-I

X.509 Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks
International Telecommunication Union ITU-T
http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-X.509

Public-Key Infrastructure (X.509) (pkix)
The Internet Engineering Task Force Working Group
http://www.ietf.org/html.charters/pkix-charter.html

Internet X.509 Public Key Infrastructure: Roadmap
The Internet Engineering Task Force
Document: draft-ietf-pkix-roadmap-09.txt
http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-09.txt

Internet X.509 Public Key Infrastructure Certificate and CRL Profile
The Internet Engineering Task Force
RFC 2459
http://www.ietf.org/rfc/rfc2459.txt

PKI, Infraestructura de Claves Públicas
Andrew Nash, William Duane, Celia Joseph
RSA Press
Mc Graw Hill
México 2002

2.2 ARQUITECTURA DEL SISTEMA

Introducción

Con base en lo anteriormente expuesto, consideramos que manejar un esquema jerárquico con AC centralizada (basado en PKIX) es lo que cubre mejor nuestras expectativas, por consecuencia este será el modelo que seguiremos de aquí en adelante.
La arquitectura general de PKI-IMP se muestra en la siguiente figura (figura 2.3):

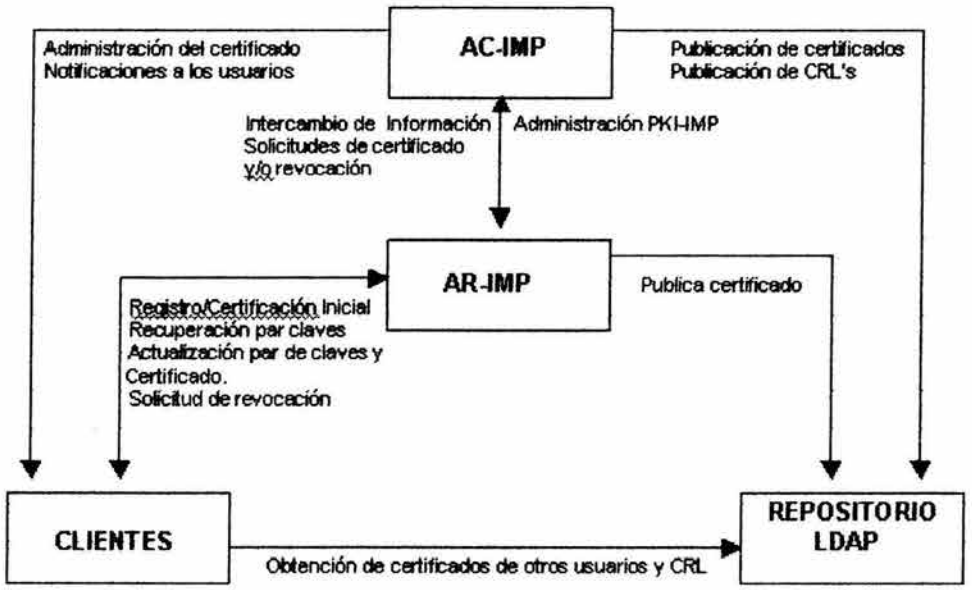


Figura 2.3 Arquitectura PKI-IMP

ESTRUCTURA DE ORGANIZACIÓN

La estructura de organización de PKI-IMP se propone basándose en que se utilice como un modelo el cual sirva en un futuro para implementarse en todo el IMP. Es flexible en el sentido de que es independiente del sistema criptográfico que se use, esto es, las herramientas criptográficas utilizadas son las que cada cliente tenga a su disposición, las cuales se encuentran disponibles en su sistema operativo.

La estructura podrá crecer gradualmente de acuerdo a las necesidades de los clientes y del Instituto. La administración de las claves y del certificado será centralizada, mas no así el proporcionar los servicios y facilidades de PKI-IMP que pueden recaer en varias entidades (otras AR's) interconectadas para satisfacer en forma ágil los requerimientos y necesidades de los usuarios.
El modelo general de organización para el funcionamiento y administración de la PKI-IMP se mostró anteriormente (figura 2.3), en él se encuentran los siguientes módulos participantes:

- Autoridad Certificadora — AC-IMP
- Autoridad Registradora — AR-IMP
- Repositorio de certificados — LDAP
- Clientes o Usuarios

FUNCIONES DE LOS PARTICIPANTES

Las funciones que desempeñan en principio cada uno de los participantes de la PKI-IMP se citaron en la sección anterior,(sección 2.1) a continuación haremos un recuento de dichas funciones pero de manera más simplificada. Estas funciones y tareas se detallan en el siguiente capítulo y se retomarán en los manuales de usuario y administración correspondientes.

Usuario

- Establece su frase de seguridad (password, PIN, contraseña, etc) que proteja su par de claves y genera con ayuda de la AR las mismas (pública y privada).
- Elabora, con ayuda de la AR, su solicitud de certificado digital, esto es, un requerimiento electrónico.
- Acude ante la AR, se identifica y valida su solicitud.
- Recibe su certificado digital.

Autoridad Registradora AR

- Valida la identidad del usuario con base en documentos oficiales de identificación.
- Elabora el requerimiento digital del usuario (solicitud).
- Aprueba dicha solicitud y emite un precertificado digital con base en el requerimiento.
- Envía el precertificado a la AC para su firma
- Entrega al usuario su certificado digital, firmado por la AC, junto con su clave privada.
- Actualización del servicio de directorio con nuevos certificados y CRL emitidas por la AC.

Autoridad Certificadora AC

- Recibe el precertificado y valida la firma electrónica de la AR y del usuario.
- Emite un certificado digital con su firma electrónica.
- Registra la solicitud así como el certificado generado.
- Verifica la unicidad de la clave pública del usuario.
- Registra el certificado en la base de datos, cuando la clave pública es única.
- Registra la clave pública en la base de datos, siempre y cuando ésta sea única.
- Envía el certificado y la clave privada a la AR para su registro y entrega al usuario.
- Da respuesta a la AR sobre la unicidad o duplicidad de la clave pública.
- Lleva el control y administración de los certificados emitidos y de las CRL's

ORGANIZACIÓN

La administración de la PKI-IMP está organizada de la siguiente manera:

La administración de las claves, de la AC, del repositorio de certificados, así como de los certificados mismos se realiza de manera centralizada por el área de seguridad informática del IMP y de las personas que se encuentren al frente de dicha área.

La AC bajo la responsabilidad del Laboratorio de Tecnología Informática (LTI) y en particular del encargado del área de seguridad informática del IMP (M en C. Uriel Tirado Ríos)

La AR cuya operación está bajo la responsabilidad del Área de Seguridad Informática del IMP (Gabriela Espinosa Castillo, Uriel Tirado Ríos), pudiéndose delegar esta responsabilidad a otra persona.

El repositorio de Certificados a cargo del área de Seguridad Informática y en particular de Gabriela Espinosa Castillo o en su defecto de alguna otra persona contemplada para efectuar dicha tarea.

Los clientes tendrán privilegios de uso de herramientas y aplicaciones ofrecidas por PKI-IMP y serán éstas y solo éstas las que los usuarios podrán utilizar. De ninguna manera un usuario acceder a al administración de la AR y mucho menos de la AC.

Como los clientes son todas aquellas aplicaciones y usuarios que utilicen a PKI-IMP, en principio todos los empleados, aplicaciones y servicios existentes en el IMP son clientes potenciales de PKI-IMP.

Con esta sección damos fin a la parte de diseño. La tarea que tenemos a continuación es la construcción e implementación de los módulos de PKI-IMP. El capítulo siguiente está dedicado precisamente a esa actividad. En él detallaremos las actividades necesarias para realizar esto.

REFERENCIAS

Public-Key Infrastructure (X.509) (pkix)
The Internet Engineering Task Force Working Group
http://www.ietf.org/html.charters/pkix-charter.html

Internet X.509 Public Key Infrastructure: Roadmap
The Internet Engineering Task Force
Document: draft-ietf-pkix-roadmap-09.txt
http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-09.txt

Internet X.509 Public Key Infrastructure Certificate and CRL Profile
The Internet Engineering Task Force
RFC 2459
http://www.ietf.org/rfc/rfc2459.txt

Infraestructura Extendida de Seguridad (IES) Banco de México
Dirección General de Operaciones de Banca Central
Dirección de Sistemas Operativos y de Pagos
Banco de México
http://www.banxico.org

PKI, Infraestructura de Claves Públicas
Andrew Nash, William Duane, Celia Joseph
RSA Press
Mc Graw Hill
México 2002

Capítulo 3

Construcción de los Módulos

Introducción

Hasta este momento nos hemos concentrado en la parte teórica del proyecto. Es el momento de poner en práctica lo aprendido hasta ahora y construir los módulos y entidades mencionadas en el capítulo 2. Para esto utilizaremos las herramientas a nuestro alcance y algunas otras que vayan apareciendo durante el transcurso de dicha construcción.
Conviene recordar que todo esto es realizado en la plataforma y con los recursos de software disponibles, los cuales son:

Arquitectura Intel Pentium I 100 MHz
128 MB RAM
1 Disco duro de 2.0 GB
1 Disco duro (extra) de 2.1 GB
Tarjeta de Red 3com 10/100
Sistema Operativo Linux RedHat 7.2
Conjunto de herramientas de desarrollo estándar de Linux

3.1 ENTIDAD CERTIFICADORA

En la sección de Propuesta de este trabajo (Sección 1.2), mencionamos una Entidad Certificadora, dijimos que este sería el módulo encargado de generar, almacenar y revocar los certificados digitales. Con lo estudiado hasta este momento, y siguiendo la arquitectura propuesta en la sección 2.2, vemos que esta entidad está constituida por la totalidad de lo que será el módulo de la Autoridad Certificadora IMP (AC-IMP) y parte de lo que será el módulo de la Autoridad Registradora IMP (AR-IMP).

Retomando la arquitectura del sistema, propuesta en la sección 2.2 (Figura 2.3), identificamos en ella la Entidad Certificadora; esto se muestra en la Figura 3.1-1

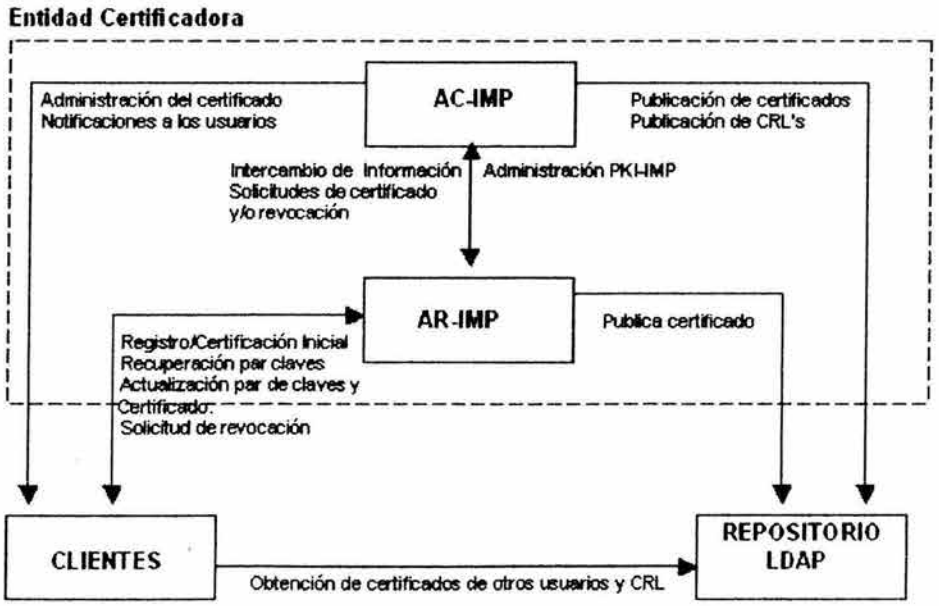


Figura 3.1-1
Entidad Certificadora

Es conveniente anotar, antes de continuar, que la Figura 3.1-1 hace referencia a la "Entidad Certificadora" respecto a los módulos de la arquitectura propuesta de PKI-IMP. Estos módulos en su construcción son independientes entre sí. Los módulos AC-IMP y AR-IMP interoperan para constituir lo que llamamos "Entidad Certificadora". Los módulos "Clientes" y "Repositorio LDAP" aparecen únicamente como referencia para una mejor comprensión, ya que en secciones posteriores se tratará su construcción e implementación.

A continuación, y en referencia a la Figura 3.1-1, presentamos el trabajo que se debe realizar para implementar una Autoridad Certificadora (parte primordial de la Entidad Certificadora), algunas consideraciones que hay que tomar en cuenta y algunas opciones para lograr esto.

3.1.1 AUTORIDAD CERTIFICADORA AC-IMP

La Autoridad Certificadora IMP (AC-IMP) es el módulo principal de la arquitectura propuesta de PKI-IMP (ver Figura 2.3) y es la parte más importante de la Entidad Certificadora (ver figura 3.1-1). Hemos dicho que, tanto para este caso como para todo el proyecto, existen muchas opciones para implementar la solución propuesta. La mayor parte de estas opciones se encuentran en el sector privado, en empresas que se dedican a vender soluciones con productos de software que, aunque en la mayoría de los casos son soluciones fáciles de implementar, resultan bastante costosas e implican una inversión económica y en recursos bastante fuerte tanto para el área responsable como para toda la empresa. Es necesario por tanto, contemplar su adquisición con anticipación cuando la empresa necesita hacer un plan de egresos y/o de inversión de manera anual, así como el justificar dicha adquisición por parte del área responsable en la asignación del presupuesto anual, como es el caso del Laboratorio de Tecnologías de la Información en el IMP, y en particular del Área de Seguridad Informática.

Basándonos en eso, la solución propuesta contempla el uso de herramientas de software libre. Dentro de las cuales, por su importancia, disponibilidad y por las facilidades que ofrecen, seleccionamos dos para este propósito.

La primera de ellas, consideramos, es la que últimamente ha ganado bastante popularidad dentro de la comunidad de software libre. El proyecto OpenSSL es una implementación libre no comercial del protocolo SSL. Además, este incluye herramientas para la administración de certificados y una implementación de clave pública la cual puede ser utilizada fuera de los Estados Unidos sin preocuparse por cuestiones de patentes o derechos reservados. OpenSSL puede ser instalado y utilizado para los siguientes propósitos, distintos pero muy relacionados:

1- Administración de Certificados
2- Implementar SSL en un servidor.

Haremos uso de ambas facilidades al igual que de la implementación de clave pública incluida en el producto.

La segunda (OpenCA), más reciente, hace uso de OpenSSL y de otras herramientas igualmente de libre distribución, para implementar una infraestructura de clave pública PKI. OpenCA cuenta con un proyecto de desarrollo e implementación de una PKI. Para esto OpenCA ha liberado varias distribuciones de su proyecto, el cual está en continuo desarrollo y al mismo tiempo agregando nuevas facilidades. OpenCA incluye las herramientas necesarias para implementar una AC, una AR, un repositorio basado en LDAP, una interfaz web con los clientes y otras herramientas, las cuales, en conjunto son una muy buena opción para los objetivos que perseguimos. Esta es quizá la opción más completa a nuestro alcance y la que quizá adoptaremos para la construcción de PKI-IMP. Dedicaremos una sección completa al estudio e implementación de esta herramienta.

En la medida de lo posible trataremos de utilizar ambas herramientas a la par, comparando ventajas y desventajas, evaluando el rendimiento y potencial de las mismas.

Proyecto OpenSSL

Para llevar a cabo la implementación de AC-IMP utilizando OpenSSL, nos basamos en el documento electrónico llamado "OpenSSL Certifcate Cookbook". Para mayor información se puede consultar este documento en : http://www.pseudonym.org/ssl/sslcertificatecookbook.html

Implementación de una Autoridad Certificadora (AC-IMP)

A continuación indicamos los pasos básicos para implementar una autoridad certificadora, para esto utilizaremos la herramienta OpenSSL.

1- Obtención e Instalación de OpenSSL

La instalación de OpenSSL es muy sencilla. Basta con descargar el paquete de la página de OpenSSL en: http://www.openssl.org/source/index.html y compilarlo. La última versión disponible es la 0.9.7b, cuando se realizó esto por primera vez se hizo con la versión 0.9.7, que es la que actualmente se encuentra instalada y en uso.

Los pasos necesarios para instalar el módulo de OpenSSL se resumen a continuación. El procedimiento completo detallado y la configuración del servicio se encuentran en la Sección I del Anexo B-1 al final de este trabajo.

1) Descomprimir el software
2) Configurar el Software
3) Compilar el software
4) Probar el software
5) Instalar el software

En caso de encontrar errores en la fase de configuración, prueba y/o instalación, habrá que verificar las instrucciones dadas y en su defecto corregir o referirse a la documentación para solucionar el problema.

El archivo de configuración default de OpenSSL está en /usr/local/ssl/openssl.cnf; vale la pena revisarlo para asegurarnos que funcione como lo deseamos o en su defecto modificarlo.

2- Estructura de directorio para la Administración de Certificados

Para poder utilizar a OpenSSL como una herramienta para la creación y administración de una AC, es necesario primeramente crear la estructura de directorios necesaria para esto. El procedimiento a seguir se encuentra en la Sección II del Anexo B-1.

Esta estructura es la recomendada por el grupo de trabajo de OpenSSL, sin embargo se puede seleccionar alguna otra estructura alterna. Es muy importante no olvidar realizar las modificaciones correspondientes en el archivo de configuración de OpenSSL.

Modificando el archivo de configuración de OpenSSL

La actividad complementaria de la estructura de directorio para la Administración de una AC es la modificación del archivo de configuración de OpenSSL. Como se mencionó en el párrafo anterior, es en este archivo donde se indican las opciones y directivas con las que OpenSSL trabajará.

El archivo de configuración de OpenSSL (openssl.cnf) tiene múltiples secciones. Cada sección es utilizada para diferentes propósitos, las secciones incluyen lo siguiente:

CA, CA_default – Define la configuración de la configuración de la Autoridad Certificadora.
Policy_match, policy_anything – Define las diferentes políticas para las solicitudes.
Req, req_distinguished_name, req_attributes – Define los defaults de los requerimientos.

Estas secciones de la configuración deben estar actualizadas antes de que la autoridad certificadora pueda ser utilizada, especialmente la especificación "dir" en la configuración de la AC, la cual define dónde es y será almacenado todo, y que debe ser /usr/local/ssl.

En nuestro caso modificamos la sección de solicitud de certificado ([req],[req_distinguished_name]), los cambios realizados se encuentran en la Sección III del Anexo B-1

Las demás secciones las dejamos sin cambios para poder utilizar el script CA.pl y generar un certificado autofirmado.

3- Creación y auto-firma de nuestro certificado

El siguiente paso que debemos dar es crear un certificado firmado por nosotros mismos. Será con este certificado que en el futuro nosotros podremos firmar los certificados que nos sean solicitados. Utilizamos el script CA.pl localizado en /usr/local/ssl/misc para poder concentrarnos más en el procedimiento que en los detalles sintácticos.

A grandes rasgos se realizan los siguientes pasos:

1) Generar clave privada
2) Generar requerimiento (solicitud) de certificación
3) Firmar el requerimiento con la clave privada generada en el paso 1
4) Reubicar clave privada y certificado donde OpenSSL lo requiera.

Los detalles y resultados de este procedimiento se encuentran en la Sección IV del Anexo B-1

El archivo de configuración de OpenSSL para AC(openssl-ca.cnf) se encuentra en la sección V del Anexo B-1.

El script CA.pl se encuentra en el Anexo B-2

Hasta este momento lo que obtuvimos es una clave privada y un certificado firmado con esta clave privada. Esto es, la clave privada de la AC-IMP y un certificado a su nombre autofirmado.

Estos dos elementos conforman la base de la AC. Es el momento de hacer cambios en el archivo de configuración de openssl en la sección de [CA_default] y de asegurarnos de que utilice estos elementos. Por comodidad podemos renombrar estos archivos y/o modificar su ubicación. Hay que tener muy en cuenta que cualquier cambio que queramos hacer sobre estos archivos hay que reportarlo en el archivo de configuración openssl.cnf, de lo contrario cuando queramos utilizarlos y openssl no los encuentre o no concuerde con lo especificado en su configuración generará un error.

Por cuestiones de administración, y a fin de tener un mejor control y cuidado tanto del certificado de la AC como de su clave privada, dentro de /usr/local/ssl/ creamos un directorio llamado CAServer y ahí movimos tanto el certificado como la clave privada de la AC. Adicionalmente realizamos una copia de newcert.pem y la guardamos como cacert.pem. Será este último archivo el que utilicemos como certificado de la AC. Igualmente creamos los archivos SerialCA e IndexCA.txt dentro de /usr/local/ssl a

fin de utilizar estos con la nueva AC. Recordamos que el archivo de configuración openssl.cnf así como el script CA.pl se encuentran en los Anexo B-1 y B-2 respectivamente.

Al llegar a este punto damos por terminado el procedimiento para la implementación de una Autoridad Certificadora utilizando el conjunto de herramientas de OpenSSL, lo que hemos hecho son a grandes rasgos los paso básicos para realizar esto.

Hasta este momento solo hemos utilizado una de las tantas funciones que OpenSSL soporta porque así lo hemos requerido. Una referencia completa y detallada de OpenSSL puede ser consultada en la documentación incluida en la distribución del software o en la sección de documentación del sitio oficial del proyecto (ver referencias al final de esta sección).

Nótese que el procedimiento lleva un orden secuencial de actividades, a simple vista parecieran pasos muy sencillos y sin tanta complejidad. La Figura 3.1-2 corresponde al diagrama de flujo de lo que se realiza para llevara cabo la implementación de una AC utilizando OpenSSL :

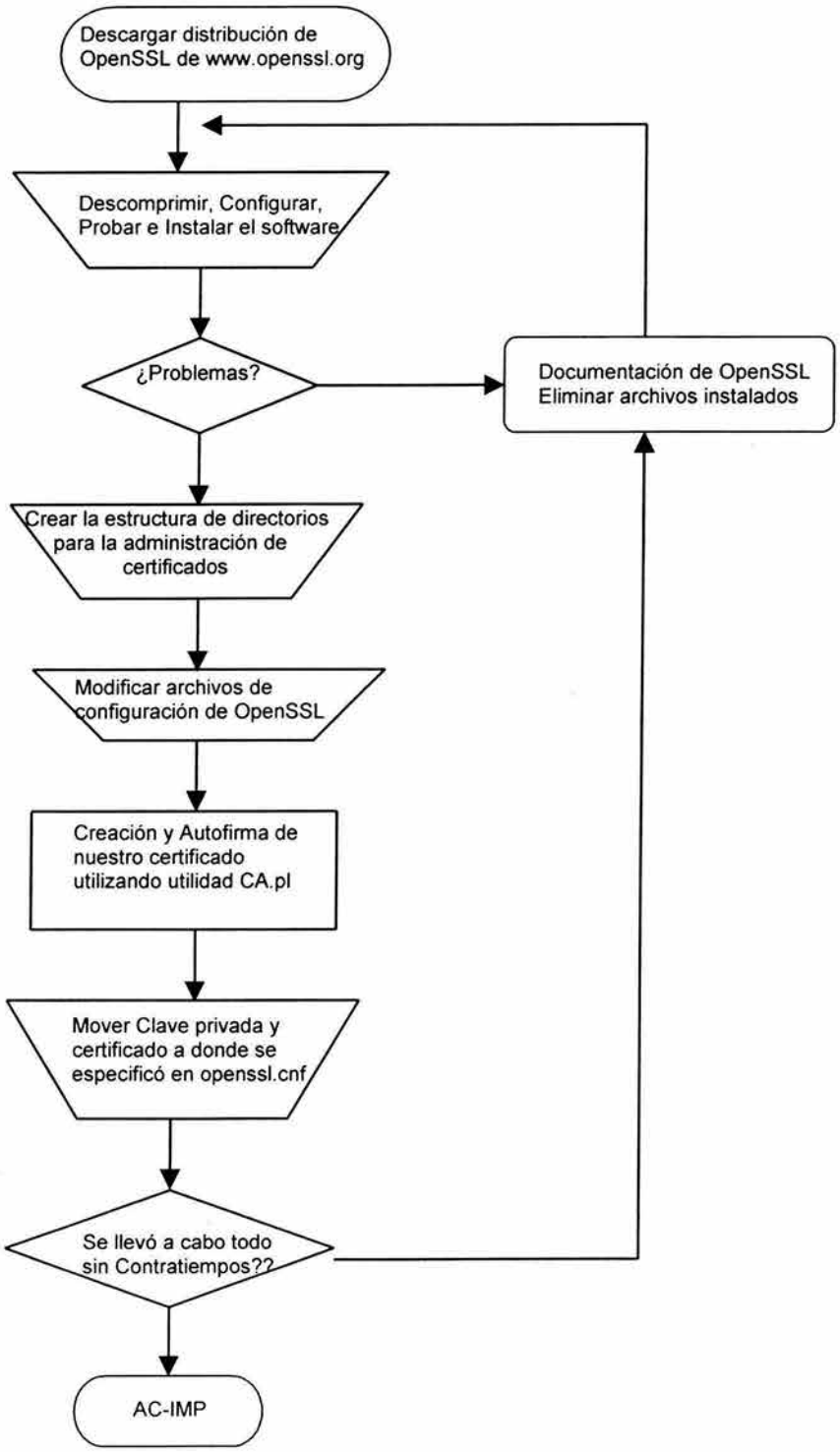


Figura 3.1-2 Diagrama de Flujo
Implementación de una AC con OpenSSL

Creación e instalación de certificados para servidor Web

Este paso es muy similar al anterior, o aún más sencillo. La única particularidad es que tendremos que utilizar un servidor Web capaz de entender y procesar peticiones cifradas, utilizando el protocolo https en vez del http normal.

Para esto instalamos el servidor web Apache con el módulo de ssl mod_ssl, ya que es software 100% libre, su instalación y configuración no presenta gran complejidad, y ofrece todas las características que podemos encontrar en cualquier servidor Web comercial.

A continuación incluimos los pasos necesarios para descargar e instalar el paquete que contiene a Apache+mod_ssl.

Obtención, configuración e Instalación del Servidor Apache

El servidor web de apache se puede descargar desde el sitio de Apache: (http://www.apache.org/distribution/apache-2.0.44.html). En este sitio se encuentran igualmente las últimas versiones del producto así como los respectivos parches de seguridad.

De igual manera se puede descargar desde cualquier otro sitio "espejo" (mirror) disponible. Se recomienda utilizar el de la UNAM ya que se encuentra actualizado, su ubicación es: ftp://mirrors.unam.mx/apache

La documentación se refiere a la versión 2.0.44, aunque actualmente se encuentra instalada la 2.0.45 y la última versión en el sitio es la 2.0.46

De la misma manera que para OpenSSL, a grandes rasgos se realizan los siguientes pasos:

1) Descomprimir el software
2) Configurar el Software
3) Compilar el software
4) Instalar el software
5) Probar el software

El procedimiento detallado se encuentra en el Anexo B-3 al final de este trabajo.

Es conveniente tener a la mano la documentación de la distribución en caso de contratiempos o problemas encontrados en esta fase. De igual manera en la documentación se encuentran las diferentes opciones de configuración de los servicios de Apache.

El siguiente paso es modificar el archivo httpd.conf en /usr/local/apache2/ de tal manera que el servidor trabaje como nosotros queremos.

Las directivas de configuración de Apache están agrupadas en tres secciones básicas principales:

1- Directivas que controlan la operación de los procesos del servidor Apache como un todo. (el "ambiente global").
2- Directivas que definen los parámetros del servidor "principal" o del "default", las cuales responden a peticiones que no son manejadas o atendidas por un host virtual. Estas directivas también proporcionan valores por defecto para la configuración de todos los host virtuales.
3- Configuración para los host virtuales, la cual permite a las solicitudes web ser enviadas a distintas direcciones IP o hostnames y manejarlas con el mismo proceso del servidor.

El primer paso es indicarle al servidor la dirección IP y/o los puertos a los cuales se va a vincular o a "escuchar", esto es, el o los puertos por los cuales el servidor apache nos podrá atender. Esto se realiza mediante la directiva Listen. Refiérase a la documentación para mayor información.

A continuación, y solo unas líneas mas adelante, se encuentra la directiva ServerName. Esta directiva da el nombre y puerto que el servidor utiliza para identificarse así mismo. Esto a menudo puede determinarse automáticamente, pero se recomienda especificar esto específicamente a fin de prevenir problemas durante el inicio del servicio.

Si el servidor no cuenta con un nombre registrado en el DNS de la organización, entonces se introduce la dirección IP. Se podrá acceder al servicio de igual manera pero tecleando la dirección IP en lugar del nombre del servidor.

```
ServerName raguel.imp.mx:80
# ServerName 192.168.144.102:80
```

La segunda opción es para el caso de que el nombre del servidor no esté dado de alta en el DNS.

Tal vez la directiva más importante sea la que se encuentra justo al final de la sección 2, esta se refiere al módulo ssl y es así:

```
<IfModule mod_ssl.c>
        Include conf/ssl.conf
<IfModule>
```

Esta directiva le indica al servidor Apache que, si encuentra el módulo mod_ssl compilado y disponible, procese el archivo de configuración ssl.conf dentro del directorio apache2/conf .Es en este archivo en donde le indicamos al servidor cómo debe manejar las solicitudes que se hagan utilizando el protocolo SSL.

El archivo de configuración ssl.conf lo trataremos más adelante, ya que para esto es necesario antes contar con un certificado para servidor web y su respectiva clave privada.

Obtención de un Certificado para servidor web

Un certificado digital y una clave privada, permite a un servidor web establecer conexiones seguras (cifradas) con los clientes. Además sirve como identificador por medio del cual el servidor se autentica con el cliente. El cliente tiene la seguridad de estar "hablando" con un servidor en el cual puede depositar toda su confianza ya que ha sido certificado por una Autoridad Certificadora.

Para obtener un certificado para servidor web, a grandes rasgos diremos que se realizan los siguientes pasos:

1) Generar clave privada del servidor
2) Generar requerimiento (solicitud) de certificación
3) Enviar el requerimiento a la AC para que esta lo firme con su clave privada y emita el correspondiente certificado.
4) Reubicar clave privada y certificado donde Apache lo requiera.

Los detalles y resultados de este procedimiento se encuentran en la Sección II del Anexo B-3

Lo que a continuación toca es explicar y modificar el archivo de configuración ssl.conf en:
/usr/local/apache2/conf /

El archivo ssl.conf contiene la configuración que le permite al servidor Apache soportar comunicaciones SSL. Esto le indica al servidor como debe servir páginas sobre una conexión https. Las modificaciones hechas a la configuración de ssl.conf se encuentran en la sección III del Anexo B-3 al final de este trabajo.

Como ya lo mencionamos anteriormente, si bien el certificado que acabamos de emitir es completamente válido y las sesiones SSL que se establezcan viajarán siempre encriptadas, los navegadores nos enviarán advertencias al entrar a un sitio protegido por éste certificado, puesto que la autoridad certificadora que lo emite no es reconocida internacionalmente o no se encuentra dentro de las Autoridades de confianza.

Hasta este momento tenemos:

- Una Autoridad Certificadora de autoconfianza capaz de generar solicitudes y certificados digitales, así como de crear el par de claves pública/privada.
- Un conjunto de herramientas en la AC basadas en PKI, que permiten la administración de dichos certificados.
- Un servidor web capaz de atender y procesar peticiones hechas mediante el protocolo http y https, el segundo nos permite tener una comunicación segura ya que todo la información intercambiada viajará cifrada.

Con lo hecho hasta ahora podemos estar seguros de que contamos con una AC con toda la infraestructura necesaria para actuar como tal. Si bien es cierto que puede resultar un tanto cuanto complicado el manejo de solicitudes y la generación y firma de certificados, consideramos que esta es una manera muy sencilla de implementar una AC.

En el anexo B-4 se encuentran los archivos completos de configuración httpd.conf así como el ssl.conf. Conviene echarle un vistazo a ambos para entender mejor como funciona esto.

3.1.2 AUTORIDAD REGISTRADORA AR-IMP

Dentro de la Entidad Certificadora se encuentra una parte del módulo de la Autoridad Registradora (ver Figura 3.1-1), aunque el módulo de la AR-IMP no emite un certificado como tal, sí genera el requerimiento del mismo así como el par de claves, además de ser esta (la AR-IMP) quien se encarga de validar la identidad de los usuarios, de distribuir los certificados emitidos y de mantener una continuidad operativa de la interfaz con los usuarios tanto actuales como nuevos.

La manera de implementar una AR con OpenSSL, la podemos reducir a contar con un servidor web seguro, el cual ya lo tenemos, y el crear un certificado a nombre de Autoridad Registradora IMP, con estos elementos creamos una estructura idéntica que a la que utilizamos cuando creamos AC-IMP. La diferencia radica en que AR-IMP no genera certificados ni firma el mismo, solo crea solicitudes y el par de claves.

La razón del servidor web seguro es que esta sería la interfaz con los clientes. Por medio de este servidor la AR recibe las solicitudes de certificación, revocación, petición de CRL, etc; así como de los datos de los usuarios durante el proceso de solicitud y envía igualmente el certificado mismo de él y/o de los usuarios cuando éste ya se encuentre listo.

Existe un punto importante que no debemos olvidar. Al momento de generar el certificado para la AR, es importante hacer algunos cambios al archivo de configuración openssl.cnf, esto con el objeto de darle al certificado generado los atributos necesarios para poder ser utilizado como un certificado de AR. No es necesario realizar cambios en el archivo de configuración de ssl cada vez que se quiera utilizar, lo recomendable es contar con varios archivos de configuración tales como: Web_server.cnf,

User.cnf, VPN_server.cnf, AR_server.cnf, CA_Server.cnf, etc, y simplemente indicarle a ssl en la línea de comandos cuál debe utilizar.

En el anexo B-1 secciones V, VI y VII se presentan varias opciones de configuración basadas en la finalidad del certificado.

Es conveniente mencionar, aunque aquí no se utilizará, el caso en el que se requiera contar con más de una Autoridad Registradora. Si esto fuera necesario, el procedimiento a seguir para implementar otra AR es muy sencillo, dado que el modelo jerárquico que estamos manejando. Se procede como a continuación se indica:

- Crear un nuevo archivo de configuración de ssl (por ejemplo AR-2.cnf) o modificar el existente.
- Generar solicitud de certificación a nombre de la nueva AR.
- Enviar la solicitud a la AC para su firma y emisión del nuevo certificado.
- Configurar e instalar OpenSSL y Apache en el caso de que la nueva AR vaya a ser implementada en otro equipo.
- Crear la estructura de directorios para la AR nueva.
- Descargar certificado y par de claves y llevarlos a donde la nueva AR lo requiera.
- Levantar el servidor web seguro (https)

En este caso la Entidad Certificadora incluiría más de una AR, para efectos de operación esto no representa mayor problema ya que ahora en lugar de un solo punto, tendríamos dos puntos de atención a los usuarios. Lo que si representaría un cambio es en las labores de administración, ya que tendríamos que administrar no solo a uno sino a más servidores, todos dependientes de una sola AC.

Con lo anterior concluimos los trabajos referentes a Entidad Certificadora utilizando como herramienta principal a OpenSSL. En esta sección se han utilizado términos y conceptos como Autoridad Certificadora, Autoridad Registradora, Servidor web seguro, conexiones SSL, autenticación del servidor, entre otras muchas cosas. Todas ellos fueron tratadas y definidos en la Sección 1.1 por lo cual no nos detuvimos en la parte teórica. Si tiene alguna duda en cuanto a teoría, refiérase a esta sección o a las referencias incluidas al final de la misma.

En la siguiente sección trataremos al elemento más importante, por su actividad, dentro de la infraestructura de PKI-IMP: Los clientes.

REFERENCIAS

The OpenSSL Certifcate Cookbook
http://www.pseudonym.org/ssl/sslcertificatecookbook.html

Proyecto OpenSSL
http://www.openssl.org

Implementación de una autoridad certificadora con OpenSSL
http://www.gwolf.cx/seguridad/pki/node1.html

Apache Software Foundation
Project HTTP-Server
http://www.apache.org

3.2 CLIENTES

En esta sección toca referirnos a los clientes de la infraestructura. En la arquitectura de PKI-IMP propuesta en la sección 2.2 (ver ref. Figura 3.1-1) se identifica claramente este módulo, retomamos la arquitectura y mostramos dicho módulo en la figura 3.2-1.

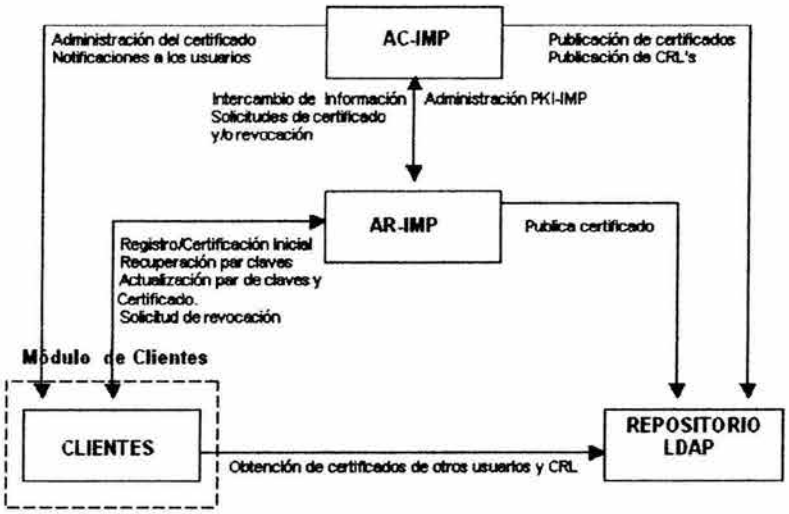


Figura 3.2-1
Módulo Clientes

Hemos mencionado que un cliente puede ser tanto un empleado de la organización, como algún equipo o aplicación en ejecución. Debido a esto, en principio no existe un modelo fijo definido de lo que es un cliente. En principio un cliente será toda aquella entidad final que haga uso de los certificados emitidos por la infraestructura PKI-IMP y de sus claves pública/privada asociadas.

Hemos dicho igualmente que lo más conveniente es contar con una interfaz web para el intercambio de información con los clientes. Esta Interfaz estará levantada en un servidor apache seguro, de manera que la información intercambiada sea segura y confiable tanto para el usuario como para la Infraestructura misma.

Cuando se concibió PKI-IMP se tuvo la intención y el deseo de contar con una aplicación que fuese capaz de utilizar tanto el certificado emitido como la clave privada del usuario para poder brindar servicios de firma digital y verificación de la misma, cifrado y descifrado de documentos electrónicos y de manera muy ambiciosa servicio de no repudio por medio de auditorias. Esto sin embargo queda fuera de los alcances y de los objetivos de esta tesis, por lo cual quedará como un trabajo futuro.

Sin embargo, lo que si podemos proporcionar, es una serie de "servicios mejorados". Esto es, servicios que actualmente están en uso pero que, con la utilización de un certificado digital y de sus claves asociadas, pueden hacer que dicho servicio ofrezca herramientas que nos ayuden a explotar servicios de seguridad que tienen habilitados.

El primer servicio que contemplamos, por su facilidad de implementación y su disponibilidad, es el de Correo Seguro.

Actualmente los clientes más populares de correo (MS Outlook, MS OutlookExpress, Netscape Web mail, Eudora[1] y algunos otros en LINUX) incluyen la opción de manejar correo seguro, previa configuración del servicio y condicionado a contar con un certificado digital y con una clave privada.

[1] Solo en algunas distribuciones

Las aplicaciones anteriores pueden ser nuestros primeros "clientes" ya que ellos cuentan con librerías y herramientas criptográficas para crear y verificar firmas digitales, para cifrar y descifrar el contenido de los mensajes, y para crear y abrir los denominados "sobres digitales".

Otro servicio que puede hacer uso de certificados es el control de acceso a sitios web o a directorios por medio de la inclusión de directivas en el servidor que indiquen que la autenticación del cliente es requerida y que dicha autenticación será realzada mediante el uso de un certificados digitales. Por el lado del cliente esto es muy sencillo ya que la mayoría de los navegadores (browser) actuales soportan esta opción, ellos cuentan con librerías y procedimientos criptográficos transparentes para el usuario y no requieren de complicadas tareas de configuración. Del lado del servidor, este debe simplemente configurarse adecuadamente para que sea capaz de manejar y negociar un conjunto de protocolos criptográficos necesarios en la autenticación por medio de certificados. Con esto se garantiza que exista una autenticación bidireccional, del servidor hacia el cliente como del cliente hacia el servidor, con esto se garantiza que solo los usuarios autorizados tiene acceso a la información solicitada.

Lo anterior requiere adicionalmente un trabajo arduo y extenso de administración de políticas así como de configuración de servicios, de manera que el servicio final proporcionado sea de calidad y que garantice que cumple con las tareas por las cuales fue concebido.

Igualmente se tiene en mente el poder configurar y gestionar el uso de certificados en el servicio de VPN.

El IMP cuenta actualmente con un servicio de VPN para sus usuarios viajeros o usuarios fuera de la red IMP. Dicho servicio actualmente se maneja mediante autenticación basada en la asignación y uso de un nombre de usuario y un password. Este mecanismo como ya lo hemos dicho antes, es deficiente y presenta muchas debilidades en cuanto a seguridad y privacidad se refiere, los nombres de usuario y passwords en muchas ocasiones se comparten entre usuarios VPN y no hay una certeza de que quien está solicitando el servicio sea realmente quien dice ser.

Adicionalmente el servicio de VPN soporta la autenticación vía certificado digital. En este caso la dificultad radica en que el software que proporciona y administra dicho servicio de VPN es un producto bajo licencia, y posiblemente no sea tan "abierto" como para poder permitir el uso de certificados de Autoridades Certificadoras no reconocidas Internacionalmente en las conexiones y validaciones que él realice.

Por la naturaleza del software, el servicio de VPN basado en certificado digital no se tratará en este trabajo, pero sí será contemplado como un trabajo futuro. Sin embargo si recalcamos que este servicio es un cliente potencial de PKI-IMP.

Existen muchas más aplicaciones en el medio comercial que pueden hacer uso de los certificados digitales. Todas ellas son clientes potenciales de la infraestructura PKI-IMP. Lo que a nosotros concierne es garantizar que las facilidades y/o condiciones necesarias ó requeridas por dichas aplicaciones para el uso de certificados, estén disponibles para poder explotar las herramientas criptográficas que ellas ofrezcan.

En la siguiente sección trataremos un módulo que pudiese ser cliente y no solo parte de PKI-IMP (dependiendo de la configuración) que es el servicio de directorio LDAP. LDAP es capaz de proporcionar servicios utilizando conexiones SSL.

Lo referente a la construcción, configuración y operación de dicho módulo lo veremos a continuación.

3.3 SERVICIO DE DIRECTORIO

Introducción

A continuación abordaremos un módulo muy importante dentro de la infraestructura de PKI que es el Repositorio de Certificados. Este repositorio contiene, además de los certificados digitales de los usuarios de la infraestructura, la lista de revocación actual (vigente). Como veremos en el desarrollo de la presente sección, el servicio de directorio LDAP cuenta con todas las facilidades y herramientas necesarias para actuar como un Repositorio de Certificados y de CRL. En la arquitectura propuesta de PKI-IMP se distingue fácilmente este módulo (ver ref. Figura 3.1-1). Retomamos dicha arquitectura y mostramos este módulo en la Figura 3.3-1

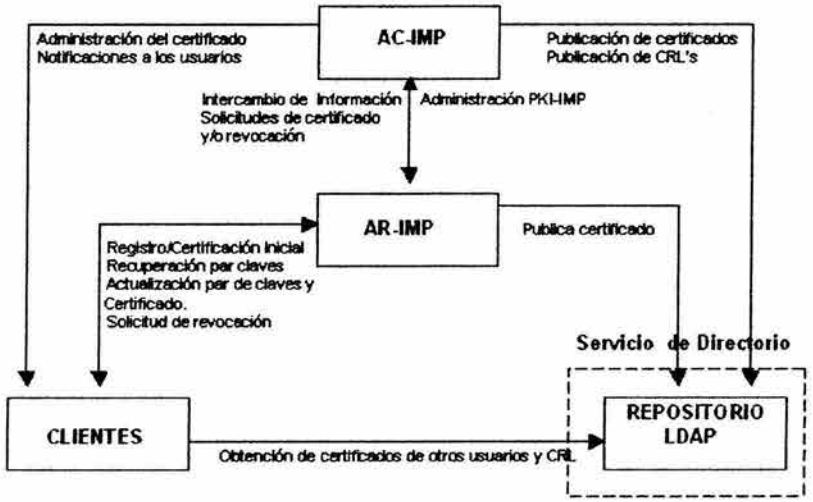


Figura 3.3-1
Servicio de Directorio

Servicio de Directorio

Un **Directorio** es una base de datos especializada optimizada para la lectura, exploración y búsqueda[1]. Los directorios tienden a contener información descriptiva y basada en atributos que además soportan sofisticadas capacidades de filtrado. Los directorios generalmente no soportan transacciones complicadas o esquemas de reducción o roll-back[2] (retorno de listas) encontrados comúnmente en los sistemas manejadores de bases de datos (RDBMS) diseñados para manejar grandes volúmenes de datos con complejas actualizaciones.

Las actualizaciones (updates) de los directorios son usualmente simples cambios de "todo o nada", si es que les es permitido. Los directorios son ajustados para proporcionar una respuesta rápida para grandes volúmenes de operaciones de búsquedas o consultas. Estos pueden tener la capacidad de duplicar (replicar) la información en otros servidores o copiar la información ampliamente de manera que se incremente la disponibilidad y confiabilidad de la misma, reduciendo con ello el tiempo de respuesta.
Cuando la información de un directorio es replicada, temporalmente algunas inconsistencias entre réplicas pudiesen ser aprobadas o aceptadas entre la información que hay en las réplicas, siempre y cuando se encuentren eventualmente en sincronía.

[1]The Directory Service, OpenLDAP Software http://www.openldap.org/doc/admin21/intro.html
[2] Término empleado para definir el resultado que se obtiene cuando se realiza una consulta a una base de datos relacional (RDB) a través de un manejador de bases de datos relacionales (RDBM)

Hay muchas otras maneras de proporcionar un servicio de directorio. Los diversos métodos permiten almacenar diferentes clases de información en el directorio, establecen diversos requisitos sobre cómo se puede uno referir a dicha información, cómo puede ser solicitada, actualizada, o cómo es protegida de accesos no autorizados, etc. Algunos servicios de directorio son locales, proporcionando servicio en un contexto restringido (e.g. el servicio de finger[3] en una terminal Unix simple). Algunos otros servicios son globales, proporcionando servicio a un amplio contexto o grupo (e.g. Internet). Los servicios globales son generalmente distribuidos, esto implica que los datos que ellos contienen se difunden a través de muchas máquinas, todas ellas colaboran para proporcionar el servicio de directorio. Típicamente un servicio global define un "espacio de nombres" uniforme el cual da la misma visión de los datos sin importar dónde se esté en relación con el dato mismo. El Sistema de Nombres de Dominio de Internet (DNS) es un ejemplo de un servicio de directorio global distribuido.

Servicio de Directorio LDAP

LDAP es la representación de Lightweight Directory Acces Protocol (Protocolo Ligero de Acceso a Directorio). Como el nombre lo sugiere, este es un protocolo "ligero" (sencillo) para accesar a los servicios de un directorio, específicamente a los servicios basados en el servicio de directorio X500. LDAP corre sobre TCP/IP o sobre algún otro servicio de transferencia orientado a conexión. Los detalles específicos de LDAP están definidos en **RFC[4] 2251** "The Lightweight Directory Access Protocol (v3)" y otros documentos más comprenden la especificación técnica **RFC 3377**. En esta sección damos una visión general de LDAP desde la perspectiva del usuario.

Tipo de Información que puede ser almacenada en el directorio

El modelo de información de LDAP está basado en "entradas". Una entrada es una colección de atributos que tienen un único Nombre Distinguido global (Distinguished Name: DN). El DN es utilizado para referirse a dicha entrada de manera no ambigua. Cada uno de los atributos de las entradas tiene un "tipo" y uno o más "valores". Los "tipos" son comúnmente cadenas de mnemónicos, como "cn" para el nombre común (common name) , o "mail" para la dirección de correo. La sintaxis de valores depende del tipo de atributo. Por ejemplo, un atributo "cn" puede quizá contener el valor Barbara Jensen. Un atributo "mail" puede contener el valor babs@ejemplo.com, un atributo jpegPhoto pudiese contener una fotografía en formato binario JPEG.

Organización de la Información

En LDAP, las entradas al directorio son acomodadas en una estructura de árbol jerárquica. Tradicionalmente, esta estructura refleja el límite geográfico u organizacional. Las entradas representando países aparecen en el tope (raíz) del árbol. Abajo de ellas se encuentran las entradas representando estados u organizaciones nacionales. Abajo de ellas se pueden encontrarse entradas representando unidades organizacionales, personas, impresoras, documentos, o cualquier otra cosa que uno se pueda imaginar.

La figura 3.3-2 muestra un ejemplo de un árbol de directorio LDAP que utiliza el nombrado tradicional.

[3] finger .- programa de ambiente UNIX que toma una dirección de un correo como una entrada y devuelve la información sobre el usuario de esa dirección de e-mail

[4] RFC- Request For Comments, estándares publicados por la IETF "Internet Engineering Task Force" www.ietf.org

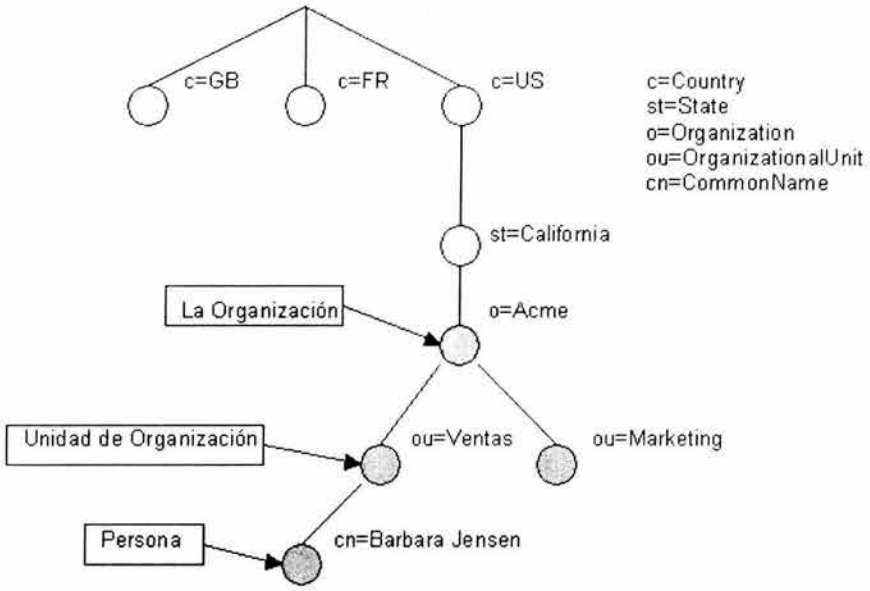


Figura 3.3-2
Árbol de directorio de LDAP (nombrado tradicional)

El árbol puede también estar organizado basándose en los nombres de dominio de Internet. Este enfoque de nombrado está incrementando su popularidad ya que permite a los servicios de directorio ser localizados utilizando el DNS.

La figura 3.3-3 muestra un ejemplo de un árbol de directorio LDAP que utiliza nombrado basado en el componente de dominio (dc).

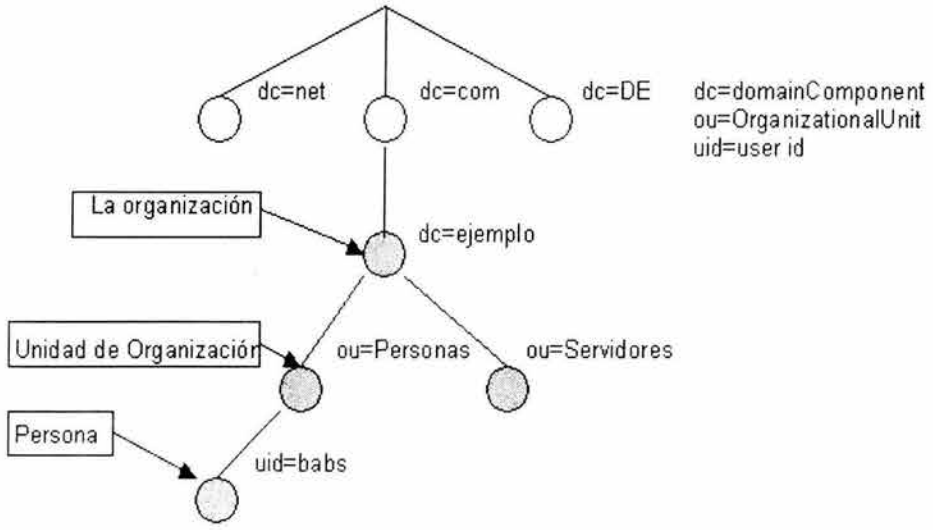


Figura 3.3-3: Árbol de directorio LDAP
(Nombrado basado en componentes de Internet, dc)

Además, LDAP nos permite controlar qué atributos son requeridos y permitidos en una entrada a través del uso de un atributo especial llamado "clase de objeto" u objectClass. Los valores de un atributo objectClass determinan las reglas de "diseño" (schema) que la entrada debe obedecer.

Referencias a la Información

Una entrada está referenciada (es referida) por su nombre distinguido, el cual se construye tomando el nombre de la entrada misma (llamado el Nombre Distinguido Relativo RDN) y concantenando el nombre de sus entradas predecesoras.

Por ejemplo, la entrada para Barbara Jensen en la Figura 3.3-3 basado en nombrado de Internet, tiene un RDN de uid=babs y un DN: uid=babs,ou=people,dc=example,dc=com. El formato completo de DN esta descrito en RFC2253, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names."

Acceso a la información

LDAP define operaciones para interrogar o actualizar el directorio. Las operaciones son proporcionadas para agregar y borrar una entrada del directorio, para cambiar una entrada existente, y para cambiar el nombre de una entrada. La mayor parte del tiempo, sin embargo, LDAP es utilizado para buscar información en el directorio. La operación de búsqueda de LDAP autoriza cierta porción del directorio para realizar la búsqueda de las entradas que empatan con algún criterio especificado por un filtro de búsqueda. La información puede ser solicitada en cada entrada que empate con el criterio.

Protección de la información en accesos no autorizados

Algunos servicios de directorio no proporcionan protección, permitiendo a cualquiera ver la información que contienen. LDAP proporciona un mecanismo para que el cliente se autentique, o proporcione su identidad al servidor de directorio, facilitando las cosas para un control de acceso "sustancioso" que proteja la información contenida en el servidor. LDAP también soporta los servicios de seguridad de privacidad y de integridad.

Funcionamiento de LDAP

El servicio de directorio LDAP esta basado en el modelo cliente-servidor. Uno o más servidores de LDAP contienen los datos de la estructura del "árbol de información del directorio " (Directory Information Tree, DIT). El cliente se conecta a los servidores y realiza una pregunta. El servidor contesta con una respuesta y/o con un apuntador a donde el cliente puede obtener información adicional (típicamente otro servidor LDAP). No importa a cual servidor LDAP se conecte el cliente, este tiene la misma vista del directorio, un nombre presentado a un servidor LDAP referencía la misma entrada que pudiese tener en otro servidor LDAP. Esta es una característica importante de un servicio de directorio global, como lo es LDAP.

X.500

Técnicamente, LDAP es un protocolo de acceso al directorio para un servicio de directorio X.500, el servicio de directorio OSI[5]. Inicialmente los clientes LDAP accesaban gateways al servicio de directorio X.500. Este gateway ejecutaba LDAP entre en cliente y el gateway, y el protocolo de acceso al directorio (Directory Acces Protocol, DAP) X.500 entre el gateway y el servidor X.500. DAP es un protocolo demasiado pesado que opera sobre una completa pila de protocolos OSI y requiere una significativa cantidad de recursos computacionales. LDAP esta diseñado para operar sobre TCP/IP y proporcionar muchas de las funcionalidades de DAP a un costo mucho más bajo.

Aunque LDAP continua siendo utilizado para accesar servicios de directorio X.500 vía gateways, LDAP es ahora comúnmente implementado directamente en servidores X.500.

[5] OSI.- Open Source Interconection

Diferencias entre LDAPv2 y LDAPv3

LDAPv3 fue desarrollado a finales de 1990 para sustituir a LDAPv2. LDAPv3 añade las siguientes facilidades a LDAP:

- Autenticación fuerte vía SASL
- Protección de la Integridad y confidencialidad vía TLS(SSL)
- Internacionalización a través del uso de Unicode
- Referencias y Continuación
- Descubrimiento de Esquemas.
- Extensibilidad (controles, operaciones extendidas y más.)

LDAPv2 es considerado histórico, El desplegar servicios con LDAPv2 y LDAPv3 simultáneamente puede ser un tanto problemático.

SLAPD y sus utilidades

Slapd es un servidor de directorio LDAP incluido en la distribución de OpenLDAP (http://www.openldap.org), que corre en varias y distintas plataformas. Podemos utilizar este para proporcionar un servicio e directorio nosotros mismos. Nuestro directorio puede contener lo que nosotros queramos colocar en él. Lo podemos conectar al servicio de directorio global, o simplemente correr un servicio nosotros mismos. Algunas de las características y posibilidades que ofrece slapd incluyen:

- LDAPv3: slapd implementa la versión 3 de LDAP. Slapd soporta LDAP sobre IPv4 como IPv6.

- SASL: slapd soporta servicios de autenticación fuerte a través del uso de SASL.

- TLS: slapd proporciona protección de la Integridad y privacidad de la información por medio del uso de TLS o SSL. Esto lo hace con ayuda de OpenSSL.

- Control de topología: slapd permite restringir el acceso al servidor basándose en la topología de la red.

- Control de Acceso: slapd proporciona una rica y poderosa facilidad en el control de acceso, permitiéndonos controlar los accesos a la información en nuestras bases de datos. Se puede controlar el acceso a las entradas basado en la autorización de Información de LDAP, direcciones IP, nombre de dominio etc. Slapd soporta controles de acceso tanto estáticos como dinámicos.

- Internacionalización: slapd permite el uso de Unicode.

- Selección del modulo terminal (backend) de base de datos: slapd viene con una variedad de diferentes módulos terminales de base de datos para elegir. Entre estos se incluye BDB (Berkeley Data Base) este es un backend de transacciones de base de datos de alto desempeño; LDBM, un administrador de base de datos (DBM) "ligero" y otras más. BDB utiliza la base de datos de SleepyCat BerkeleyDB.

- Múltiples instancias de base de datos: slapd puede ser configurado para atender múltiples bases de datos al mismo tiempo. Esto significa que un servidor slapd puede responder a solicitudes de muchas porciones lógicas del árbol de LDAP, utilizando el mismo o un backend de base de datos diferente.

Existen aún varias características más que proporciona slapd, sin embargo creemos que las anteriores, para nosotros, son las más importantes.

Por esta y por otras razones elegimos a LDAP como el servicio de directorio que utilizaremos para los fines que se persiguen en este trabajo de tesis.

El recurso a utilizar tiene el nombre de OpenLDAP, nombre que recibe el proyecto OpenLDAP[6] Software el cual es una implementación de libre distribución (open source) del protocolo LDAP (Lightweight Directory Access Protocol).

Esta característica adicional hace de OpenLDAP la herramienta ideal para la implementación que se requiere. La obtención, requisitos, configuración, instalación y customización del software es muy sencilla. En la distribución del software se incluye la documentación del mismo, igualmente en el sitio del proyecto se encuentra la documentación correspondiente, así como algunas consideraciones extras. En la siguiente sección resumiremos lo que el Proyecto PKI-IMP realizó para poder contar con el servicio de directorio OpenLDAP.

Descarga, Instalación y Configuración de OpenLDAP

El procedimiento es similar al realizado con OpenSSL y Apache, a continuación resumimos los pasos que se realizan.

Para el caso de PKI-IMP, los detalles y recomendaciones para implementar un servicio de directorio utilizando OpenLDAP se encuentran en la Sección I del Anexo C-1 (Ver).

A continuación resumimos el procedimiento realizado.

Servicio de Directorio de PKI-IMP

1. Descargar el Software
2. Desempaquetar el software
3. Cumplir con los pre-requisitos de software
4. Configurar OpenLDAP
5. Instalar OpenLDAP
6. Configurar el Servidor

La Figura 3.3-4 corresponde al diagrama de flujo de lo que se realiza para llevar a cabo la implementación del Servicio de Directorio con OpenLDAP.

Una vez que llegamos a este punto, ya tenemos un servidor LDAP funcionando, listo para ser llenado con información.

La manera estándar para introducir información a un directorio LDAP, es crear un archivo LDIF (Formato de intercambio de directorios LDAP, LDAP Directory Interchange Format).

Muy brevemente podemos decir que ldif es la representación textual de las entradas de LDAP. Estas entradas están en un formato legible por el hombre e intercambiable entre dos servidores LDAP diferentes de diferentes fabricantes, usando motores de bases de datos diferentes, y ejecutándose en sistemas operativos distintos.

La manera de realizar esto la dejaremos para el siguiente capítulo. En la siguiente sección se abordará una herramienta alternativa para la implementación de PKI y que, de alguna manera, abarca y hace uso de lo hecho hasta este momento.

[6] OpenLDAP.- Es un proyecto de libre distribución cuyo sitio oficial se encuentra en http://www.openldap.org

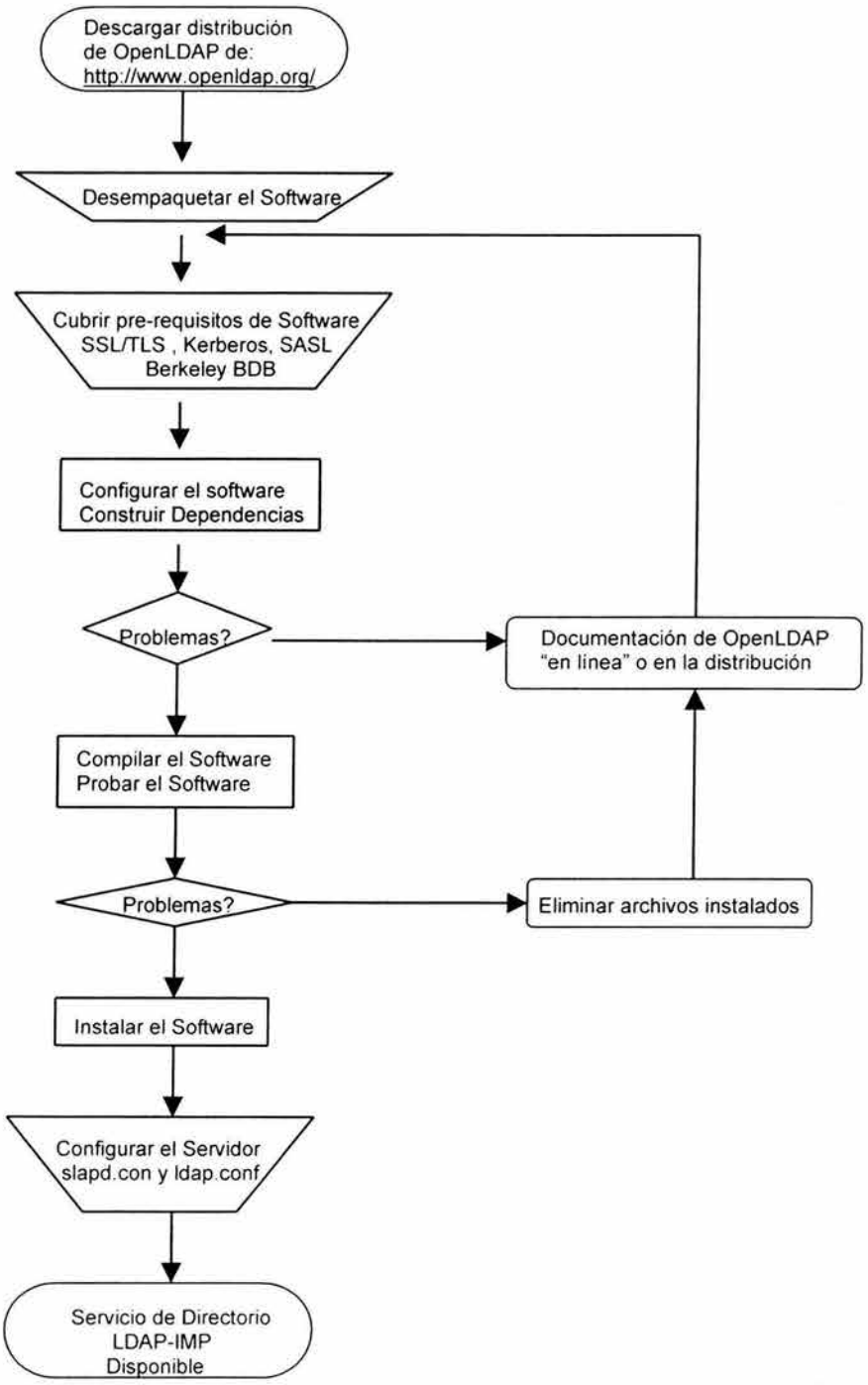


Figura 3.3-4
Implementación de Servicio de Directorio con OpenLDAP

FUENTES Y REFERENCIAS :

RFC 2251: Lightweight Directory Access Protocol (v3)
http://www.ietf.org/rfc/rfc2251.txt

RFC 2252 : http://www.ietf.org/rfc/rfc2252.txt

RFC 2253: http://www.ietf.org/rfc/rfc2253.txt

RFC 2254: http://www.ietf.org/rfc/rfc2254.txt

RFC 2255: http://www.ietf.org/rfc/rfc2255.txt

RFC 2829 : http://www.ietf.org/rfc/rfc2829.txt

RFC 2830 : http://www.ietf.org/rfc/rfc2830.txt

RFC 3377: Lightweight Directory Access Protocol (v3): Technical Specification
http://www.ietf.org/rfc/rfc3377.txt

The OpenLDAP Project Software

Página principal del proyecto
http://www.openldap.org

Documentación
http://www.openldap.org/doc/admin21/intro.html

3.4 PROYECTO PKI-OPENCA

La siguiente herramienta que analizaremos es un proyecto de la comunidad de software libre OpenCA Labs[1], orientado para la implementación de una Infraestructura de Claves Públicas, PKI. En la sección 3.1 mencionamos esta herramienta como una segunda alternativa para la implementación de lo que sería AC-IMP, y en general la Entidad Certificadora (ver Figura 3.1-1) tratada en dicha sección. Dijimos que la llevaríamos a la par con el trabajo realizado con OpenSSL, e igualmente mencionamos que esta era quizá la opción más completa a nuestro alcance y la que quizá adoptaríamos. En la presente sección trataremos de justificar esto.

Este proyecto, PKI-OpenCA, quisimos tratarlo al final de éste capítulo ya que a diferencia de las herramientas hasta ahora presentadas, está orientado para trabajar explotando las facilidades ofrecidas por un gran número de herramientas de software, igualmente de libre distribución, sirviendo este de enlace entre dichos módulos que, en conjunto, implementan y brindan servicios de PKI.

PKI-OpenCA abarca y se acopla muy bien a la arquitectura de PKI-IMP propuesta en la sección 2.2 (ver Figura 2.3). Esta y otras cuestiones se mencionan en la siguiente sección.

INTRODUCCIÓN

Las infraestructuras de llave pública (PKI's) son una de las necesidades más ampliamente aceptadas para el futuro. El problema es que más y más aplicaciones pueden hacerse seguras con cosas tan simples como lo son los certificados y las llaves digitales, pero es realmente difícil organizar y estructurar PKI's y es también realmente caro, debido a que las herramientas de seguridad flexibles confiables para Unix, principalmente, son caras.

Este fue el punto de partida de OpenCA. La meta del grupo consiste en la producción de un sistema autoconfiable de código abierto para dar soporte a su comunidad con una solución buena, barata y útil en el futuro en su infraestructura base.

OpenCA comenzó en 1999. La idea original consistía de tres partes principales. Una interfaz web basada en Perl, un backend basado en OpenSSL para las operaciones criptográficas y una base de datos. Este simple concepto sigue siendo la base del proyecto en al actualidad. Casi todas las operaciones pueden ser realizadas mediante alguna interfaz web, lo cual la hace bastante "amigable" para los usuarios y los administradores. La única diferencia es que cuenta con una interfaz web preconfigurada, pero además se pueden crear tantas interfaces como uno desee o modificar la existente[2].

En las bases de datos se almacena toda la información necesaria sobre los objetos criptográficos de los usuarios, como son: solicitudes de firma de certificado, certificados, solicitudes de revocación de certificados y CRL's.

Hay sin embargo muchas cosas que aún no se han implementado o que están en proceso de implementarse, es por esto que el proyecto está en continuo desarrollo apoyado por un gran número de colaboradores alrededor del mundo.

Nosotros trabajaremos con la versión 0.9.1-1, aunque actualmente está en fase de prueba la versión 0.9.2.

Hoy en día PKI-OpenCA soporta los siguientes servicios, los cuales consideramos son los más importantes:

---

[1] Sitio oficial del proyecto http://www.openca.org

[2] OpenCA Guide; Chris Covell, Michael Bell; PKI Project, OpenCA Labs http://www.openca.org/docs/opencaguide

- Una Interfaz Pública
- Una Interfaz LDAP
- Una interfaz de AR
- Una interfaz de AC
- El protocolo de Administración de PKI SCEP
- El protocolo OCSP (aún en pruebas)
- Filtros IP para las interfaces
- Control de Acceso Basado en Roles, RBAC
- Flexibilidad en el manejo de sujetos de certificado
- Revocación basada en PIN
- Emisión y manejo de CRL's
- Soporte para casi todos los navegadores (los más populares)

OpenCA está diseñada para trabajar en una infraestructura distribuida, de igual manera es capaz de manejar no solo un esquema de AC fuera de línea y una AR en línea. Se puede construir una jerarquía con tres o más niveles. El objetivo es lograr una máxima flexibilidad para que pueda ser utilizada en organizaciones grandes como Universidades, redes de compañías y compañías globales. OpenCA no es solo una pequeña solución para pequeños y medianos proyectos de investigación.

Para dar un mejor seguimiento a lo que vamos a realizar, conviene ilustrar lo que representa cada uno de los módulos de OpenCA respecto al modelo PKI-IMP propuesto en la sección 2.2. Esto se muestra en la Figura 3.4-1.

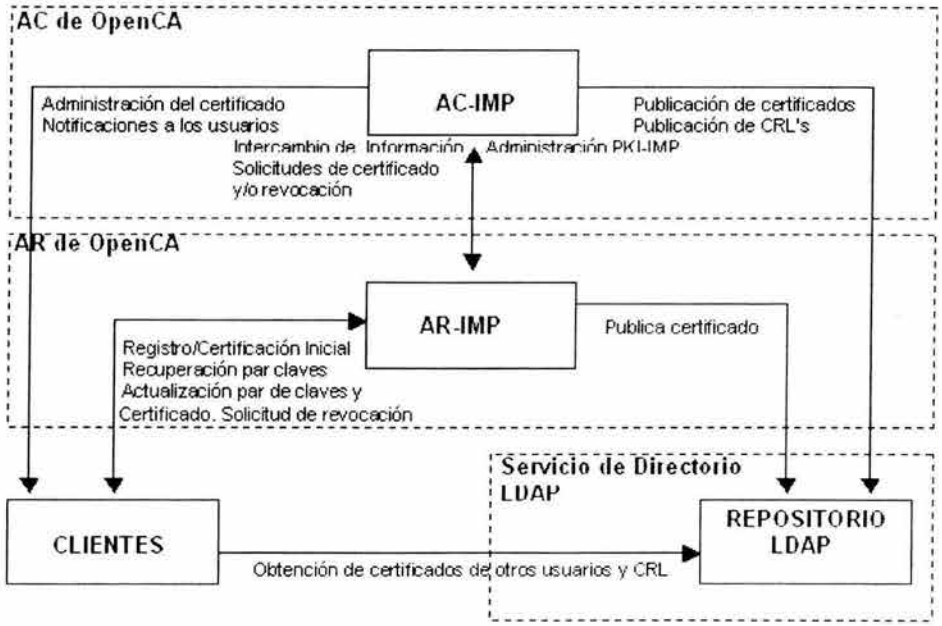


Figura 3.4-1
Módulos PKI-IMP y OpenCA

DESCARGANDO Y DESEMPAQUETANDO EL SOFTWARE

El primer paso es obtener una copia de la última distribución de OpenCA, esto se puede realizar fácilmente de : http://www.openca.org/download.html, ó de http://www.sourceforge.org/openca

En nuestro caso la distribución descargada fue openca-0.9.1.

Está por demás decir que la distribución viene comprimida y "empaquetada". Así que el primer paso es descomprimir y desempaquetar la distribución. Para esto se recomienda crear un directorio para ello, por cuestiones de control de software. Seleccione el directorio en cuestión o cree uno nuevo y ahí descomprima la distribución.

Se recomienda leer la documentación asociada con la distribución a fin de entender mejor la estructura y las diferentes opciones de configuración que se permiten.

REQUERIMIENTOS DE SOTWARE

OpenCA no es un sistema monolítico. Para su correcta configuración e instalación , OpenCA requiere de varios productos de software en módulos proporcionados por terceras partes, todos dentro de la comunidad de Software Libre (OpenSource).

Perl 5.6.1 o superior.
Este módulo es uno de los más importantes y por tanto se recomienda tener instalada la última versión de Perl con todos sus componentes, principalmente los criptográficos. La última versión de Perl así como la documentación relacionada se pueden descargar de: http://www.perl.org. El presente trabajo utiliza la distribución 5.8.0, se recomienda ampliamente utilizar esta versión. Dependiendo del sistema con el que se trabaje, Perl 5.8.0 puede requerir la instalación o actualización de algunas herramientas de software. Una herramienta muy importante es el compilador de C. Algunos errores o advertencias en la compilación del software se deben a versiones no actualizadas del compilador. Se recomienda realizar la instalación de la última versión del compilador gnu de C (caso nuestro), gcc, este se puede descargar de http://www.gcc.org en la distribución 3.3.gcc.core.tar.gz. Adicionalmente se recomienda la instalación del motor Make en su última versión. Este se puede obtener de: http://www.make.org en la distribución make-3.8.0.tar.gz. En ambos casos el software descargado se compila e instala según la documentación incluida en la distribución.

OpenSSL 0.9.7 o superior.
Por este módulo no nos preocupamos ya que dicha distribución ya la tenemos instalada en nuestro sistema.

Apache http Server + mod_ssl
Este módulo es indispensable ya que este es quien soporta todas las interfaces web de OpenCA. El módulo mod_ssl en necesario para poder atender las peticiones que se realicen mediante el protocolo https. Al igual que para OpenSSL, esto no representa mayor preocupación, ya que este servicio con el módulo requerido lo tenemos ya disponible en nuestro sistema.

OpenLDAP
Este módulo es necesario solo si se desea levantar el servicio de directorio. Esto en nuestro caso es un requisito indispensable, mas sin embargo en este momento ya no nos preocupa porque en la sección anterior obtuvimos la distribución, compilamos e instalamos OpenLDAP.

CONFIGURACIÓN DEL SOFTWARE DE OPENCA

OpenCA utiliza el método usual de configuración e instalación de los fuentes de OpenSource. La configuración de la distribución es necesaria para poder compilar e instalar el software mas no para configurar la manera en cómo trabajará el sistema ya instalado. Durante la configuración se realizan algunas configuraciones por defecto si es que no se especifica lo contrario, pero la configuración real se realiza en la fase de post-instalación.

3.4.1 Autoridad Certificadora AC-IMP

Siguiendo lo ilustrado y referido con la Figura 3.4-1 (ver) y con base en la documentación del proyecto, a nuestro juicio y conveniencia, seleccionamos y ejecutamos la configuración para AC-IMP. Previamente creamos el directorio de instalación el cual para este trabajo fue: /srv/ca

Los detalles y las opciones con las que se configuró el software para la AC-IMP y algunos comentarios a estas se encuentran en la Sección I del Anexo C-3 que se encuentra al final del presente trabajo.

Se debe poner especial atención a los mensajes de salida en el proceso de la configuración del software, ya que pueden existir mensajes de alerta (warnings) o mensajes de error durante la configuración, los mensajes de alerta muchas veces son solo informativos y se refieren en su mayoría al uso de librerías no actualizadas o totalmente compatibles con el software que se instala, generalmente no afectan al proceso de configuración y este puede terminar sin mayor problema.

Los mensajes de error son diferentes y muy importantes, ya que es por medio de éstos que el proceso de configuración envía información cuando detecta algún problema grave. Dentro de la información desplegada se incluye el módulo que generó el error, la posible causa y eventualmente una recomendación para solucionar el problema. En cualquiera de los casos el proceso de configuración terminará y enviará un mensaje de falla (configuration failed).

En cualquiera de los casos, se recomienda referirse a la documentación tanto de OpenCA como del módulo relacionado a fin de evitar cualquier contratiempo en la configuración, instalación y/o funcionamiento.

Compilar e Instalar AC de OpenCA

Una vez que el proceso de configuración termine con éxito, lo siguiente es compilar e instalar el software. Como ya mencionamos, OpenCA utiliza método usual de instalación de los fuentes de OpenSource (*make* y *make install*). Se recomienda revisar la Sección I del Anexo C-3 para ver los detalles de la instalación.

Si por ejemplo se configura una instalación de una AC y se solicita la instalación de una AR, lo más probable es que dicho proceso no se pueda completar y/o que lo haga con errores ya que la configuración no ha sido la adecuada.

Si se desea o es necesario realizar todo el proceso de nueva cuenta, no olvidar eliminar previamente los archivos instalados, de lo contrario se obtendrán errores.

Configuración del servidor apache para la AC

Ahora toca configurar el servidor apache para poder utilizar la interfaz web de la AC. Los pasos necesarios a seguir son a grandes rasgos los siguientes:

1. Modificar archivo de configuración httpd.conf (Servidor http de Apache)
   - Indicar nombre del host y puerto por donde se atenderá el servicio
   - Indicar el procesar archivo de configuración del servidor web para la AC
2. Crear archivo de configuración del servidor web de la AC (apache.conf)
3. Guardar cambios

Los detalles de estos pasos y el archivo de configuración apache.conf se encuentran en la Sección II del Anexo C-3 que se encuentra al final del presente trabajo (Ver).

Al llegar a este punto podemos decir que tenemos todo listo para levantar el servicio de la AC. Por el momento dejamos esto pendiente y a continuación hacemos lo propio para la AR.

3.4.2 Autoridad Registradora AR-IMP

La Autoridad Registradora es la interfaz "externa" de la infraestructura (ver Figura 3.4-1). Como se muestra en la Figura 3.4-1, por medio de ésta se pueden generar solicitudes de certificación, de revocación, y otras tareas de administración; por lo cual es muy importante configurar e instalar perfectamente el software de la AR.

Al igual que para la AC, y con base en la documentación del proyecto, a nuestro juicio y conveniencia, seleccionamos y ejecutamos la configuración para la AR. Previamente creamos el directorio de instalación el cual para este trabajo fue: /hdc1/ra

Los detalles y las opciones con las que se configuró el software para la AR y algunos comentarios a estas se encuentran en la Sección III del Anexo C-3 que se encuentra al final del presente trabajo.

Al igual que para la configuración de la AC, se debe poner especial atención a los mensajes de salida en el proceso de la configuración del software, ya que pueden existir mensajes de alerta (warnings) o mensajes de error durante la configuración. De igual manera, los mensajes de alerta muchas veces son solo informativos y se refieren en su mayoría al uso de librerías no actualizadas o totalmente compatibles con el software que se instala, generalmente no afectan al proceso de configuración y este puede terminar sin mayor problema.

Refiérase a la documentación del software OpenCA o a la de los módulos implicados en la configuración, con el fin de evitar o solucionar problemas en esta sección.

Compilar e Instalar AR de OpenCA

Una vez que el proceso de configuración haya terminado con éxito, lo siguiente es compilar e instalar el software. Igual que en para el caso de la AC, la instalación de los fuentes de la AR se realiza utilizando el método habitual para OpenSource (*make y make install*). Se recomienda revisar la Sección III del Anexo C-3 para ver los detalles de la instalación.

Si se desea o es necesario realizar todo el proceso de nueva cuenta, no olvidar eliminar antes los archivos instalados, de lo contrario se obtendrán errores.

Configuración del servidor apache para la AR

Lo siguiente es configurar el servidor apache para poder utilizar la interfaz web de la AR. De la misma manera que hicimos para la AC, los pasos necesarios a seguir son a grandes rasgos los siguientes:

1. Modificar archivo de configuración httpd.conf (Servidor http de Apache)
   - Indicar nombre del host y puerto por donde se atenderá el servicio
   - Indicar el procesar archivo de configuración del servidor web para la AR
2. Crear archivo de configuración del servidor web de la AR (apachera.conf)
3. Guardar cambios

Los detalles de estos pasos, el archivo de configuración apachera.conf, así como algunos comentarios a los mismos se encuentra en la Sección IV del Anexo C-3 (Ver).

Al llegar a este punto podemos decir que tenemos todo listo para levantar el servicio de la AR.

Recapitulando lo hecho hasta el momento, lo que tenemos es una AC y una AR configuradas y listas para trabajar mediante una interfaz web (módulos principales de una infraestructura de PKI).
A continuación realizamos la configuración del servicio de directorio LDAP para poder utilizarlo con OpenCA. Esta configuración es bastante sencilla y no debe presentar mayor complicación. Los detalles de ésta se encuentran en el Anexo C-3 Sección V. Se recomienda ver esta sección así como

todas las demás que conforman el Anexo C-3 para una mejor comprensión de lo que se ha hecho hasta ahora. Con esto terminamos la configuración de la AR y de LDAP.

Para finalizar esta sección, solo queda "levantar" el servicio de LDAP en la AR y levantar el servidor web apache en la AR y en la AC (si es que se encuentran en máquinas separadas).

La Figura 3.4-2 representa el diagrama de flujo que resume lo que se realiza para llevar a cabo la implementación de la AC y la AR con OpenCA[3].

Lo realizado hasta este momento, de manera detallada y comentada, forma parte de la **Guía de Administración PKI-IMP.** Un fragmento de esta se encuentra en el Anexo C-4 (Ver).

Lo que sigue es Inicializar los servicios tanto de la AC como de la AR. Esto no tiene mayor complicación ya que todo el proceso se basa en una interfaz web y en comandos y funciones que se ejecutan en cada paso y con instrucciones claras de lo que ocurre y de lo que hay que hacer en cada fase.

[3] Considerando que tanto la configuración e instalación de la AC como de la AR se realizan en la misma máquina.

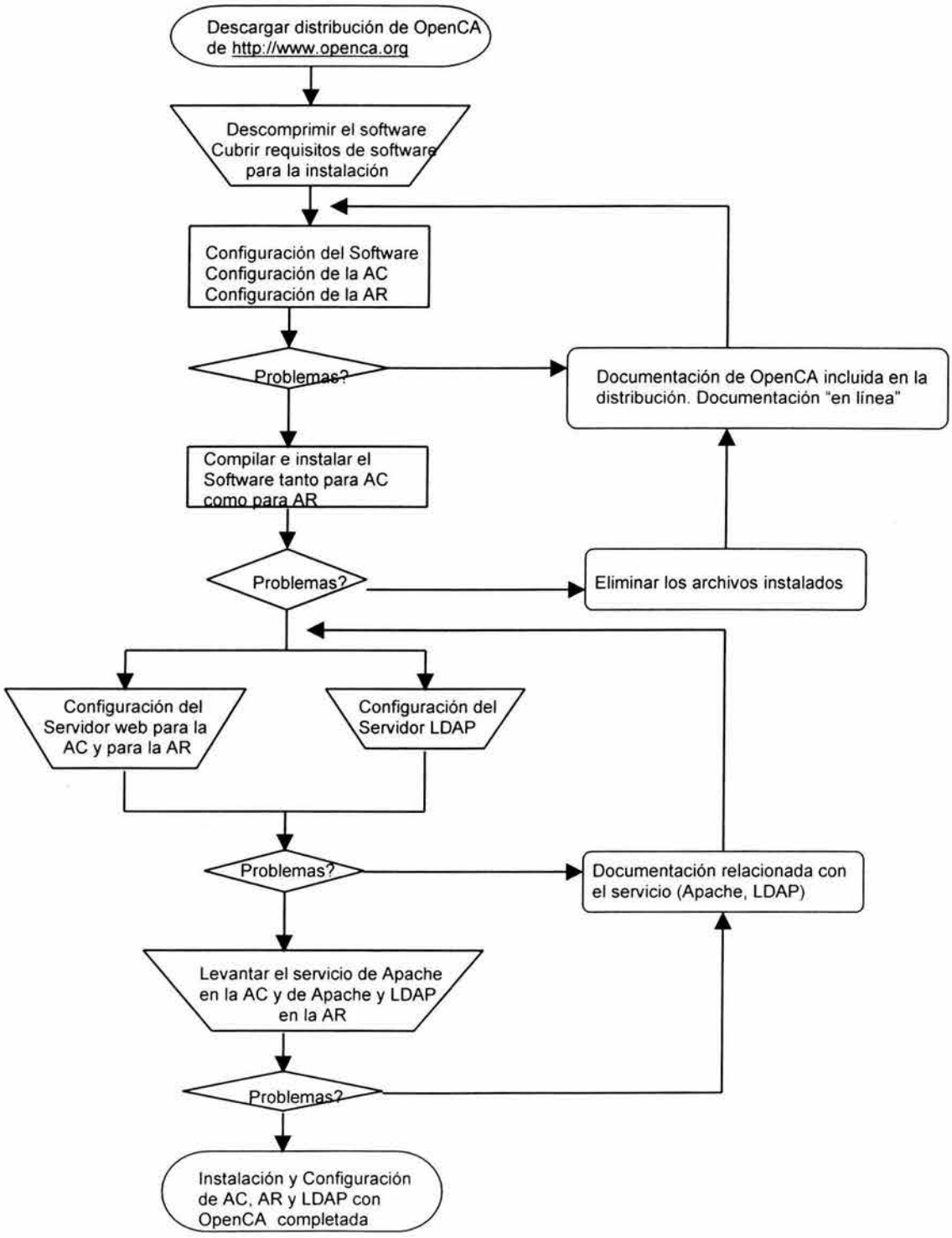


Figura 3.4-2
Implementación de AC y AR con OpenCA

Inicialización de los módulos de PKI-IMP-OpenCA

Para poder continuar, antes es necesario asegurarse de que la configuración e instalación fueron realizadas sin ningún error y que los módulos y servicios se encuentran levantados y disponibles.

Haciendo uso de lo que se mencionó en la Introducción de la presente sección (3.4), lo primero que realizamos fueron modificaciones y adaptaciones a la interfaz web de la AC, AR, interfaz pública de usuario e interfaz web de LDAP (selección de componentes y herramientas disponibles, así como la traducción y rediseño de todas las interfaces web).

Lo que se muestra a continuación, corresponde a los pasos básicos y resumidos para la inicialización de los módulos de PKI-IMP utilizando la herramienta OpenCA.

Los pasos completos y detallados se encuentran en el Anexo C-4 al final de este trabajo.

Es importante aclarar que lo que contiene el Anexo C-4 corresponde solo a las secciones de Instalación, Configuración e Inicialización de PKI-IMP, de la "**Guía de Administración PKI-IMP**" que por cuestiones de seguridad no se incluye completa en este trabajo pero que se encuentra a disposición del administrador de la AC y de la AR de PKI-IMP. Dicho documento se encuentra actualmente bajo custodia del responsable del Área de Seguridad Informática del IMP, M.C. Uriel Tirado Ríos.

Inicializar la AC

Fases de Inicialización AC-IMP

Preparaciones

Inicialización de la AC Fase 1

- AC en http://raguel.imp.mx:88/ca
- Inicializar la Base de Datos
- Crear Nueva clave privada
- Generar nueva solicitud de certificado de AC.
- Creación de certificado de AC autofirmado.
- Reconstruir la cadena de certificación.
- Exportar la configuración de la AC.

Inicialización de la AC Fase 2

- Inicializar y crear al administrador inicial.
- Crear nueva solicitud
- Editar la solicitud
- Manejar el certificado
- Descargar el archivo
- Cargar el certificado y la llave en el navegador.

Inicialización de la AC Fase 3

- Inicializar y crear el primer certificado de AR
- Crear nueva solicitud
- Editar la solicitud
- Manejamos el nuevo certificado y el par de claves

Inicializar la AR

Fases de Inicialización AR-IMP

Preparaciones

Exportar la configuración desde la AC

- Nodo de intercambio de información de OpenCA
  http://raguel.imp.mx:88/ca_node .
  Intercambio de datos.
  Registrar/Enviar datos a un nivel inferior de la jerarquía.
- Enviar Configuración

Inicializar la Base de Datos de la AR

- https://raguel.imp.mx/ra_node
- Inicializar Base de Datos.

Importar Configuración en la AR

- https://raguel.imp.mx/ra_node
- Liga de Inicializar Servidor
- Importar Configuración

Finalmente, con lo hecho hasta este momento, podemos decir que tenemos lista la infraestructura PKI-IMP para probarla y trabajar con ella.

Es conveniente leer la documentación correspondiente con el fin de entender lo que se está haciendo, prever y solucionar problemas encontrados en los procesos de configuración e Inicialización de la AC y de la AR.

Con lo que hemos visto hasta este momento de la herramienta PKI de OpenCA, siguiendo el método descrito en la sección de Justificación y luego de evaluar las facilidades que proporciona y la utilidad que ellas representan para este trabajo, tomamos la decisión de concentrarnos en ella y dedicar nuestros esfuerzos en entenderla y adaptarla a nuestras condiciones y necesidades. Con esto no queremos decir que lo hecho con OpenSSL, con Apache y con OpenLDAP de manera independiente no sea útil, sino mas bien que OpenCA nos brinda esa conexión y vínculo entre los módulos anteriores de una manera óptima, potente y fácil de utilizar.

Para concluir el presente capítulo, haremos un resumen del software instalado en este trabajo e incluiremos alguna información adicional importante. Dicho resumen se muestra en la tabla 3.4-1.

Software Descargado	Versión	Usado en Módulo	URL Fuente	Archivos de Configuración	Directorio de Instalación
OpenSSL	0.9.7b	AC-IMP AR-IMP	http://www.openssl.org	openssl.cnf openssl-ca.cnf	/usr/local/ssl
Apache HTTP Server + mod_ssl	2.0.45	AC-IMP AR-IMP LDAP	http://www.apache.org	httpd.conf ssl.conf	/usr/local/apache2
OpenLDAP	2.1.17	AR-IMP LDAP	http://www.openldap.org	slapd.conf ldap.conf	/usr/local/etc/openldap
OpenCA	0.9.1-1	AC-IMP AR-IMP LDAP	http://www.openca.org	apache.conf apachera.conf slapd.conf ldap.conf openssl.cnf openca.conf	AC-IMP: /srv/ca AR-IMP: /hdc1/ra

Tabla 3.4-1

En el siguiente capítulo nos dedicaremos a probar lo que hemos construido y a validar los resultados que podamos obtener.

REFERENCIAS

OpenCA Labs
PKI Project
http://www.openca.org/openca

The OpenCA Guide
Formato PDF
Chris Covell, Michael Bell; Febrero 2003
PKI Project, OpenCA Labs
http://www.openca.org/openca/docs

Capítulo 4

Pruebas y Validación

Introducción

Toca tratar en este capítulo lo concerniente a las pruebas del trabajo hecho y validar el mismo. El propósito de las pruebas es verificar y validar la información generada e intercambiada, verificar la interacción entre los módulos y el funcionamiento adecuado de los mismos, verificar la propia integración de todos los componentes de la infraestructura de software, verificar que todos los requerimientos han sido correctamente implementados y los objetivos cubiertos, e identificar posibles problemas y defectos.

4.1 ESPECIFICACIÓN DE PRUEBAS

Las pruebas del software pueden significar la mayor parte del costo de un proyecto en tiempo y recursos. Es común que esta actividad se realice después del desarrollo del mismo y se da principalmente como consecuencia de dos razones principales:

- Las pruebas de software son en extremo difíciles (ya que las diferentes opciones que pueden resultar en cualquier producto de software no son cuantificables), la validación formal no está aún ampliamente difundida y las herramientas para ello son escasas y caras; y
- Las pruebas son hechas sin una metodología clara.

Nuestro caso no es la excepción. Existe una gran cantidad de aspectos y opciones que resultan del uso de una infraestructura basada en certificados que probablemente derivarían en otro trabajo de tesis completo como el que se presenta.

En un proyecto como este, es conveniente realizar pruebas que nos proporcionen una medición o una referencia de propiedades como la vivacidad o seguridad del mismo[1]. La realización de pruebas de medición formales de las propiedades anteriores están fuera de los alcances y objetivos del presente trabajo de tesis, además de que ellas implican una inversión mayor en tiempo y recursos; por tanto, trataremos la vivacidad del sistema y la evaluaremos en función de si los procedimientos y procesos que se inician concluyen de la manera en que se esperaba. En cuanto a la seguridad se refiere, incluiremos únicamente un análisis de vulnerabilidades de red realizado con la herramienta Nessus para detectar y corregir debilidades y vulnerabilidades (hoyos de seguridad) en los servidores de PKI-IMP

Nos concentraremos en pruebas funcionales útiles para nosotros, basándonos en los objetivos principales de esta tesis, que nos indiquen que lo implementado corresponde a lo esperado.

Para efectos de los objetivos que se persiguen, abordaremos los siguientes puntos:

- Probar y validad el funcionamiento adecuado de los módulos de PKI-IMP y de los datos intercambiados.
- Probar y validar los certificados y las claves generadas.
- Evaluar y validad los procedimientos administrativos de PKI-IMP.
- Probar y validar los aspectos de la seguridad informática relacionados con el uso de certificados; casos específicos: autenticación (como punto principal), confidencialidad e integridad de la información.

[1] Una de muchas referencias a estas propiedades se pueden consultar en las siguientes direcciones:
http://antareja.rvs.uni-bielefeld.de/avinanta/Publication/SurveyCrypto/surveycrypto.pdf
http://dalila.sip.ucm.es/concurrencia/introduccion.html

A simple vista parecieran aspectos muy simples de examinar y valuar, en el resultado final tal vez si, más no en el procedimiento que se debe seguir para llegar hasta él. Esto es, hay una gran cantidad de trabajo detrás de cada uno de estos puntos, en el presente trabajo reportaremos aspectos importantes de cada prueba, pero nos concentraremos más en los resultados de las mismas.

Para cada uno de los puntos anteriores consideraremos los siguientes aspectos:

- El módulo o servicio funciona y se encuentra disponible.
- Las operaciones básicas de uso y/o intercambio de información se realizan sin contratiempos (los procesos que se inician terminan como se esperaba).
- La interacción entre módulos se lleva a cabo sin contratiempos.
- El servicio y la información se encuentran disponibles solo para aquellos usuarios autorizados, y
- Principalmente que el certificado (elemento central de este trabajo de tesis) así como el par de claves asociadas, son útiles y válidos en los ámbitos en los que se les utiliza.

Las pruebas de rendimiento en los servicios que se proporcionan los dejaremos como un trabajo futuro que deberá realizarse si se desea atender a una comunidad de usuarios mayor a la que en principio se contempló servir.

Conviene recordar y tener en mente que este modelo que se prueba y valida se construyó como un prototipo. En esta fase los errores y pruebas fallidas son totalmente permisibles y deseables con el objetivo de que en el trabajo final se eviten, en la medida de lo posible, cualquier tipo de fallas o contratiempos no contemplados.

Una prueba la consideraremos exitosa si cumple con los fines para los cuales se realiza. En caso contrario, en la medida de lo posible, se realizarán cambios con el objetivo de superar los obstáculos encontrados y se volverá a realizar dicha prueba. Si no es posible superar el contratiempo entonces se propondrá una solución y se dejará como trabajo pendiente.

El obtener un resultado exitoso en una prueba no implica que el procedimiento u objetivo sea alcanzado y/o validado. En este punto es importante mencionar que pudiésemos obtener excelentes resultados en las pruebas, más al final no validar estas. Pudiese darse el caso de que los objetivos iniciales no fuesen alcanzados en su totalidad y por esto concluir que el trabajo realizado no es totalmente satisfactorio.

En la sección siguiente realizaremos las pruebas correspondientes a los puntos que aquí hemos mencionado y fijado. Los resultados que obtengamos nos serán de gran utilidad para elaborar las conclusiones finales de este trabajo.

4.2 Realización de Pruebas

Con base en el plan de pruebas presentado en la sección anterior, procedemos a realizar éstas.

4.2.1 Creación de AC-IMP con OpenSSL

El primer módulo que abordaremos es el de la infraestructura de AC creada con OpenSSL.

El primer punto es la creación de la infraestructura misma de la AC. De este punto podemos decir que siguiendo al pie de la letra los procedimientos y recomendaciones de la documentación relacionada, cuyas fuentes y procedimientos se detallaron en la sección 3.1.1, y con ayuda de la utilidad CA.pl incluida en la distribución de OpenSSL, podemos fácilmente crear y administrar una infraestructura de AC de autoconfianza, esto es, una AC que hace uso de un certificado digital y una clave privada creada por nosotros mismos en la cual, de común acuerdo, depositamos nuestra confianza para la emisión y administración de los certificados digitales que utilizamos para llevar a cabo las operaciones de confidencialidad e integridad de la información intercambiada, pero principalmente utilizados como un mecanismo para la autenticación de los participantes en dichas transacciones. El procedimiento se realiza una sola vez para cada estructura de AC que se desee crear.

La tabla 4.2-1 contiene el resultado de las pruebas realizadas a este módulo.

Procedimiento	Resultado	Estado Actual	Observaciones
Creación de la estructura de administración de AC.	Terminado sin contratiempos	Activo, disponible	Todo el proceso se realiza desde una terminal de manera "manual".
Generación y autofirma de certificado para AC-IMP	Correcto	Disponible	El proceso se realiza desde una terminal de manera "manual".

Tabla 4.2-1
Pruebas realizadas en el módulo AC-IMP

Diremos que tanto el certificado como la clave privada se encuentran en formato PEM (*Private Enhanced Mail*), esto debido a que así lo requiere OpenSSL para poder utilizar este certificado como certificado raíz y la clave privada para firmar los certificados futuros.

La tabla 4.2-2 contiene el resultado de la evaluación de este módulo

Módulo / **Aspecto a evaluar**	Autoridad Certificadora AC-IMP
El módulo funciona y se encuentra disponible	Sí
Las operaciones de uso y/o intercambio de información se realizan sin contratiempos	Sí
La interacción entre módulos se lleva a cabo sin contratiempos	No aplica
El servicio y la información se encuentra disponible solo para usuarios autorizados	Sí, solo para el administrador
El certificado, así como el par de claves asociadas es útil y válido en los ámbitos en los que se utiliza.	Sí

Tabla 4.2-2

Consideramos esta prueba como exitosa y damos paso al siguiente procedimiento.

4.2.2- Procedimiento de Certificación por medio de AC-IMP con OpenSSL

El procedimiento detallado que se sigue referente a esta parte se encuentra en el Anexo D-1.

Con este procedimiento generamos certificados para servidor web y para usuario final. Igualmente se pueden crear certificados para AC's subsidiarias de la AC Raíz. En esta sección somos libres de "experimentar" y jugar con las diferentes opciones que nos brinda OpenSSL. De cualquier manera se recomienda referirse a la documentación para evitar algunos contratiempos sorpresivos.

La tabla 4.2-3 contiene el resultado de las pruebas realizadas a este procedimiento.

Procedimiento	Resultado	Estado Actual	Observaciones
Generación de solicitud y par de claves	Terminado sin contratiempos	Disponible	Todo el proceso se realiza desde una terminal de manera "manual".
Firma de la solicitud y emisión del certificado por parte de la AC-IMP	Correcto, terminado sin contratiempos	Disponible	El proceso se realiza desde una terminal de manera "manual".

Tabla 4.2-3
Pruebas de Procedimiento de certificación

Este procedimiento, al igual que el de la creación de la AC, operacionalmente es poco eficiente, ya que es el operador-Administrador de la AC quien debe realizar todo el trabajo desde una ventana de terminal. Igualmente es necesario que, previo a la generación de la solicitud, se tengan los datos de la entidad final que se certificará, además de introducir y registrar un password de seguridad durante el proceso de generación. El procedimiento recomendado implica que fuese el solicitante quien introdujese y conociese dicho password, lo cual deriva en un procedimiento más confiable y transparente para el destinatario, la anterior recomendación complicaría bastante las cosas dentro de este esquema.

Lo ideal sería poder contar con un mecanismo que permitiese recibir dicha información por parte del destinatario de manera más eficiente o inclusive que fuese él quien generase la solicitud y la AC se limite simplemente a validarla y aprobarla. Se puede pensar en una interfaz web para la captura y envío de dicha información o de alguna aplicación ejecutándose del lado del cliente .

Al final obtenemos un certificado firmado por nuestra AC y una clave privada correspondiente al certificado. La clave pública se encuentra en certificado mismo.

Por medio de este procedimiento generamos el certificado web del servidor Apache que se presenta al visitar el sitio https://192.168.144.102, así como varios certificados de prueba para usuarios finales.

La tabla 4.2-4 contiene el resultado de la evaluación del procedimiento de certificación con AC-IMP OpenSSL.

Procedimiento / Aspecto a evaluar	Generación de solicitud y claves	Firma de la solicitud y emisión del certificado
El servicio funciona y se encuentra disponible	Sí	Sí
Los procesos que se inician terminan como se esperaba	Sí	Sí
La interacción entre módulos se lleva a cabo sin contratiempos	No aplica	No aplica
El servicio y la información se encuentra disponible solo para usuarios autorizados	Sí, solo para el administrador	Sí, solo para el administrador
El certificado, así como el par de claves son útiles y válidos en los ámbitos en los que se utilizan.	Sí	Si

Tabla 4.2-4
Evaluación del procedimiento

Consideramos la prueba como exitosa ya que finalmente obtenemos un certificado perfectamente válido y utilizable, objetivo primordial de este trabajo.

4.2.3 Servicio de Directorio LDAP-IMP

En la sección 3.3 definimos lo que es un servicio de directorio y los pasos necesarios para poder instalar y configurar OpenLDAP, y lo dejamos listo para comenzar a introducirle información.

Para esto mencionamos que utilizaríamos un formato llamado ldif, igualmente dijimos que la manera estándar de llenar de información un directorio por medio del servidor LDAP, es crear un archivo LDIF (Formato de intercambio de directorios LDAP, LDAP Directory Interchange Format).

Comentamos que ldif es la representación textual de las entradas de LDAP. Estas entradas están en un formato legible por el hombre e intercambiable entre dos servidores LDAP diferentes de diferentes fabricantes, usando motores de bases de datos diferentes, y ejecutándose en sistemas operativos distintos. Podemos leer el manual de ldif (man ldif) para conocer más sobre este formato.

A grandes rasgos podemos decir que este formato se basa en entradas en las que se definen tipos de atributos y clases de objetos. Para saber mas de ellos refiérase a RFC 2252/2256 (LDAPv3). Si deseamos saber cuales tenemos disponibles en nuestro servidor conviene echarle un vistazo al archivo de configuración core.schema, que en nuestro caso se encuentra ubicado en /usr/local/etc/openldap/schema junto con otros archivos de configuración.

Igualmente se debe tener claro el esquema de nombrado que se desea manejar, ya que en base a este se realizarán las entradas y consultas del servicio de directorio. Tenemos dos opciones principales el sistema de nombrado basado en componente de dominio (domain component,dc) y el típico basado en nombrado tradicional (límites geográficos u organizacionales).

Convine leer la documentación relacionada con estos puntos para comprender mejor el tema y explotar las opciones que se nos brindan.

Para este trabajo, utilizamos aquí un sistema de nombrado tradicional teniendo como raíz del árbol de LDAP a "o=IMP, c=MX"

EL procedimiento de realización de pruebas de este módulo se encuentra en el Anexo D-2. Siguiendo este procedimiento se realizaron varias pruebas de inserción de entradas al directorio. En esta sección analizaremos únicamente los resultados

Para ver la estructura del directorio ponemos la siguiente dirección en el navegador web Konqueror:

ldap://raguel.imp.mx/o=IMP,c=MX

Nos deberá mostrar algo parecido a lo siguiente (Figura 4.2-1):

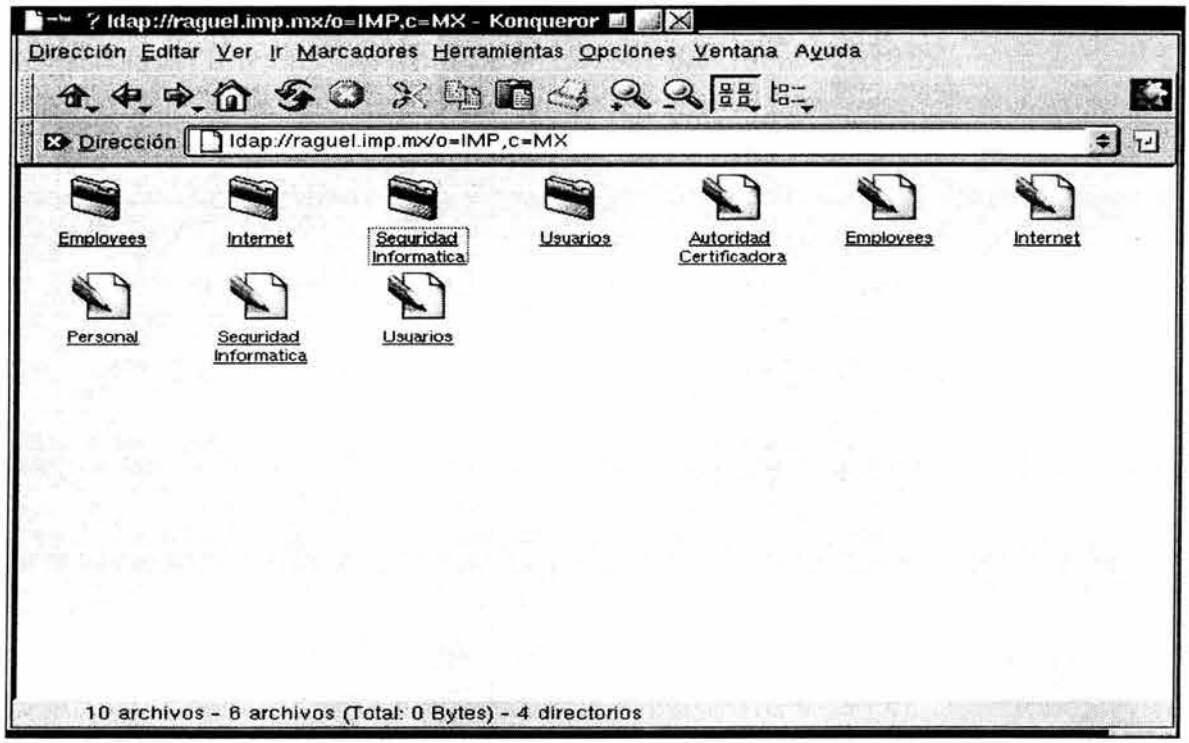


Figura 4.2-1
Árbol de Directorio LDAP-IMP visto con Navegador web Konqueror

La figura anterior es una representación "visual" de cómo está constituido el árbol de directorio de LDAP-IMP. Quisimos utilizar el navegador web Konqueror ya que este nos permite ver la estructura de los directorios y no solo el contenido de los mismos. En la estructura anterior podemos identificar los siguientes elementos:

Primeramente cuatro ramas principales de unidad de organización, OU (Employees, Internet, Usuarios y Seguridad Informática) Las tres primeras pertenecen o fueron creadas cuando se instaló la herramienta OpenCA, la última fue generada mediante nuestros archivos de prueba ldif.

En segundo lugar identificamos nodos terminales los cuales fueron insertados mediante los archivos de prueba ldif, en estos se encuentra toda la información que nosotros le especificamos en cada entrada en el archivo ldif.

Si lo anterior lo realizamos mediante la libreta de direcciones de Microsoft lo que obtendremos será algo como lo siguiente (Figura 4.2-2):

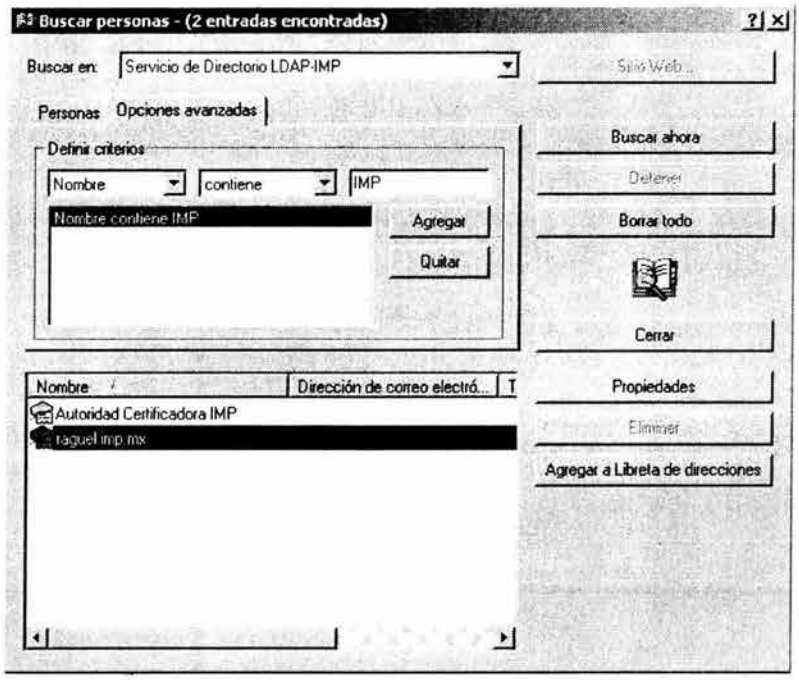


Figura 4.2-2
Árbol de Directorio LDAP-IMP visto con libreta de direcciones Microsoft

Como comentamos hace un momento, este tipo de aplicación no nos permite ver la estructura completa del directorio, solo aquellas entradas que correspondan a nodos terminales con alguna información útil que se pueda mostrar dependiendo de los parámetros de búsqueda. Para ver el contenido del nodo, basta con seleccionarlo y dar clic en el botón de Propiedades.

El procedimiento para realizar esta tipo de búsquedas, así como la configuración del servicio utilizando tanto la libreta de direcciones de Microsoft como el libro de direcciones de Netscape, se encuentra en la sección 4 del Manual de Procedimientos de Usuario de la Infraestructura PKI-IMP que se encuentra en el anexo D-3 de este documento.

La tabla 4.2-5 contiene el resultado de la evaluación de este módulo

Módulo / Aspecto a evaluar	Servicio de Directorio LDAP-IMP
El módulo funciona y se encuentra disponible	Sí
Las operaciones de uso y/o intercambio de información se realizan sin contratiempos	Sí
La interacción entre módulos se lleva a cabo sin contratiempos	No aplica
El servicio y la información se encuentra disponible solo para usuarios autorizados	Sí, para todos los usuarios
El certificado, así como el par de claves asociadas es útil y válido en los ámbitos en los que se utiliza.	Si

Tabla 4.2-5
Evaluación del módulo

Consideramos esta prueba al módulo LDAP como exitosa.

Con esto terminan las pruebas de lo realizado y experimentado con los módulos trabajando de manera independiente, ya que, como lo mencionamos al final de la sección 3.4, decidimos concentrarnos en el proyecto PKI de OpenCA que integra los módulos anteriores de una manera potente, óptima y fácil de usar.

4.2.4 PKI-IMP utilizando el conjunto de herramientas de PKI OpenCA

En la sección 3.4 detallamos el procedimiento necesario para la Configuración, Instalación e Inicialización de la herramienta PKI OpenCA. Al final de esa sección lo que obtuvimos es una infraestructura de PKI lista para probarla. Como lo mencionamos en dicha sección, se realizaron cambios y adaptaciones en los archivos de configuración, en las librerías y en las interfaces web de PKI-IMP. Es desde este punto del que partimos, siguiendo lo indicado en la sección 4.1, para realizar lo que a continuación se menciona.

4.2.4.1 Módulos de PKI-IMP

Primero probaremos el estado de las interfaces de PKI-IMP

1-Interfaz Pública

La interfaz pública comprende el conjunto de páginas web disponibles y accesibles para todos los usuarios de la infraestructura.

Las pruebas y resultados se encuentran en la Tabla 4.2-6.

Procedimiento	Resultado	Estado Actual	Observaciones
Acceso al servicio vía navegador web en la siguiente dirección: https://raguel.imp.mx/pub.	Carga sin problemas	Disponible	Se carga de diferente manera dependiendo del navegador que se utilice.
Visita a todas las ligas disponibles	Cargan sin problemas	Disponible	Ninguna

Tabla 4.2-6
Pruebas de la Interfaz Pública

2- Interfaz LDAP-IMP

La interfaz LDAP comprende el conjunto de páginas web por medio de las cuales se administra el servidor LDAP-IMP. Este conjunto de páginas deben estar disponibles solo para el administrador del servicio.

Las pruebas y resultados se encuentran en la Tabla 4.2-7.

Procedimiento	Resultado	Estado Actual	Observaciones
Administrador accesa al servicio vía navegador web en la siguiente dirección: https://raguel.imp.mx/ldap.	Carga sin problemas	Disponible	Se carga de diferente manera dependiendo del navegador que se utilice.
Visita a todas las ligas disponibles	Cargan sin problemas	Disponible	Ninguna
Usuario normal solicita el servicio	Se tuvo acceso al servicio	Sin cambios, servicio disponible	Solo los usuarios dentro de la subred 192.168.144 pueden hacerlo debido a que así está configurado en apachera.conf para efectos de prueba (Ver Anexo C-3)

Tabla 4.2-7
Pruebas de la interfaz LDAP

3- Interfaz AR-IMP

La interfaz AR-IMP comprende el conjunto de páginas web por medio de las cuales se administra la AR y el nodo de intercambio ra_node. Este conjunto de páginas deben estar disponibles solo para el administrador de AR-IMP.

Las pruebas y resultados se encuentran en la Tabla 4.2-8.

Procedimiento	Resultado	Estado Actual	Observaciones
Administrador accesa a los servicio vía navegador web en las siguientes direcciones: https://raguel.imp.mx/ra y https://raguel.imp.mx/ra_node	Cargan sin problemas	Disponibles	Se cargan de diferente manera dependiendo del navegador que se utilice.
Visita a todas las ligas disponibles	Cargan sin problemas	Disponible	Ninguna
Usuario normal solicita el servicio en /ar y /ar_node	Se tuvo acceso al servicio	Sin cambios, servicio disponible	Solo los usuarios dentro de la subred 192.168.144 pueden hacerlo debido a que así está configurado en apachera.conf para efectos de prueba (Ver Anexo C-3)

Tabla 4.2-8
Pruebas de la interfaz AR-IMP

4- Interfaz AC-IMP

La interfaz AC-IMP comprende el conjunto de páginas web por medio de las cuales se administra la AC y el nodo de intercambio ac_node. Este conjunto de páginas deben estar disponibles solo para el administrador de AC-IMP. Las pruebas y resultados se encuentran en la Tabla 4.2-9.

Procedimiento	Resultado	Estado Actual	Observaciones
Administrador accesa a los servicios vía navegador web en las siguientes direcciones: http://raguel.imp.mx:88/ca y http://raguel.imp.mx:88/ca_node	Cargan sin problemas	Disponibles	Se cargan de diferente manera dependiendo del navegador que se utilice.
Visita a todas las ligas disponibles	Cargan sin problemas	Disponible	Ninguna
Usuario normal solicita el servicio en /ac y /ac_node	Se tuvo acceso al servicio	Sin cambios, servicio disponible	Solo los usuarios dentro de la subred 192.168.144 pueden hacerlo sabiendo el nombre del servidor y el puerto debido a que así está configurado en apache.conf para efectos de prueba (Ver Anexo C-3)

Tabla 4.2-9
Pruebas de la interfaz AC-IMP

4.2.4.2 Funciones de los participantes

A continuación probaremos las funciones de los participantes según lo especificado en el capítulo 2

1- Clientes o Usuarios

Realizamos pruebas de las principales funciones de los usuarios. Nos valimos para ello del Manual de Procedimientos para usuarios de PKI-IMP (Ver Anexo D-3)

Los resultados de estas pruebas se encuentran en la Tabla 4.2-10

Procedimiento y/o Función	Resultado	Estado Actual	Observaciones
Solicitar su certificado digital a la AC-IMP a través de la AR-IMP	Completado sin contratiempos	Disponible	La solicitud se realiza con la ayuda de la AR-IMP por medio de una interfaz web
Establece frase de seguridad para clave privada y genera el par pública/privada con la ayuda de la AR	Completado sin contratiempos	Disponible	Realizado mediante una interfaz web. Dependiendo de la versión del SO se tienen varias opciones de CSP para realizar esto.
Acude ante la AR-IMP, se identifica y valida su solicitud	Completado sin contratiempos	Disponible	Para efectos de prueba, la identificación y validación ante la AR no se realizó.
Recibe su certificado digital ya registrado en la AR-IMP	Completado sin contratiempos	Disponible	El usuario recibe vía e-mail una notificación cuando está listo así como el procedimiento para recogerlo
Solicita certificados de terceros a la AR-IMP o al Servicio de Directorio LDAP-IMP	Completado	Disponible vía AR o vía LDAP	Los usuarios tienen complicaciones en el uso del servicio de directorio
Descarga de CRL's actualizadas	Correcto	Disponible	Es responsabilidad el usuario realizar esto
Firmar, verificar firmas, cifrar, descifrar mensajes, autenticarse con otros usuarios	Sí se pueden realizar	Disponible dependiendo de los recursos del usuario	Esto se llevó a cabo en los clientes de correo MS Outlook, OutLook Express y Netscape Web Mail
Cuando así se requiera, solicita revocación de certificado ante la AR-IMP	Completado sin contratiempos	Disponible	Se realiza mediante una interfaz web. El usuario debe justificar y validar dicha solicitud

Tabla 4.2-10
Pruebas de Funciones de Usuarios

Para evaluar el penúltimo punto de la tabla 4.2-10 nos valimos del documento "**Correo Seguro**" y de los resultados obtenidos con él (ver Anexo D-4).

2- Autoridad Registradora AR-IMP

A continuación realizamos pruebas de las principales funciones de la AR. Los resultados se muestran en la Tabla 4.2-11.

Procedimiento y/o Función	Resultado	Estado Actual	Observaciones
Recibe solicitudes de certificación y elabora la solicitud	Completado sin contratiempos	Disponible	La solicitud se realiza por medio de una interfaz web
Genera junto con el cliente el par pública/privada y la solicitud de certificación	Completado sin contratiempos	Disponible	En la interfaz web se dan varias opciones para llevar a cabo esto.

Verifica y valida la identidad del usuario y los atributos solicitados	Completado sin contratiempos	Disponible	Para efectos de prueba, la verificación y validación del usuario no se realizó
Aprueba y firma la solicitud, inicia el proceso de registro y envía la misma a la AC para su firma	Problemas al firmar la solicitud.	En proceso de solución	Problema en navegador del operador de la AR. No es crítico. Lo demás funciona y está esta disponible
Recibe el certificado emitido por la AC, notifica y entrega este al usuario	Completado sin contratiempos	Disponible	La notificación y el procedimiento de descarga se realizan vía e-mail
Recibe solicitudes de revocación, verifica y valida la solicitud y envía esta a la AC para su revocación	Completado sin contratiempos	Disponible	Se realiza mediante una interfaz web.
Recibe y distribuye cada nueva CRL emitida por la AC	Completado sin contratiempos	Disponible vía un CDP o vía web	El CDP (Crl Distribution Point) se encuentra en el certificado del usuario
Administra y Actualiza el servicio de directorio con nuevos certificados y CRL emitidas por la AC	Completado	Disponible	Algunas ocasiones la CRL no se importa con éxito a LDAP al primer intento

Tabla 4.2-11
Pruebas de Funciones de AR-IMP

3- Autoridad Certificadora AC-IMP

Siguiendo con el trabajo, realizamos pruebas de las funciones principales de la AC. Los resultados se muestran en la Tabla 4.2-12

Procedimiento y/o Función	Resultado	Estado Actual	Observaciones
Crea su propio certificado digital y certifica a la AR	Completado sin contratiempos	Disponible	El certificado de la AC está disponible desde la AR vía web y LDAP
Recibe el precertificado enviado por la AR, valida la firma de la misma y emite el certificado	Completado sin contratiempos	Disponible	Todo el procedimiento se realiza mediante una interfaz web
Emite y Firma certificados digitales	Completado sin contratiempos	Disponible	Esta opción es para certificados de AC subsidiaria, AR, servidores web y para operadores de AC y de AR
Registra el certificado y la clave pública del usuario	Completado sin contratiempos	Disponible	Se realiza de manera automática cuando se emite el certificado
Envía el certificado y la clave privada a la AR para su registro y entrega al usuario	Completado sin contratiempos	Disponible	El procedimiento se realiza a través del módulo de intercambio ac_node
Revoca certificados y emite listas de revocación	Completado sin contratiempos	Disponible	La CRL se envía a la AR para su publicación

Tabla 4.2-12
Pruebas de Funciones AC-IMP

4- Repositorio de Certificados LDAP-IMP

Finalmente realizamos pruebas complementarias al servicio de directorio LDAP-IMP ya que en la sección 4.2.3 se realizaron las más importantes.

Los resultados de estas pruebas se encuentran en la Tabla 4.2-13

Procedimiento y/o Función	Resultado	Estado Actual	Observaciones
Inserción de certificados y de CRL	Se realiza sin contratiempos	Disponible solo para el administrador	Se realiza de manera automática al recibir los datos desde la AC en la AR. Se puede realizar también desde la interfaz web
Eliminar entrada correspondiente cuando el certificado es revocado	Se realiza sin contratiempos	Disponible solo para el administrador	Se realiza de manera automática al recibir la CRL desde la AC en la AR. Se puede realizar también desde la interfaz web
Funciones de búsqueda en el directorio y resultados de las mismas	Correcto	Disponible para todos los usuarios	Se realizaron desde la libreta de direcciones de Microsoft y desde el libro de direcciones de Netscape

Tabla 4.2-13
Pruebas de Funciones Repositorio LDAP-IMP

4.2.4.3 Pruebas de Servicios de PKI-IMP

En base a lo especificado en la sección 2.1, realizamos las siguientes pruebas valiéndonos para ello de los documentos "**Manual de Procedimiento de usuarios de PKI-IMP**" y "**Correo Seguro**", que se encuentran en los Anexos D-3 y D-4 respectivamente, y de los resultados que con ellos se obtuvieron.

1- Servicio de Manejo de claves para firmas digitales y para confidencialidad

Esto punto en específico se realizó basándonos en el documento "Correo Seguro" y en los resultados obtenidos con él.

Los resultados se muestran en la Tabla 4.2-14

Procedimiento y/o Función	Resultado	Estado Actual	Observaciones
Validación de una firma digital así como del certificado digital correspondiente	Correcto	Disponible en los clientes mencionados	Realizado en los clientes de correo MS Outlook, Outlook Express y Netscape Web Mail
Almacenamiento de claves privadas y de certificados (Repositorio de certificados de Windows y Netscape)	Correcto	Disponible en los clientes mencionados	Realizado con ayuda de los Repositorios de certificados de MS Windows y de Netscape
Recuperación de certificados en caso de pérdida	Realizado sin contratiempos	Disponible	Certificados válidos disponibles en la interfaz web pública o vía LDAP

Verificación del estado de un certificado durante el proceso de verificación de firma	Correcto	Disponible	Realizado con la ayuda de los clientes de correo por medio de CDP y de CRL
Servicio de logs (registros) de las acciones en todas las entidades	Correcto	Disponibles solo para el administrador	Realizado en la configuración de las interfaces web de los módulos y en los logs de cada entidad

Tabla 4.2-14
Pruebas de manejo de claves

2- Servicio de manejo de certificados

La mayor parte de estos aspectos ya han sido cubiertos con las pruebas pasadas. En esta sección se abordan los citados en la sección2.1. Los resultados se muestran en la Tabla 4.2-15

Procedimiento y/o Función	Resultado	Estado actual	Observaciones
Generación de Certificados digitales que se enlazan de manera simple con la AC	Correcto	Disponible	Se realiza por medio de cadenas de certificados creadas automáticamente en los repositorios de certificados mencionados
Generación de certificados de AR para agregar AR's locales a la infraestructura de manera sencilla rápida y eficiente	Correcto	Disponible	Disponible desde la AC, ya se ha abordado este punto en pruebas anteriores
Generación y publicación de CRL's de manera eficiente	Correcto	Disponible	Ya se cubrió con las pruebas anteriores
Mecanismo de verificación de certificados	Correcto salvo verificación "en línea"	Disponible	Realizado correctamente mediante CRL y CDP, pero falla utilizando la interfaz web pública

Tabla 4.2-15
Pruebas de Servicio de Manejo de certificados

3- Servicio de Publicación y almacenamiento de claves, certificados y CRL

Al igual que para el punto anterior, la mayor parte de los puntos citados en la sección 2.1 han sido cubiertos con las pruebas pasadas. Los resultados se muestran en la Tabla 4.2-16

Procedimiento y/o Función	Resultado	Estado actual	Observaciones
Publicación de CRL's actualizadas y de certificados digitales de usuarios, de la AC, de la AR y que sean fácilmente accesibles por todos los usuarios	Correcto	Disponible	Casi todos los puntos ya han sido cubiertos. El certificado de la AR se presenta al visitar la interfaz web pública, simplemente hay que descargarlo
Entradas en las BD para obtener los certificados con sus datos	Correcto	Disponible	Disponible solo para la administración de PKI-IMP

Asegurarse de que exista un solo nombre para cada objeto en la PKI-IMP	Correcto	Disponible	Automático desde la AC al momento de emitir el certificado
Servicio de directorio confidencial	No aplica	No Disponible	El servicio de directorio es público y está disponible para cualquier usuario dentro de la red IMP
Control estricto sobre la modificación de la información, solo permitido al personal autorizado	Deficiente	Disponible	Para efectos de prueba se permite el acceso al sistema desde cualquier ubicación dentro de nuestra subred, se requiere del password de la AC para poder realizar operaciones administrativas
Respaldo de clave privada	Por cuestiones de seguridad no se realiza	No disponible	La clave privada no se respalda por cuestiones de seguridad, es responsabilidad de cada usuario realizar esto

Tabla 4.2-16
Pruebas de publicación y almacenamiento de claves, certificados y CRL

4- Servicio de Interfaz con el cliente

Este punto se cubrió por completo con las pruebas del punto uno de la sección 4.2.4.1, por lo tanto no se volverá a abordar.

4.2.5 Reestructuración y evaluación de resultados

Como punto final, reestructuramos los resultados obtenidos para poder evaluar los puntos indicados en la sección 4.1. Los resultados se muestran a continuación.

Pruebas de Módulos de PKI-IMP y datos intercambiados (Tabla 4.2-17).

Módulo / Aspecto A evaluar	Clientes	AR-IMP	AC-IMP	Repositorio LDAP-IMP
El módulo funciona y se encuentra disponible	Sí	Sí	Sí	Sí
Las operaciones básicas de uso y/o intercambio de información se realizan sin contratiempos	Sí	Sí	Sí	Sí
La interacción entre módulos se lleva a cabo sin contratiempos	Sí	Sí	Sí	Sí
El servicio y la información se encuentran disponibles solo para los usuarios autorizados	Sí	No, por cuestiones de configuración	No, por cuestiones de configuración	Sí

El certificado y el par de claves son útiles y válidos en los ámbitos en los que se les utiliza	Sí	Sí	Sí	Sí

Tabla 4.2-17
Pruebas de Módulos de PKI

Certificados y las claves generadas (Tabla 4.2-18)

Certificado y claves en / Aspecto A evaluar	Clientes	AR-IMP	AC-IMP	Repositorio LDAP-IMP
El servicio funciona y se encuentra disponible	Si, certificado y claves	Sí, certificado y claves	Sí, certificado y claves	Sí, solo el certificado
Las operaciones básicas de uso se realizan sin contratiempos	Sí	Sí	Sí	No aplica, funciona solo como "llavero"
La interacción entre módulos se lleva a cabo sin contratiempos	No aplica	No aplica	No aplica	No aplica
El servicio y la información se encuentran disponibles solo para los usuarios autorizados	Si, protegidos por contraseña	No, por cuestiones de configuración	Sí, clave privada protegida por contraseña	Sí, solo aplica para el certificado
El certificado y el par de claves son útiles y válidos en los ámbitos en los que se les utiliza	Sí, en los clientes de correo y conexiones seguras	Sí, servidor web seguro y firma de solicitudes de certificación	Sí, emisión de certificados, verificación de firma de solicitudes	No aplica, aunque se puede implementar un servicio de directorio seguro vía SSL

Tabla 4.2-18
Resultados de Pruebas de Certificados y claves generadas

Pruebas de Procedimientos administrativos de PKI-IMP (Tabla 4.2-19)

Módulo / Procedimiento	Clientes	AR-IMP	AC-IMP	Repositorio LDAP-IMP
Inicialización	Descarga certificado raíz de la AC	Importa configuración de la AC e inicializa su base de datos	Crea y autofirma el certificado raíz, Inicializa su base de datos y exporta su configuración a la AR	Creación de la estructura del árbol de directorio

Registro	Solicitud de certificado ante la AR	Valida y aprueba solicitud	Crea solicitudes de AC subsidiaria, AR, servidor web y operadores de AR y AC	No aplica
Generación de par de claves	Conjuntamente con la AR	Conjuntamente con el usuario	Solo para certificados autofirmados, de AR, y de servidor web	No aplica
Certificación	No aplica	No aplica	Creación y firma de certificados	Crea nueva entrada y ahí coloca al nuevo certificado
Actualización de claves, claves expiradas o clave privada comprometidas	Nuevo par de claves y nueva solicitud	No aplica	Genera nuevas claves, emite nuevos certificados	Actualiza su base de datos
Claves expiradas	No implementado	No implementado	No implementado	No aplica
Solicitud de revocación	Genera solicitud, justifica la misma	Valida y aprueba solicitud	Revoca certificado y emite CRL	Introduce la nueva CRL

Tabla 4.2-19
Procedimientos administrativos de PKI-IMP

Aspectos de la seguridad informática relacionados con el uso de certificados; casos específicos: autenticación, confidencialidad e integridad de la información (Tabla 4.2-20).

Servicios en / Aspecto a evaluar	Clientes	AR-IMP	AC-IMP	Repositorio LDAP-IMP
El servicio funciona y se encuentra disponible	Autenticación, Integridad y confidencialidad en correo seguro y conexiones seguras	Autenticación, integridad y confidencialidad en servidor web seguro	Todos por defecto	No aplica
Las operaciones básicas de uso se realizan sin contratiempos	Firma digital y verificación de firma, cifrado y descifrado de mensajes en correo seguro, cifrado y descifrado de conexiones	Firma digital y verificación, cifrado y descifrado de conexiones	Sí, todos por defecto	No aplica

La interacción entre módulos se lleva a cabo sin contratiempos	Si, todos los procesos se completan correctamente	Si, todos los procesos se completan correctamente	Sí, todos los procesos se completan correctamente	No aplica
El servicio y la información se encuentran disponibles solo para los usuarios autorizados	Todos los servicios, clave privada protegida por contraseña	Sí, autenticación, integridad y confidencialidad	Todos por defecto, clave privada protegida por password	No aplica
El certificado y el par de claves son útiles y válidos en los ámbitos en los que se les utiliza	Sí, en todos	Sí en todos	Sí, en todos	No aplica

Tabla 4.2-20
Pruebas de Autenticación, Integridad y Confidencialidad con el uso de certificados

Con lo anterior finalizamos la reestructuración de resultados y la realización de pruebas.

Para concluir, solo resta tratar lo mencionado en la sección 4.1 respecto a las pruebas de vivacidad y seguridad del sistema.

Todos los procesos y procedimientos que se llevaron a cabo con PKI-IMP, referentes a las pruebas y puntos especificados en la sección 4.1, tanto en el lado de los clientes como en el lado de los servidores y en la administración de los mismos, iniciaron y concluyeron de manera correcta. El sistema respondió de manera correcta a errores de procedimiento y procedimientos no válidos. No se han reportado errores graves por parte de los usuarios y no se experimentó ninguno en la administración de PKI-IMP.

Con lo anterior, evaluamos y damos por superada la prueba de vivacidad del sistema.

En cuanto a la seguridad del sistema se refiere, realizamos un análisis de vulnerabilidades del sistema en el que se encuentra implementado PKI-IMP. Esto consistió en un escaneo de puertos realizado con la herramienta Nessus[1], esta herramienta, así como todas las utilizadas en la implementación de PKI-IMP, es de libre distribución.

El resultado de este análisis se encuentra en el Anexo E-1. El análisis de los resultados lo realizamos a continuación.

Solo una vulnerabilidad grave corresponde a PKI-IMP y es el servidor LDAP. Hemos seguido las recomendaciones mostradas y revisamos la configuración del servicio a fin de evitar que alguien pueda explotar esta debilidad.

Aunque las demás vulnerabilidades graves (ssh y snmp), notas informativas y alertas(warnings) no corresponden directamente a PKI-IMP, son importantes y se pondrá atención en ellas ya que son parte del sistema en el cual esta levantada la infraestructura. Estas se solucionarán y evaluarán nuevamente cuando se separen la AC y la AR en máquinas distintas. Actualmente todo está implementado en una sola máquina y esta es la principal causa de los resultados obtenidos en el análisis.

Sería deseable poder realizar un análisis del tráfico de la red y de los paquetes que por ella viajen con el objeto de poder ver y analizar algunos otros aspectos relacionados con la seguridad de PKI-IMP.

[1] Para mayor información y referencia de esta herramienta visite http://www.nessus.org

Lo anterior será propuesto como un trabajo futuro, ya que no se contempló en el inicio y no es un asunto primordial, considerando el objetivo y los alcances del proyecto.

Consideramos la seguridad de PKI-IMP como satisfactoria para efectos de ataques al sistema y/o de la explotación de vulnerabilidades a nivel operativo, y la prueba como exitosa.

En la siguiente sección realizaremos la validación de los resultados obtenidos en la presente, tomando en cuenta los objetivos iniciales del trabajo y los que después fueron propuestos como una extensión el mismo.

4.3 VALIDACIÓN

Como parte final de este capítulo, presentamos la validación de los resultados obtenidos con el trabajo realizado.

En la sección 4.1 comentamos que la validación formal no está aún ampliamente difundida y que las herramientas para ello resultan caras y escasas. Igualmente mencionamos que realizaríamos pruebas sencillas funcionales con las que pudiésemos darnos cuenta y evaluar que por lo menos el trabajo realizado hace lo esperado.

Basándonos en esto y en lo reportado hasta ahora , podemos decir lo siguiente:

Los certificados digitales, elemento principal de este trabajo de tesis, generados mediante la infraestructura de PKI-IMP (certificado de AC, certificados de servidor web, certificados de operador de AC, de AR y de usuario final) se han podido utilizar en los diferentes ámbitos en los que se les ha probado sin ningún problema. Estos ámbitos han sido principalmente:

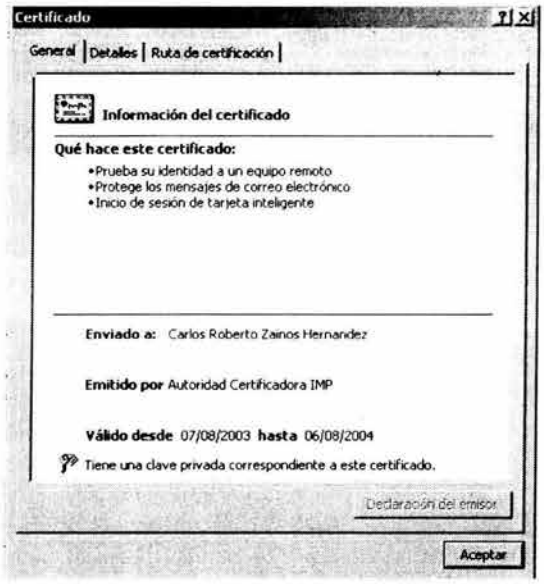
- Clientes de correo como Outlook, Outlook Express, Netscape Web Mail (en plataforma Windows) y Kmail (en plataforma Linux)
- Servidores web como lo son Apache y IIS (Nosotros no lo realizamos pero hay reportes que indican que es posible realizarlo).
- El Conjunto de librerías criptográficas tanto de plataformas Windows como de Unix/Linux cuando se utilizan éstas conjuntamente con el certificado y la clave privada para encriptar, desencriptar, firmar digitalmente, validar firmas, calcular mensajes de digestión y verificar estos.
- Almacenamiento de éstos en el servicio de directorio LDAP y disponibilidad de los mismos vía un navegador web o libros de direcciones.
- Portabilidad de los certificados así como de las claves privadas sin ningún problema.
- Navegadores web, principalmente Internet Explorer y Netscape navigator, así como Mozilla y Konqueror.

En todos ellos tanto el certificado como las claves se utilizan sin ningún problema, basta con realizar la configuración correcta para que el servicio en cuestión pueda hacer uso de ellos y brindar los servicios deseados.

Durante la fase de pruebas del uso de certificados, algunos usuarios tuvieron alguna dificultad durante la configuración de sus clientes de correo y de sus navegadores, más esto se debió principalmente a la falta de conocimiento y práctica en el uso y manejo de los certificados digitales. El uso cotidiano de éstos, así como la documentación relacionada, eliminará en el futuro la mayor parte de este tipo de problemas.

Tal vez el medio más sencillo para validar los certificados, es entrar al repositorio de certificados de Windows o de Netscape y ahí ver los que hemos generado y almacenado así como su estado.

A continuación mostramos lo que podremos ver si realizamos esto en el repositorio de certificados de un sistema Windows (Figuras 4.3-1 y 4.3-2).



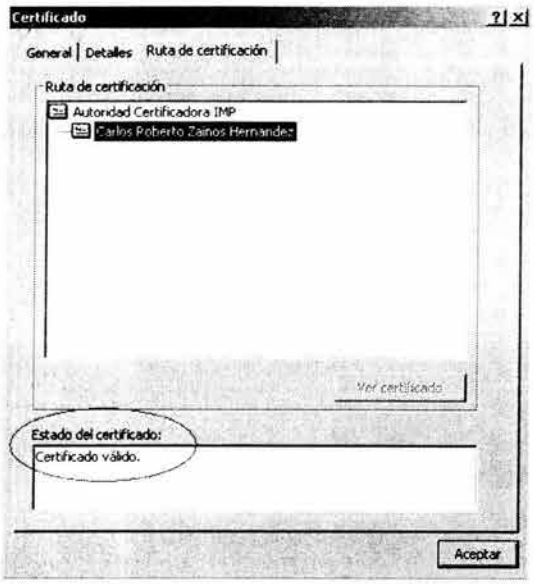
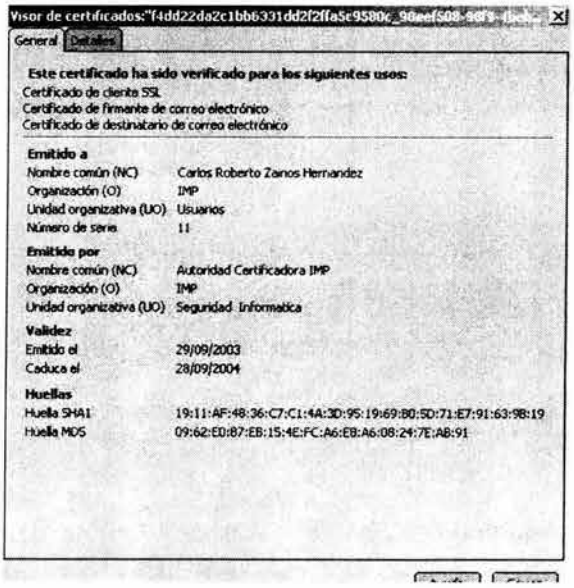


Figura 4.3-1 **Figura 4.3-2**

Certificado en el repositorio de certificados de Windows

En ellas podemos ver tanto la información general, como los detalles del mismo.

En Netscape lo que veremos es algo como esto (Figuras 4.3-3 y 4.3-4):



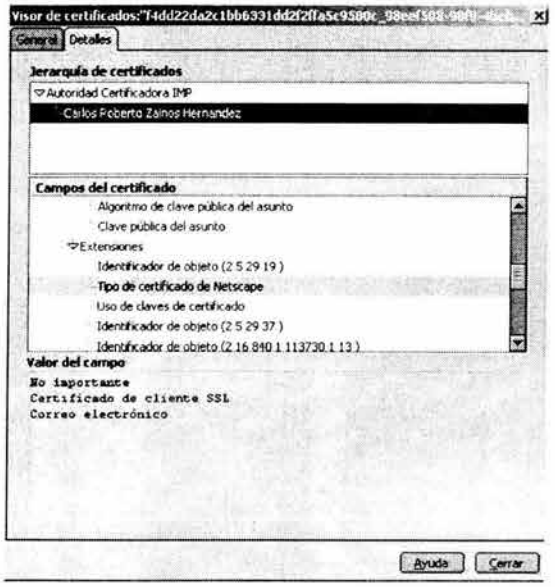


Figura 4.3-3 **Figura 4.3-4**

Certificado en el repositorio de certificados de Netscape

En ambos casos vemos que los certificados son perfectamente válidos, en las figuras anteriores (Figuras 4.3-1 a 4.3-4), mostramos un certificado de usuario; en secciones pasadas y en los anexos del presente trabajo, hemos mostrado el de un servidor web y de la autoridad certificadora, todos ellos útiles y funcionando a la perfección.

Los principales problemas se encontraron en la fase de diseño e implementación de PKI-IMP. En las pruebas de operación, los problemas encontrados fueron pequeños y fácilmente superables; estos se debieron principalmente a errores en la realización de procedimientos tanto de usuarios como de administración.

Las pruebas realizadas fueron útiles para evaluar que lo implementado, salvo que la AR no puede firmar la solicitud de certificación de los clientes, funciona como debe de ser y hace lo que tiene que hacer.

Basándonos en los resultados de las pruebas realizadas y con lo reportado en este trabajo, nos lleva a concluir que las pruebas y el trabajo realizado en todos los aspectos son válidos.

En el siguiente capítulo, indicaremos los resultados obtenidos con el presente trabajo de tesis, las conclusiones a las que podamos llegar así como los trabajos futuros.

Capítulo 5

Resultados Obtenidos

5.1 INTERPRETACIÓN DE PRUEBAS

Los resultados obtenidos en las pruebas fueron muchos y muy diversos. No todos se reportaron en este trabajo, solo los que consideramos más importantes por su trascendencia dentro del trabajo mismo, sin embargo, tanto los problemas grandes como los pequeños detalles ayudaron mucho en la comprensión y modificación de configuraciones, esquemas y/o procedimientos del proyecto.

El elemento central de este trabajo (el certificado digital), finalmente se obtuvo y se pudo trabajar con él. Se utilizó en varios ámbitos y esquemas y verificamos que realmente sirviera para los fines que se creó.

Respecto a este último punto, no debemos olvidar que el objetivo central inicial era la implementación de un esquema de autenticación basado en certificados digitales. Después de un amplio estudio y análisis, se decidió ampliar nuestras perspectivas de estudio y los alcances del proyecto hasta finalmente proponer un esquema de servicios basados en PKI. Si bien es cierto que un esquema de PKI resulta una solución muy atractiva en la búsqueda de incrementar los niveles de seguridad en una organización, respecto a la autenticación en particular, conviene anotar lo siguiente:

- Existe un concepto errado de que PKI reemplaza la necesidad de otras formas de autenticación; esto se debe a que la mayoría de los protocolos PKI implican algún mecanismo para pasar u obtener acceso a un certificado. El nombre distinguido de un certificado identifica al propietario de la clave pública que se encuentra dentro de un certificado. Dado que los mecanismos (como el que se describió en la sección 1.3 respecto a SSL/TLS) existen para hacer que la parte con la que nos estamos comunicando demuestre que tiene la clave privada correspondiente, podríamos pensar que todo está resuelto.

- A partir de aquí se encuentra uno de los pequeños detalles y secretos de PKI. Ha habido mucha discusión en la industria acerca de la necesidad de tener una fuente de confianza que expida el certificado, incluido el proceso de validar al propietario de la clave pública antes de firmar el certificado. De ahí proceden los Enunciados de las Prácticas de Certificación, CPS (*Certification Practices Statement).* Lo que no se discute tan ampliamente es la necesidad de proteger la clave privada para que alguien no la copie y la necesidad de autenticar fuertemente al propietario contra su clave privada.

- La seguridad de cualquier solución solo es tan buena como lo sea el enlace más débil de la cadena de seguridad. La criptografía de los sistemas PKI es fuerte. Nosotros podemos garantizar que tenemos un certificado expedido por una fuente de confianza (nuestra AC-IMP); podemos verificar las CRL's publicadas para comprobar que un certificado sigue siendo válido; sin embargo nada de eso es seguro si la clave privada (de cualquier elemento de la infraestructura) es vulnerable a la copia o se puede tener fácil acceso a ella.

- Como resultado de esto, una PKI fuerte necesita basarse en autenticación fuerte del propietario frente a su clave privada. Hemos mencionado algunas técnicas principales de autenticación como contraseñas, identificadores de prenda (tokens) de autenticación, tarjetas inteligentes y biométricas. La mayoría de ellas, dentro del ámbito de estudio/investigación del que se trata, están fuera de nuestro alcance y por lo tanto no fueron consideradas. Si realmente se tiene el deseo y el objetivo de implementar una solución basada en PKI de manera seria y formal, será necesario considerar esto como una parte primordial.

- Cada forma de autenticación cuenta con diferentes niveles de seguridad, distintas características de facilidad de uso, diferentes costos de adquisición y distintos costos de administración. Podríamos preguntarnos cuánta seguridad es necesaria o hasta dónde es suficiente. Eso dependerá de las políticas de cada organización, del nivel de cultura informática que se tenga o de la importancia que para nosotros representen los datos e

información que estemos manejando. Es bien sabido que el incrementar los niveles de seguridad trae como consecuencia una caída en el rendimiento. Este tal vez sea el costo más grande que haya que pagar y que muchas empresas y organizaciones no están dispuestas a hacerlo, ya que para recuperarse de esa pérdida en la mayoría de los casos es necesario una inversión extra en equipos que la mayoría de las veces no están contemplados en los planes de inversión del negocio.

- PKI-IMP delega la responsabilidad de la protección de la clave privada al propietario de la misma, para esto sugiere el uso de passwords para el uso y protección de dicha clave, estamos seguros de que este no es el mejor mecanismo de protección de dicha clave, pero es lo que por el momento está a nuestro alcance.

Hemos visto como es extremadamente sencillo ingresar a una computadora, accesar al repositorio de certificados y obtener una copia del certificado y de la clave privada cuando esta no está protegida. Aún cuando en las declaraciones de uso y/o manuales de procedimiento se recomiende lo contrario, es casi un hecho que los usuarios con el objeto de evitar "complicaciones" en el uso y manejo de passwords de encripción opten por no proteger su clave privada con una contraseña. Es aquí donde se encuentra nuestro eslabón más débil en la cadena de seguridad.

El trabajo de tesis se concentra en la autenticación de los participantes en transacciones electrónicas mediante el uso de certificados digitales y claves públicas/privadas, mas no en la autenticación del propietario de la clave privada frente a la misma. Como se dijo antes, PKI-IMP delega esta responsabilidad al propietario de dicha clave. Para esto PKI-IMP podría en un futuro proporcionar facilidades para esto mediante el uso de tarjetas inteligentes, llaves USB o algún otro dispositivo o esquema similar que exista en el mercado.

Técnicamente, los resultados obtenidos en las pruebas son muy buenos, válidos y nos permiten utilizar con toda confianza el certificado y la infraestructura. Existen, es cierto, algunas deficiencias en la misma, sin embargo consideramos que el principal problema y debilidad, como en la mayoría de los casos, radicará en el usuario final.

El proyecto realizado y los resultados obtenidos con el presente trabajo de tesis, son buenos y muy útiles. La tesis cubre los objetivos que se planteó, además sirve como una fuente de conocimiento dentro de la cultura de la seguridad informática del IMP; este y otros puntos se tratan en la siguiente sección, referente a las conclusiones del trabajo.

5.2 CONCLUSIONES

Las conclusiones a las que llegamos con el trabajo realizado, basándonos en los objetivos del mismo, en las pruebas y en los resultados obtenidos, las enunciamos a continuación.

- Respecto a la ampliación del objetivo inicial del trabajo hacia la concepción, diseño e implementación de una PKI, podemos decir que es un trabajo por demás benéfico, útil y explotable para el IMP y en particular para el área de seguridad informática, ya que detrás de esto existe un amplio estudio e investigación de estándares, métodos, modelos y soluciones que por su importancia tendrán mayor desarrollo y difusión en el futuro.

- Inicialmente se planteó un problema y una propuesta de solución a dicho problema. Dicha solución se centraba en un objetivo específico principal: la autenticación de los participantes en transacciones electrónicas basada en certificados digitales. Propusimos el diseño y la implementación de una PKI (como una solución a dicho problema), porque ésta además de brindar principalmente servicios de autenticación, brinda facilidades y servicios de seguridad que resultan más atractivos y útiles para el Área de Seguridad Informática del IMP y para el IMP mismo.

- Lo que se propuso e implementó como solución, se encuentra perfectamente fundamentado en bases sólidas de las directrices de la seguridad en cómputo que se encuentran actualmente vigentes o que guían las futuras tendencias de la misma. Podemos decir sin temor a equivocarnos que el rumbo que hemos marcado tiene grandes expectativas de investigación, crecimiento y desarrollo; que el riesgo de llegar a un camino sin salida o al final de este es muy pequeño y que podemos estar seguros de que la inversión hecha en este rubro dará muy buenos resultados y situará al IMP a la vanguardia en cuanto a soluciones y servicios de seguridad se refiere.

- Durante el desarrollo del trabajo, fuimos conociendo modelos, esquemas, directrices y tendencias que se siguen en el campo de la seguridad informática. Gracias a esto la visión y perspectivas, en cuanto a seguridad se refiere, se ampliaron considerablemente para el área de Seguridad Informática del IMP. Esto nos dio la capacidad de identificar deficiencias y fallas en los esquemas de seguridad que actualmente se aplican en el IMP.

- El proyecto logró establecer una entidad de confianza, la Autoridad Certificadora del IMP (AC-IMP) que es quién emite los certificados digitales correspondientes, sobre la cual descansa toda la infraestructura de PKI-IMP.

- Uno de los principales logros de PKI-IMP, es que logró establecer una identidad digital en la que se puede confiar (el certificado digital emitido y firmado por AC-IMP). Este certificado se pudo utilizar conjuntamente con el par de claves asociadas a él y con los mecanismos criptográficos disponibles, para brindar servicios de seguridad como autenticación (principalmente), integridad y confidencialidad.

- El uso de firmas digitales y de certificados emitidos y validados por una autoridad certificadora de confianza, AC-IMP, constituye un mecanismo de autenticación de participantes en transacciones electrónicas mucho más fuerte que el actualmente utilizado en el IMP basado en usuario-password.

- La infraestructura de PKI resultante consideramos que es buena, cubre y cumple a la perfección con los objetivos iniciales y va más allá, ya que, adicionalmente al servicio de autenticación, brinda servicios de integridad y confidencialidad en el manejo de la información intercambiada. En el futuro, podría también proporcionar servicios de auditoria en transacciones electrónicas (No repudio).

- PKI-IMP brinda servicios y soluciones a problemas que en el futuro próximo harán su aparición dentro del IMP, tales servicios serán herramientas indispensables como ahora lo es el servicio de correo electrónico institucional.

- Con el fin de evaluar y validar el trabajo realizado, se propuso un esquema de pruebas y se realizaron estas. Las dificultades y problemas encontrados fueron resueltos, en su mayoría, de manera satisfactoria. Hubo algunos detalles, en cuanto a la administración de PKI-IMP, que no se pudieron resolver y que se citan en la sección de Trabajos Futuros, pero que por su naturaleza, no representan un impacto negativo al proyecto en conjunto.

- Las pruebas realizadas al proyecto PKI-IMP y los resultados obtenidos, garantizan que esta es una buena solución y que se puede implementar a nivel institucional.

Como parte final del presente trabajo de tesis, en la siguiente sección mencionamos los trabajos futuros que el proyecto PKI-IMP tiene por delante.

5.3 TRABAJOS FUTUROS

Para este, como para todo trabajo de desarrollo e investigación, los trabajos futuros son importantes; ya que ellos representan la continuidad de la solución o del trabajo realizado. En este sentido podríamos pensar que entonces el trabajo se vuelve interminable, y de hecho lo es.

Son muy pocos los casos en los que una organización invierte en desarrollo e investigación y al final abandona o descarta los resultados obtenidos. Es deseable que exista una continuidad en lo que se realiza ya que eso nos garantizaría, al menos en teoría, presencia e impacto en los procesos de negocio o investigación en los que nos basamos. Igualmente no podemos quedarnos solo con los resultados obtenidos ya que resultarían poco atractivos en la planeación de la continuidad del negocio y lo más seguro es que caigan en desuso o en el olvido.

El proyecto PKI-IMP consideramos que tiene aún mucho trabajo por delante.

Existe un tipo de trabajos que, por su naturaleza, están fuera de nuestro alcance o de nuestro campo de acción, pero que impactan negativamente y representan en gran medida obstáculos para los trabajos que si están en nuestras manos. Estos se refieren a la manera de operar, distribuir el trabajo, delegar responsabilidades, asignar atributos y facilidades en el IMP.

Aunque dijimos que lo que haríamos sería un modelo prototipo de estudio y experimentación, hubiese sido deseable contar con un equipo de cómputo para realizar un prototipo más robusto. No queremos decir que el actual no sirva, sino que podríamos eliminar de nuestras consideraciones cuestiones referentes a espacio en disco duro, velocidad de procesamiento, rendimiento de las aplicaciones, velocidad de respuesta, las cuales estuvieron presentes e impactaron mucho de manera negativa en el prototipo construido en la plataforma y con los recursos de software descritos en la sección 3.1

Lo primero que recomendamos es separar a la AC y a la AR en equipos distintos con un nivel tecnológico aceptable. Proponemos una arquitectura Intel Pentium II para la AC y una Pentium III para la AR, las arquitecturas PowerPC de Macintosh y Ultrasparc de SUN son igualmente bienvenidas. En ambas plataformas, AC y AR, seria bueno contar con una distribución Linux RedHat 8 o superior, o cualquier otro sistema UNÍX-Like.

Además de esto, y con mayor prioridad, es importante abordar y trabajar sobre los siguientes puntos:

Primeramente definir, diseñar y emitir las políticas y los lineamientos a los cuales estará sujeta la infraestructura PKI-IMP. De entre estos por su importancia y como un requisito de diseño y administración de PKI, se encuentra las Declaraciones de Prácticas de Certificación (Cetification Practices Statement, *CPS*). Las reglas que describen la manera como las diferentes facetas de una AC están limitadas y operan se definen en el documento de las CPS's. Una Declaración de Prácticas de Certificación para la AC que expidió el certificado debe estar disponible para el usuario del certificado. Si no se encuentra disponible una CPS, esto puede producir una duda razonable sobre la veracidad de la AC y reducir la confianza en la entidad que lo expide. Esto pensando en que tal vez un día la infraestructura PKI-IMP podría prestar servicios de certificación a usuarios fuera de la red IMP, certificados de servidor para sitios seguros, certificados a usuarios finales, etc.

Dentro de la estructura de PKI-IMP quedan varios pendientes que sería conveniente realizar en el corto plazo.

Uno de ellos es el probar una configuración de AC en malla, esto con el objetivo de poder hacer frente a una contingencia mayor en el acaso de que una AC se viera comprometida, actualmente trabajamos en un esquema jerárquico de AC, más si la AC raíz se ve comprometida, entonces toda la infraestructura se vendría abajo. En una configuración de AC's en malla, si alguna de ellas se ve

comprometida lo que procede es eliminar el nodo correspondiente y toda la rama que de él se derive, los demás nodos y ramas correspondientes se conservan y la infraestructura se mantiene en pie.

Los módulos de PKI-IMP en general trabajan bien, mas en cada uno de ellos existen algunos detalles que debieran mejorar. Por ejemplo, en la AC no existe un mecanismo que permita validar la identidad del operador de la misma y con ello se controle el acceso a la interfaz de la AC y al nodo de administración de los datos, esto es, actualmente aunque se cuenta con certificados de operador de AC, el mismo se encuentra instalado en el navegador con el que se accesa a la interfaz de la AC, así que cualquier persona que tenga acceso a la computadora en la que reside la AC y sepa la ubicación y el puerto por el cual se presta el servicio, es capaz de ejecutar operaciones administrativas. Lo que en principio se propondría es un mecanismo de password que controle y permita el acceso a las interfaces de administración de la AC basándose en el conocimiento del mismo. Lo ideal sería contar con dispositivos de identificación de prenda (tokens criptográficos) de autenticación en las cuales se almacene el certificado de Operador de AC con el objetivo de que este funcione como una llave.

Hay algunos otros esquemas en los que se requiere de más de un usuario para operar la AC, en estos casos los encargados de la administración de la AC deben estar todos presentes para poder utilizar la AC. Cada uno de ellos cuenta con un password de autorización para realizar cualquier operación de la AC. Se requiere de todos los passwords para ejecutar cualquier acción en la AC. Con esto lo que ganamos es que se requiere de un consenso en la tomar las decisiones de lo que hará la AC. Esto logra delegar responsabilidad y evita que todo recaiga en una sola persona.

Actualmente tenemos una sola AR y esta pudiese verse comprometida en cuanto a seguridad se refiere, o simplemente verse rebasada en sus capacidades administrativas o de respuesta. Para tal caso no contamos con una AR de "reserva" que pudiese atender a los usuarios mientras se reestablece el servicio en la primera. Para el caso de un compromiso de seguridad lo que hacemos es eliminar la AR y crear una nueva desde la AC, sería la solución más rápida. Para el caso de que se viera rebasada en cuento a capacidades, no tenemos manera de enfrentar dicha contingencia. Por esto, un pendiente que se debe atender en el corto plazo es levantar una nueva configuración de AR y de ser posible llevarla a algún lugar fuera de la ubicación actual. Podríamos pensar por ejemplo en una distribución de AR's por regiones geográficas o por gerencias, de esta manera distribuimos el trabajo y el tráfico de los usuarios de PKI-IMP, e igualmente evitamos una interrupción en los servicios que ella presta. Esta configuración, junto con el de las AC's en malla serían un muy buen caso de estudio e investigación.

En el caso del control de la validez de los certificados, actualmente manejamos una configuración basada en CRL's disponibles en los Puntos de distribución de CRL (CRL Distribution Point, CDP). Esta configuración es deficiente ya que se requiere que el navegador del cliente esté configurado de manera que cada vez que se haga uso de un certificado, éste vaya al CDP y verifique el estado del mismo. Otra opción es que el usuario sea quien descargue periódicamente la última CRL emitida. Esto, al igual que el caso anterior, presenta deficiencias si consideramos que hoy en día existen usuarios que ni siquiera saben que existen actualizaciones del servicio de antivirus institucional. En este sentido el administrador de la AC juega un papel muy importante, ya que es él el responsable de publicar la CRL correspondiente cuando el periodo de validez de ésta haya expirado o cada vez que un certificado sea revocado y notificarlo a los usuarios. Puede darse el caso de que un certificado haya sido revocado y no se haya emitido la CRL correspondiente, por tanto, el certificado podría seguir siendo utilizado y pasar este como válido.

Para este caso lo que proponemos es implementar un servicio de verificación en línea del estado del certificado (On-line Certificate Status Protocol, OCSP). OCSP es un protocolo que está definido en el RFC 2560 de la IETF. El objetivo de OCSP es superar limitaciones en los esquemas de revocación basados en CRL y ofrecer respuesta inmediata y actualizada a la consulta de estado de los certificados. La información específica de revocación se devuelve, en lugar de convertirse en una larga lista lineal de búsqueda en la forma de una CRL. Esto se logra por medio de un servicio conocido como un "respondedor" de OCSP, por medio del cual se devuelve el estado del certificado.

La respuesta indica el estado del certificado regresando los valores "bueno", "revocado" y "desconocido" (cuando no se puede identificar el certificado). La última distribución de OpenCA (0.9.2) incluye un respondedor de OCSP. Convendría probarlo y adoptarlo.

Respecto a esto último, al momento de terminar este trabajo se encontraba ya disponible la versión de evaluación 0.9.2 de OpenCA. Paralelamente al trabajo implicado con PKI-IMP, sería conveniente descargar esta distribución y probarla, ya que estamos seguros de que incluye mejoras importantes, algunas de las cuales solucionarían algunos de los problemas hasta ahora mencionados.

Es necesario realizar pruebas y analizar resultados (e incluir conclusiones de estos en las CPS's) referentes a los ciclos de vida de las claves y del certificado.

Otro aspecto ligado con el anterior es el de la administración del certificado. Es necesario especificar claramente en las CPS's bajo qué casos el certificado será revocado e igualmente cómo se manejará la renovación de los mismos, incluyendo el de la AC, AR y servidores web. Nosotros lo tratamos de manera muy superficial, más esto debe profundizarse de manera que la administración de los mismos sea sencilla, rápida y no consuma demasiados recursos tanto computacionales como administrativos.

Los protocolos de administración soportan la transferencia de solicitudes e información administrativa entre las entidades destino PKI y las entidades de administración de PKI, y para comunicación entre las entidades administrativas de PKI. Los protocolos pueden, por ejemplo, transportar solicitudes de registro, estados de revocación o solicitudes de certificación cruzada y las correspondientes respuestas. En la actualidad los protocolos de administración se definen en tres documentos: los protocolos de administración de certificado (Certificate Management Protocols, CMP) en el RFC 2510 de IETF, los mensajes de administración de certificados sobre CMS (Certificate Management Messages over CMS[1], CMC) en el RFC 2779, y el Formato de Mensaje de Solicitud de Certificado (Certificate Request Message Format, CRMF) definido en RFC 2511.

Los protocolos anteriores son los más importantes, mas no los únicos. Actualmente están en estudio y pruebas algunos otros. Conviene estudiar mas a fondo estos de manera que nos ayuden en el funcionamiento y administración de PKI-IMP.

Otro aspecto importante es el probar los certificados de PKI-IMP en el servicio de VPN con el que cuenta actualmente el IMP. El IMP cuenta con un servicio de Firewall por medio del cual se prestan servicios de VPN a usuarios viajeros. En dicho servicio es posible manejar un control de acceso basado en nombre de usuario y certificado digital. Este esquema no se encuentra en uso por falta de estudio primero, y por falta de certificados en segundo lugar; por lo que el esquema de autenticación manejado es el típico de nombre de usuario y password.

El servicio de Firewall cuenta con una herramienta de emisión de certificados y con una AC, sin embargo hay que tener en cuenta que para poder hacer uso de esto es necesario solicitarlo en nuestra licencia de uso y pagando el correspondiente costo que ello implique. Si fuese posible configurar el servicio del Firewall de manera que acepte los certificados digitales emitidos y firmados por nuestra AC, entonces el mecanismo de autenticación se reforzaría considerablemente sin necesidad de realizar costosas inversiones. Este trabajo está pendiente.

Otro pendiente es la elaboración de documentos informativos que promuevan el uso de PKI-IMP y/o el refinamiento de los manuales de usuario (principalmente) y administración de PKI-IMP.

Los anteriores puntos mencionados se basan en el trabajo hecho con la herramienta OpenCA, pero no hay que olvidar que contamos con otra opción la cual hicimos de lado para concentrarnos en PKI-OpenCA y que es nuestra AC levantada con el conjunto de herramientas de OpenSSL.

[1] CMS- Cryptographic Message Syntax, forma parte de S/MIME en el mundo del protocolo de correo electrónico.

Para ambos casos, en un principio adicionalmente se contempló la construcción de una aplicación que fuese capaz de tomar como entrada un archivo, programa o aplicación y, haciendo uso de los certificados digitales, generase como salida un archivo cifrado y/o firmado digitalmente; de manera que, haciendo uso de la misma aplicación y de un certificado digital o una clave privada, pudiese realizar la operación inversa descifrando el archivo cifrado que toma como entrada y produciendo como salida un archivo de "texto en claro". Igualmente esto serviría para verificar la firma digital, si es el caso. Lo anterior es una manera de proporcionar autenticación, integridad y confidencialidad en el intercambio de archivos. Esto ya no se realizó, pero consideramos que es importante y muy necesario contar con dicha aplicación.

Los algoritmos y protocolos utilizados en este trabajo fueron los que se tenían al alcance. Principalmente los soportados por las aplicaciones en las que se probaron los certificados y el par de claves. Existe un esquema de seguridad basado en las llamadas curvas elípticas. Es conveniente realizar un estudio profundo al respecto y proponer una implementación de ello y realizar pruebas correspondientes en la actual infraestructura de PKI-IMP.

Existen muchas más cosas por hacer en busca de una mejora al modelo PKI-IMP. En esta sección hemos mencionado los puntos que a nuestro juicio son los más importantes para su estudio e implementación en el corto plazo.

Anexos

Anexo A-1
Public Key Criptography Standards (PKCS's)[1]

Inicialmente, RSA Laboratories desarrolló los *Estándares de la Infraestructura de Claves Públicas* (*Public Key Cryptography Standards*,PKCS) en colaboración con la industria, la academia y representantes del gobierno par avanzar en la interoperabilidad de la criptografía de claves públicas. Encabezada todavía por RSA, el trabajo de PKCS se ha expandido a través del tiempo para cubrir un creciente grupo de estándares de formato, algoritmos y las API's de PKCS. Los estándares PKCS ofrecen definiciones fundamentales de formatos de datos y algoritmos que se encuentran virtualmente en todas las implantaciones PKI actuales.

Los estándares PKCS se enuncian a continuación:

PKCS #1 Estándar de cifrado RSA

PKCS #1 define reglas de formateo básicas para las funciones RSA de claves públicas, específicamente firmas digitales. Define la manera como se calculan las firmas digitales, incluidos el formato de los datos que se van a firmar y el de la firma misma. También define la sintaxis de las claves pública y privada RSA.

PKCS #2

Cubre el cifrado RSA de las reseñas de mensajes (hashes) y fue incorporado al PKCS #1.

PKCS #3 Estándar de acuerdo de clave Diffie-Hellman

PKCS #3 describe un método para implantar el acuerdo de clave Diffie-Hellman.

PKCS #4

Originalmente especificó la sintaxis de claves RSA, pero al igual que PKCS #2, fue incluido en el PKCS #1.

PKCS #5 Estándar de cifrado basado en contraseñas

PKCS #5 describe un método para cifrar una cadena octeto con una clave secreta, derivada de una contraseña para producir una cadena octeto cifrada. PKCS #5 se puede usar para cifrar claves privadas y permitir el transporte seguro de las mismas, como se describe en PKCS #8.

PKCS #6 Estándar de sintaxis de certificado extendido

Define una sintaxis para certificados X.509 extendidos con atributos que ofrecen información adicional acerca de la entidad. (Cuando PKCS #6 se publicó por primera vez, X.509 no se había revisado para soportar extensiones. Desde entonces estas extensiones se han incorporado en X.509).

PKCS #7 Estándar de sintaxis de mensaje criptográfico

PKCS #7 especifica una sintaxis general para datos que pueden tener criptografía, tales como firmas digitales y sobres digitales. PKCS #7 ofrece varias opciones de formateo, incluidos mensajes formateados sin ningún cifrado o firma, mensajes en sobre (cifrados), mensajes firmados, y mensajes a la vez cifrados y firmados.

[1] Uso de PKI en aplicaciones, Capítulo 7; PKI Infraestructura de claves públicas; Andrew Nash, William Duane, Cecilia Joseph y Derek Brink; RSA Press- Mc Graw Hill 2002

PKCS #8 Estándar de sintaxis de información de clave privada

PKCS #8 define la sintaxis de información de clave privada y la sintaxis de clave privada cifrada, en la que el cifrado de clave privada emplea PKCS #5.

PKCS #9 Tipos de atributos seleccionados

PKCS #9 define tipos de atributos seleccionados para usar en certificados extendidos PKCS #6, mensaje firmados digitalmente PKCS #7, información de clave privada PKCS #8 y solicitudes de firmas de certifica PKCS #10. Los atributos de certificados definidos incluyen dirección de correo electrónico, nombre sin estructura, tipo de contenido, reseña del mensaje, tiempo de firma, contrafirma, contraseña de pregunta y atributos de certificado extendido.

PKCS #10 Estándar de sintaxis de solicitud de certificación

PKCS #10 define una sintaxis para solicitudes de certificación. Una solicitud de certificación consta de un nombre distinguido, una clave pública y, opcionalmente, un conjunto de atributos, firmados colectivamente por la entidad que solicita la certificación.

PKCS #11 Estándar de interfaz criptográfica de señal

PKCS #11 o "Cryptoki" especifica una interfaz de programación de aplicación (*Application Programming Interface* ,API) para dispositivos de usuario único que contiene información criptográfica (tales como claves de cifrado y certificados) y realiza funciones criptográficas. Las tarjetas inteligentes son dispositivos típicos que implantan Cvryptoki. Observe que Cryptoki define la interfaz para funciones criptográficas y no especifica la manera como el dispositivo va a implementar las funciones. Además, Criptoki solamente especifica interfaces criptográficas y no define otras interfaces que pueden ser útiles para el dispositivo, como el acceso al sistema de archivo de dicho dispositivo.

PKCS #12 Estándar de sintaxis de intercambio de información personal

PKCS #12 define un formato para la información de identificación personal, incluidas claves privadas, certificados, secretos varios y extensiones. PKCS #12 facilita la transferencia de certificados y claves privadas asociadas, de modo que los usuarios puedan mover la información de su identificación, de dispositivo a dispositivo.

PKCS #13 Estándar de criptografía de curva elíptica

En la actualidad, PKCS #13 continúa en desarrollo. Cubre la generación y validación de parámetros de curva elíptica, generación y validación de claves, firmas digitales y cifrado público, lo mismo que acuerdo de claves, sintaxis ASN.1 para parámetros claves e identificación de esquema.

PKCS #14 Estándar de generación de números pseudoaleatorios

PKCS #14 en la actualidad está en desarrollo. Podríamos preguntarnos el por qué requiere este proceso su propio estándar. Muchas funciones criptográficas fundamentales usadas en PKI, como la generación de claves y la negociación de secreto compartido Diffie-Hellman, usan datos aleatorios. Sin embargo, si los datos aleatorios no son aleatorios, sino que en realidad se seleccionan a partir de un conjunto de valores predecibles, la función criptográfica ya no es del todo segura, dado que sus valores están restringidos a un campo reducido de posibilidades. Por lo tanto, la generación segura de números pseudoaleatorios es vital para la seguridad de PKI.

PKCS #15 Estándar de la sintaxis de información criptográfica de señal.

PKCS #15 promueve la interoperabilidad de señales criptográficas mediante la definición de un formato común para objetos criptográficos almacenados en una señal. Los datos guardados en un dispositivo que implementa PKCS #15 parecerán iguales a cualquier aplicación que use el dispositivo, aunque en realidad pueden estar implementados en un formato interno diferente. La implementación de PKCS #15 actúa como un intérprete que traduce el formato interno de la tarjeta y el formato que esperan las aplicaciones.

Anexo B-1

Sección I

Procedimiento de Instalación de OpenSSL

La secuencia de comandos que se deben teclear desde la línea de comandos para compilarlo e instalarlo siguiendo la configuración por default es:

Para descomprimirlo:

$ gzip –d openssl-0.9.7.tar.gz
$ tar xvf openssl-0.9.7.tar

Lo anterior crea un directorio en la ubicación actual llamado OpenSSL-0.9.7 y dentro de él varios subdirectorios, continuamos y desde la terminal tecleamos:

$ cd openssl-0.9.7

Configuramos la distribución en nuestro sistema:
$ ./config

Si el sistema no marca ningún error, procedemos con:
$ make

Nuevamente, si todo funciona correctamente y no se desplegó ningún error
$ make test

Por último, y si las pruebas anteriores concluyeron exitosamente, procedemos a instalarlo. Hay que tener en cuenta que debemos hacerlo como *root* o tener sus privilegios para poder hacerlo.

$ make install

La instrucción anterior instalará OpenSSL en nuestro sistema. El directorio de instalación por default es /usr/local/ssl, aunque se puede indicar algún otro directorio. Conviene leer la documentación de instalación si es que se requiere esto o si se desean utilizar algunas otras opciones de configuración. En este directorio encontraremos los binarios base del programa en /bin y varios scripts para ayudarnos a crear y manejar una autoridad certificadora en /misc.

Sección II

Creación de la Estructura de directorio para la Administración de Certificados

Creamos la estructura de directorio necesaria para la administración de certificados:

Dentro de /usr/local/ssl :

$ mkdir /certs
$ mkdir /crl
$ mkdir /newcerts
$ mkdir /private
$ echo "01" > /serial
$ touch /index.txt

La penúltima línea crea un archivo de texto llamado serial y escribe en él el valor 01, este archivo llevará la "cuenta" de los certificados emitidos, dicha cuenta corresponderá al serial del certificado que se esté emitiendo. La última línea crea un archivo de texto llamado index.txt, este archivo es el "índice" de los certificados que se han emitido. Cada que se quiera generar un nuevo certificado, OpenSSL verificará que en dicho archivo no exista un serial o un DN igual al que se está generando, una vez que ha verificado esto emite el nuevo certificado y escribe una nueva línea a este archivo con la información del nuevo certificado.

Sección III

Modificando el archivo de configuración de OpenSSL

Para poder utilizar correctamente OpenSSL con la estructura de directorio creada anteriormente, es necesario realizar algunos cambios en el archivo de configuración en la sección de solicitud de certificado con la siguiente información:

```
####################################################################
[ req ]
default_bits                = 1024              # Longitud de la clave privada
default_keyfile             = privkey.pem       # Nombre bajo el cual se guardará si no se
                                                # especifica uno
distinguished_name          = req_distinguished_name    # Extensiones utilizadas para el
                                                        # nombre distinguido.
attributes                  = req_attributes    # Atributos extras para la solicitud
x509_extensions             = v3_ca             # Las extensiones que se agregan al
                                                # certificado autofirmado

[ req_distinguished_name ]          # Especificaciones para el DN en la solicitud

countryName                 = Country Name (2 letter code)
countryName_default         = MX                # Identificador de país, C
countryName_min             = 2                 # Longitud mínima del identificador
countryName_max             = 2                 # Longitud máxima del identificador

stateOrProvinceName         = State or Province Name (full name)  stateOrProvinceName_default
        = Distrito Federal  # Identificador de estado o provincia, L

localityName                = Locality Name (eg, city)
localityName_default        = Mexico            # Identificador de Localidad

0.organizationName          = Organization Name (eg, company)
0.organizationName_default  = IMP               # Identificador de la Organización, O

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default = Seguridad Informatica      # Identificador de la Unidad
                                                            # Organizacional OU

commonName                  = Common Name (eg, YOUR name)
commonName_max              = 64                # Longitud del Nombre común, CN

emailAddress                = Email Address
emailAddress_max            = 64                # Longitud dirección de correo, E
```

La demás secciones permanecen sin cambios

Sección IV

Creación y autofirma de un certificado digital de AC utilizando el script CA.pl

Desde la línea de comandos dentro de /usr/local/ssl/misc/ tecleamos:

$ CA.pl –newca

```
CA certificate filename (or enter to create)          # Dejamos en blanco
Making CA certificate ...
Using configuration from /usr/local/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to './demoCA/private/cakey.pem'   # La clave privada generada y su ubicación
Enter PEM pass phrase: caimp0                         # Contraseña que protege la clave privada, muy importante
Verifying password - Enter PEM pass phrase: caimp0    # Confirmar contraseña
---
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some
blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
---
Country Name (2 letter code) [MX]:MX                  # Identificador del país según X.500, MX para México
State or Province Name (full name) [Distrito Federal]:Distrito Federal  # Identificador de estado o provincia
Locality Name (eg, city) [Mexico]:Mexico              # Identificador de localidad
Organization Name (eg, company) [IMP]:IMP             # Nombre de la organización
Organizational Unit Name (eg, section) [Seguridad Informatica]:Seguridad Informatica # Area o Unidad Organizacional OU
Common Name (eg, YOUR name) []:acimp                  # Nombre del sujeto o CommonName (CN)
Email Address []:root@raguel.imp.mx                   # e-mail del sujeto
```

La contraseña que dimos arriba (PEM pass phrase) es de vital importancia - la utilizaremos siempre que queramos firmar un certificado. Sin esta, o con una contraseña incorrecta, OpenSSL generará un error y abortará la operación.
Hasta este momento podemos considerar que terminó al primera fase, el final de la cual generamos una clave privada que en este momento se encuentra en:
/usr/local/ssl/misc/private/cakey.pem.

Ahora generamos una petición de certificado, desde la línea de comandos:

$ CA.pl –newreq

Lo anterior le dice al script que genere un nuevo requerimiento para certificado.

```
Using configuration from /usr/local/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.++++++
.....++++++
writing new private key to 'newreq.pem'               # nombre del archivo del requerimiento
Enter PEM pass phrase: caejem                         # contraseña para el requerimiento
Verifying password - Enter PEM pass phrase: caejem    # confirmar contraseña
---
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
---
Country Name (2 letter code) [MX]:MX                  # Identificador del pais
State or Province Name (full name) [Distrito Federal]:Distrito Federal # Identificador de estado o provincia
Locality Name (eg, city) [Mexico]:Mexico              # Identificador de Localidad
Organization Name (eg, company) [IMP]:IMP             # Nombre de la organización
Organizational Unit Name (eg, section) [Seguridad Informatica]:Seguridad Informatica  # Area o Unidad Organizacional OU
```

```
Common Name (eg, YOUR name) []:acimp                    # Nombre del sujeto o Common Name (CN)
Email Address []:root@raguel.imp.mx                     # e-mail del sujeto

Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:secreta
An optional company name []:UNAM
Request (and private key) is in newreq.pem               # Nombre y ubicación del requerimiento y de la clave privada
```

Queda entonces hecha la petición de certificado, así como nuestra clave privada, en el archivo newreq.pem.

El siguiente paso es firmarlo:

$ CA.pl –sign

```
Using configuration from /usr/local/ssl/openssl.cnf
Enter PEM pass phrase: caimp0                           # Solicita contraseña para la clave privada de la AC
Check that the request matches the signature            # Verifica que la contraseña concuerde así como la firma
Signature ok
Certificate details:                                    # A continuación muestra la información contenida en el cert
        Serial Number: 1(0x1)
        Validity
          Not Before: May 23 18:43:31 2003
          Not After  : May 22 18:43:31 2004
        Subject:
        countryName :PRINTABLE:'MX'
        stateOrProvinceName :PRINTABLE:'Distrito Federal'
        localityName :PRINTABLE:'Mexico'
        organizationName :PRINTABLE:'IMP'
        organizationalUnitName:PRINTABLE:'Seguridad Informatica'
        commonName :PRINTABLE:'acimp'
        emailAddress :IA5STRING:'root@raguel.imp.mx'
Certificate is to be certified until May 22 18:43:31 2004 GMT (365 days)
Sign the certificate? [y/n]:y                           # Pregunta si deseamos firmar el certificado, decimos que si

1 out of 1 certificate requests certified, commit? [y/n]y   # confirmamos el procedimiento
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem                    # Nombre del nuevo certificado, el de la AC
```

Por último, toca poner nuestro certificado firmado por nosotros mismos y la clave privada en donde OpenSSL espera encontrarlos, en el directorio especificado en /usr/local/ssl/openssl.cnf.

$ *cp newcert.pem /usr/local/ssl/private/*
$ *cp demoCA/private/cakey.pem /usr/local/ssl/private/*

Sección V

Archivo de configuración de OpenSSL para Autoridad Certificadora

openssl-ca.cnf en /usr/local/ssl

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.

# Archivo de configuracion de openSSL para una AC por Carlos Roberto  Zainos Hdez Para IMP

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = /usr/local/ssl/.rand
```

```
# Extra OBJECT IDENTIFIER info:
#oid_file             = $ENV::HOME/.oid
oid_section           = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions          =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca     = CA_default          # The default ca section

####################################################################
[ CA_default ]

dir            = /usr/local/ssl      # Directorio base de la infraestructura de la CA
certs          = $dir/certs          # Where the issued certs are kept
crl_dir        = $dir/crl            # Where the issued crl are kept
database       = $dir/index.txt      # database index file.
new_certs_dir  = $dir/newcerts               # default place for new certs.

certificate    = $dir/private/cacert.pem     # The CA certificate "acimp" en formato PEM
serial         = $dir/serial                 # The current serial number
crl            = $dir/crl.pem                      # The current CRL
private_key    = $dir/private/cakey.pem      # The private key encriptado
RANDFILE       = $dir/private/.rand          # private random number file

x509_extensions       = v3_ca        # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt       = ca_default          # Subject Name options
cert_opt       = ca_default          # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions      = crl_ext

default_days   = 365                 # how long to certify for
default_crl_days= 30                 # how long before next CRL
default_md     = md5                 # which md to use.
preserve       = no                  # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy         = policy_match

# For the CA policy
[ policy_match ]
countryName            = match
stateOrProvinceName    = match
organizationName       = match
organizationalUnitName= optional
commonName             = supplied
```

```
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName= optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca   # The extentions to add to the self signed cert

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req  # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = MX
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Distrito Federal

localityName                    = Locality Name (eg, city)
localityName_default            = Mexico

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = IMP

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default= Seguridad Informatica

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 64

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name
```

```
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment


[ v3_ca ]

# Extensions for a typical CA

# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
keyUsage = cRLSign, keyCertSign

# Some might want this also
nsCertType = sslCA, emailCA

nsComment                      = "Certificado Autoridad Certificadora AC-IMP"

# Include email address in subject alt name: another PKIX recommendation
subjectAltName=email:copy
# Copy issuer details
issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

Sección VI

Archivo de configuración de OpenSSL para Usuarios Finales

openssl.cnf en /usr/local/ssl

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
# Archivo de configuracion de openSSL por Carlos Roberto  Zainos Hdez Para IMP

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
```

```
RANDFILE              = /usr/local/ssl/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file             = $ENV::HOME/.oid
oid_section           = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions          =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca    = CA_default          # The default ca section

####################################################################
[ CA_default ]

dir           = /usr/local/ssl      # Directorio base de la infraestructura de la CA
certs         = $dir/certs          # Where the issued certs are kept
crl_dir       = $dir/crl            # Where the issued crl are kept
database      = $dir/index.txt      # database index file.
new_certs_dir = $dir/newcerts       # default place for new certs.

certificate   = $dir/private/cacert.pem     # The CA certificate "acimp" en formato PEM
serial        = $dir/serial                 # The current serial number
crl           = $dir/crl.pem                # The current CRL
private_key   = $dir/private/cakey.pem      # The private key encriptado
RANDFILE      = $dir/private/.rand          # private random number file

x509_extensions       = usr_cert    # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt      = ca_default          # Subject Name options
cert_opt      = ca_default          # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions      = crl_ext

default_days   = 365                # how long to certify for
default_crl_days= 30                # how long before next CRL
default_md     = md5                # which md to use.
preserve       = no                 # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy        = policy_match

# For the CA policy
[ policy_match ]
countryName           = match
stateOrProvinceName   = match
organizationName      = match
organizationalUnitName= optional
```

```
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName= optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca   # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret
# req_extensions = v3_req  # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = MX
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Distrito Federal

localityName                    = Locality Name (eg, city)
localityName_default            = Mexico

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = IMP

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default= Seguridad Informatica

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 64
# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.
```

```
basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment             = "Certificado Generado por openSSL CA-IMP"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl            = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

Sección VII

Archivo de configuración de OpenSSL para Servidor Web

ssl-web-server.cnf en /usr/local/ssl

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
# Archivo de configuracion de openSSL para servidor por Carlos Roberto  Zainos Hdez Para IMP

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = /usr/local/ssl/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids


[ new_oids ]
```

```
# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca    = CA_default          # The default ca section

####################################################################
[ CA_default ]

dir           = /usr/local/ssl      # Directorio base de la infraestructura de la CA
certs         = $dir/certs          # Where the issued certs are kept
crl_dir       = $dir/crl            # Where the issued crl are kept
database      = $dir/index.txt      # database index file.
new_certs_dir = $dir/newcerts               # default place for new certs.

certificate   = $dir/private/cacert.pem     # The CA certificate "acimp" en formato PEM
serial        = $dir/serial                 # The current serial number
crl           = $dir/crl.pem                      # The current CRL
private_key   = $dir/private/cakey.pem      # The private key encriptado
RANDFILE      = $dir/private/.rand          # private random number file

x509_extensions       = usr_cert     # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt      = ca_default          # Subject Name options
cert_opt      = ca_default          # Certificate field options

# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions      = crl_ext

default_days   = 365                # how long to certify for
default_crl_days= 30                # how long before next CRL
default_md     = md5                # which md to use.
preserve       = no                 # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy        = policy_match

# For the CA policy
[ policy_match ]
countryName            = match
stateOrProvinceName    = match
organizationName       = match
organizationalUnitName= optional
commonName             = supplied
emailAddress           = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName            = optional
stateOrProvinceName    = optional
localityName           = optional
organizationName       = optional
organizationalUnitName= optional
commonName             = supplied
emailAddress           = optional
```

```
#####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca   # The extentions to add to the self signed cert

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req  # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = MX
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Distrito Federal

localityName                    = Locality Name (eg, city)
localityName_default            = Mexico

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = IMP

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default= Seguridad Informatica

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_max                = 64

SET-ex3                 = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints = CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
```

```
nsCertType                      = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth

# This will be displayed in Netscape's comment listbox.
nsComment                       = "Servidor Web IMP"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always


# Copy subject details
issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

Anexo B-2

Script CA.pl en /usr/local/ssl/misc para la creación de una Autoridad Certificadora con OpenSSL. Utilizado sin modificaciones

```
#!/usr/bin/perl
#
# CA - wrapper around ca to make it easier to use ... basically ca requires
#      some setup stuff to be done before you can use it and this makes
#      things easier between now and when Eric is convinced to fix it :-)
#
# CA -newca ... will setup the right stuff
# CA -newreq[-nodes] ... will generate a certificate request
# CA -sign ... will sign the generated request and output
#
# At the end of that grab newreq.pem and newcert.pem (one has the key
# and the other the certificate) and cat them together and that is what
# you want/need ... I'll make even this a little cleaner later.
#
#
# 12-Jan-96 tjh    Added more things ... including CA -signcert which
#                  converts a certificate to a request and then signs it.
# 10-Jan-96 eay    Fixed a few more bugs and added the SSLEAY_CONFIG
#                 environment variable so this can be driven from
#                 a script.
# 25-Jul-96 eay    Cleaned up filenames some more.
# 11-Jun-96 eay    Fixed a few filename missmatches.
# 03-May-96 eay    Modified to use 'ssleay cmd' instead of 'cmd'.
# 18-Apr-96 tjh    Original hacking
#
# Tim Hudson
# tjh@cryptsoft.com
#

# 27-Apr-98 snh    Translation into perl, fix existing CA bug.
#
#
# Steve Henson
# shenson@bigfoot.com

# default openssl.cnf file has setup as per the following
# demoCA ... where everything is stored

$SSLEAY_CONFIG=$ENV{"SSLEAY_CONFIG"};
$DAYS="-days 365";
$REQ="openssl req $SSLEAY_CONFIG";
$CA="openssl ca $SSLEAY_CONFIG";
$VERIFY="openssl verify";
$X509="openssl x509";
$PKCS12="openssl pkcs12";

$CATOP="./demoCA";
$CAKEY="cakey.pem";
$CACERT="cacert.pem";

$DIRMODE = 0777;

$RET = 0;

foreach (@ARGV) {
        if ( /^(-\?|-h|-help)$/ ) {
            print STDERR "usage: CA -newcert|-newreq|-newreq-nodes|-newca|-sign|-verify\n";
            exit 0;
        } elsif (/^-newcert$/) {
            # create a certificate
            system ("$REQ -new -x509 -keyout newreq.pem -out newreq.pem $DAYS");
            $RET=$?;
            print "Certificate (and private key) is in newreq.pem\n"
```

```
} elsif (/^-newreq$/) {
    # create a certificate request
    system ("$REQ -new -keyout newreq.pem -out newreq.pem $DAYS");
    $RET=$?;
    print "Request (and private key) is in newreq.pem\n";
} elsif (/^-newreq-nodes$/) {
    # create a certificate request
    system ("$REQ -new -nodes -keyout newreq.pem -out newreq.pem $DAYS");
    $RET=$?;
    print "Request (and private key) is in newreq.pem\n";
} elsif (/^-newca$/) {
        # if explicitly asked for or it doesn't exist then setup the
        # directory structure that Eric likes to manage things
    $NEW="1";
    if ( "$NEW" || ! -f "${CATOP}/serial" ) {
        # create the directory hierarchy
        mkdir $CATOP, $DIRMODE;
        mkdir "${CATOP}/certs", $DIRMODE;
        mkdir "${CATOP}/crl", $DIRMODE ;
        mkdir "${CATOP}/newcerts", $DIRMODE;
        mkdir "${CATOP}/private", $DIRMODE;
        open OUT, ">${CATOP}/serial";
        print OUT "01\n";
        close OUT;
        open OUT, ">${CATOP}/index.txt";
        close OUT;
    }
    if ( ! -f "${CATOP}/private/$CAKEY" ) {
        print "CA certificate filename (or enter to create)\n";
        $FILE = <STDIN>;

        chop $FILE;

        # ask user for existing CA certificate
        if ($FILE) {
            cp_pem($FILE,"${CATOP}/private/$CAKEY", "PRIVATE");
            cp_pem($FILE,"${CATOP}/$CACERT", "CERTIFICATE");
            $RET=$?;
        } else {
            print "Making CA certificate ...\n";
            system ("$REQ -new -x509 -keyout " .
                "${CATOP}/private/$CAKEY -out ${CATOP}/$CACERT $DAYS");
            $RET=$?;
        }
    }
} elsif (/^-pkcs12$/) {
    my $cname = $ARGV[1];
    $cname = "My Certificate" unless defined $cname;
    system ("$PKCS12 -in newcert.pem -inkey newreq.pem " .
              "-certfile ${CATOP}/$CACERT -out newcert.p12 " .
              "-export -name \"$cname\"");
    $RET=$?;
    exit $RET;
} elsif (/^-xsign$/) {
    system ("$CA -policy policy_anything -infiles newreq.pem");
    $RET=$?;
} elsif (/^(-sign|-signreq)$/) {
    system ("$CA -policy policy_anything -out newcert.pem " .
                                            "-infiles newreq.pem");
    $RET=$?;
    print "Signed certificate is in newcert.pem\n";
} elsif (/^(-signCA)$/) {
    system ("$CA -policy policy_anything -out newcert.pem " .
                              "-extensions v3_ca -infiles newreq.pem");
    $RET=$?;
    print "Signed CA certificate is in newcert.pem\n";
} elsif (/^-signcert$/) {
    system ("$X509 -x509toreq -in newreq.pem -signkey newreq.pem " .
                                                  "-out tmp.pem");
    system ("$CA -policy policy_anything -out newcert.pem " .
```

```
                                                    "-infiles tmp.pem");
            $RET = $?;
            print "Signed certificate is in newcert.pem\n";
        } elsif (/^-verify$/) {
            if (shift) {
                foreach $j (@ARGV) {
                    system ("$VERIFY -CAfile $CATOP/$CACERT $j");
                    $RET=$? if ($? != 0);
                }
                exit $RET;
            } else {
                    system ("$VERIFY -CAfile $CATOP/$CACERT newcert.pem");
                    $RET=$?;
                    exit 0;
            }
        } else {
            print STDERR "Unknown arg $_\n";
            print STDERR "usage: CA -newcert|-newreq|-newreq-nodes|-newca|-sign|-verify\n";
            exit 1;
        }
}

exit $RET;

sub cp_pem {
my ($infile, $outfile, $bound) = @_;
open IN, $infile;
open OUT, ">$outfile";
my $flag = 0;
while (<IN>) {
        $flag = 1 if (/^-----BEGIN.*$bound/) ;
        print OUT $_ if ($flag);
        if (/^-----END.*$bound/) {
                close IN;
                close OUT;
                return;
        }
}
}
```

ANEXO B-3

SECCIÓN I

Instalación y configuración del servidor web Apache.

Descargamos el paquete httpd-2.0.44.tar.gz y lo descomprimimos:

$ gzip –d httpd-2.0.44.tar.gz
$ tar –xvf httpd-2.0.44.tar

Lo anterior crea un directorio llamado httpd-2.0.44 en la ubicación actual con varios subdirectorios. Nos movemos a este directorio:

$ cd httpd-2.0.44

Es importante destacar que para los fines que requerimos necesitamos compilar e instalar el modulo de ssl incluido en apache (mod_ssl). En el caso de que ya se haya compilado e instalado Apache, será necesario re-compilar el servidor e incluir este módulo. Para realizar esto, dentro del directorio httpd-2.0.44 tecleamos lo siguiente:

Para configurar:
$ ./configure –enable-ssl

Para compilarlo en nuestro sistema:
$ make

para instalarlo:
$ make install

En este momento, y si no recibimos mensajes de error, el servidor web apache debió quedar instalado en nuestro sistema ,con el módulo de ssl incluido, el directorio /usr/local/apache2.

Para probarlo, desde el directorio /usr/local/apache2/bin tecleamos:
$ ./apachectl start
Abrimos un navegador web y en la URL teclear: http:localhost
Apache debe desplegar una página informativa de éxito en la instalación del servidor web.
Para parar el servicio :
$ ./apachectl stop

Refiérase a la documentación de Apache en caso de encontrar problemas en esta fase o para referencias detalladas de los módulos del mismo.

SECCIÓN II

Obtención de un certificado para servidor web

A continuación detallamos los pasos para realizar dicho procedimiento:

El primer paso que debemos dar es crear una solicitud de certificado. Crearemos nuestra llave privada y la petición. Normalmente, queremos que la llave este tan segura como sea posible, por lo cual la cifraremos con una contraseña. Tendremos que escribir esta contraseña cada que iniciemos nuestro servidor Web, lo cual puede ser bastante molesto, por lo que queda como opción el no encriptarla (por cuestiones de seguridad esto no se recomienda), para lo cual le agregaríamos la opción -nodes al final de la línea de comando.

$ /usr/local/ssl/bin/openssl req -new -keyout apachekey.pem -out solicitudapache.pem -days 365 -config /usr/local/ssl/ssl-web-server.cnf

La línea anterior indica lo siguiente: Utilizando el motor de openssl (openssl) crea un nuevo requerimiento de certificado (req -new) y una clave privada llamada llavea.pem (-keyout), la salida completa de lo anterior (-out) ponla en el archivo solicituda.pem, la validez de la solicitud será de 365 dias (-days 365) y para ello utiliza el siguiente archivo de configuración (-config /usr/local/ssl/ssl-web-server.cnf). Ver sección VII Anexo B-1

Esto produce la siguiente salida en la terminal de trabajo:

```
Using configuration from /usr/local/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'solicituda.pem'                    # Nombre del archivo de salida de la solicitud
Enter PEM pass phrase: httpserver                              # Contraseña para la clave privada
Verifying password - Enter PEM pass phrase: httpserver         # Confirmar contraseña
—
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
—
Country Name (2 letter code) [MX]:MX                           # Introduzco los datos  del servidor
State or Province Name (full name) [Distrito Federal]:Distrito Federal
Locality Name (eg, city) [Mexico]:Mexico
Organization Name (eg, company) [IMP]:IMP
Organizational Unit Name (eg, section) [Seguridad Informática]:Seguridad Informatica
Common Name (eg, YOUR name) []:raguel.imp.mx
Email Address []:root@raguel.imp.mx

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: otroserver
An optional company name []: UNAM
```

Tenemos ya los dos archivos necesarios, una solicitud de certificado llamada solicitudapache.pem y apachekey.pem. En solicitudapache.pem está la solicitud de certificado, y en apachekey.pem nuestra llave privada. Enviamos la solicitud a la autoridad certificadora para que la firme.
Para firmar una solicitud, tecleamos lo siguiente :

$ /usr/local/ssl/bin/openssl ca -policy policy_anything -out apachecert.pem –config /usr/local/ssl/openssl.cnf -infiles solicitudapache.pem

La línea anterior de comandos indica lo siguiente: Utilizando el motor de openssl, actúa como ca (openssl ca), aplicando para ello la política de policy_anything (-policy policy_anything) que se encuentra definida en /usr/local/ssl/openssl.cnf (-config) y genera un certificado llamado certificadoa.pem (-out certificadoa.pem), para esto utiliza como entrada la solicitud llamada solicitudapache.pem (-infiles).

La autoridad certificadora hará:

```
Using configuration from /usr/local/ssl/openssl.cnf
Enter PEM pass phrase: caimp0                    # La contraseña que protege a la clave privada de la AC
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows country    # Datos del certificado
Name :PRINTABLE:'MX'
stateOrProvinceName :PRINTABLE:'Distrito Federal'
localityName :PRINTABLE:'Mexico'
organizationName :PRINTABLE:'IMP'
organizationalUnitName:PRINTABLE:'Seguridad Informática'
commonName :PRINTABLE:'raguel.imp.mx'
emailAddress :IA5STRING:'root@raguel.imp.mx'
Certificate is to be certified until May 23 14:41:40 2004 GMT (365 days)
Sign the certificate? [y/n]:y                    # Pide confirmación para firmar el certificado, decimos que si

1 out of 1 certificate requests certified, commit? [y/n]y    # Indica: 1 salida de 1 solicitud, entregar? Indicar que si
Write out database with 1 new entries            # Actualiza la base de datos
Data Base Updated
```

Tenemos entonces por fin un archivo llamado apachecert.pem, conteniendo aquello por lo que tanto nos hemos esforzado: Un certificado firmado por una autoridad certificadora. Ponemos ahora este archivo donde lo requiera nuestro servidor de Web, así como la llave privada apachekey.pem, y todo listo. Claro, no podemos dejar de recalcar la importancia de vigilar la seguridad de nuestra llave privada, pues si cae en manos ajenas, podrá suplantar nuestra identidad sin ningún problema, por lo que la podemos proteger dando permiso de lectura solo para root:

```
$ chmod 500 ssl.key
$ chmod 400 ssl.key/apachekey.pem
```

ssl.key es un directorio creado dentro de usr/local/apache2 el cual contiene tanto el certificado como la clave privada del servidor web seguro.

Los archivos de configuración utilizados en esta sección se encuentran en el Anexo B-1 secciones VI y VII, refiérase a ellos para mayor información.

SECCIÓN III

Configuración del archivo ssl.conf

El archivo ssl.conf contiene todos los parámetros necesarios para que Apeche pueda establecer sesiones utilizando el protocolo SSL/TLS. En este archivo, entre otras cosas, se indica la ubicación de la clave privada y del certificado; igualmente los protocolos que puede soportar y algunos otros detalles importantes. Refiérase a la documentación de mod_ssl para mayor información.

ssl.conf se procesa al momento de levantar el servidor web apache si es que así está indicado en httpd.conf. (Ver Anexo B-4)

Lo primero es indicarle al servidor por que puerto atenderá dichas peticiones. Por defecto, y es lo que utilizaremos, el servidor atiende peticiones por el puerto 443.

```
Listen 443
```

Al igual que para el servidor web, se puede establecer una dirección IP específica y un puerto, para nosotros y a manera de ejemplo esto sería:

```
Listen 192.168.144.102:443
```

A continuación le indicamos al servidor un host virtual el cual será el encargado de atendernos, esto debido a que el servidor puede atender peticiones tanto por el protocolo http como por el https.

Esto aparece de la siguiente manera:

```
## SSL Virtual Host Context
<VirtualHost _default_:443>
DocumentRoot "usr/local/apache2/htdocs"
ServerName raguel.imp.mx:443
#ServerName 192.168.144.102:443
ServerAdmin root@raguel.imp.mx
```

A continuación localizamos y modificamos la línea donde se indica el archivo que contiene el certificado SSL, esto queda de la siguiente manera:

SSLCertificateFile /usr/local/apache2/conf/ssl.crt/apachecert.pem

De la misma manera indicamos donde encontrar la clave privada:

SSLCertificateKeyFile /usr/local/apache2/ssl.crt/apachekey.pem

Ahora la cadena de certificados:

SSLCertificateChainFile /usr/local/apache2/ssl.crt/cachain.pem

Ahora realizamos lo mismo pero para la AC:

SSLCACertificatePath /usr/local/ssl/CAServer
SSLCACertificateFile /usr/local/ssl/CAServercacert.pem

Ahora el directorio y la lista de certificados revocados, por el momento conviene comentar estas líneas ya que no contamos con crl alguna pero en un futuro la tendremos:

SSLCARevocationPath /usr/local/ssl/crl/

SSLCARevocationFile /usr/local/ssl/crl/cacrl1.crl

Finalmente le indicamos que "sabores" de protocolo puede utilizar cuando establece su ambiente de servidor. Los clientes entonces pueden conectarse con alguno de los protocolos provistos.

SSLProtocol all

Donde "all" quiere decir: SSLv2, SSLv3 y TLSv1

Es conveniente recordar que ssl.conf está estrechamente ligado con httpd.conf. En el Anexo B-4 se encuentran los archivos completos httpd.conf y ssl.conf.

Anexo B-4

Archivos de configuración del servidor Apache y del módulo mod_ssl con ssl.conf
Nota: Se omitieron los comentarios

http.conf en /usr/local/apache2/conf :

```
# Archivo de configuracion del servidor apache al 23/06/03 por ZAINOS FINAL

# This is the main Apache server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs-2.0/> for detailed information about
# the directives.

### Section 1: Global Environment
#
ServerRoot "/usr/local/apache2"

# The accept serialization lock file MUST BE STORED ON A LOCAL DISK.
#
<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#LockFile logs/accept.lock
</IfModule>
</IfModule>
#
<IfModule !mpm_netware.c>
<IfModule !perchild.c>
#ScoreBoardFile logs/apache_runtime_status
</IfModule>
</IfModule>
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
#
<IfModule !mpm_netware.c>
PidFile logs/httpd.pid
</IfModule>
#
# Timeout: The number of seconds before receives and sends time out.
Timeout 300
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
KeepAlive On

# We recommend you leave this number high, for maximum performance.
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
KeepAliveTimeout 15

# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule prefork.c>
StartServers         5
MinSpareServers      5
MaxSpareServers     10
MaxClients         150
MaxRequestsPerChild  0
</IfModule>

# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule worker.c>
StartServers         2
MaxClients         150
MinSpareThreads     25
```

```
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>

# MaxRequestsPerChild: maximum number of connections per server process
<IfModule perchild.c>
NumServers           5
StartThreads         5
MinSpareThreads      5
MaxSpareThreads     10
MaxThreadsPerChild  20
MaxRequestsPerChild  0
</IfModule>

# WinNT MPM
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum  number of requests a server process serves
<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild  0
</IfModule>

<IfModule beos.c>
StartThreads                10
MaxClients                  50
MaxRequestsPerThread        10000
</IfModule>

<IfModule mpm_netware.c>
ThreadStackSize      65536
StartThreads           250
MinSpareThreads         25
MaxSpareThreads        250
MaxThreads            1000
MaxRequestsPerChild      0
</IfModule>

<IfModule mpmt_os2.c>
StartServers           2
MinSpareThreads        5
MaxSpareThreads       10
MaxRequestsPerChild    0
</IfModule>
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)

Listen 80
Listen 88
Listen 443

### Section 2: 'Main' server configuration

<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#
User nobody
#Group #-1
Group nobody
</IfModule>
</IfModule>
ServerAdmin utirado@imp.mx
#
#ServerName new.host.name:80
#
```

```
UseCanonicalName Off
#
DocumentRoot "/usr/local/apache2/htdocs"
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
#
<Directory "/usr/local/apache2/htdocs">
#
    Options Indexes FollowSymLinks
#
    AllowOverride None
#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all

</Directory>
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
UserDir public_html
#
# Control access to UserDir directories.  The following is an example
# for a site where these directories are restricted to read-only.
#
DirectoryIndex index.html index.html.var
#
# AccessFileName: The name of the file to look for in each directory
# for additional configuration directives.  See also the AllowOverride
# directive.
#
AccessFileName .htaccess
#
# The following lines prevent .htaccess and .htpasswd files from being
# viewed by Web clients.
#
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
#
# TypesConfig describes where the mime.types file (or equivalent) is
# to be found.
#
TypesConfig conf/mime.types
#
DefaultType text/plain
#
# The mod_mime_magic module allows the server to use various hints from the
# contents of the file itself to determine its type.  The MIMEMagicFile
# directive tells the module where the hint definitions are located.
#
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
#
HostnameLookups Off
#
ErrorLog logs/error_log
#
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel warn
```

```
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
#
CustomLog logs/access_log common
#
ServerTokens Full
#
ServerSignature On
#
Alias /icons/ "/usr/local/apache2/icons/"

<Directory "/usr/local/apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
#
Alias /manual "/usr/local/apache2/manual"

<Directory "/usr/local/apache2/manual">
    Options Indexes FollowSymLinks MultiViews IncludesNoExec
    AddOutputFilter Includes html
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
#
ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"

<IfModule mod_cgid.c>
#
# Additional to mod_cgid.c settings, mod_cgid has Scriptsock <path>
# for setting UNIX socket for communicating with cgid.
#
#Scriptsock            logs/cgisock
</IfModule>
#
# "/usr/local/apache2/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "/usr/local/apache2/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
#
IndexOptions FancyIndexing VersionSort
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
#
# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
#
DefaultIcon /icons/unknown.gif

ReadmeName README.html
HeaderName HEADER.html
# IndexIgnore is a set of filenames which directory indexing should ignore
```

```
# and not include in the listing.  Shell-style wildcarding is permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+) uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
#
# LanguagePriority allows you to give precedence to some languages
#
LanguagePriority en da nl et fr de el it ja ko no pl pt pt-br ltz ca es sv tw
#
# ForceLanguagePriority allows you to serve a result page rather than
# MULTIPLE CHOICES (Prefer) [in case of a tie] or NOT ACCEPTABLE (Fallback)
# [in case no accepted languages matched the available variants]
#
ForceLanguagePriority Prefer Fallback
#
AddDefaultCharset ISO-8859-1
#
# Commonly used filename extensions to character sets. You probably
# want to avoid clashes with the language extensions, unless you
# are good at carefully testing your setup after each change.
# See http://www.iana.org/assignments/character-sets for the
# official list of charset names and their respective RFCs.
#
AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset ISO-8859-2  .iso8859-2  .latin2 .cen
AddCharset ISO-8859-3  .iso8859-3  .latin3
AddCharset ISO-8859-4  .iso8859-4  .latin4
AddCharset ISO-8859-5  .iso8859-5  .latin5 .cyr .iso-ru
AddCharset ISO-8859-6  .iso8859-6  .latin6 .arb
AddCharset ISO-8859-7  .iso8859-7  .latin7 .grk
AddCharset ISO-8859-8  .iso8859-8  .latin8 .heb
AddCharset ISO-8859-9  .iso8859-9  .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5        .Big5       .big5
AddCharset WINDOWS-1251 .cp-1251   .win-1251
AddCharset CP866       .cp866
AddCharset KOI8-r      .koi8-r .koi8-ru
AddCharset KOI8-ru     .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8       .utf8
AddCharset GB2312      .gb2312 .gb
AddCharset utf-7       .utf7
AddCharset utf-8       .utf8
AddCharset big5        .big5 .b5
AddCharset EUC-TW      .euc-tw
AddCharset EUC-JP      .euc-jp
AddCharset EUC-KR      .euc-kr
AddCharset shift_jis   .sjis
#
# AddType allows you to add to or override the MIME configuration
# file mime.types for specific file types.
#
AddType application/x-tar .tgz
AddType image/x-icon .ico

AddHandler type-map var

# Lo  siguiente respecto a ifmodule yo lo puse y descomente las 30  lineas siguientes

<IfModule mod_negociation.c>
```

```
<IfModule mod_include.c>

    Alias /error/ "/usr/local/apache2/error/"

    <Directory "/usr/local/apache2/error">
        AllowOverride None
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en cs de es fr it nl sv pt-br ro
        ForceLanguagePriority Prefer Fallback
    </Directory>

    ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
    ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
    ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
    ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
    ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
    ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
    ErrorDocument 410 /error/HTTP_GONE.html.var
    ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
    ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
    ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
    ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
    ErrorDocument 415 /error/HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
    ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
    ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
    ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
    ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
    ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var

</IfModule>
</IfModule>

# The following directives modify normal HTTP response behavior to
# handle known problems with browser implementations.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
#
# The following directive disables redirects on non-GET requests for
# a directory that does not include the trailing slash.  This fixes a
# problem with Microsoft WebFolders which does not appropriately handle
# redirects for folders with DAV methods.
# Same deal with Apple's DAV filesystem and Gnome VFS support for DAV.
#
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[012]" redirect-carefully
BrowserMatch "^gnome-vfs" redirect-carefully

# Aqui está el archivo que me interesa que se procese al levantar el servicio
#
<IfModule mod_ssl.c>
    Include conf/ssl.conf
</IfModule>

# Incluir el archivo de configuracion de la CA y de  la RA para OpenCA

Include /srv/ca/apache.conf
Include /hdc1/ra/apachera.conf

########################################################################################
```

Archivo de configuración ssl.conf

Nota: Se omitieron los comentarios

ssl.conf en /usr/local/apache2/conf :

```
# Archivo de configuracion  actualizado al 18/08 por ZAINOS para IMP FINAL si sirve
#
# This is the Apache server configuration file providing SSL support.
# It contains the configuration directives to instruct the server how to
# serve pages over an https connection. For detailing information about these
# directives see <URL:http://httpd.apache.org/docs-2.0/mod/mod_ssl.html>
#
<IfDefine SSL>
#
Listen 443

#  SSL Global Context
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#   terminal dialog) has to provide the pass phrase on stdout.
SSLPassPhraseDialog  builtin

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).

#SSLSessionCache         none
#SSLSessionCache         shmht:logs/ssl_scache(512000)
#SSLSessionCache         shmcb:logs/ssl_scache(512000)
SSLSessionCache         dbm:logs/ssl_scache
SSLSessionCacheTimeout  300

#   Semaphore:
#   Configure the path to the mutual exclusion semaphore the
#   SSL engine uses internally for inter-process synchronization.
SSLMutex  file:logs/ssl_mutex

#   Pseudo Random Number Generator (PRNG):
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
##
## SSL Virtual Host Context
##
<VirtualHost _default_:443>

#  General setup for the virtual host
DocumentRoot "/usr/local/apache2/htdocs"
#ServerName new.host.name:443
ServerName 192.168.144.102:443
#ServerAdmin you@your.address
ServerAdmin root@raguel.imp.mx
ErrorLog logs/error_log
TransferLog logs/access_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.
SSLEngine on

#   SSL Cipher Suite:
#   List the ciphers that the client is permitted to negotiate.
#   See the mod_ssl documentation for a complete list.
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

#   Server Certificate:
```

```
#SSLCertificateFile /usr/local/apache2/conf/ssl/apachecert.pem
SSLCertificateFile /usr/local/apache2/conf/ssl/apachesslcert.pem

#SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server-dsa.crt

#   Server Private Key:

#SSLCertificateKeyFile /usr/local/apache2/conf/ssl/apachekey.pem
SSLCertificateKeyFile /usr/local/apache2/conf/ssl/apachessl.pem

#SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server-dsa.key

SSLCertificateChainFile /usr/local/ssl/private/acimp.pem
#SSLCertificateChainFile /usr/local/apache2/conf/ssl.crt/ca.crt

#   Certificate Authority (CA):
#SSLCACertificatePath /usr/local/apache2/conf/ssl.crt
#SSLCACertificateFile /usr/local/apache2/conf/ssl.crt/ca-bundle.crt

SSLCACertificatePath /usr/local/ssl/certs
SSLCACertificateFile /usr/local/ssl/private/acimp.pem

#   Certificate Revocation Lists (CRL):
#SSLCARevocationPath /usr/local/apache2/conf/ssl.crl
#SSLCARevocationFile /usr/local/apache2/conf/ssl.crl/ca-bundle.crl

SSLCARevocationPath /usr/local/ssl/crl
SSLCARevocationFile /usr/local/ssl/crl/crl.pem

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
    SSLOptions +StdEnvVars
</Files>
<Directory "/usr/local/apache2/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:

SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

CustomLog logs/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

# SSL Protocol Directive
# Esta directiva puede ser utilizada para controlar los diferentees "sabores"
# del protocolo SSL que mod_ssl puede  utilizar cuando establece su ambiente
# de servidor. Los clientes entonces  pueden solo conectarse con alguno de
# los protocolos provistos.

# SSL protocol  "opcioon"
# donde  "opcion" puede ser:
# SSLv2, SSLv3, TLSv1, All
SSLProtocol all

</VirtualHost>

</IfDefine>
```

ANEXO C-1

SECCIÓN I

Descarga, Instalación y Configuración de OpenLDAP

En esta parte nos dedicaremos a detallar el procedimiento que se siguió para el caso específico de PKI-IMP.

OBTENIENDO OPENLDAP

Se puede descargar la última versión de OpenLDAP siguiendo las instrucciones de la página de descarga de OpenLDAP en http://www.openldap.org/software/download/
En nuestro caso, PKI-IMP utiliza OpenLDAP-2.1.17

DESEPAQUETANDO EL SOFTWARE

Es conveniente crear un directorio en el cual se desee extraer la distribución, esto con el objeto de tener un mejor control en la administración de los archivos. Cree el directorio en cuestión y ahí descomprima el software. PKI-IMP utilizó /hdc1/openldap-2.1.17

Se recomienda ampliamente revisar la documentación en el caso de que se deseen ajustar parámetros de configuración así como características adicionales:

PRE-REQUISITOS DE SOFTWARE

OpenLDAP para su instalación y funcionamiento, necesita de varias herramientas de software proporcionadas por terceras partes. Dependiendo de las características deseadas en la instalación serán los requisitos de software que se deban cubrir. Si así se requiere, se deberán descargar e instalar estas herramientas para poder completar la instalación. En nuestro caso, la instalación se realiza sin características especiales más que las que por defecto debe tener. Esto sin embargo implicó la descarga e instalación de varios paquetes de software.

Seguridad en la Capa de Transporte TLS

Los clientes y servidores OpenLDAP requieren la instalación de las librerías de OpenSSL-TLS para proporcionar servicios de seguridad utilizando protocolos SSL y TLS. Esto no representa un problema ya que previamente hemos configurado e instalado OpenSSL. Si no se cuenta con este conjunto de librerías vale la pena recordar que se encuentran disponibles en http://www.openssl.org/

Servicios de Autenticación Kerberos

Los clientes y servidores de OpenLDAP soportan los servicios de autenticación basados en Kerberos. Esto es opcional pero ampliamente recomendado. La mayoría de los sistemas actuales incluyen un conjunto de librerías para soportar este tipo de autenticación y no habrá por que preocuparse, en caso contrario se puede obtener una copia de la última distribución de Kerberos en :
http://web.mit.edu/kerberos/www/.

Simple Authentication and Security Layer, SASL

Los clientes y servidores de OpenLDAP requieren la instalación de las librerías de **Cyrus SASL**[1] para proporcionar servicios de capa simple de seguridad y autenticación, SASL.. AL igual que para el caso

[1] Se puede obtener una distribución de este en http://asg.web.cmu.edu/sasl/sasl-library.html

de Kerberos, la mayoría de los sistemas actuales cuentan con este conjunto de librerías, sin embargo en ocasiones no es así o simplemente se encuentran obsoletas.
Cyrus SASL se encuentra disponible para su descarga en: http://asg.web.cmu.edu/sasl/sasl-library.html.
La documentación relacionada con el mismo se puede obtener igualmente en el mismo sitio.

Software de la Base de Datos

El backend primario de slapd - OpenLDAP es BDB, por lo tanto en necesario contar con el software correspondiente: Sleepycat Software BerkeleyDB en su versión 4. Si no se encuentra disponible en el sistema, el script de configuración no podrá preconfigurar el software y prepararlo para su instalación.
La última versión de BerkeleyDB puede ser descargada de http://www.sleepycat.com/download.html
Se recomienda la versión 4.1. En nuestro caso se descargó e instaló la versión 4.1.25

Para saber más, refiérase a la documentación incluida en la distribución o directamente en el sitio de Sleepycat (http://www.sleepycat.com).

CONFIGURACIÓN E INSTALACIÓN

Lo siguiente es configurar el software de manera que se pueda instalar en nuestro sistema. Esto es muy sencillo ya que la distribución incluye un script de auto configuración, basta con ejecutar este script y listo.

Por defecto, el software OpenLDAP se instala en /usr/local. Aunque se le puede especificar cualquier otra ubicación mediante el uso de un prefijo.

En el caso de encontrar algún error, refiérase a la documentación de OpenLDAP, esta se encuentra disponible en: http://www.openldap.org/doc/admin21/

Configuración

Lo primero que hay que hacer es configurar el software, esto se realiza de la siguiente manera:

```
/openldap-2.1.17]# ./configure
```

Si desea ver las opciones de configuración debe teclear lo siguiente:

```
/openldap-2.1.17]# ./configure --help
```

En nuestro caso específico, al momento de configurar el software encontramos algunos problemas, la mayoría ocasionados por la falta del cumplimiento de los requisitos de software. Por el momento simplemente asumiremos que la configuración pasó sin mayores complicaciones.

Compilación

Lo siguiente es instalar el software, para esto previamente hay que construir las dependencias del mismo. Al terminar de ejecutar el script de configuración, la última línea de salida debe ser la siguiente:

```
Please "make depend" to build dependencies
```

Para construir las dependencias se teclea lo siguiente:

```
Openldap-2.1.17]# make depend
```

Ahora "construimos" el software, este paso compilará OpenLDAP :

Openldap-2.1.17]# make

En este momento de debe prestar especial atención a las salidas que arroja el comando para asegurarse de que todo se compile correctamente. Este comando compila y crea las librerías de LDAP y los clientes asociados, así como a slapd.

Probar el Software

Una vez que el software ha sido configurado adecuadamente y se ha compilado con éxito, se debe correr el conjunto de pruebas para verificar la instalación de los fuentes. Esto se realiza tecleando :

openldap-2.1.17]# make test

Las pruebas respectivas a la configuración realizada comenzarán, y en principio deben pasar con éxito.

En nuestro caso, tuvimos bastantes complicaciones en esta parte ya que tanto en la configuración como en las pruebas había elementos no soportados por nuestro sistema, por lo se tuvo que leer detalladamente la documentación tanto de OpenLDAP como del software asociado.

Instalación

Finalmente, y luego de que todas las pruebas fueron pasadas con éxito, nos encontramos listos para instalar el software. Para esto debemos contar con permiso de escritura, nos logeamos entonces como *root* y solo resta teclear lo siguiente:

Openldap-2.1.17]# make install

Ahora es muy importante examinar cuidadosamente la salida que este comando arroja, para asegurarnos de que todo se instale correctamente. Los archivos de configuración para slapd se pueden encontrar en /usr/local/etc/openldap.

Respecto a esta sección esto es todo lo que diremos. Si se desea mayor información o referencias consultar el manual incluido en la distribución o consultar el que se encuentra en la sección de Documentación en la página principal del Proyecto (Ver referencias al final de la sección 3.3)

SECCIÓN II

Configuración del servidor

Lo que a continuación hacemos es configurar el servidor. Para esto es necesario modificar, de acuerdo a nuestras necesidades, los archivos de configuración de slapd (slapd.conf) y ldap.conf que se encuentran en /usr/local/etc/openldap.

Configuración del Servidor LDAP

La manera en cómo OpenLDAP trabajará se encuentra basada principalmente en dos archivos de configuración ldap.conf y slapd.conf. Es necesario realizar algunas modificaciones a estos archivos con la finalidad de que el servicio funcione como lo deseamos.

Para slap.conf, en la sección de ldbm database definition :

database – esto define el tipo de base de datos que vamos a utilizar. En nuestro caso tiene el siguiente valor:
database bdb (utilizamos Berkeley Data Base)

suffix – Esta directiva especifica el sufijo, o raíz del directorio, de nombre distinguido (DN) de las peticiones o "queries" que serán pasadas o enviadas a esta base de datos (backend) y atendidas por el servidor. Se pueden dar múltiples líneas de sufijo, y al menos una es requerida para cada definición de base de datos.
En nuestro caso tiene el siguiente valor:

suffix "o=IMP, c=MX"

directory - Aquí es donde se guardará el dbs de ldap, para nosotros tiene el siguiente valor:

directory /usr/local/var/openldap-data

rootdn – Esta directiva especifica el nombre distinguido (DN) que no está sujeto a ningún control de acceso (algo así como el administrador) o restricciones limitadas de administración para operaciones en esta base de datos. En nuestro servidor es el siguiente:

rootdn "cn=Manager,o=IMP,c=MX"

rootpw – Esta directiva puede ser utilizada para especificar un password para el DN del rootdn (cuando rootdn es colocado con un DN dentro de la base de datos).

En nuestro caso tiene el siguiente valor:
rootpw secretimp

Ahora editamos el archivo ldap.conf

Este archivo pertenece al cliente de LDAP, en este caso nosotros utilizamos la misma máquina como cliente y servidor. Esto es posible, pero pueden ser dos máquinas diferentes.

Lo siguiente define el servidor ldap. Se puede usar un hostname o la dirección IP.
host 192.168.144.102: 389

En principio estos son los principales cambios que hay que realizar al archivo de configuración, ahora solo queda guardar los cambios e iniciar el servicio.

Para esto levantamos el servicio de LDAP con la siguiente instrucción:

```
usr/local/libexec]# ./slapd
```

Para detener el servicio tecleamos lo siguiente:

```
]# kill –INT 'cat usr/local/var/slapd.pid'
```

Los dos procedimientos anteriores no deben causar mayor problema.

Si se requiere mayor referencia respecto a las opciones de configuración, dudas o problemas relacionados con la misma refiérase a la documentación de OpenLDAP. Ver referencias al final de la sección 3.3.

ANEXO C-2

Base de Datos Berkeley DB de SleepyCat Software

La base de datos Berkeley DB es una herramienta desarrollada por SleepycatSoftware (www.sleepycat.com). BerkeleyDB es el software de administración de datos de aplicación específica líder en el mundo, con más de 200 millones de implementaciones. Esta herramienta corre en sistemas operativos UNIX/LINUX así como en sistemas WIN32.

Esta es una de las bases de datos con las que puede trabajar la herramienta OpenCA, y que usa por defecto. Esta herramienta debe estar instalada previamente antes de compilar e instalar OpenCA.

Descarga del software

Si no se cuenta con dicho software instalado, se puede descargar la última versión de Internet de la siguiente dirección:

http://www.sleepycat.com/download/index.shtml

El presente trabajo se encuentra hecho con la versión 4.1 corriendo sobre un RedHat 7.2.

Berkeley DB es una librería de Base de datos "incrustada" de código abierto(Open Source) que proporciona servicios de administración de datos escalables, de alto rendimiento, con transacciones protegidas a aplicaciones. Berkeley DB proporciona una API simple de llamada a función para accesar y administrar información.

Berkeley DB es "incrustado" porque esta se liga directamente dentro de la aplicación que la usa. Esta corre en el mismo espacio de dirección que la aplicación. Como resultado, no son requeridas comunicaciones interprocesos, sobre la red o entre procesos dentro de la misma máquina, para operaciones de base de datos. Berkeley DB proporciona una simple API de llamada a función para un amplio número de lenguajes, dentro de los que se incluyen C, C++, java, Perl, Tcl, Phyton y PHP.
Todas las operaciones de base de datos se llevan a cabo dentro de la librería.

Por "código abierto", queremos decir que Berkeley DB es distribuida bajo una licencia conforme a las definiciones de Open Source (www.opensource.org/osd.html) Esta licencia garantiza que Berkeley DB se encuentra disponible de manera gratuita para su uso y redistribución en otros productos Open Source. Sleepycat Software vende licencias comerciales para redistribución en aplicaciones propietarias, pero en todos los casos el código fuente completo de Berkeley DB se encuentra disponible de manera gratuita para ser descargado y utilizado.

Compilación e instalación del software

La compilación e instalación es muy sencilla. La distribución Berkeley DB se conforma de cuatro librerías separadas:

- la librería base en C
- API Berkeley DB (Interfaz de Programación de Aplicación Berkeley DB) y las librerías opcionales C++,
- Java y
- la API Tcl.

Por razones de portabilidad, cada librería es independiente y contiene el soporte completo de Berkeley DB necesario para construir aplicaciones; esto es, la librería C++ API Berkeley DB no requiere ninguna otra librería Berkeley DB para construir y correr aplicaciones C++.

La distribución de Berkeley DB utiliza las herramientas autoconf y libtool de la fundación de Software Libre para instalarse en plataformas UNIX/LINUX. En general, las opciones estándar de compilación e instalación para estas herramientas aplican para las distribuciones de Berkeley DB.

Para realizar una instalación estándar en UNIX, cámbiese al directorio build_unix dentro del directorio raíz de Berkeley DB y teclee los siguientes comandos:

```
]# ../dist/configure
]# make
```

Esto construirá la librería Berkeeley DB.

Para instalar la librería Berkeley DB, introduzca el siguiente comando:

```
]# make install
```

Esto es todo lo necesario para construir e instalar Berkeley DB.

Configuración de Berkeley DB

Hay muchos argumentos que se pueden especificar cuando se configura Berkeley DB. Sin embargo únicamente los argumentos específicos de Berkeley DB se encuentran descritos en la documentación, la mayoría de los argumentos estándares GNU autoconf se encuentran disponibles y soportados.

Si tiene dudas o problemas con alguno de los comandos anteriores, refiérase a la documentación incluida en la distribución.

Visite el sitio web arriba indicado para saber más de este software utilizado, una explicación detallada del mismo queda fuera de los alcances del presente trabajo.

ANEXO C-3

SECCIÓN I

CONFIGURACIÓN E INSTALACIÓN DE OPENCA PARA AC

Configuración

Lo que a continuación se indica se refiere a la configuración del software de OpenCA para crear una estructura de AC.

Desde la línea de comandos, dentro del directorio de desempaque de OpenCA:

```
]# ./configure    --prefix=/srv/ca                                    \
                    --with-apache=/usr/local/apache2                  \
                    --with-openssl-prefix=/usr/local/ssl              \
                    --with-web-host=raguel.imp.mx                     \
                    --with-httpd-user=nobody                          \
                    --with-httpd-group=nobody                         \
                    --with-dist-user=zainos                           \
                    --with-dist-group=openca                          \
                    --enable db                                       \
                    --enable-ocspd                                    \
                    --with-ca-organization=IMP                        \
                    --with-ca-locality=Mexico                         \
                    --with-ca-country=MX                              \
                    --with-service-mail-account=root@raguel.imp.mx    \
                    --with-mail-program="usr/sbin/sendmail –t"        \
                    --with-hierarchy-level=ca                         \
                    --with-db-type=berkeley                           \
                    --with-db-name=openca                             \
                    --with-db-host=localhost                          \
                    --with-db-port=3306                               \
                    --with-db-user=openca                                 ENTER
```

Lo anterior realiza la configuración de la instalación del software de acuerdo a nuestro sistema. Dentro de esta configuración podemos distinguir, en orden descendente:

Un prefijo de instalación, esto es, un directorio creado para que dentro de él sean colocados los directorios y archivos necesarios de la CA, esto con el objetivo de tener un mejor control y administración de los archivos así como de las versiones.
Dos directivas que indican la ubicación de los servicios tanto de apache como de OpenSSL.

Una directiva que indica el nombre de host por medio del cual se brindarán los servicios web.

Dos directivas que indican el propietario y el grupo al que pertenece el servicio de httpd (apache web server). Esto resultado de los privilegios de usuarios y de grupo para hacer uso de los servicios, así como de los respectivos permisos de lectura/escritura. Es necesario verificar en la configuración de Apache (httpd.conf) que esto concuerde, es decir, indicarle a Apache el usuario y el grupo al cual pertenece el servicio, esto debe aparece de la siguiente manera:

```
User  nobody
Group nobody
```

Dos directivas que indican el nombre del usuario y del grupo propietarios de la instalación de OpenCA.
Dos directivas que indican, una que se debe habilitar el servicio de base de datos (db), y la otra que indica que se debe habilitar el uso de protocolo de estado de certificado en línea, OCSPD.

Tres directivas que indican la configuración de la AC de acuerdo al estándar X.500, la cual incluye el identificador de la organización, de la localidad o región y del país (organization, locality, country)
Estos tres elementos conformarán la raíz del DN de todos los elementos dentro del árbol de directorio de la AC.

Dos directivas que indican, tanto la cuenta de correo que se utilizará, como el servicio de envío de mensajes de correo electrónico deseado para distribuir información vía e-mail.

Una directiva que es muy importante es especificar la jerarquía que se está configurando. En este caso es de CA (Certification Authority), lo que indica que esta infraestructura tendrá las capacidades de actuar como AC, y que dentro de la jerarquía ocupará el lugar más alto.
Mas adelante veremos que si modificamos este parámetro por el de una RA (Registration Authority) entonces las directivas de configuración y herramientas instaladas serán distintas.

Las últimas directivas se refieren a la base de datos que será habilitada y utilizada. En esta configuración los aspectos relacionados con esta son: el tipo de base de datos, el nombre la base, el servidor, el puerto, y el nombre del usuario.

Compilar e Instalar AC

Una vez que el proceso de configuración termine con éxito, lo siguiente es compilar e instalar el software. OpenCA utiliza método usual de instalación de los fuentes de OpenSource (*make* y *make install*). La pequeña variante que se utiliza en la instalación se encuentra en la fase propia de la instalación del software, la cual debe ser indicada de la siguiente manera:

```
]# make

]# make install-ca
```

Esta última instrucción es muy importante, ya que con esto le indicamos al proceso de instalación que lo que deseamos instalar es el software de la AC, el que previamente se haya configurado el software para dicho módulo no quiere decir que automáticamente se instale la AC, ya que existen otras opciones, que mencionaremos más adelante, que instalan componentes distintos.

ANEXO C-3

SECCIÓN II

Configuración del servidor apache para la AC

Es conveniente manejar la configuración y el servicio mediante un host virtual dentro de apache. Para esto es necesario realizar algunas modificaciones en el archivo de configuración httpd.conf y agregar lo siguiente:

En la sección donde indicamos al servidor el o los puertos que debe de "escuchar" (Listen) se agrega:

```
Listen 88
```

Con esto estamos indicando que el servicio web de la AC será atendido por el puerto 88.

Adicionalmente, al final de la sección 2 (Section 2:'Main' server configuration) debemos agregar la siguiente línea:

```
Include /srv/ca/apache.conf
```

Con esto indicamos que cuando se levante el servicio de apache y se procese el archivo de configuración de apache (http.conf) se incluya el de la AC, que se encuentra ubicado en el "path" indicado, en este caso es en /srv/ca y se llama apache.conf.
Una vez hecho esto, es necesario crear el archivo de configuración del servicio web de la AC. Para esto se crea el archivo apache.conf en cualquier editor y lo colocamos en la ubicación dada anteriormente.

```
Archivo de configuración apache.conf en /srv/ca/

# Configuracion del servidor  apache para la CA  por Carlos R Zainos H
# Ligado directamente en httpd.conf, trabaja  como  host virtual

<VirtualHost 192.168.144.102:88>
        ServerAdmin root@raguel.imp.mx
        DocumentRoot /srv/ca/apache/htdocs
        ServerName 192.168.144.102
        #ErrorLog logs/dummy-host.example.com-error_log
        SetenvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-
        response-1.0
        CustomLog /srv/log/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
        <Directory "/srv/ca/apache/htdocs">
                Options Indexes FollowSymlinks Multiviews
                AllowOverride None
                Order allow,deny
                Allow from 192.168.144
        </Directory>
        ScriptAlias /cgi-bin/  "/srv/ca/apache/cgi-bin/"
        <Directory "/srv/ca/apache/cgi-bin">
                AllowOverride  None
                Options None
                Order allow,deny
                Allow  from 192.168.144
        </Directory>
</VirtualHost>
```

La configuración anterior contiene información importante para la manera de trabajar con la AC. A grandes rasgos comentaremos los puntos más importantes, los detalles se pueden ver en la documentación de Apache.

El primer punto importante es la definición del host virtual :

```
<VirtualHost 192.168.144.102:88>
```

Con este encabezado estamos indicando que deseamos manejar un host virtual, el cual será atendido en la dirección IP y por el puerto especificado. Nosotros seleccionamos esta opción ya que deseamos accesar al servicio de la AC desde una ubicación remota (AC en línea). Si se elige por una configuración de AC no conectada (AC offline) entonces probablemente sea conveniente definir el host virtual como "localhost".

Después de esto encontramos directivas que indican entre otras cosas, el administrador del servidor, la ubicación de la "documentación", el nombre del servidor, la opciones con las que se levantará el servidor, directorios dentro del servidor, opciones y permisos de éstos, y algunos "alias" que se puedan manejar.

Para finalizar, el servidor web se "levanta mediante la siguiente instrucción:

```
/usr/local/apache2/bin]# ./apachectl start
```

ANEXO C-3

SECCIÓN III

CONFIGURACIÓN E INSTALACIÓN DE OPENCA PARA AR

Configuración

Lo que a continuación se indica se refiere a la configuración del software de OpenCA para crear una estructura de AR.

Desde la línea de comandos, dentro del directorio de desempaque de OpenCA:

```
]# ./configure    --prefix=/hdc1/ra                                    \
                    --with-apache=/usr/local/apache2                   \
                    --with-openssl-prefix=/usr/local/ssl               \
                    --with-web-host=raguel.imp.mx                      \
                    --with-httpd-user=nobody                           \
                    --with-httpd-group=nobody                          \
                    --with-dist-user=utirado                           \
                    --with-dist-group=openca                           \
                    --enable db                                        \
                    --enable-ocspd                                     \
                    --with-ca-organization=IMP                         \
                    --with-ca-locality=Mexico                          \
                    --with-ca-country=MX                                    \
                    --with-service-mail-account=root@raguel.imp.mx     \
                    --with-mail-program="usr/sbin/sendmail –t"              \
                    --with-hierarchy-level=ra                          \
                    --with-db-type=berkeley                            \
                    --with-db-name=openca-ra                           \
                    --with-db-host=localhost                           \
                    --with-db-port=3306                                \
                    --with-db-user=openca                              \
                    --with-ldap-host=raguel.imp.mx                     \
                    --with-ldap-root="cn=Manager,o=IMP,c=MX"           \
                    --with-ldap-port=389                               \
                    --with-ldap-root-pwd=secretimp                     \
```

Igual que para la AC, lo anterior realiza la configuración de la instalación del software de acuerdo a nuestro sistema. Dentro de esta configuración podemos distinguir algunas diferencias respecto a la de la AC y son las que comentamos a continuación, en orden descendente:

La primera, y tal vez la más importante diferencia, es la jerarquía. En esta configuración lo que deseamos es crear una AR, con esto obtenemos una infraestructura que tendrá todas las capacidades y facilidades para actuar como tal. Ello quiere decir dentro del árbol jerárquico de PKI, esta se encuentra debajo de la AC, esto implica que se encuentra ligada a una AC para trabajar.

La siguiente diferencia importante, es que es en la configuración de la AR donde indicamos que utilizaremos el servicio de directorio LDAP. Para esto indicamos algunos parámetros importantes en las siguientes directivas:

Una directiva que indica el host que brindará el servicio, en nuestro caso al igual que para la interfaz web es raguel.imp.mx, la diferencia es que el primero se brinda mediante el protocolo ldap, y el segundo mediante http ó https.

Una directiva que indica el usuario que cuenta con todos los privilegios de "root", este usuario tendrá los permisos necesarios para agregar nuevas entradas al directorio o eliminar algunas existentes.

Una directiva que indica el puerto por el cual se atenderá el servicio, uno de los puertos dedicados para dicho servicio es el 389 y es el que aquí utilizamos.

Finalmente una directiva que indica la clave o password de acceso que el servidor ldap pide para poder brindar las facilidades de escritura y borrado de las entradas del directorio.

Existen aún más directivas que podemos incluir, estas dependerán de la configuración que se desee, todas las opciones de configuración se encuentran en la documentación de la distribución.

Compilar e Instalar AR

Una vez que el proceso de configuración termine con éxito, lo siguiente es compilar e instalar el software. OpenCA utiliza método usual de instalación de los fuentes de OpenSource (*make* y *make install*). Al igual que para la AC, la pequeña variante que se utiliza en la instalación se encuentra en la fase propia de la instalación del software, la cual debe ser indicada de la siguiente manera:

```
]# make install-ext
```

Con esta instrucción, se indica que deseamos realizar una instalación de los componentes "externos" de la Infraestructura. Como ya lo hemos mencionado antes, OpenCA está contemplado para trabajar en dos módulos principales, uno offline (interno) y otro online (externo). El módulo externo a su vez se conforma de todos aquellos servicios con los que puede interactuar o de los que puede disponer un usuario normal de PKI. En este caso el módulo externo se compone de los servicios de la AR y del los del directorio LDAP.

Si no se desea instalar el módulo de LDAP, simplemente se utiliza:

```
]# make install-ra
```

ANEXO C-3

SECCIÓN IV

Configuración del servidor apache para la AR

Manejamos la configuración y el servicio mediante un host virtual dentro de apache. Para esto es necesario realizar algunas modificaciones en el archivo de configuración httpd.conf y agregar lo siguiente:

En la sección donde indicamos al servidor él o los puertos que debe de "escuchar" (Listen) agregamos:

```
Listen 443
```

Con esto estamos indicando que el servicio web de la AR será atendido por el puerto 443 (servicio de conexión segura SSL/TLS).

Adicionalmente, al final de la sección 2 (Section 2:'Main' server configuration) debemos agregar la siguiente línea:

```
Include /hdc1/ra/apachera.conf
```

Con esto indicamos que cuando se levante el servicio de apache y se procese el archivo de configuración apache (http.conf) se incluya el de la AR, que se encuentra ubicado en el "path" indicado, en nuestro caso es en /hdc1/ra y se llama apachera.conf.

Siguiendo los mismos pasos que para la AC, es necesario crear el archivo de configuración del servicio web de la AR. Para esto se crea el archivo apachera.conf en cualquier editor y lo colocamos en la ubicación dada anteriormente.

```
Archivo de configuración apachera.conf en /hdc1/ra/

# Configuración del servidor  apache para la RA  por Carlos R Zainos H, actualizado 18/08
# Ligado directamente en http.conf, trabaja  como  host virtual en el puerto 443
<VirtualHost 192.168.144.102:443>
    ServerAdmin root@raguel.imp.mx
    DocumentRoot /hdc1/ra/apache/htdocs
    ServerName raguel.imp.mx
    #ErrorLog logs/dummy-host.example.com-error_log
    SetenvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-
response-1.0
    CustomLog /hdc1/ra/log/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\"  %b"

    SSLEngine on
    SSLCertificateFile /hdc1/ra/ssl.crt/server.pem
    SSLCertificateKeyFile /hdc1/ra/ssl.key/key.pem

    <Directory "/hdc1/ra/apache/htdocs/pub/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from all
     </Directory>
#Solo  permite  el acceso al directorio desde "adentro(subred)"

     <Directory "/hdc1/ra/apache/htdocs/ra/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     <Directory "/hdc1/ra/apache/htdocs/ra_node/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     <Directory "/hdc1/ra/apache/htdocs/ldap/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     ScriptAlias /cgi-bin/  "/hdc1/ra/apache/cgi-bin/"
     <Directory "/hdc1/ra/apache/cgi-bin">
        AllowOverride  None
       Options None
       Order allow,deny
       Allow  from all
     </Directory>
</VirtualHost>
```

La configuración anterior del servicio "externo" contiene información importante para la manera de trabajar e interactuar con la AR y con LDAP. A grandes rasgos comentaremos los puntos más importantes, los detalles se pueden ver en la documentación de Apache.

En primer lugar encontramos la definición del host virtual:

```
<VirtualHost 192.168.144.102:443>
```

Con esta directiva le indicamos al servidor que implementará un servicio web en la dirección IP indicada y por el puerto 443. Esto es debido a que toda el tráfico de información se hará con el protocolo SSL/TLS , por cuestiones de seguridad.

Después de esto, y de manera similar a la AC, encontramos directivas que indican entre otras cosas, el administrador del servidor, la ubicación de la "documentación", el nombre del servidor, la opciones con las que se levantará el servidor y archivos logs que se generarán.

Lo siguiente es una sección muy importante dentro de la configuración, se trata del servicio de SSL.Primeramente indicamos que deseamos tener activo este servicio con la directiva :

```
SSLEngine on
```

A continuación le decimos al servidor la ubicación del certificado de servidor web, así como de la clave privada que debe utilizar. Esto es necesario para que el servidor pueda negociar conexiones SSL con los clientes e igualmente pueda cifrar/descifrar la información que se intercambie.

```
SSLCertificateFile /hdc1/ra/ssl.crt/server.pem
SSLCertificateKeyFile /hdc1/ra/ssl.key/key.pem
```

A continuación encontramos directivas que configuran los directorios dentro del servidor. Estos son "pub", "ra", ra_node" y "ldap". Estos son los directorios que conforman los servicios web "externos" de OpenCA.

El primero es un directorio que está destinado para que cualquier usuario pueda entrar en él. El segundo es el directorio de trabajo de la AR, el tercero es el directorio de trabajo del nodo de intercambio de información AR-AC y el tercero es el directorio de trabajo de LDAP.

Para los tres últimos, debemos limitar el acceso a ellos solo al personal autorizado, esto se logra con directivas de control e acceso mediante IP, en nuestro caso se dejó libre acceso a todo el segmento de red que conforma Seguridad Informática por cuestiones de prueba. En un futuro solo debe de permitirse el acceso al administrador de la AR y de LDAP.

Finalmente, el servidor web se "levanta" mediante la siguiente instrucción:

```
/usr/local/apache2/bin]# ./apachectl start
```

ANEXO C-3

SECCIÓN V

Configuración del servicio LDAP

Retomamos para este punto lo realizado en las sección 3.3 del presente trabajo.

Ahora toca configurar el servidor LDAP para poder utilizarlo conjuntamente con OpenCA, y en particular con la AR.

En el archivo de configuración slapd.conf en /usr/local/etc/openldap/ verificar que las siguientes líneas se encuentren, de lo contrario agregarlas:

```
Include /etc/openldap/schema/core.schema
Include /etc/openldap/schema/cosine.schema
Include /etc/openldap/schema/inetorgperson.schema

pidfile usr/local/var/slapd.pid
argsfile usr/local/var/slapd.args

allow bind_v2
database bdb
suffix "o=IMP, c=MX"
rootdn "cn=Manager,o=IMP,c=MX"
rootpw  secretimp
```

Lo anterior debe concordar con las líneas en /hdc1/ra/OpenCA/etc/servers/ldap.conf que a continuación se muestran:

```
LDAP            "yes"
Ldapserver      raguel.imp.mx
Ldapversion     2
Ldapport        389
Ldaplimit       100
Basedn "o=IMP,c=MX"
Ldaproot        "cn=Manager,o=IMP,c=MX"
Ldappwd         "secretimp"
Ldapbasedir     "/usr/local/ldap"
```

Son todas las modificaciones que se deben realizar en los archivos de configuración de LDAP. Existen muchas más opciones y facilidades que proporciona OpenLDAP. Revise la documentación para una referencia más completa.

Finalmente solo queda "levantar" el servidor de OpenLDAP mediante la siguiente instrucción:

```
/usr/local/libexec/]# slapd
```

ANEXO C-4
Guía de Administración de PKI-IMP V0.2
Noviembre 2003 (Fragmento)

El presente documento tienen como objetivo servir como una referencia ilustrada de la instalación y configuración de la herramienta PKI-OpenCA para utilizarla como base en la implantación de la infraestructura de PKI-IMP.

Por "referencia" queremos decir que esto es solo un ejemplo, ya que esta documentación se realizó utilizando la versión 0.9.1, actualmente la versión 0.9.2 ha sido liberada para efectos de pruebas.

Lo que a continuación se presenta es lo que se realizó para levantar la infraestructura PKI-IMP. Esta refleja los cambios, adaptaciones y modificaciones realizados respecto a la instalación original. Para mayor información refiérase a la documentación del proyecto que se encuentra en http://www.openca.org, ó a la incluida en la distribución. Igualmente pueden consultarse las listas de discusión, parches, documentación, últimas versiones y distribuciones de evaluación en http://ww.sourceforge.org/openca

1- Instalación y Configuración de Software

1.1 Instalación

La instlación de OpenCA está basada en las bien conocidas herramientas GNU autoconf y automake. Los pasos básicos son muy sencillos:

1- Descargue una versión de OpenCA
2- Desempaquete la distribución (gzip –d *distribución.tar.gz*)
3- Descomprima la distribución (tar –xvf *distribución.tar*)
4- Cámbiese al directorio de la distribución
5- Configure los fuentes
6- Ejecute Make
7- Instale el software

Los primeros cuatro pasos son muy simples. Los problemas comienzan con la configuración del software, así que antes de que comience a leer sobre los problemas por favor lea los requerimientos del software cuidadosamente, ya que un buen número de problemas son causados por versiones incorrectas o por módulos no encontrados.

1.1.1 Requerimientos

OpenCA no es un sistema monolítico. Para su correcta configuración e instalación, OpenCA requiere de varios productos de software en módulos proporcionados por terceras partes, todos dentro de la comunidad de Software Libre (OpenSource).

Perl 5.6.1 o superior.
Este módulo es uno de los más importantes y por tanto se recomienda tener instalada la última versión de Perl con todos sus componentes, principalmente los criptográficos.
La última versión de Perl así como la documentación relacionada se pueden descargar de http://www.perl.org.

OpenSSL 0.9.7 o superior.

Apache http Server + mod_ssl
Este módulo es indispensable ya este es quien soporte todas las interfaces web de OpenCA. El módulo mod_ssl en necesario para poder atender las peticiones que se realicen mediante el protocolo https.

OpenLDAP
Este módulo es necesario solo si se desea levantar el servicio de directorio.

Compilador de C (gcc estará bien)

1.2Configuración de los fuentes

OpenCA utiliza el método usual de configuración de los fuentes de OpenSource. La configuración de la distribución es necesaria para poder compilar e instalar el software mas no para configurar la manera en cómo trabajará el sistema ya instalado. Durante la configuración se realizan algunas configuraciones por defecto si es que no se especifica lo contrario, pero la configuración real se realiza en la fase de post-instalación. En la sección 1.4 de este documento se presentan ejemplos de esta configuración

1.3 Instalación del Software compilado

Si desea instalar el software, entonces se debe seleccionar que partes o módulos se desean instalar:

- *Install-ca* instalará solo la AC y el nodo de administración de la misma
- *Install-ext* instalará la AR, el servicio LDAP, la interfaz pública y el nodo de administración.
- *Install-ra* instalará solo la AR y el nodo de administración de la misma.
- *Install-pub* instalará solo la interfaz pública y el nodo de administración

El nodo de administración se incluye siempre debido a que se necesitará administrar la base de datos y el intercambio de información con los otros nodos de la jerarquía de PKI-IMP.

1.4 Ejemplo de Instalación de OpenCA para PKI-IMP

Antes que nada definamos que es cada cosa así como su ubicación

- Autoridad Certificadora, AC
  - Ubicación http://raguel.imp.mx:88/ca
  - No necesita conexión a Internet o a alguna otra red (opcional y dependiendo de la configuración).

- Autoridad Registradora, AR
  - Ubicación: https://raguel.imp.mx/ra
  - Necesita conexión a Internet o a la red corporativa de la organización.
  - Contienen la interfaz pública para la solicitud de certificados
  - Contiene la interfaz de la AR para realizar la s tareas de pre-certificación.

1.4.1 Instalación de los fuentes de la AC

Es en la AC donde en realidad se emiten los certificados digitales. Por razones de seguridad, se recomienda que la AC no debe de estar conectada a alguna red. Debiera ser una computadora "aislada". En este caso por razones de prueba no fue así. Los certificados y las solicitudes deben de ser transportadas en un floppy (disquete). Para efectos de prueba del software, se pueden instalar la AC y la AR en el mismo servidor (nuestro caso).

1.4.1.1 Configuración del software

Con base en la documentación del proyecto, a nuestro juicio y conveniencia, seleccionamos y ejecutamos la siguiente configuración para la AC. Previamente creamos el directorio de instalación el cual para nosotros fue: /srv/ca

Desde la línea de comandos, dentro del directorio de desempaque de OpenCA:

```
]# ./configure    --prefix=/srv/ca                                  \
                    --with-apache=/usr/local/apache2                \
                    --with-openssl-prefix=/usr/local/ssl            \
                    --with-web-host=raguel.imp.mx                   \
                    --with-httpd-user=nobody                        \
                    --with-httpd-group=nobody                       \
                    --with-dist-user=zainos                         \
                    --with-dist-group=openca                        \
                    --enable db                                     \
                    --enable-ocspd                                  \
                    --with-ca-organization=IMP                      \
                    --with-ca-locality=Mexico                       \
                    --with-ca-country=MX                            \
                    --with-service-mail-account=root@raguel.imp.mx  \
                    --with-mail-program="usr/sbin/sendmail –t"      \
                    --with-hierarchy-level=ca                       \
                    --with-db-type=berkeley                         \
                    --with-db-name=openca                           \
                    --with-db-host=localhost                        \
                    --with-db-port=3306                             \
                    --with-db-user=openca                           ENTER
```

1.4.1.2 Compilar e Instalar AC-IMP

- Make
- Make install-ca

1.4.1.3 Configuración del servidor web apache de la AC

Para este punto usamos host virtuales en http.conf, incluios las siguientes líneas:

```
Listen 88
Include /srv/ca/apache.conf
```

Una vez hecho esto toca crear el archivo de configuración del servicio web de la AC. Para esto creamos el archivo apache.conf y lo colocamos en la ubicación dada anteriormente. Para esto en cualquier editor tecleamos lo siguiente:

```
# Configuracion del servidor  apache para la CA  por Carlos R Zainos H
# Ligado directamente en httpd.conf, trabaja  como  host virtual

<VirtualHost 192.168.144.102:88>
        ServerAdmin root@raguel.imp.mx
        DocumentRoot /srv/ca/apache/htdocs
        ServerName 192.168.144.102
        #ErrorLog logs/dummy-host.example.com-error_log
        SetenvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-
        response-1.0
        CustomLog /srv/log/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
        <Directory "/srv/ca/apache/htdocs">
                Options Indexes FollowSymlinks Multiviews
                AllowOverride None
                Order allow,deny
                Allow from all
        </Directory>
```

```
        ScriptAlias /cgi-bin/  "/srv/ca/apache/cgi-bin/"
        <Directory "/srv/ca/apache/cgi-bin">
                AllowOverride  None
                Options None
                Order allow,deny
                Allow  from all
        </Directory>
</VirtualHost>
```

1.4.2 Instalación de los fuentes de la AR

La interfaz externa: aquí se puede crear una solicitud de certificado. Después de esto la solicitud o requerimiento debe de ser transportado a la AC.

1.4.2.1 Configuración del software

Al igual que para la AC, y con base en la documentación del proyecto, a nuestro juicio y conveniencia, seleccionamos y ejecutamos la siguiente configuración para la AR. Previamente creamos el directorio de instalación el cual para nosotros fue: /hdc1/ra

Desde la línea de comandos, dentro del directorio de desempaque de OpenCA:

```
]# ./configure    --prefix=/hdc1/ra                                  \
                    --with-apache=/usr/local/apache2                 \
                    --with-openssl-prefix=/usr/local/ssl             \
                    --with-web-host=raguel.imp.mx                    \
                    --with-httpd-user=nobody                         \
                    --with-httpd-group=nobody                        \
                    --with-dist-user=utirado                         \
                    --with-dist-group=openca                         \
                    --enable db                                      \
                    --enable-ocspd                                   \
                    --with-ca-organization=IMP                       \
                    --with-ca-locality=Mexico                        \
                    --with-ca-country=MX                             \
                    --with-service-mail-account=root@raguel.imp.mx   \
                    --with-mail-program="usr/sbin/sendmail –t"       \
                    --with-hierarchy-level=ra                        \
                    --with-db-type=berkeley                          \
                    --with-db-name=openca-ra                         \
                    --with-db-host=localhost                         \
                    --with-db-port=3306                              \
                    --with-db-user=openca                            \
                    --with-ldap-host=raguel.imp.mx                   \
                    --with-ldap-root="cn=Manager,o=IMP,c=MX"         \
                    --with-ldap-port=389                             \
                    --with-ldap-root-pwd=secretimp                   ENTER
```

1.4.2.2 Compilar e Instalar AR-IMP e interfaz externa

- Make
- Make install-ext

1.4.2.3 Configuración del servidor web apache de la AR

De la misma manera que hicimos para la AC, manejamos la configuración y el servicio mediante un host virtual dentro de apache. Para esto es necesario realizar algunas modificaciones en el archivo de configuración httpd.conf y agregar lo siguiente:
En la sección donde indicamos al servidor él o los puertos que debe de "escuchar" (Listen) agregamos:

```
        Listen 443
```

Con esto estamos indicando que el servicio web de la AR será atendido por el puerto 443 (servicio de conexión segura SSL/TLS).

Adicionalmente, al final de la sección 2 (`Section 2:'Main' server configuration`) debemos agregar la siguiente línea:

```
Include /hdcl/ra/apachera.conf
```

Siguiendo los mismos pasos que para la AC, toca crear ahora el archivo de configuración del servicio web de la AR. Para esto creamos el archivo apachera.conf y lo colocamos en la ubicación dada anteriormente. Para esto en cualquier editor tecleamos lo siguiente:

```
# Configuración del servidor  apache para la RA  por Carlos R Zainos H, actualizado 18/08
# Ligado directamente en http.conf, trabaja  como  host virtual en el puerto 443
<VirtualHost 192.168.144.102:443>
    ServerAdmin root@raguel.imp.mx
    DocumentRoot /hdcl/ra/apache/htdocs
    ServerName raguel.imp.mx
    #ErrorLog logs/dummy-host.example.com-error_log
    SetenvIf  User-Agent  ".*MSIE.*"  nokeepalive  ssl-unclean-shutdown  downgrade-1.0  force-
response-1.0
    CustomLog /hdcl/ra/log/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\"  %b"

    SSLEngine on
    SSLCertificateFile /hdcl/ra/ssl.crt/server.pem
    SSLCertificateKeyFile /hdcl/ra/ssl.key/key.pem

    <Directory "/hdcl/ra/apache/htdocs/pub/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from all
     </Directory>
#Solo  permite  el acceso al directorio desde "adentro"

     <Directory "/hdcl/ra/apache/htdocs/ra/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     <Directory "/hdcl/ra/apache/htdocs/ra_node/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     <Directory "/hdcl/ra/apache/htdocs/ldap/">
        Options Indexes FollowSymlinks Multiviews
        AllowOverride None
       Order allow,deny
       Allow from 192.168.144
     </Directory>

     ScriptAlias /cgi-bin/  "/hdcl/ra/apache/cgi-bin/"
     <Directory "/hdcl/ra/apache/cgi-bin">
        AllowOverride  None
       Options None
       Order allow,deny
       Allow  from all
     </Directory>
</VirtualHost>
```

Nota: Las interfaces para la AR y para LDAP deben de ser solamente accesibles para la AR. Aquí nosotros utilizamos un método simple de protección. Se permite el acceso solo para una subred, la nuestra 192.168.144.

1.4.3 Configuración del servicio LDAP

Ahora toca configurar el servidor LDAP para poder utilizarlo conjuntamente con OpenCA, y en particular con la AR.

En el archivo de configuración slapd.conf en /usr/local/etc/openldap/ verificar que las siguientes líneas se encuentren, de lo contrario agregarlas:

```
Include /etc/openldap/schema/core.schema
Include /etc/openldap/schema/cosine.schema
Include /etc/openldap/schema/inetorgperson.schema
Include /usr/local/etc/openldap/schema/PKI-OpenCA.schema

pidfile usr/local/var/slapd.pid
argsfile usr/local/var/slapd.args

allow bind_v2
database bdb
suffix "o=IMP, c=MX"
rootdn "cn=Manager,o=IMP,c=MX"
rootpw  secretimp
directory       /usr/local/var/openldap-data
```

Lo anterior debe concordar con las líneas en /hdc1/ra/OpenCA/etc/servers/ldap.conf que a continuación se muestran:

```
LDAP            "yes"
Ldapserver      raguel.imp.mx
Ldapversion     2
Ldapport        389
Ldaplimit       100
Basedn  "o=IMP,c=MX"
Ldaproot        "cn=Manager,o=IMP,c=MX"
Ldappwd         "secretimp"
Ldapbasedir     "/usr/local/ldap"
```

Para finalizar y dar paso a la fase de pruebas, solo queda "levantar" el servicio de LDAP mediante la siguiente instrucción:

```
/usr/local/libexec/]# slapd
```

Igualmente levantamos el servidor web Apache mediante la instrucción :

```
/usr/local/apache2/bin]# ./apachectl start
```

2- Inicialización de los módulos de PKI-IMP-OpenCA

2.1 Inicializar la AC

2.1.1 Preparaciones

Es necesario asegurarse, antes de continuar, de que tenemos los permisos de lectura y escritura sobre el dispositivo fd0 (floppy) ya que será por medio de él que transportaremos la configuración de la AC y de la AR.
Para esto tecleamos en la ventana de terminal lo siguiente :

```
]# chmod 777 /dev/fd0
```

Lo anterior le indica a nuestro sistema que permita la lectura, escritura y ejecución sobre el dispositivo fd0 (floppy) al usuario-grupo-todos.

2.1.2 Inicialización de la AC Fase 1

La AC se inicializa en http://raguel.imp.mx:88/ca, recordemos que la configuración se hizo para que el servicio fuese atendido por el puerto 88.

El sistema nos mostrará la página de inicio (Figura 1.1)

Nos dirigimos a la liga **Inicialización** que nos lleva a la página que muestra la Figura 1.2.

Se muestran algunas indicaciones en pantalla, léalas cuidadosamente

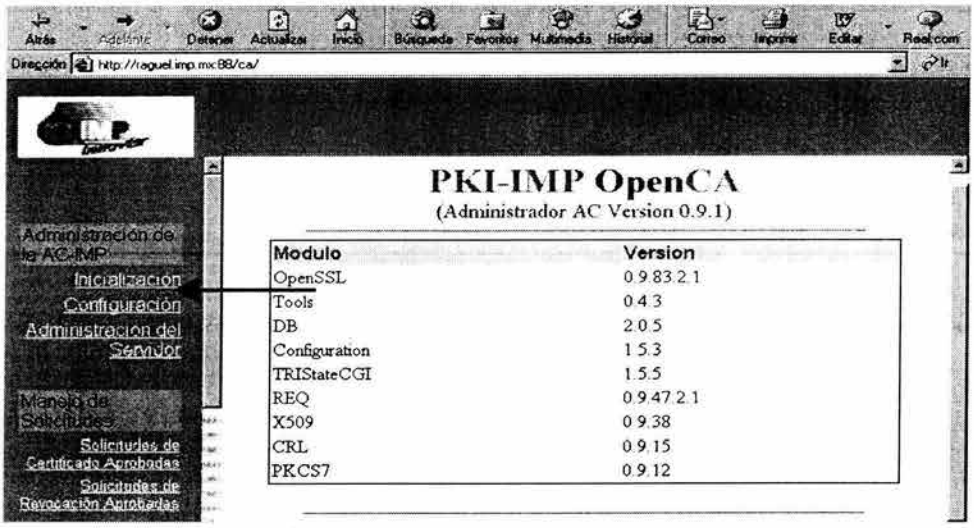


Figura 1.1

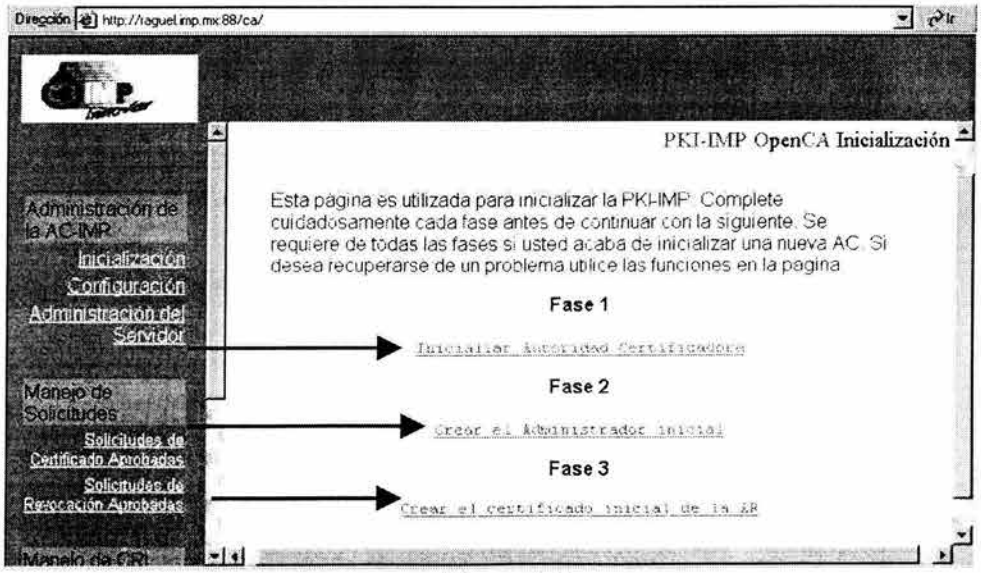


Figura 1.2

Ahí seleccionamos la liga de la sección **Fase 1** que nos llevará a la página que se muestra en la Figura 1.3.

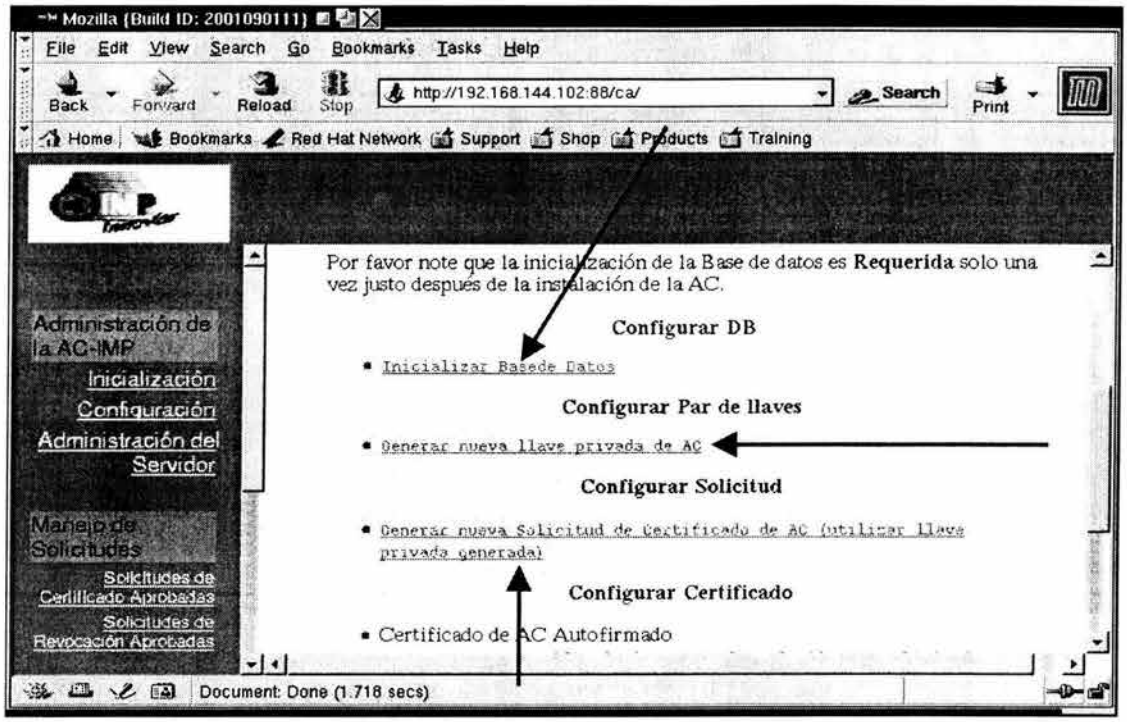


Figura 1.3

Seleccionamos la opción **Inicializar Base de Datos** de la sección **Configurar DB**, esto se realiza sólo una vez después de la instalación de OpenCA.

El proceso no debe de generar ningún contratiempo y el sistema enviará un mensaje de "La base de Datos fue Inicializada exitosamente".

A continuación en la sección **Configurar Par de llaves (Figura 1.3),** seguimos la liga **Generar nueva llave privada de AC** e introducimos en los datos solicitados los siguientes parámetros:

- Algoritmo simétrico 3DES,
- Longitud de clave privada 2048,
- Password de protección : impca00

Como siguiente paso vamos a la liga **Generar nueva solicitud de certificado de AC** (utilizando la clave secreta generada anteriormente) de la sección **Configurar Solicitud** (Figura 1.3) y completamos o modificamos la forma con los siguientes datos:

- Email: root@raguel.imp.mx
- Common Name : Autoridad Certificadora IMP
- Organizational Unit: Seguridad Informática
- Organization: IMP
- Country: MX

Como resultado nos arroja el siguiente DN: DN= emailAddress=root@ragul.imp.mx, CN=Autoridad Certificadora IMP, OU=Seguridad Informática, O=IMP, C=MX

El anterior corresponde al DN de la Autoridad Certificadora del IMP creada mediante OpenCA.

A continuación creamos un certificado de AC autofirmado(a partir de la solicitud creada en el punto anterior), en la sección **Configurar Certificado** (misma página de la Figura 1.3). Dar click en la liga correspondiente.

Introducimos el password que protege la clave privada (impca00)

Finalmente reconstruimos la cadena de certificación (click en la liga correspondiente de la Figura 1.3)), en este caso el certificado creado estará en la cima de la cadena. Al final de esto habremos obtenido un certificado llamado cacert.crt

A continuación colocamos un disquete en la unidad de 3 ½ y exportamos la configuración de la AC, figura 1.3 (los parámetros dados anteriormente y el certificado).

2.1.3 Inicialización de la AC Fase 2

Inicializamos y creamos al administrador inicial (certificado de usuario para firmar transacciones) en la siguiente ubicación : http://raguel.imp.mx:88/ca

El sistema nos llevará a la página que se muestra en la Figura 1.2

Ahí seleccionamos la liga de la sección **Fase 2** que nos llevará a la página que se muestra en la Figura 1.4.

Vamos a la liga de **Crear una nueva solicitud** (Paso 1 Figura 1.4):

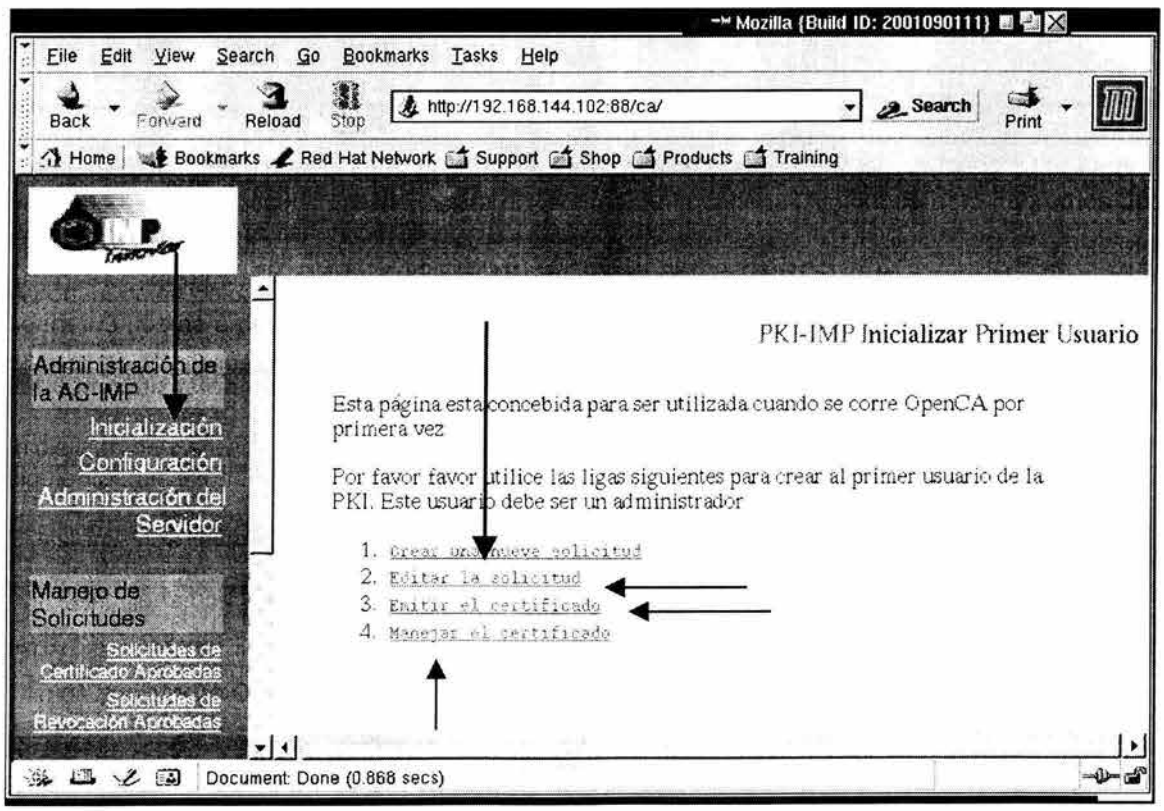


Figura 1.4

Introducimos los siguientes valores en los campos correspondientes:

- Email: root@raguel.imp.mx
- Nombre : Uriel Tirado Ríos
- Grupo del certificado solicitado: Internet
- Rol : RA Operator
- Autoridad Registradora: Autoridad Registradora IMP
- PIN: adminutr123
- Longitud de claves 1024

Obtendremos un resultado como el siguiente (Figura 1.5):

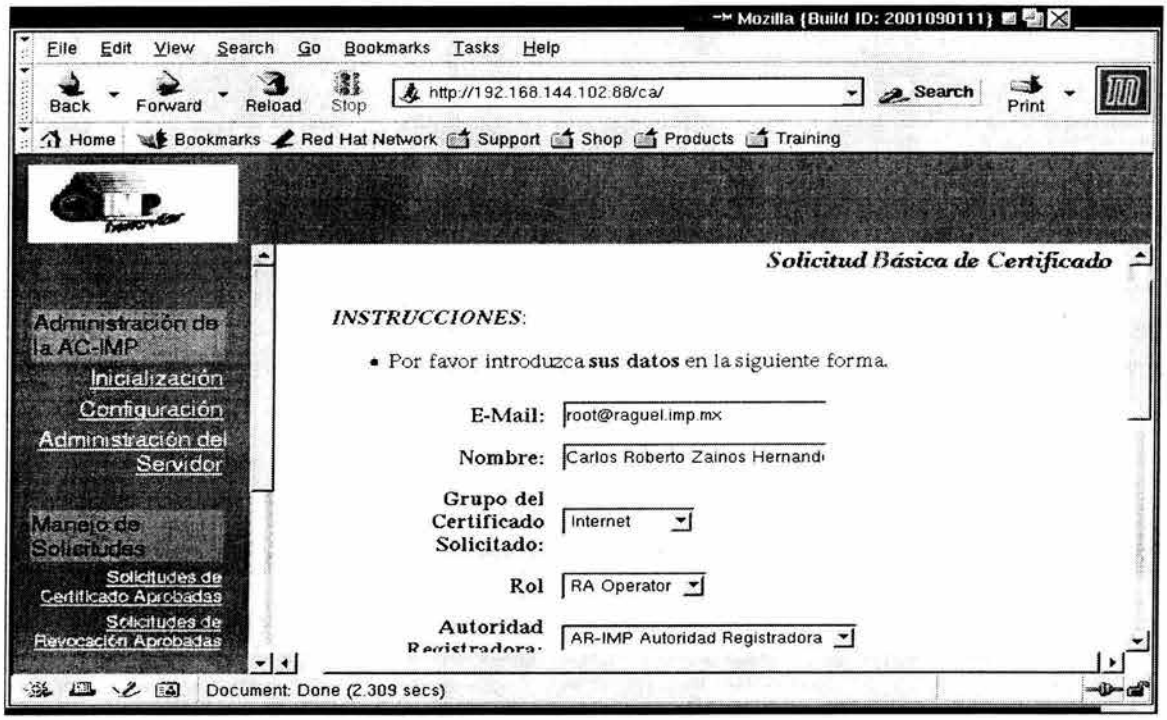


Figura 1.5

A continuación damos clic en los botones de **continuar** hasta finalizar con éxito al solicitud (Figura 1.6).

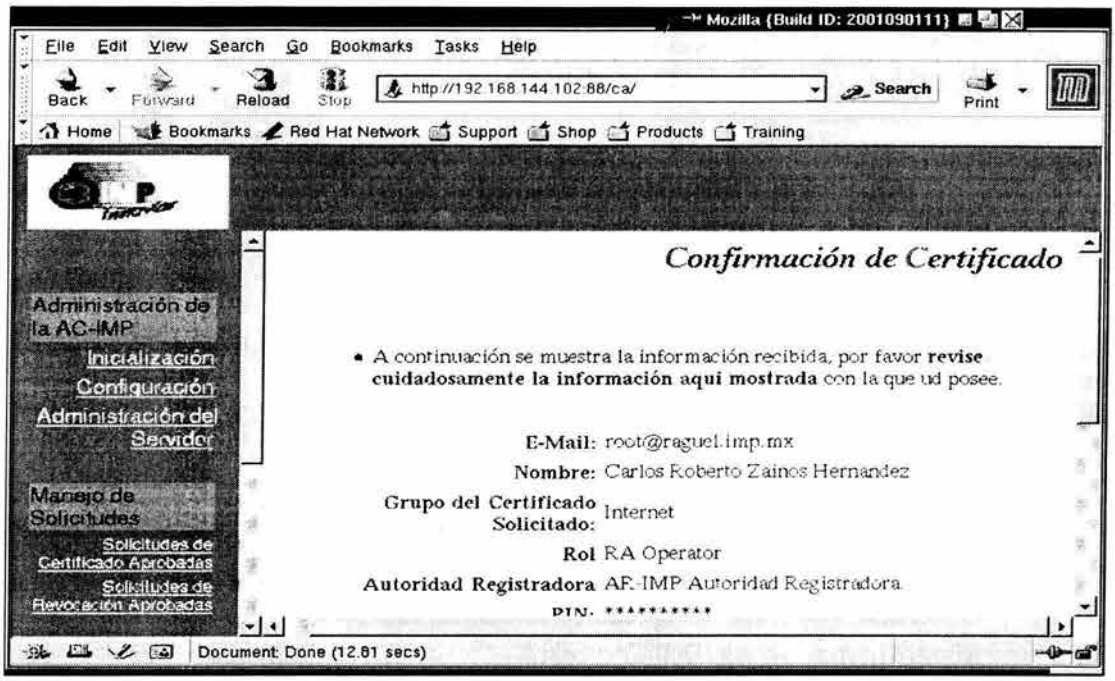


Figura 1.6

A continuación damos click en **Editar la Solicitud** (Paso 2 Figura 1.4). nos deberá enviar una página como la siguiente (Figura 1.7)

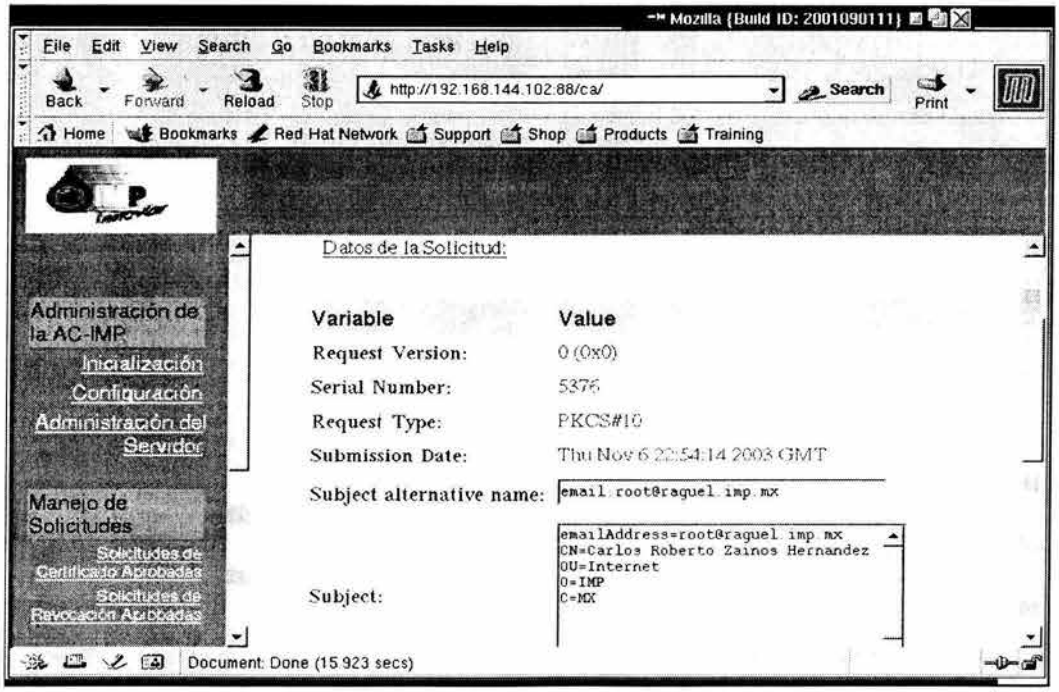


Figura 1.7

Sí todos los datos son correctos, damos click en el botón OK.

Finalmente daos click en la liga **Emitir Certificado** (Paso 3 Figura 1.4)

Al momento de realizar esto se nos pide la clave que protege la clave privada de la AC para firmar el certificado. Lo introducimos y entonces se genera el certificado.

Al terminar el procedimiento deberemos ver una ventana como la que se muestra en la Figura 1.8.

Figura 1.8

En este momento contamos con un certificado de operador de AR. Lo que nos permitirá firmar las solicitudes de certificado que nos lleguen.

A continuación tenemos que manejar el certificado, para esto seguimos la liga correspondiente (Paso 4 Figura 1.4) y a continuación se nos presentarán varias opciones. Seleccionamos descargar el certificado y el par de claves en formato PKCS#12. Aparecerá un diálogo de descarga de archivo, asignar extensión p12 (*archivo.p12*)

Después de esto descargamos el archivo e introducimos el password que protege al nuevo certificado (adminutr123)

Lo siguiente es cargar el certificado y la llave en el navegador del operador de la AC, se puede seleccionar IExplorer o Netscape.

2.1.4 Inicialización de la AC Fase 3

A continuación inicializamos y creamos el primer certificado de AR (certificado web para apache) en http://raguel.imp.mx:88/ca

El sistema nuevamente nos llevará a la página de la Figura 1.2

Seleccionamos la liga de la sección **Fase 3,** el sistema nos mostrará la página siguiente (Figura 1.9)

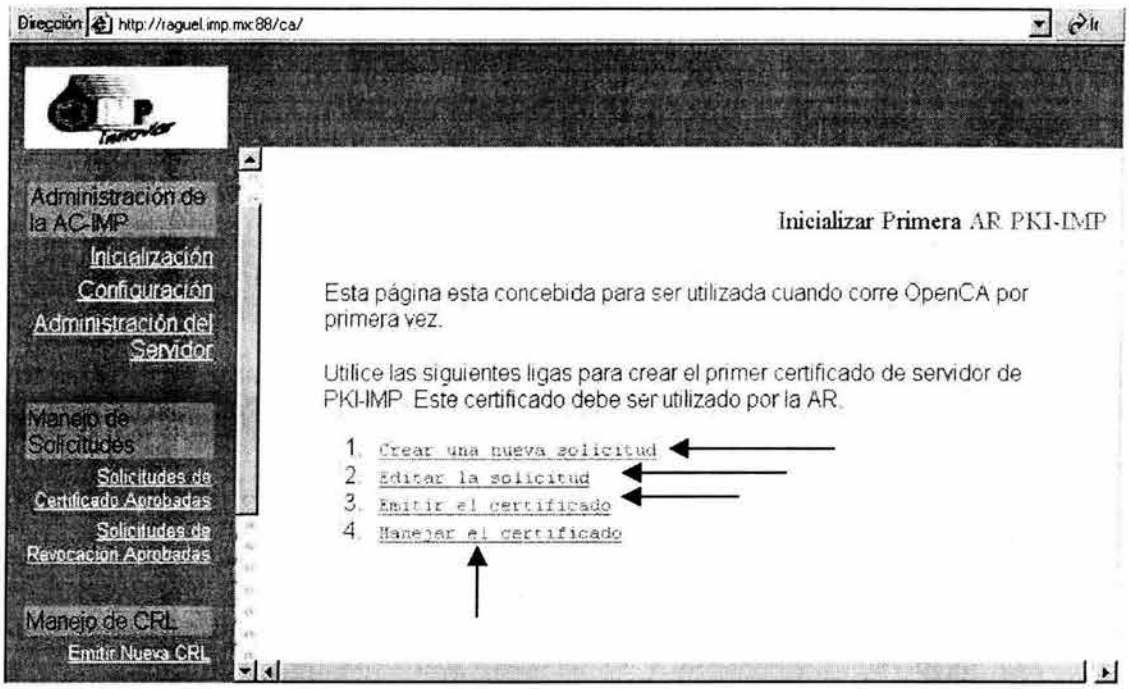


Figura 1.9

En la liga correspondiente creamos una nueva solicitud

Introducimos los siguientes valores para la solicitud:

- Email: root@raguel.imp.mx
- Nombre : raguel.imp.mx
- Grupo del certificado solicitado: Internet
- Rol : Web Server
- Autoridad Registradora: Autoridad Registradora IMP
- PIN: racertserver
- Longitud de claves 1024

A continuación damos clic en los botones de continuar hasta finalizar con éxito al solicitud.

En esta solicitud distinguimos la siguiente particularidad: como este será el certificado que presente el servidor web Apache a los usuarios que lo contacten, el nombre del sujeto debe ser el nombre del servidor o del host que se teclea en la barra de direcciones. En nuestro caso este nombre corresponde a raguel.imp.mx; este nombre, hay que decirlo, fue dado de alta previamente en el DNS del IMP y se ligó a la dirección 192.168.144.102 que corresponde a la máquina en la cual se levanta toda la infraestructura PKI-IMP.

Lo que sigue es editar la solicitud para hacer un pequeño cambio en la misma (Paso 2 figura 1.9).

En la sección que muestra los datos de la solicitud, cambiamos el nombre alterno del sujeto (Subject Alternate Name) de "email:root@raguel.imp.mx" por este otro:

"DNS:raguel.imp.mx;email:root@raguel.imp.mx" .

Lo anterior tiene como objetivo indicarle a OpenSSL los parámetros que tiene que utilizar para emitir el certificado correspondiente.

Damos clic en OK para guardar los cambios y a continuación emitimos el certificado (Paso 3 Figura 1.9)

Introducimos el password que protege la clave privada de la AC y damos clic en aceptar.

A continuación manejamos el nuevo certificado y el par de claves (Paso 4 Figura 1.9) de la siguiente manera:

- Siguiendo la liga correspondiente seleccionamos SSLeay (mod_ssl) en la ventana de opciones.
- Seleccionamos descargar
- Introducimos el password que protege al clave privada (racertserver}

A continuación el certificado debe aparecer en el navegador. La manera de mostrarse es en formato de servidor, lo cual indica que lo que veremos será una larga cadena de caracteres alfanuméricos limitados por los encabezados:

```
      -----BEGIN CERTIFICATE-----
      -----END CERTIFICATE-----
y
      -----BEGIN RSA PRIVATE KEY-----
      -----END RSA PRIVATE KEY-----
```

Los que corresponde al certificado digital en cuestión en formato PEM, así como la clave privada correspondiente.

Este certificado y esta llave es la que Apeche utilizará para cifrar las conexiones realizadas con SSL/TLS , por lo que a continuación seleccionamos todo y realizamos un respaldo de esto en un lugar seguro. Enseguida creamos los siguientes directorios: ssl.crt y ssl.key dentro del directorio de la AR (/hdc1/ra). Luego seleccionamos solo la parte que se refiere al certificado y desde cualquier editor de texto lo guardamos con un nuevo nombre, en nuestro caso fue server.pem. Realizamos lo mismo para la parte de la clave privada , esta fue guardada como key.pem.

Recordemos que en la parte de la configuración del servidor Apache para la AR dimos esta ubicación y estos archivos. Si estos no corresponden o no se encuentran, al momento de levantar el servidor Apache para la AR se generará un error.

Con esto terminamos, por el momento, lo referente a la AC. Lo que sigue es realizar algo parecido para la AR.

2.2 Inicializar la AR

2.2.1 Preparaciones

Lo que hay que tener listo antes de iniciar con el procedimiento de inicialización de la AR es el servidor de la Base de Datos, en este caso tener instalado Berkeley DB, y asegurarnos de que podemos utilizar el servicio de envío de e-mails. Para el caso de la base de datos puede ser también MySQL. En tal caso lo que hay que tener en cuenta es que el servicio se encuentre corriendo y verificar que este se inicie al momento de iniciar nuestro sistema. Refiérase a la documentación para notas relacionadas.

2.2.2 Exportar la configuración desde la AC

Etiquetamos con Export-CA-Conf un disquete y lo colocamos en la unidad de 3 ½ .

Nos aseguramos de que tenemos los permisos de lectura-escritura-ejecución del dispositivo (chmod 777 /dev/fd0)

Acto seguido, nos dirigimos al nodo de intercambio de información de OpenCA ubicado en http://raguel.imp.mx:88/ca_node

Nos dirigimos a la liga de **Intercambio de datos**

Seguimos la liga de **Registrar/Enviar datos a un nivel inferior de la jerarquía.**

Seleccionamos que deseamos enviar la configuración (Seguir la liga correspondiente).

El sistema nos enviará un mensaje de exportación realizada satisfactoriamente.

2.2.3 Inicializar la Base de Datos de la AR

A continuación Inicializamos la base de datos de la AR

En la barra de direcciones tecleamos la siguiente ubicación https://raguel.imps.mx/ra_node

Nótese que en este caso estamos estableciendo una conexión mediante el protocolo SSL (https) por lo que el navegador lanzará advertencias de seguridad.

El sistema nos mostrará la página siguiente (Figura 2.1)

Figura 2.1

Nos dirigimos a la liga de **Inicializar Servidor** de la sección **Administración,** el sistema nos llevará a la página siguiente (Figura 2.2).

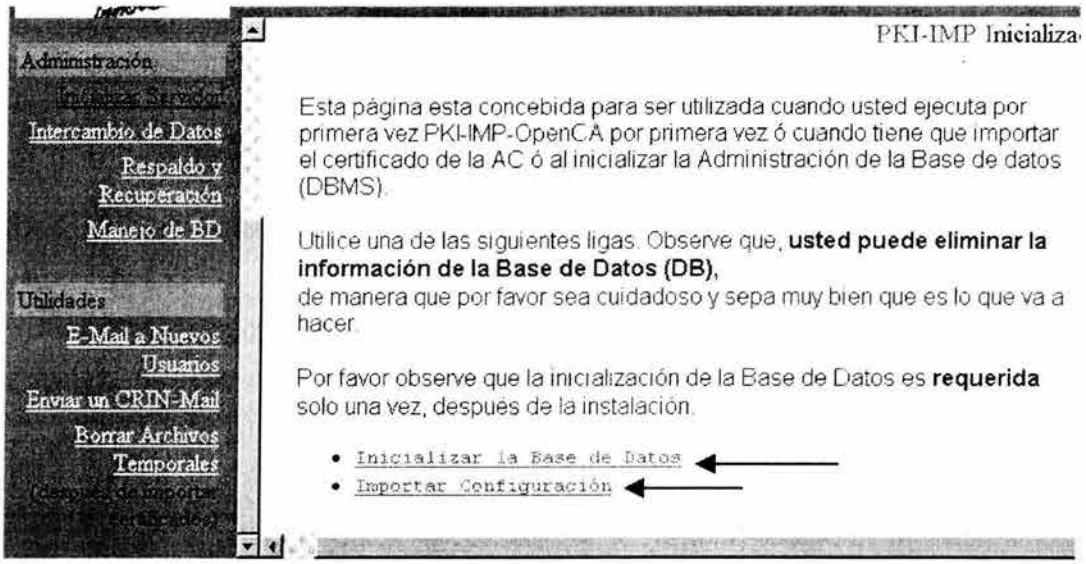


Figura 2.2

Seguimos la liga de **Inicializar la Base de Datos**.

El proceso no debe de generar ningún contratiempo y el sistema enviará un mensaje de "La base de Datos fue Inicializada exitosamente".

A continuación introducimos el disquete con la configuración de la AC, creado en la sección pasada, en la unidad de 3 ½ de la AR. En nuestro caso la AC y la AR se encuentran en la misma máquina, mas recordemos que esto esta planeado para trabajar la AC en un sistema y la AR en otro. Por razones de prueba y por motivos presupuestales tuvimos que instalarlo así, aunque no se recomienda.

Debemos asegurarnos en este punto que el servidor LDAP se encuentra correctamente configurado y que esté corriendo.

A continuación, en la página de la figura 2.2, seguimos la liga **Importar Configuración**.

El sistema leerá la información contenida y enviará un mensaje de importación exitosa o enviará algún mensaje de error.

Si ningún mensaje de error aparece podemos asegurar que la configuración se realizó de manera satisfactoria. En caso contrario es necesario revisar que las condiciones necesarias se cumplen y revisar los logs para mayor referencia del problema encontrado. Refiérase a la documentación de OpenCA, OpenLDAP o Apache para solucionar los problemas específicos.

ANEXO D-1

CONFIGURACIÓN DE OPENSSL BASADA EN PRIVILEGIOS DE USUARIO Y PROCEDIMIENTO PARA LA GENERACIÓN Y FIRMA DE UN CERTIFICADO DIGITAL(Procedimiento de certificación).

La configuración de OpenSSL basada en privilegios se realiza mediante el uso de archivos de configuración específicos para cada tipo de usuario que se pretenda atender. Como consecuencia de esto, podemos tener por ejemplo archivos de configuración para certificados de AC, de AR, de Servidor Web, de servicio y clientes VPN, de usuarios finales, etc. La ventaja de este tipo de configuración radica en las facilidades de administración que se obtienen. Una entidad final pudiese tener mas de un certificado digital, cada uno para un uso específico. En la documentación incluida en la distribución de OpenSSL se detalla este tipo de configuración así como las diferentes opciones con las que se cuenta.

Para el caso de este trabajo, solo se concibieron e implementaron archivos de configuración para AC, para servidor web y para usuario final. Estos archivos se encuentran en las secciones V, VI y VII del Anexo B-1.

A continuación se indica el procedimiento básico para generar y firmar un certificado digital con la AC implementada con OpenSSL.

Paso 1- El primer paso para la certificación es la generación de la solicitud o del requerimiento de certificación y del correspondiente par de claves.

Para esto utilizamos el comando de OpenSSL :

/usr/local/ssl/bin/]# openssl req -new -keyout llaveprivada.pem -out solicitud.pem -days 365 -config /usr/local/ssl/openssl.cnf

El anterior comando tiene varios parámetros los cuales podemos cambiar de acuerdo a nuestras necesidades. Como ya lo hemos dicho antes, *openssl req* le indica al motor de OpenSSL que deseamos realizar un requerimiento para la generación de un nuevo (-new) par de claves y una solicitud de certificado. La clave privada deberá ponerla en el archivo *llaveprivada.pem* y la solicitud en *solicitud.pem*. Los parámetros nosotros los damos y pueden variar, ya que podemos realizar una solicitud en formato PEM (por defecto) o bien una en formato PKCS# 10.

Es importante recalcar que lo anterior se realiza para cada solicitud de certificado que deseemos generar. Igualmente conviene asegurarse de que solo el administrador de la AC tiene acceso a este servicio así como al conocimiento del password que protege a la clave privada de la AC.

El punto más importante aquí a considerar es el archivo de configuración que se está utilizando (indicado por *–config openssl.cnf*), ya que es éste quien le dice a OpenSSL los parámetros y las funciones que tendrá el futuro certificado. Es conveniente contar con un archivo de configuración específica para cada tipo de certificado que se desee emitir, nosotros contamos con tres: ssl-web-server.cnf , openssl-ca.cnf y openssl.cnf; para servidor web, para AC y para usuario final respectivamente (Nosotros realizamos varias solicitudes en ambos formatos, y si todos los parámetros son incluidos correctamente, el procedimiento de generación del par de claves y la solicitud deben completarse sin contratiempos (Ver Anexo B-1).

Paso 2- El segundo paso de la certificación es firmar el requerimiento con la clave privada de la AC y emitir el certificado correspondiente, dejándolo a este y a su clave privada disponibles para su entrega al propietario.
Esto lo realizamos desde la línea de comandos con ayuda de OpenSSL de la siguiente manera:

/usr/local/ssl/bin/]# openssl ca -policy policy_anything -out certificado.pem –config /usr/local/ssl/openssl.cnf -infiles solicitud.pem

Con lo anterior le decimos a OpenSSL que haga uso de nuestra AC y que firme el requerimiento con la clave privada de la AC tomando como entrada la solicitud de certificación llamada *solicitud.pem* aplicando las políticas dadas en *–policy policy_anything,* y que el certificado resultante lo guarde en *certificado.pem* utilizando para ello la configuración dada en *openssl.cnf.*

Al igual que para el requerimiento, para la emisión del certificado tenemos varias opciones, las cuales incluyen el generar certificados X509 y certificados en formato PKCS#12.

Igualmente las políticas de certificación que se apliquen pueden cambiar dependiendo de la política seleccionada. Una referencia más completa de todo lo mencionado aquí se encuentra en la documentación de OpenSSL la cual recomendamos leer y tener a la mano. Notemos que aquí aparece nuevamente un archivo de configuración al cual se hace referencia. Las opciones y recomendaciones para ello se aplican de igual forma que para el proceso de solicitud de certificación descrito en el paso 1.

ANEXO D-2
Pruebas al módulo LDAP

Para probar el funcionamiento del servidor LDAP, primero creamos el directorio /hdc1/ldapif y en él los archivos ldif de prueba desde cualquier editor de texto. El primero con el que probamos es con IMP.ldif el cual se muestra a continuación:

```
  hdc1/ldapif/IMP.ldif :

# Organizacion para el nivel mas alto del directorio LDAP-IMP
dn: o=IMP,c=MX
o: IMP
objectclass: top
objectclass: organization
description: Servidor LDAP Instituto Mexicano del Petroleo

# Organizacion de las tres ramas principales del arbol LDAP-IMP
dn: ou=Seguridad Informatica,o=IMP,c=MX
ou: Seguridad Informatica
objectclass: organizationalUnit

dn: ou=Autoridad Certificadora,o=IMP,c=MX
ou: Autoridad Certificadora
objectclass: organizationalUnit

dn: ou=Personal, o=IMP,c=MX
ou: Personal
objectclass: organizationalUnit
```

Como ya lo dijimos, este archivo tiene un formato y un contenido definido previamente en los RFC's correspondientes. En esta parte indicaremos a grandes rasgos que es lo que estamos haciendo.

En el primer párrafo definimos el nodo raíz del árbol, esto es, el dn para la entrada de nivel superior. Es muy necesario definirlo.

En este mismo párrafo definimos la organización (o) cuyo valor es IMP, y la clase de objeto del que se trata, en este caso es top (cima).

Finalmente definimos el tipo de objeto, que en este caso es una organización (organization).

En el siguiente párrafo definimos un dn para el grupo de usuarios de Seguridad Informática (Esta es la rama hecha para los usuarios que pertenezcan a esta unidad de organización). Como esta rama, podemos tener todas las que queramos para otros propósitos. Simplemente creamos la estructura necesaria y luego la "colgamos" al árbol del directorio. En este mismo párrafo definimos explícitamente el atributo ou (unidad de organización) para Seguridad Informática. Este tipo de atributos sirven como "items buscables". Por ejemplo, si quisiéramos encontrar todos los usuarios que pertenecen a Seguridad Informática podemos buscarlos mediante "show all dn where ou=Seguridad Informatica". Si no definimos esto aquí esta entrada no se encontraría.

A continuación se define la clase de objeto (objectclass), organizationalUnit

El siguiente párrafo definimos un dn para la Autoridad Certificadora. De este se desprenderán los datos correspondientes a la AC. Al igual que para el párrafo anterior, definimos la unidad organizacional Autoridad Certificadora y la clase de objeto organizationalUnit

Finalmente el último párrafo define un dn para en nodo Personal e igualmente una unidad organizacional llamada Personal y la clase de objeto que es.

Notemos que los nodos entrada anteriores se encuentran al mismo nivel y los tres están "colgados" de la raíz o=IMP. Lo anterior por razones de organización.

Una vez que tenemos un archivo ldif de entrada de datos, tenemos dos opciones para introducirlos a LDAP. La primera de ellas es "on-line" utilizando el comando *ldapadd* y la otra es "off-line" utilizando el comando *slapadd* .

Lógicamente, para poder utilizar el modo "on-line" es necesario que el servicio de LDAP se encuentre corriendo y disponible.

Utilizaremos IMP.ldif para crear una entrada en el directorio en el modo "on-line" , para esto desde la línea e comandos tecleamos lo siguiente:

```
/usr/local/bin]# ./ldapadd -f /hdc1/ldapif/IMP.ldif
-x -D  "cn=Manager, o=IMP, c=MX" -w secretimp
```

Con esto queremos decir que :

-x → Utilizar autenticación simple en lugar de SASL
-D binddn → utiliza el nombre distinguido binddn para ligar con el directorio LDAP
-w password → utiliza *password* como autenticación simple para poder escribir datos al directorio.
-f file → Lee la entrada de modificación de información de *file* en lugar de la entrada estándar. Con entrada estándar quiere decir que en la anterior línea de comandos deberíamos incluir toda la rama de la entrada en cuestión, lo cual puede ser confuso, y provocar muchos errores.

Lo anterior está ligado estrechamente con los archivos de configuración de LDAP ldap.conf y slapd.conf (Ver Anexo C-1).

El servidor escribirá las entradas en el directorio y nos enviará los siguientes mensajes:

```
adding new entry "o=IMP,c=MX"
adding new entry "ou=Seguridd Informática, o=IMP, c=MX"
adding new entry "ou=Autoridad Certificadora, o=IMP, c=MX"
adding new entry "ou=Personal, o=IMP, c=MX"
```

Con esto garantizamos que las entradas fueron hechas exitosamente

Lo anterior es, en resumidas cuentas, el procedimiento de prueba para LDAP-IMP

En esta parte encontramos varias dificultades en el funcionamiento del servidor y en la manera de introducir y consultar los datos en el directorio. Tales problemas se debieron principalmente a configuraciones erróneas del servicio (archivos ldap.conf y slapd.conf), este es quizá el motivo más común de los errores que pueda uno experimentar cuando se levanta un servicio de directorio.

Este y otro tipo de consideraciones, problemas y soluciones, se encuentran en la documentación de OpenLDAP. Conviene leerla y tenerla cerca para cualquier referencia futura.

Realizamos algunas otras pruebas introduciendo información al directorio pero con nuevos tipos de atributos y clases de objetos. Lo que se nos complicó mucho fue el tratar de incluir el certificado digital en una entrada dada. Esto aún presenta algunas fallas pero en conclusión podemos decir que el servicio de directorio en esencia trabaja y funciona como debería hacerlo.

ANEXO D-3 **PROYECTO PKI-IMP**

MANUAL DE PROCEDIMIENTO DE USUARIOS PKI-IMP

V0.5 Noviembre 2003

INTRODUCCIÓN

Este documento tiene como propósito servirle de guía durante el proceso de obtención de un certificado digital del servicio de PKI-IMP, desde realizar la solicitud de certificación hasta descargarlo del servidor e instalarlo en su computadora. Se incluye una guía de la configuración y uso del Servicio de Directorio LDAP-IMP

Si tiene alguna duda o comentario respecto a este manual o al procedimiento, por favor acuda al área de Seguridad Informática del IMP.

Responsable del Proyecto PKI-IMP: Uriel Tirado Ríos. **utirado@imp.mx**
Administrador: Carlos Roberto Zainos H. **zainos_hcr@correo.unam.mx**

1- INICIO

El servicio de PKI-IMP utiliza una interfaz web para interactuar con sus usuarios. Para accesar a este servicio debe abrir su navegador web (Iexplorer, Nescape, Mozilla etc.) y escribir la siguiente dirección:

https://raguel.imp.mx/pub

WIN95, WIN98, WIN2000, WIN XP IEXPLORER 5.0, 6.0

Esta dirección lo llevará directamente a la página de inicio del Servidor Público PKI-IMP. Cabe hacer mención que su navegador le enviará un mensaje de alerta (Imagen 1.0) ya que estará accesando a un sitio seguro (https), el navegador le desplegará información acerca del certificado presentado por el sitio, así mismo le preguntara si desea continuar, de click en Sí.

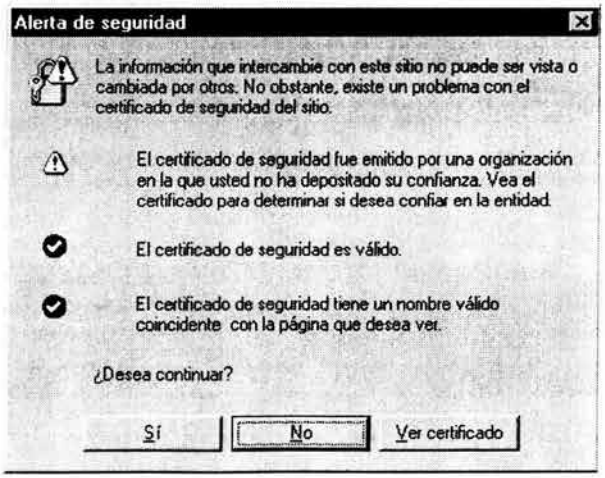


Imagen 1.0

1.1 - A continuación el servidor lo llevará a la página principal, ahí se muestran todas las opciones de lo que el usuario puede realizar (Imagen 1.1).

Existe una liga de **Introducción** la cual contiene Información importante sobre el proyecto PKI-IMP y que se recomienda leer para comprender mejor lo que esta haciendo así como las ventajas y utilidades que brinda esta infraestructura.

A continuación se presenta la página de inicio:

En la parte inferior derecha aparece un candado el cual indica que es una conexión segura, si da doble click sobre él obtendrá información mas detallada acerca del certificado utilizado por el sitio.

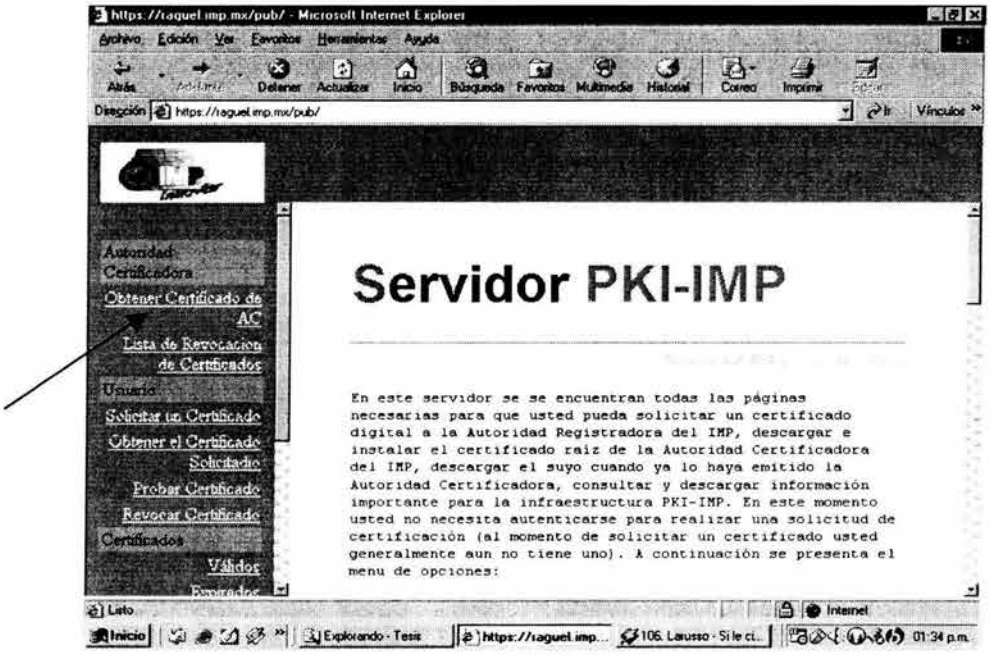


Imagen 1.1

1.2 - El primer paso es obtener el certificado raíz de la Autoridad Certificadora, e importarlo en el navegador. Para esto proceda como a continuación se indica:

Dé click en la liga **"Obtener Certificado de AC"** que aparece en la parte superior izquierda de su navegador dentro de la sección llamada "Autoridad Certificadora" o bien dentro del menú principal en la liga `Obtener certificado de la AC-IMP`

Es necesario realizar este paso previo a realizar la solicitud de certificación a fin de evitar los mensajes de "Alerta de Seguridad" enviados por su explorador al accesar a otras páginas dentro del directorio y al momento de generar y enviar la solicitud de certificación.

El servidor lo llevará a la página siguiente (Imagen 1.2):

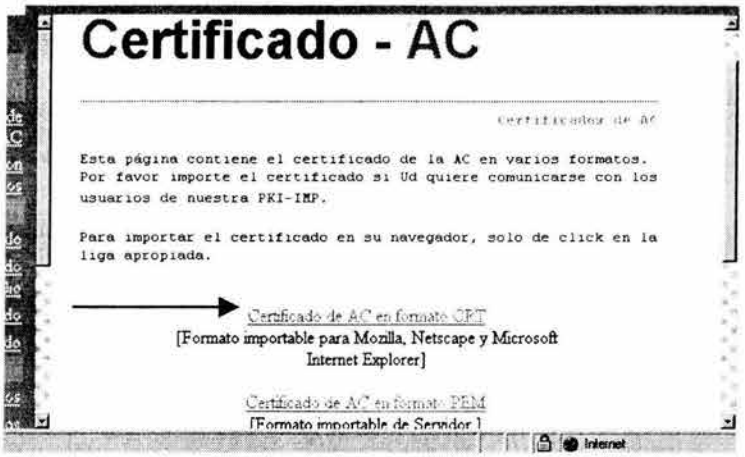


Imagen 1.2

Se le presentan varias opciones, de click en la liga **Certificado de AC en formato CRT**

WIN95 y WIN98
A continuación se le presentará un mensaje de descarga de archivos, elija **Abrir**

WIN2000 y WIN XP
Windows le mostrará la siguiente ventana (Imagen 1.3):

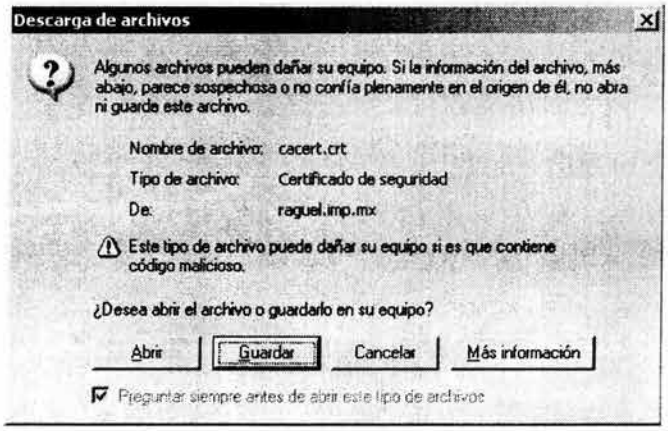


Imagen 1.3

Igual que en el caso anterior elija **Abrir**

Windows le mostrará los datos del certificado de la AC-IMP en una ventana como la siguiente (Imagen 1.4):

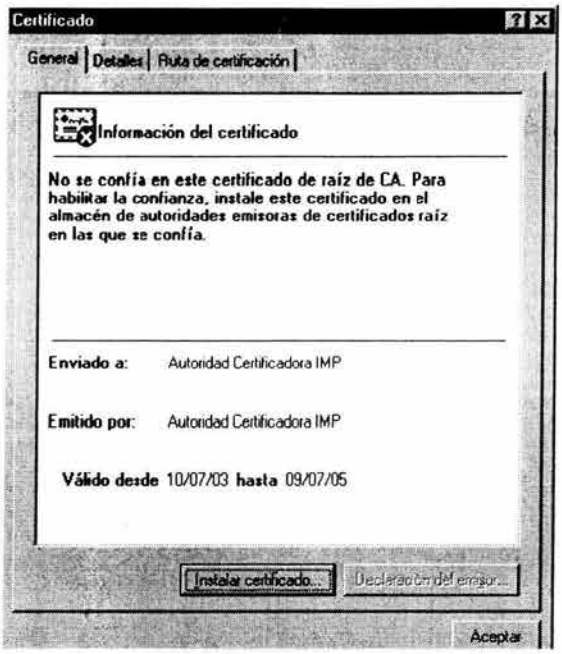


Imagen 1.4

De click en **Instalar certificado...**

Se abrirá el asistente de Importación de certificados siga los pasos que se le indican.

Le preguntará el asisten donde quiere colocar el certificado, seleccione la opción que le presenta automáticamente **"Seleccionar Automáticamente el almacén de certificados en base al tipo".**
De click en siguiente.

El asistente desplegará un mensaje de éxito, de click en **Finalizar** para continuar con el proceso.

A continuación se le preguntará si desea agregar el certificado al almacén de certificados de raíz.
De click en SI, de lo contrario la ruta de certificación no podrá validarse y tendrá problemas al intentar comunicarse con otros usuarios dentro de la infraestructura PKI-IMP (Imagen 1.5).

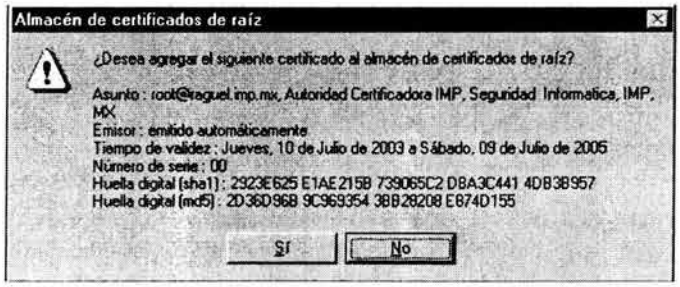


Imagen 1.5

Windows desplegará un mensaje final de "Instalación completada"

WIN95 WIN98

Si lo desea puede verificar que el certificado se encuentre realmente instalado en el navegador. Para esto en la ventana del explorador vaya al menú de **Herramientas → Opciones de Internet →**

Contenido → Certificados → Autoridades Emisoras de Certificados raíz en las que se Confía y ahí deberá encontrar el certificado de la AC-IMP (Imagen 1.6).

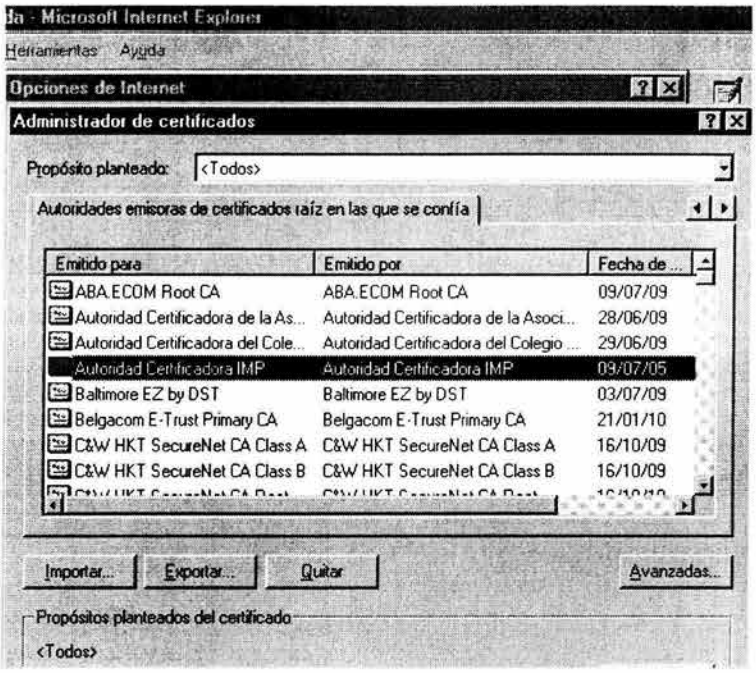


Imagen 1.6

Si da click en el botón "**Ver**" verá los detalles del certificado en cuestión.

WIN2000 y WIN XP

Puede verificar que el certificado se encuentre realmente instalado en el navegador. Para esto en la ventana del explorador vaya al menú de **Herramientas → Opciones de Internet → Contenido → Certificados → Autoridades Emisoras Raíz de Confianza** y ahí deberá encontrar el certificado de la AC-IMP.

Si no puede ver el certificado de la AC instalado en este lugar, lo más probable es que lo haya colocado en algún otro sitio o que haya ocurrido un error en el proceso de instalación del certificado raíz. Si este fuese el caso regrese a la página principal de PKI-IMP y vuelva a descargar el certificado de la AC.

Si da click en el botón "**Ver**" verá los detalles del certificado en cuestión.

Si no existe algún otro problema o complicación siga adelante con el procedimiento que le permitirá solicitar y recuperar un certificado digital personal.

2- SOLICITANDO UN CERTIFICADO

Vaya a la página principal del servicio PKI-IMP ubicada en https://raguel.imp.mx/pub.

De click en la liga **Solicitar un Certificado** que se encuentra en la sección **Administración de Certificados** del menú principal ó en la sección de **Usuario** en la barra de navegación que se encuentra a la izquierda de su pantalla.

En esta página se presentan varias opciones dependiendo del navegador que este utilizando. Si tiene dudas o no sabe que hacer hay una liga que detecta automáticamente el tipo de navegador que esta utilizando, recomendamos utilizar esta liga a fin de evitar complicaciones.

Lo mencionado anteriormente lo deberá usted ver de la siguiente manera (Imagen 2.0):

Imagen 2.0

USUARIOS DE IEXPLORER (WIN95 Y WIN98)

A continuación se le presentará una página que le solicitará datos suyos a fin de poder conformar la solicitud y enviar esta a la Autoridad Registradora AR-IMP para su aprobación y que el proceso de certificación pueda continuar.

La Autoridad Registradora AR-IMP es un elemento importante dentro de la infraestructura PKI-IMP ya que es ella quien interactúa directamente con los usuarios y además es la única que tiene contacto directo con la AC-IMP quien es quien firma su certificado digital. Una vez emitido este, la AR-IMP es quien le notificará a usted que su certificado se encuentra listo y le dirá el procedimiento que debe seguir para recuperarlo.

Llene la forma con sus datos y corrobórelos (Imagen 2.1):

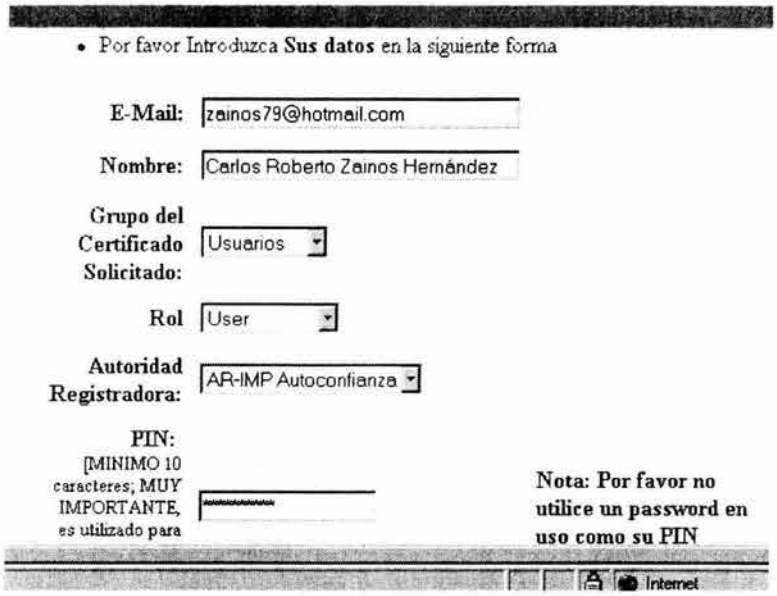


Imagen 2.1

Es importante recalcar en esta parte que todos los datos deben de estar correctamente escritos, ya que todos son utilizados como componentes importantes de su certificado digital. En la parte de Nombre omita los acentos.

En el menú de **"Grupo del Certificado Solicitado"** usted deberá seleccionar **"Usuarios"** ya que esta información se utiliza para organizar su registro en el servicio de directorio LDAP (Fines Administrativos).

De igual manera, en el menú de **"Rol"** deberá seleccionar **"User"**, esta información es muy importante ya que si elige algún otro rol su certificado será generado con otro tipo de atributos que no le serán útiles para las operaciones que usted quiere realizar (e.g. Cifrado y Descifrado de datos, Autenticación de Usuario, Correo Seguro etc.).

Observe y siga las recomendaciones que se le hacen en la página.

De click en el botón **"Continuar"**

Si usted esta utilizando una versión muy antigua de Iexplorer (e.g. versión 5.0) se desplegará un mensaje de que esta utilizando una versión con un "hoyo" de seguridad, de click en aceptar y descargue las actualizaciones del explorador. En caso contrario se le desplegará un mensaje de que esta utilizando un explorador con el "parche" de seguridad adecuado. De igual manera de click en aceptar.

A continuación el servidor lo llevará a una página la cual contiene los datos proporcionados por usted (Imagen 2.2).

Observe y siga las recomendaciones.

Imagen 2.2

Se le presentará un menú llamado **"Dispositivo Criptográfico"** el cual se refiere al tipo de dispositivo soportado por Windows para realizar las operaciones criptográficas que se requieran.

Deje la opción que se selecciona automáticamente (default).

De click en el botón **"Continuar"**

A continuación se le mostrará una ventana con información importante para usted (Imagen 2.3).

Se trata de su DN o Nombre Distinguido (Distinguished Name). Su DN es importante ya que este funciona como su "dirección" dentro del Servicio LDAP (Servicio de Directorio). Le es útil cuando alguien requiere descargar su certificado para establecer una comunicación segura, si ese alguien conoce su DN puede buscarlo rápidamente en el servicio de directorio y descargar de ahí su certificado o bien realizar una búsqueda basada en DN en el servidor PKI-IMP. Es conveniente tenerlo a la mano para futuras referencias.

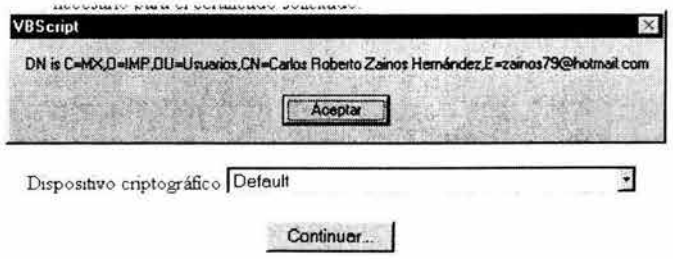


Imagen 2.3

Windows le desplegará el mensaje de que el sitio estará solicitando un certificado en su nombre (Imagen 2.4).

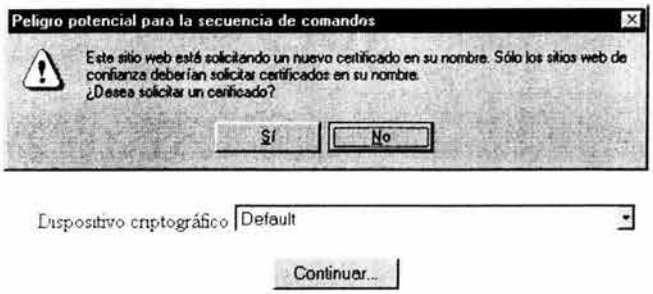


Imagen 2.4

De click en Sí

A continuación aparecerá lo siguiente (Imagen 2.5):

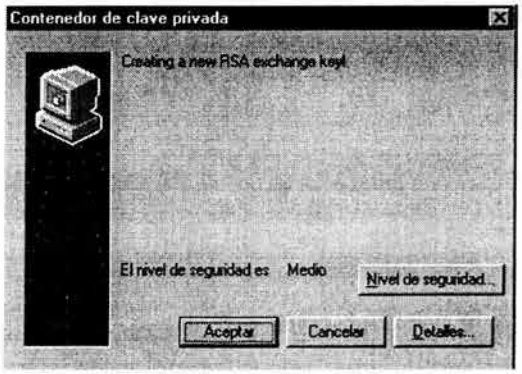


Imagen 2.5

Aquí debe de dar click en la opción "Nivel de Seguridad.." y cambiarlo a **ALTO** (Imagen 2.6)

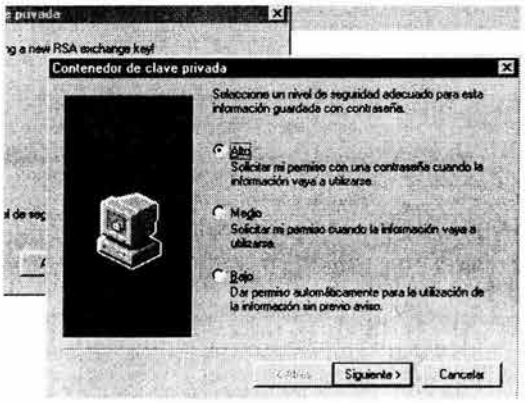


Imagen 2.6

WIN95, WIN98 y WIN2000

Windows le pedirá un nombre para la llave que se está generando y un password para proteger dicha llave y que solo usted pueda utilizarla.

WIN XP

En este caso Windows no le pedirá un nombre, simplemente le pedirá el password ya que windows automáticamente le asignará uno. Vea el nombre asignado y recuérdelo para futuras referencias.

Es responsabilidad de usted resguardar este password en un lugar seguro ya que si lo llega a olvidar o a perder no podrá utilizar su llave para firmar digitalmente y si alguien mas sabe dicho password y tiene acceso a su llave privada puede firmar documentos digitalmente en su nombre.

Como WIN XP no le solicita un nombre, la siguiente pantalla diferirá un poco de esta que fue el resultado utilizando WIN98, si es que está utilizando WIN XP (Imagen 2.6).

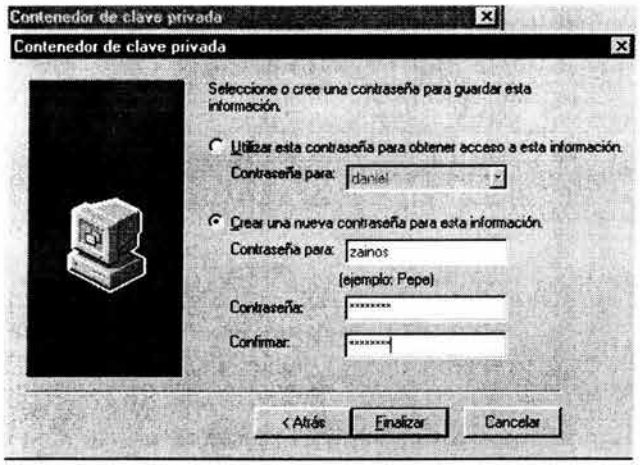


Imagen 2.6

De click en Finalizar

El proceso continúa y le muestra lo siguiente (WIN95 y WIN98) (Imagen 2.7):

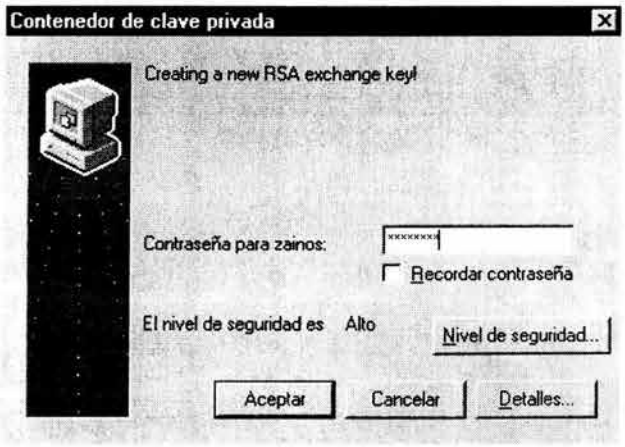


Imagen 2.7

Introduzca el password para la llave que se generará y de click en Aceptar

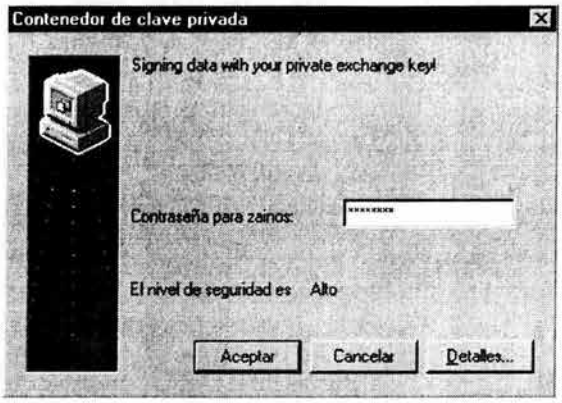


Imagen 2.8

Windows le envía el mensaje que se muestra en la Imagen 2.8, el cual le pide el password que protege a su clave privada para poder utilizar esta, introdúzcalo y de click en Aceptar.

Un script le enviará un mensaje de que la solicitud de certificado se generó con éxito, de click en Aceptar (Imagen 2.9).

En Windows XP usted no podrá ver las dos imágenes anteriores.

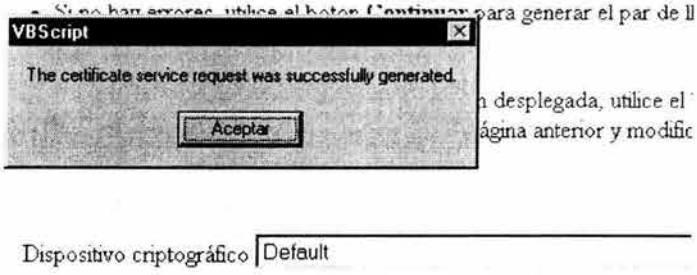


Imagen 2.9

A continuación el servidor lo llevará a una página informativa, similar a la que se muestra a continuación (Imagen 2.10), en la cual se le indica que su solicitud fue completada y del estado de su solicitud.

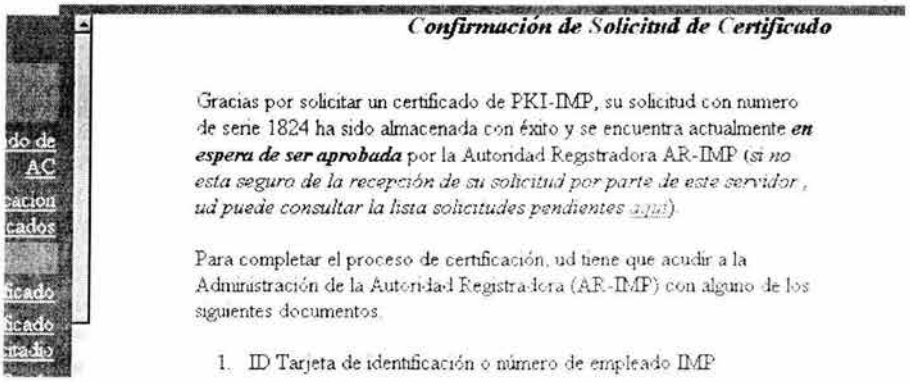


Imagen 2.10

Así mismo le presenta una liga en la cual usted puede consultar la lista de solicitudes pendientes, si usted puede ver su solicitud en dicha página entonces quiere decir que no hubo ningún contratiempo en el proceso. Si no fuese así entonces algún error debió ocurrir durante el proceso y lo más recomendable es volver a repetir desde el principio.

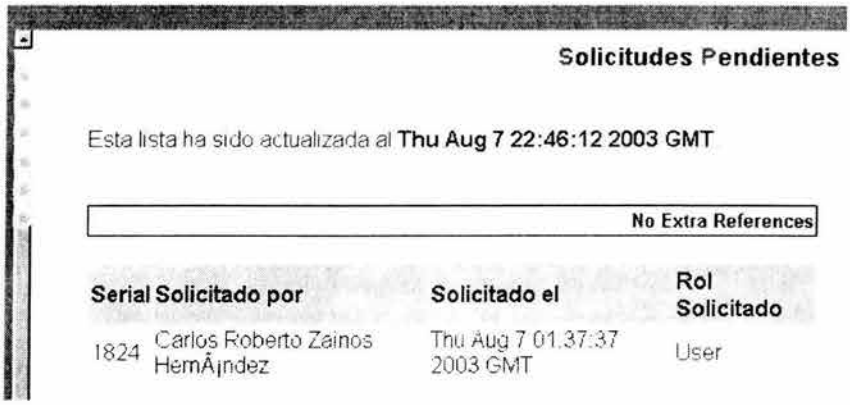


Imagen 2.11

Siga las instrucciones que se le dieron y acuda a la Autoridad Registradora lo más pronto posible a fin de que el proceso continúe y usted pueda descargar el certificado solicitado de este mismo servidor.

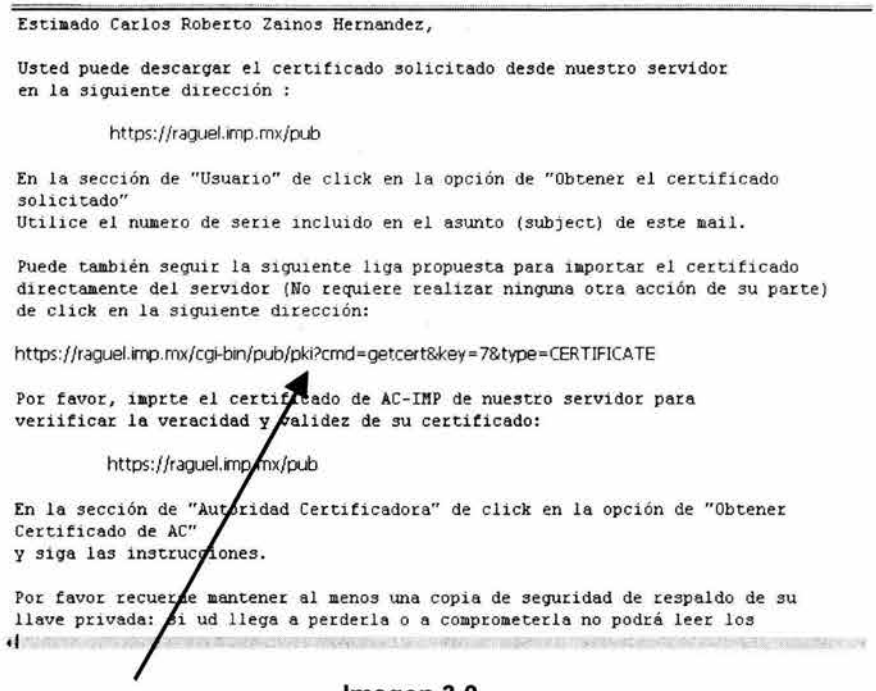
No olvide presentar alguno de los documentos que se listaron en la página, esto sirve para validar su identidad ante la PKI-IMP y los usuarios de la misma.

3- DESCARGANDO SU CERTIFICADO

Cuando su certificado se encuentre listo, la AR-IMP se lo notificará vía e-mail.
En este e-mail se le indicará el proceso que debe de seguir para descargarlo e importarlo en el navegador. La manera más fácil es seguir la liga que se le proporciona en el e-mail.

Cabe recalcar que en el Asunto (Subject) del mensaje se incluye un número de serie. Fíjese bien en ese número ya que si decide descargarlo desde la página que para ese fin se encuentra en el servidor, se le solicitará que escriba ese número.

El e-mail que usted recibe en mas o menos como el siguiente (Imagen 3.0):


Estimado Carlos Roberto Zainos Hernandez,

Usted puede descargar el certificado solicitado desde nuestro servidor
en la siguiente dirección :

https://raguel.imp.mx/pub

En la sección de "Usuario" de click en la opción de "Obtener el certificado
solicitado"
Utilice el numero de serie incluido en el asunto (subject) de este mail.

Puede también seguir la siguiente liga propuesta para importar el certificado
directamente del servidor (No requiere realizar ninguna otra acción de su parte)
de click en la siguiente dirección:

https://raguel.imp.mx/cgi-bin/pub/pki?cmd=getcert&key=7&type=CERTIFICATE

Por favor, imprte el certificado de AC-IMP de nuestro servidor para
veriificar la veracidad y validez de su certificado:

https://raguel.imp.mx/pub

En la sección de "Autoridad Certificadora" de click en la opción de "Obtener
Certificado de AC"
y siga las instrucciones.

Por favor recuerde mantener al menos una copia de seguridad de respaldo de su
llave privada: Si ud llega a perderla o a comprometerla no podrá leer los

Imagen 3.0

De click en la liga sugerida.

Windows le enviará el siguiente mensaje (Imagen 3.1):

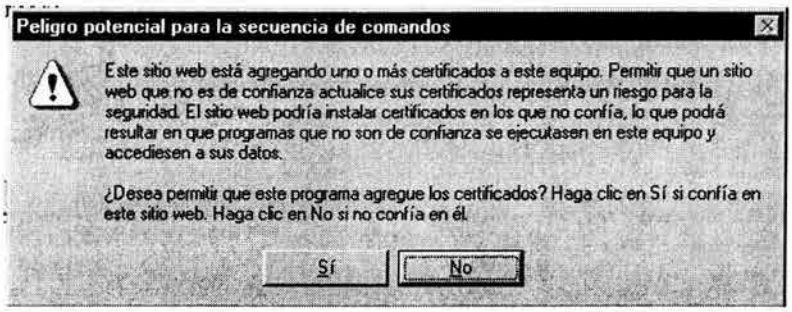


Imagen 3.1

De click en Sí

Windows Importará automáticamente el certificado digital junto con la firma en su almacén de certificados. Puede verificar que esto realmente se llevó a cabo dando click en el menú de **Herramientas → Opciones de Internet → Contenido → Certificados** en el submenú **Personal** deberá ver su certificado instalado (Imagen 3.2).

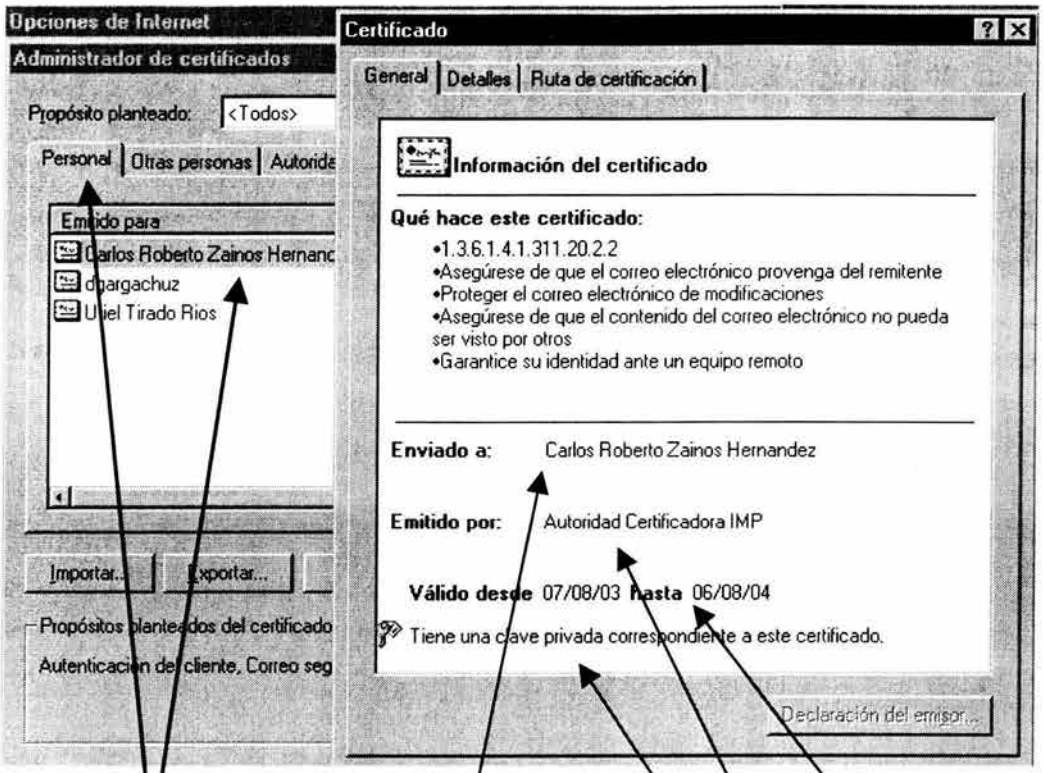


Imagen 3.2

Submenú Personal, **Certificado Solicitado** Instalado en el Navegador y guardado en el Administrador de certificados

Propietario del Certificado

Emisor

Periodo de Validez

NOTA: El administrador de certificados igualmente le hace una anotación importante: **"Tiene una clave privada correspondiente a este certificado"** esto quiere decir que usted puede utilizar ambas herramientas para intercambiar información con los demás usuarios de la PKI-IMP garantizando con esto la **Autenticación, Integridad, Confidencialidad y el No repudio** de los datos intercambiados. Cada vez que usted quiera utilizar esta llave privada, se le pedirá el password que protege a la misma, garantizando con esto que usted y solo usted podrán utilizar dicha llave, ya que nadie más que usted conoce el password.

En el documento **"CORREO SEGURO"** conocerá un ejemplo práctico de como utilizar estas herramientas criptográficas para el intercambio de información de manera segura.

La siguiente sección se refiere a la configuración y uso de otro módulo de PKI-IMP que es el Servicio de Directorio LDAP-IMP.

4- SERVICIO DE DIRECTORIO LDAP-IMP

Una vez que usted cuenta con un certificado digital y una clave privada, usted puede intercambiar información confidencial con los demás usuarios de la infraestructura. Para esto y dependiendo de lo que desee hacer, es posible que usted requiera obtener el certificado digital del o los usuarios con los que desee intercambiar información.
Una manera para realizar esto es por medio del servicio de directorio LDAP-IMP que se encuentra a su disposición. En él usted encontrará los certificados digitales de los demás usuarios de PKI-IMP. LDAP-IMP tiene ventajas en cuanto a su uso respecto a descargar el certificado deseado "en línea" , la principal de ellas es que usted lo puede consultar desde la libreta de direcciones de Microsoft o desde el cliente de correo de LDAP.

En esta sección se detallan los pasos a seguir para configurar y utilizar este servicio en WIN95,WIN98 y WIN2000.

LIBRETA DE DIRECCIONES DE MICROSOFT

Hay varias aplicaciones que utilizan la libreta de direcciones de Microsoft, las principales por la frecuencia de su uso son los clientes de correo Outlook y Outlook Express.

Para configurar la libreta de direcciones de manera que pueda esta consultar información en el Servicio de Directorio LDAP-IMP proceda como se indica a continuación:

1- Abra la libreta de Direcciones

Vaya al menú de **Inicio→Programas→Accesorios** y de click en el botón correspondiente.

2- En la libreta de direcciones vaya al menú de **Herramientas** y a continuación seleccione la opción **Cuentas**. Windows le mostrará una ventana llamada **Cuentas de Internet** con una pestaña titulada **Servicio de Directorio**, en esta ventana de click en el botón **Agregar...** , esto iniciará un asistente para la configuración del servicio.

Usted verá una pantalla como la siguiente (Imagen 4.1):

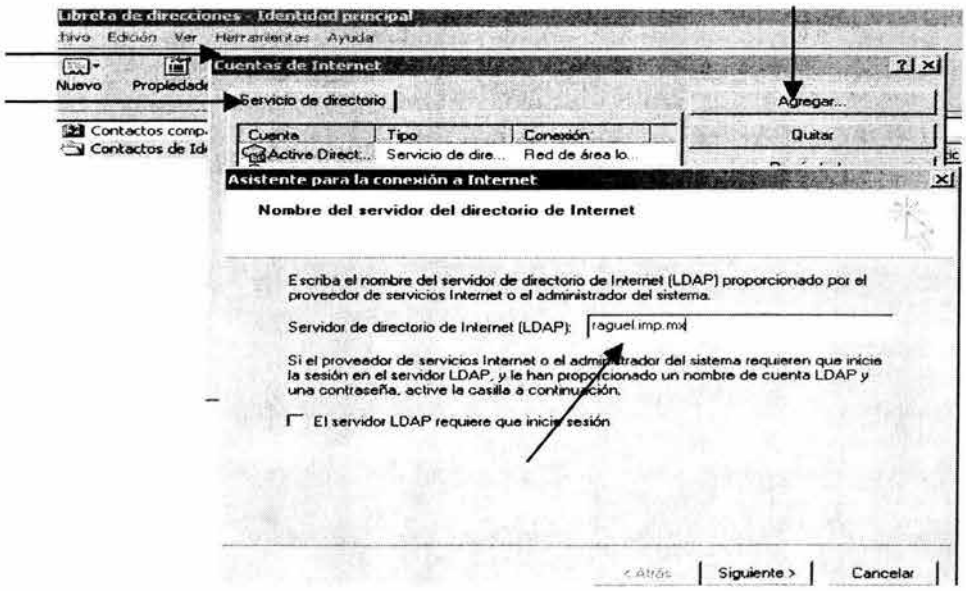


Imagen 4.1

En ella se le pedirá que escriba el nombre del servidor de directorio de Internet LDAP proporcionado por el administrador del sistema.

Usted deberá teclear el nombre **raguel.imp.mx** como lo muestra la Imagen 4.1

De click en siguiente hasta completar el procedimiento dejando los valores que por defecto le aparecen. Cuando se cierre el asistente usted deberá haber creado una cuenta nueva y a continuación ver una pantalla como la siguiente (Imagen 4.2):

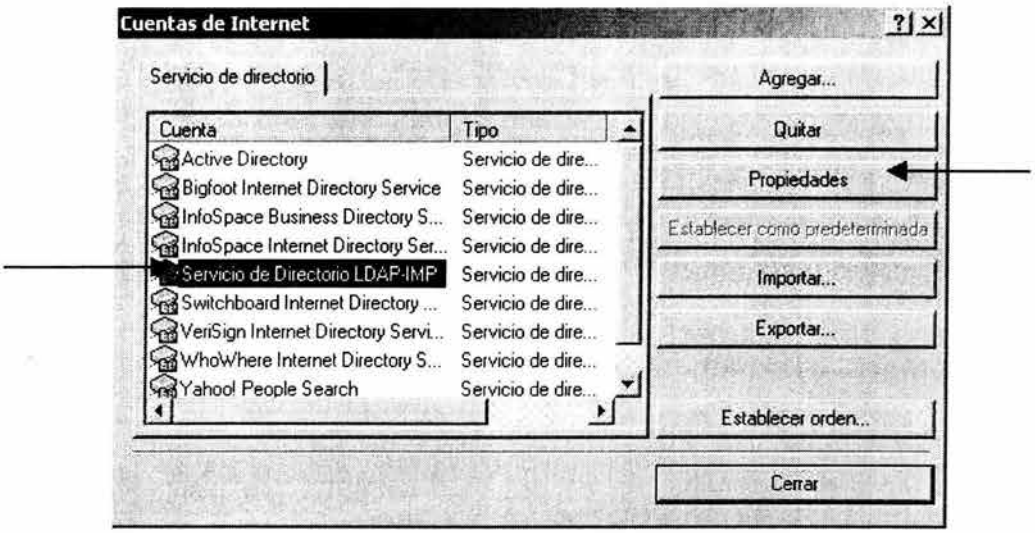


Imagen 4.2

3- Seleccione la cuenta y de click en el botón de **Propiedades**

Windows le mostrará una ventana con dos pestañas como las siguientes (Imagen 4.3 e Imagen 4.4):

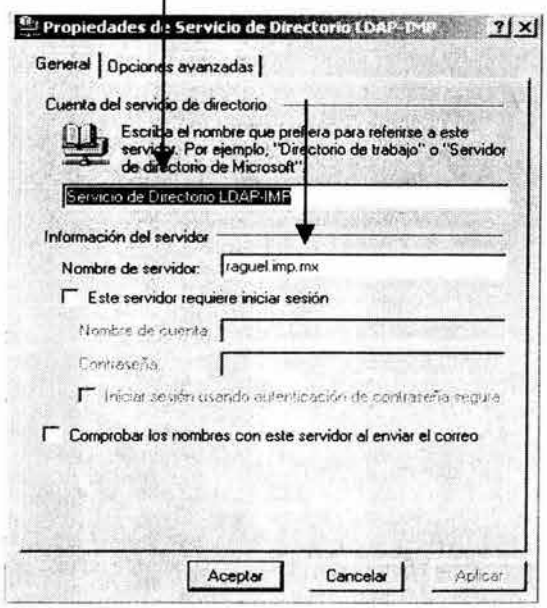


Imagen 4.3

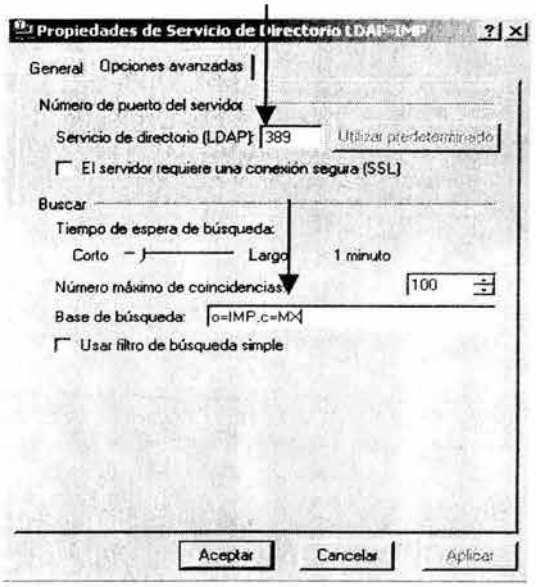


Imagen 4.4

En la Imagen 4.3 se muestran las opciones **Generales** de la cuenta, en ella usted deberá escribir un nombre descriptivo para el servicio así como el nombre del servidor.

En la Imagen 4.4 se muestran las **Opciones Avanzadas** de la cuenta, en ella usted deberá escribir el número de puerto por el cual se comunicará al servicio, escriba **389**. Igualmente deberá escribir una **Base de búsqueda**, esta se refiere a la raíz del directorio que estará consultando, escriba los valores "o=IMP,c=MX".

De click en **Aceptar** .

En la ventana de **Cuentas de Internet** de click en **Cerrar**.

REALIZAR BÚSQUEDAS EN EL DIRECTORIO DESDE LA LIBRETA DE DIRECCIONES

Una vez que se ha configurado correctamente la libreta de direcciones, podemos utilizar esta para realizar búsquedas en el servicio de directorio LDAP-IMP.
La principal utilidad de esto es recuperar el certificado digital del o los participantes en una comunicación segura.

Para esto en la ventana principal de la libreta de direcciones de click en el botón "**Buscar Personas**".

Aparecerá una ventana como la siguiente (Imagen 4.5):

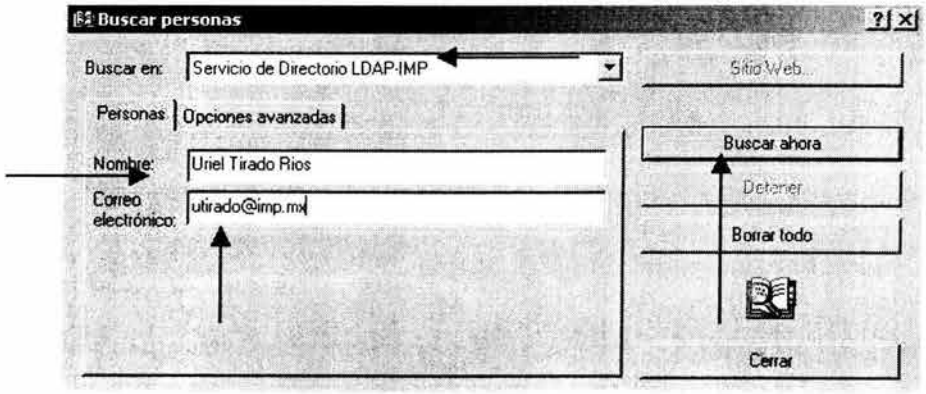


Imagen 4.5

En ella usted puede introducir el nombre de la persona y/o la dirección de correo electrónico.

Asegúrese de que está buscando en el directorio correcto (LDAP-IMP)

Finalmente, para realizar la búsqueda de click en el botón **Buscar ahora.** La búsqueda se efectuará.

Aunque esta es la manera más sencilla de realizar la búsqueda, puede no arrojar resultados útiles, esto debido a que usted deberá saber el nombre completo del usuario y/o su dirección de correo electrónico.

En tal caso se le recomienda seguir el siguiente procedimiento para buscar en todo el directorio LDAP-IMP
A partir de lo desplegado en la Imagen 4.5, vaya a la pestaña **Opciones avanzadas**, en ella usted puede especificar todo un criterio de búsqueda de manera que obtenga resultados que concuerden con alguna entrada en el directorio.

Usted verá algo como lo siguiente (Imagen 4.6):

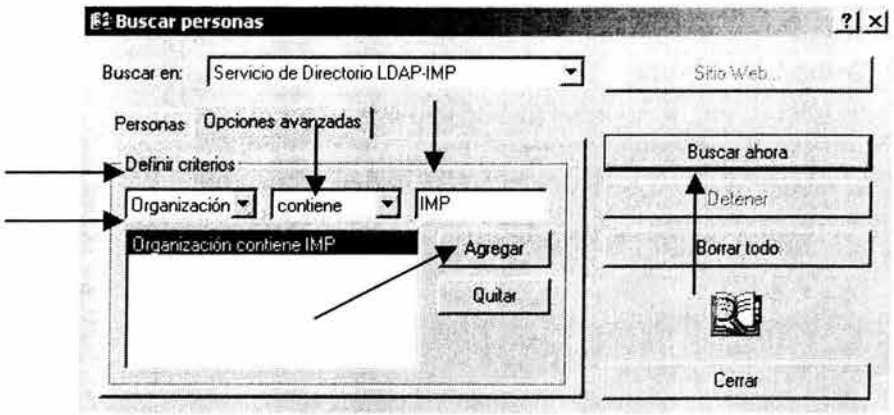


Imagen 4.6

Observe que en la sección Definir criterios usted puede seleccionar entre opciones como *Nombre, Coreo electrónico, Apellidos, Organización* y criterios como *contiene, es, empieza con, termina con, se parece a*; y a continuación indicar la palabra clave en cuestión. Finalmente pulse el botón Agregar para considerar dicho criterio y después pulse el botón Buscar ahora para efectuar la búsqueda.

En el ejemplo anterior la búsqueda se definió de la siguiente manera: "Busca en el servicio de directorio LDAP-IMP todas las entradas cuyo campo de Organización Contiene la palabra clave IMP" . Lo anterior en principio nos mostrará como resultado todos los usuarios dados de alta en el directo LDAP-IMP. Usted puede probar otros criterios de búsqueda.

Lo que se obtiene como resultado es lo siguiente (Imagen 4.7):

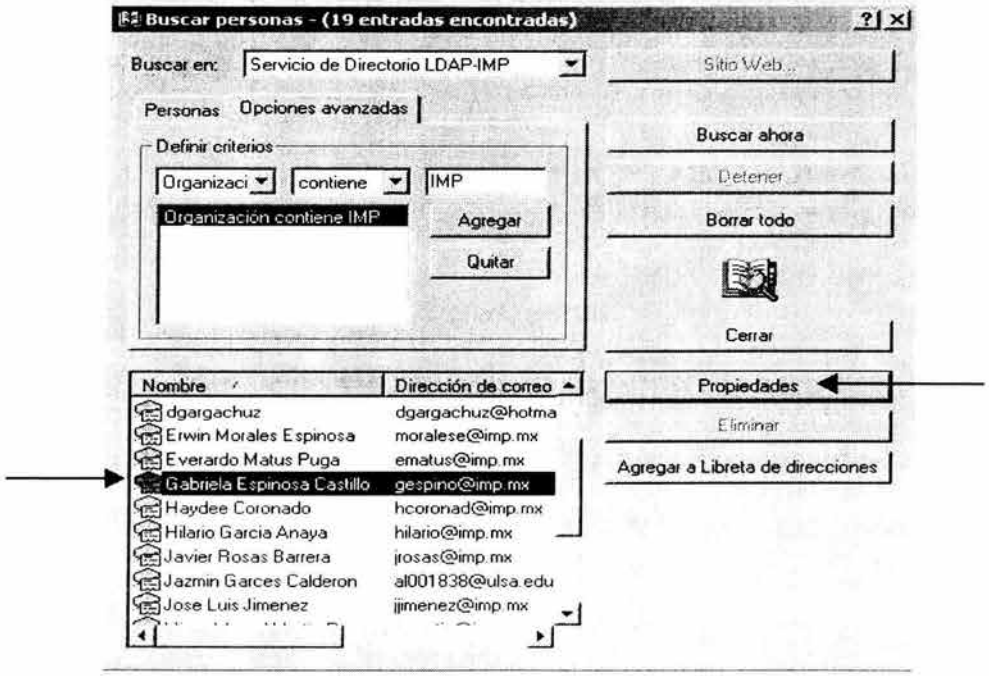


Imagen 4.7

Seleccione al usuario deseado y de click en el botón **Propiedades**

Usted verá una ventana como la siguiente con los datos del usuario (Imagen 4.8) :

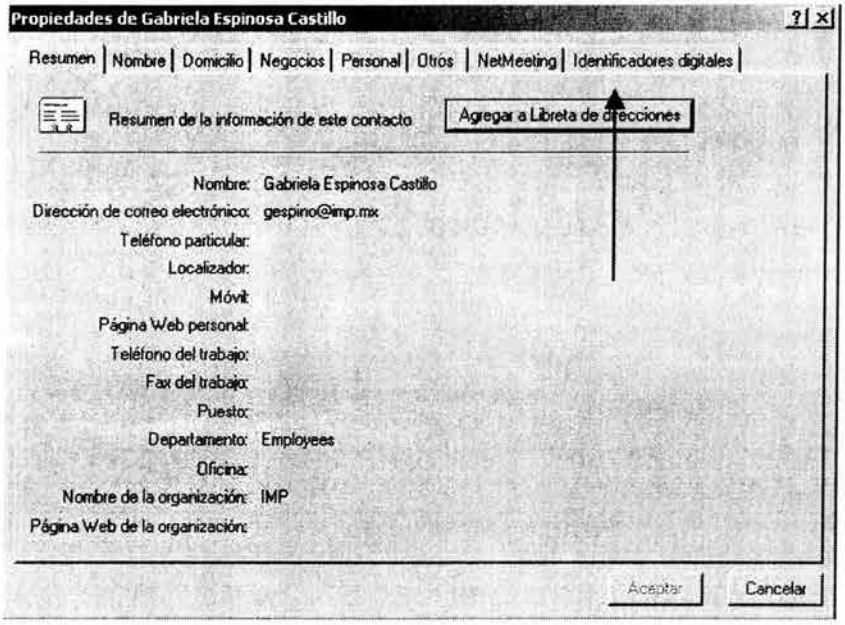


Imagen 4.8

En esta ventana se muestra el resumen de los datos del usuario. La parte importante de esto se encuentra en la etiqueta **Identificadores Digitales** , de click en la etiqueta y verá una ventana como la siguiente (Imagen 4.9):

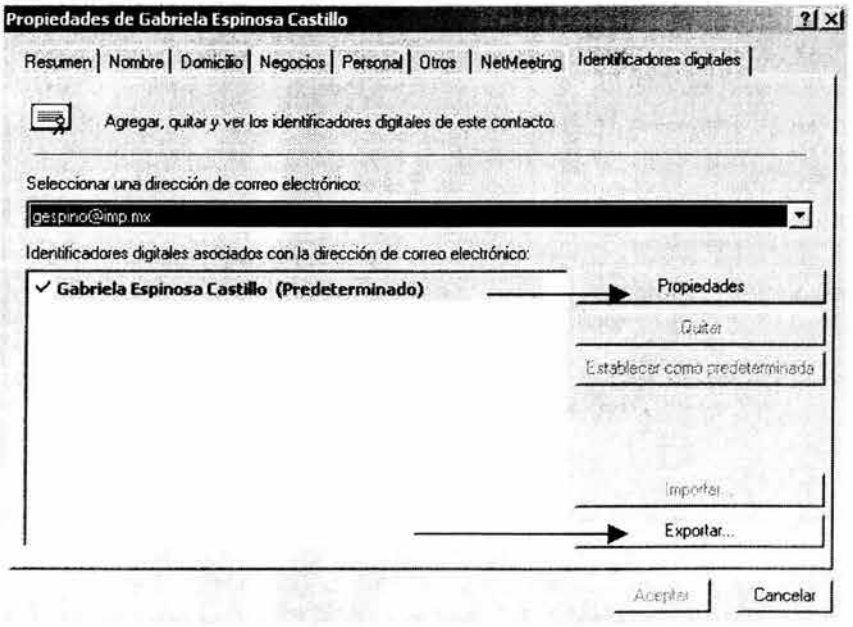


Imagen 4.9

Si el usuario cuenta con un certificado digital, como es este caso, usted podrá encontrarlo aquí.

Si presiona el botón de **Propiedades** podrá ver los detalles del mismo, en este caso lo que deseamos es obtener el certificado en cuestión.

De click en el botón **Exportar** , a continuación verá una ventana como la siguiente (Imagen 4.10):

Imagen 4.10

Especifique un nombre y una ubicación para el certificado

A continuación vaya a la ubicación del certificado y de doble click en el mismo, se abrirá una ventana como la siguiente (Imagen 4.11)

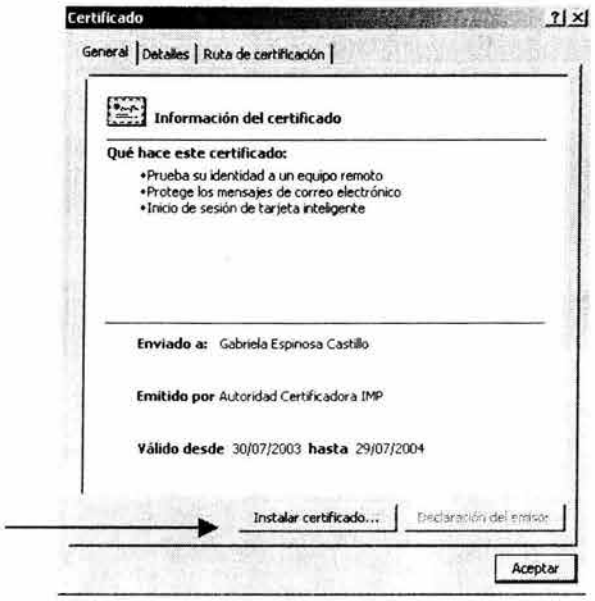


Imagen 4.11

De click en el botón **Instalar certificado.**

Siga las indicaciones del administrador de certificados.

Al final usted deberá tener instalado dicho certificado en el repositorio de certificados de Windows.

LIBRO DE DIRECCIONES DE NETSCAPE

A continuación se mencionan los pasos necesarios para configurar el libro de direcciones de Netscape para poder realizar consultas al servicio de directorio LDAP-IMP.

Abra el libro de direcciones.

Vaya al menú **Archivo→Nuevo→Directorio LDAP** e introduzca la siguiente información (Imagen 4.12) :

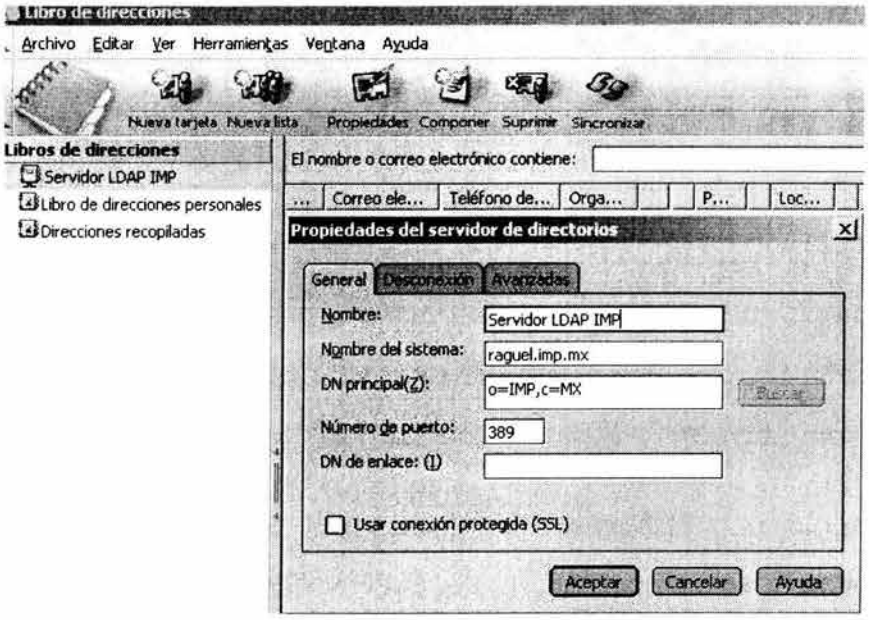


Imagen 4.12

De click en **Aceptar** y listo.

Para realizar una búsqueda simplemente vaya al menú **Herramientas→Buscar** direcciones y verá algo como esto (Imagen 4.13):

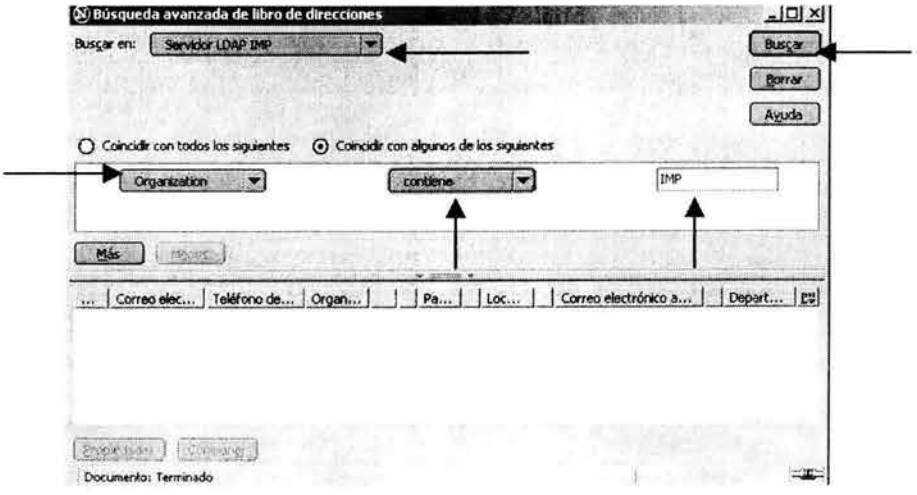


Imagen 4.13

Al igual que para el caso de la Libreta de direcciones de Microsoft, verifique que está buscando en el directorio correcto, defina un criterio de búsqueda con una o más palabras clave en uno o más campos y a continuación pulse el botón **Buscar.**

Cuando la búsqueda finalice usted deberá ver algo parecido a esto (Imagen 4.14):

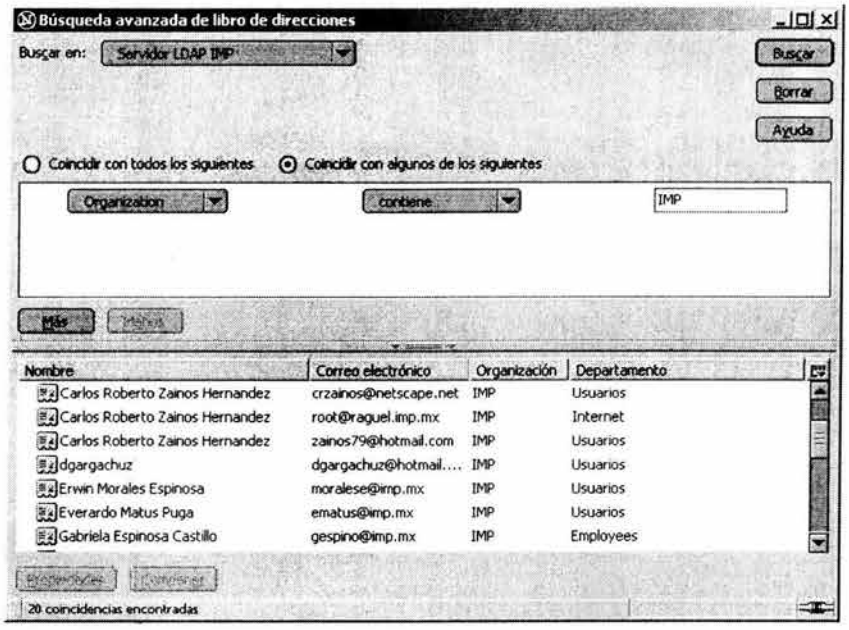


Imagen 4.14

La anterior es una manera alternativa de buscar y recuperar certificados digitales de los usuarios de PKI-IMP. Usted puede descargar los certificados de los usuarios desde la página principal de PKI-IMP, esta modalidad es conocida como "en línea" usted es libre de elegir la que mejor crea conveniente.

No olvide que para cualquier duda o comentario respecto al manual o al proyecto puede dirigirse con el responsable del mismo o con el administrador.

PROYECTO PKI-IMP

URIEL TIRADO RIOS
RESPONSABLE DEL PROYECTO

CARLOS ROBERTO ZAINOS H
ADMINISTRADOR

ANEXO D-4 PROYECTO PKI IMP

CORREO SEGURO

V0.5 Noviembre 2003

1- INTRODUCCIÓN

Este documento tiene como objetivo servir como una guía para la implementación de correo seguro entre usuarios en el IMP, utilizando para ello como herramienta principal el certificado digital del usuario así como su llave privada, ambas emitidas por la Infraestructura PKI-IMP. Si no cuenta con un certificado digital emitido por esta infraestructura, lea primero el documento "Manual de Procedimiento de usuarios PKI-IMP"

Este es un primer ejemplo práctico y tangible de una aplicación directa de dichas herramientas, en un futuro se pretende utilizar tanto el certificado como la llave privada para el intercambio de cualquier tipo de datos(archivos, aplicaciones, etc.) cifrados y autenticados por medio del correo electrónico.

Este documento se basa en la utilización de MS Outlook y MS Outlook Express v6.0 como clientes de correo. Cabe mencionar que las referencias hechas en la documentación en cuanto a mensajes del sistema, ventanas, opciones, comportamiento, etc., pueden diferir dependiendo tanto de la versión del sistema operativo que usted tenga instalado, así como de la versión del cliente de correo. En este documento se tratarán de cubrir los detalles más importantes de cada versión tanto del cliente de correo como del sistema operativo.

De la misma manera se asume (como una recomendación de seguridad de los estándares de PKI) que usted tiene un y solo un certificado digital con la llave privada correspondiente a dicho certificado en el repositorio de Certificados Personales de su computadora. Windows XP maneja esto de manera muy fácil y sin complicaciones para el usuario, mas sin embargo versiones como WIN95 y WIN98 y WIN2000 permiten mantener mas de un certificado digital con su respectiva llave privada en el repositorio de certificados personales, en este caso la complejidad del procedimiento que se documenta aumenta para el usuario. Siga las recomendaciones que se le indican a fin de evitar complicaciones durante el procedimiento.

2- CONFIGURACIÓN

Asumimos que usted tiene una cuenta activa y configurada en su cliente de correo. Si no sabe como hacer esto refiérase a la documentación del Servicio de Correo Institucional del IMP en la siguiente dirección:

http://intranet.imp.mx/tecnologias/servicios/serv_email.htm

La dirección de correo que usted tenga configurada en Outlook u Outlook Express deberá ser la misma que la dirección de correo que aparece en su certificado digital, de lo contrario tendrá problemas al cifrar o al firmar digitalmente.

Abra una ventana del Cliente de correo

2.1- WIN95, WIN98 y WIN 2000

a) MS Outlook Express

Seleccione la cuenta que desee configurar.

En el menú, vaya a la barra de **Herramientas →Cuentas → Correo**

Windows le mostrará una ventana similar a la siguiente (Imagen 2.0):

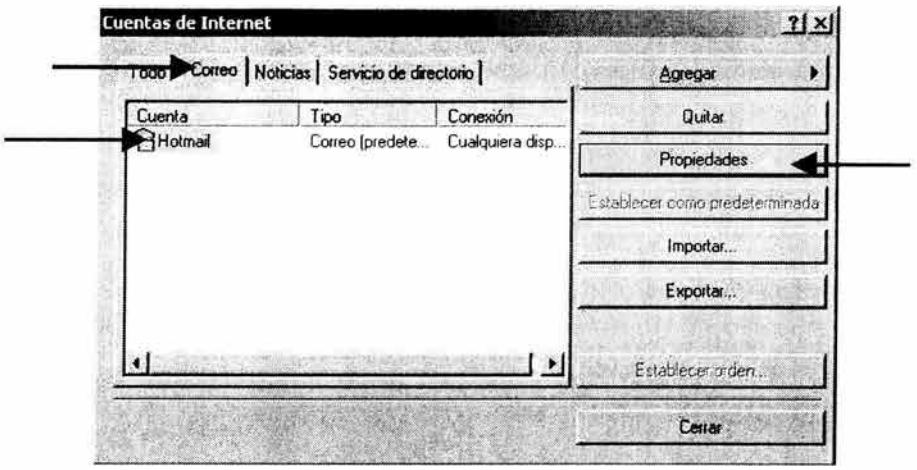


Imagen 2.0

Seleccione la cuenta y a Seguridad□s□ de click en el botón **Seguridad□s**

Segurid le mostrará la ventana siguiente (Imagen 2.1), seleccione la pestaña de **Seguridad** :

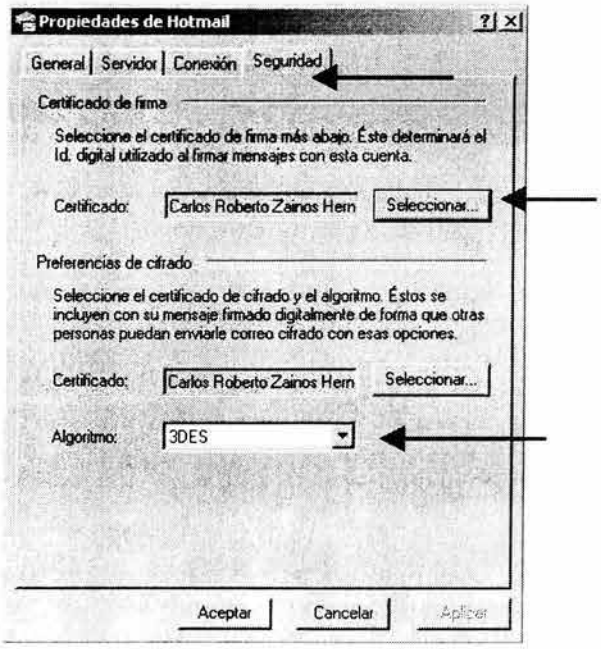


Imagen 2.1

En la sección de "Certificado de firma" de click en el botón **Seleccionar.**

Windows le mostrará en una ventana los certificados disponibles, seleccione el que desee utilizar (Imagen 2.2).

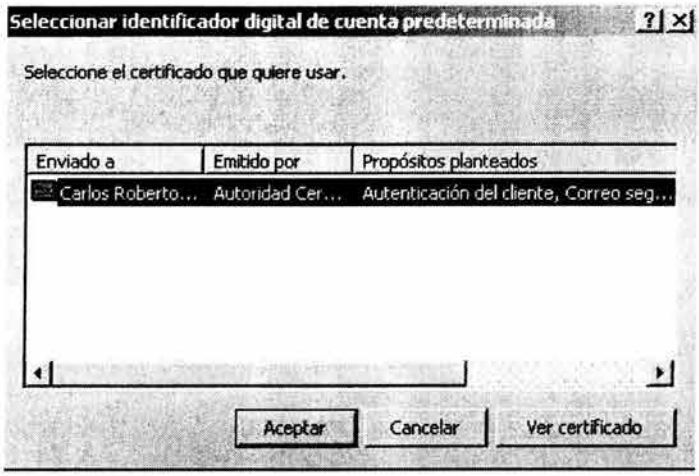


Imagen 2.2

De click en **Aceptar** y en la ventana de la Imagen 2.1 seleccione en la sección de "Preferencias de cifrado" seleccione el algoritmo 3DES (recomendado) y de click en **Aceptar**.

Finalmente en la ventana de cuentas de Internet (Imagen 2.0) de click en **Cerrar.**

b) MS Outlook

En la cuenta deseada vaya a la barra de **Herramientas →Opciones → Seguridad**

Windows le mostrará una ventana similar a la siguiente (Imagen 2.3):

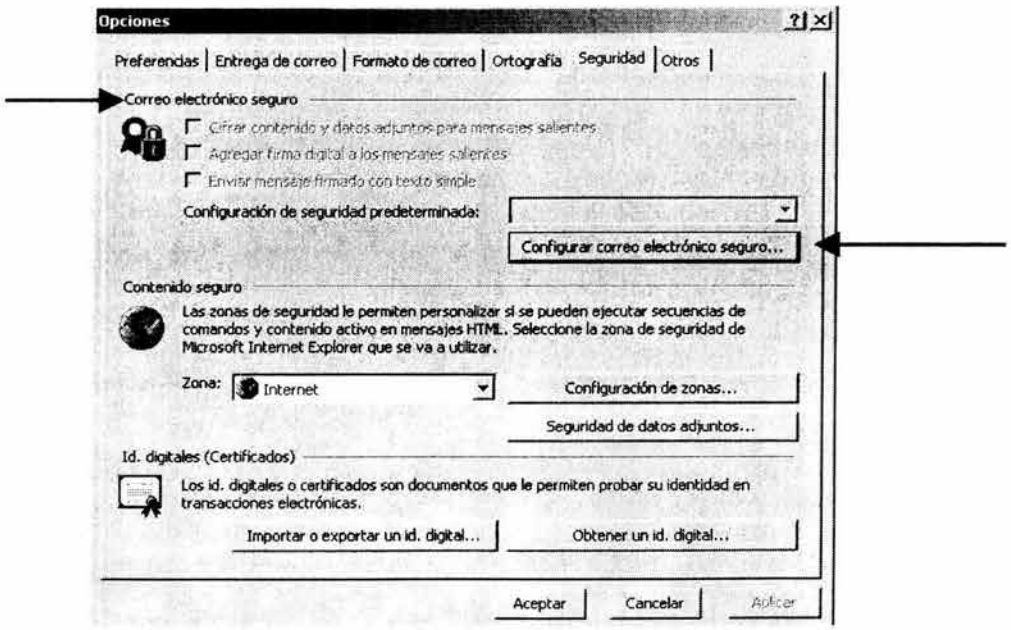


Imagen 2.3

En la sección de "Correo electrónico seguro" vaya a la opción de "Configurar correo electrónico seguro". Windows le mostrará la siguiente ventana (Imagen 2.4):

Imagen 2.4

Escriba un nombre para la configuración de seguridad y seleccione S/MIME en el formato de mensaje seguro.

A continuación de click en el botón **elegir** del campo **certificado de firma**. Verá una ventana como la siguiente (Imagen 2.5)

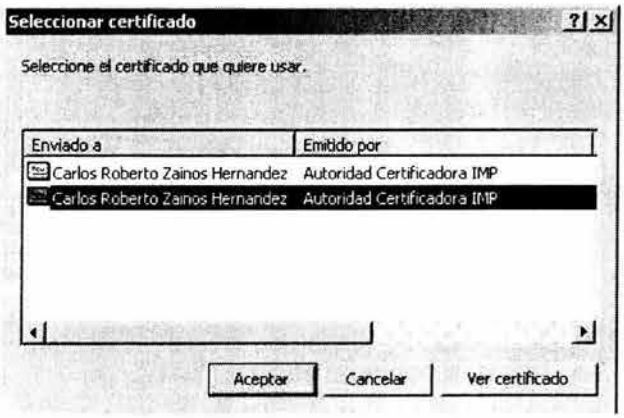


Imagen 2.5

Seleccione el certificado deseado y de click en **Aceptar.** Como resultado de esto Windows le mostrará la siguiente ventana (Imagen 2.6):

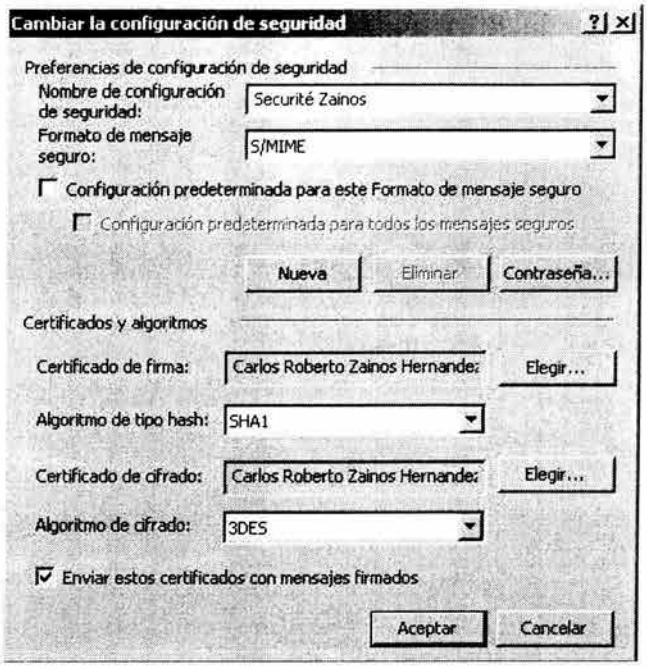


Imagen 2.6

En la Imagen 2.6 Windows le mostrará los algoritmos disponibles para llevar a cabo las operaciones de firma (Hash) y de cifrado. Seleccione SHA1 como algoritmo Hash y 3DES como algoritmo de cifrado (Recomendado).

De click en **Aceptar,** luego en **Aplicar** y finalmente en **Aceptar** nuevamente.

Una vez que ha terminado de configurar su cuenta de manera correcta, el siguiente paso es obtener los certificados digitales de los usuarios con los que desee manejar correo seguro. Estos certificados se utilizarán para agregarlos a las libretas de direcciones de los clientes y poder realizar las operaciones de firma digital, verificación de firma, cifrado y descifrado de mensajes. Usted cuenta con varias opciones para esto, las principales se mencionan a continuación.

2.1.1 Exportando un certificado del repositorio del explorador

Puede darse el caso de que usted ya cuente con el certificado digital, en el repositorio de certificados de su explorador, del o los destinatarios con los que desee intercambiar correo seguro. Si es así lo que usted debe hacer es simplemente exportar dicho certificado de ese repositorio e insertarlo en la libreta de direcciones del cliente de correo.

Para verificar esto, en una ventana de su explorador vaya al menú y siga **Herramientas→ Opciones de Internet→ Contenido→ Certificados→ Otras Personas**.

Usted debe ver una ventana como la siguiente (Imagen 2.7) :

Imagen 2.7

En la ventana que muestra la figura 2.7, seleccione el certificado deseado y de click en **Exportar...**

Se abrirá una ventana del Asistente para Exportación del Administrador de Certificados similar a la siguiente (Imagen 2.8):

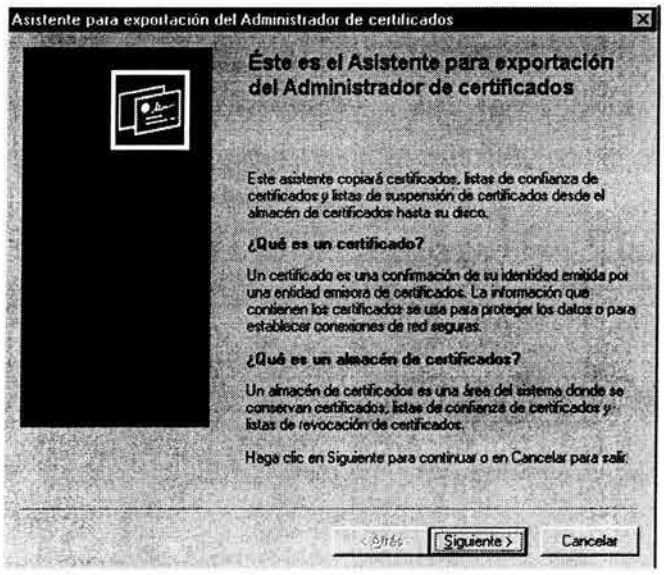


Imagen 2.8

De click en siguiente.

A continuación Windows le preguntará el formato en el cual desea guardar dicho certificado (Imagen 2.9).

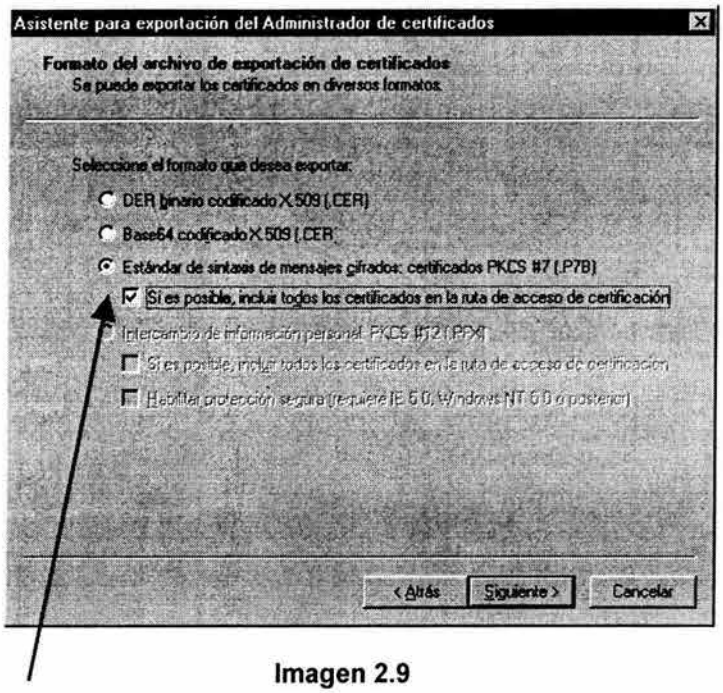


Imagen 2.9

Seleccione lo que se indica: formato "**PKCS #7**" y si es posible "**Incluir todos los certificados en la ruta de certificación".**

De click en siguiente.

Seleccione un Nombre y una Ubicación para el certificado que será exportado (Imagen 2.10):

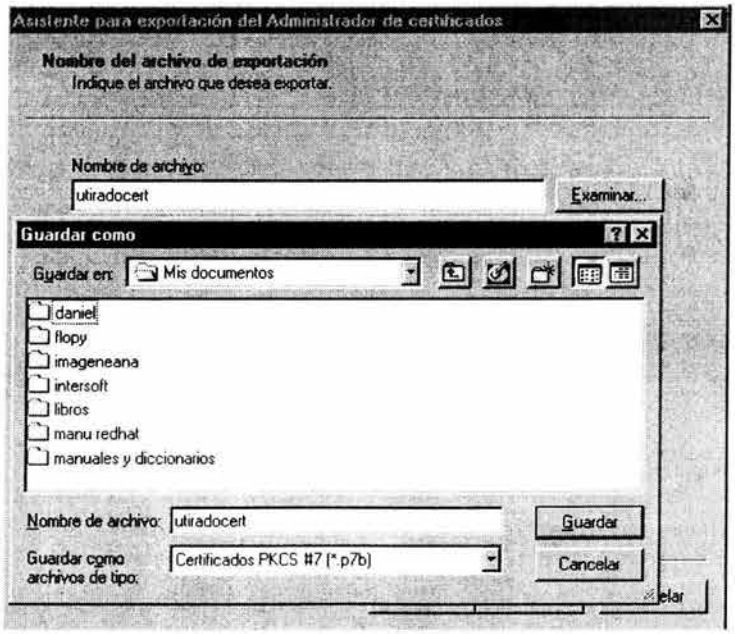


Imagen 2.10

De click en **Guardar,** una vez realizado esto de click en **Siguiente**

A continuación aparece la ventana de **Completando el asistente para exportación del Administrador de Certificados. (Imagen 2.11)**

Imagen 2.11

De click en **Finalizar** para cerrar esta ventana. Windows le enviará el siguiente mensaje (Imagen 2.12):

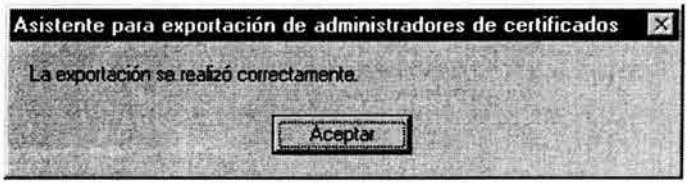


Imagen 2.12

De click en **Aceptar.**

Realice el procedimiento anterior para cada certificado que desee exportar al cliente de correo.

2.1.2 Descargando certificados de terceros (Otras Personas)

Si él o los destinatarios del correo seguro no aparecen en la lista de "Otras Personas" del repositorio de certificados del navegador, deberá descargar él o los certificados de dichas personas. Para ello proceda como se indica.

Vaya a la página principal de PKI-IMP (https://raguel.imp.mx/pub).

En la sección de **"Certificados"** de click en la liga **"Válidos".**

El navegador le mostrará la lista actualizada de certificados válidos (Imagen 2.13).

Busque y seleccione el certificado del destinatario del correo seguro.....

En este caso vamos a suponer que queremos enviarle un correo seguro a Hilario García

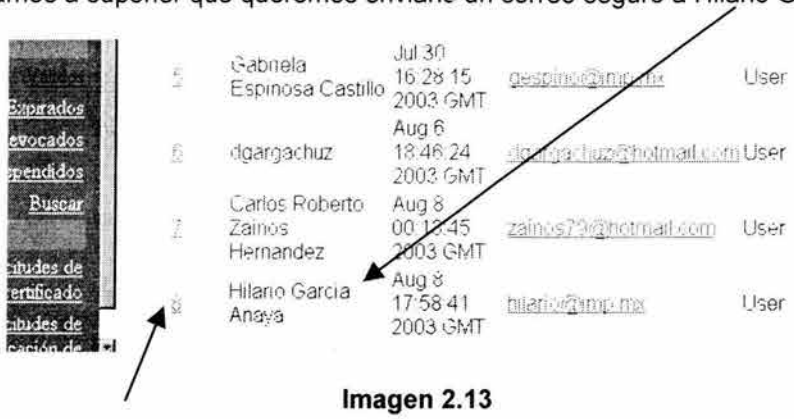


Imagen 2.13

Damos click en el número de serie del certificado del usuario deseado.

El servidor nos muestra la página de **Información del Certificado**, en ella se muestra la información relevante del certificado en cuestión (Imagen 2.14).

Imagen 2.14

Al final de la página se le presentan dos opciones: **Descargar el Certificado** o **Revocar el Certificado**.

Seleccione **Descargar el Certificado**.

Windows le desplegará una ventana de descarga de Archivos.

Seleccione un **nombre de archivo** y una ubicación (Imagen 2.15).

Vea bien la ubicación ya que posteriormente procederemos a importar este certificado en la libreta de direcciones de Outlook o Outlook Express. La extensión la da por defecto Windows.

Seleccione **Guardar.....**

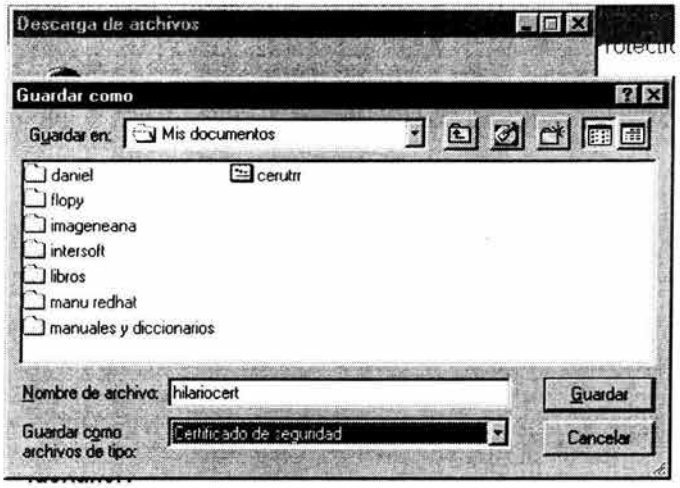


Imagen 2.15

Realice este procedimiento para cada destinatario deseado.

2.1.3 Insertando el certificado de "Otras Personas" en Outlook y Outlook Express

Vaya a la libreta de direcciones del cliente correspondiente.

De click en agregar **Nuevo** contacto.

Introduzca el nombre del contacto así como su dirección de correo electrónico, de click en **Agregar**

Lo realizado anteriormente se muestra en la siguiente ventana (Imagen 2.16):

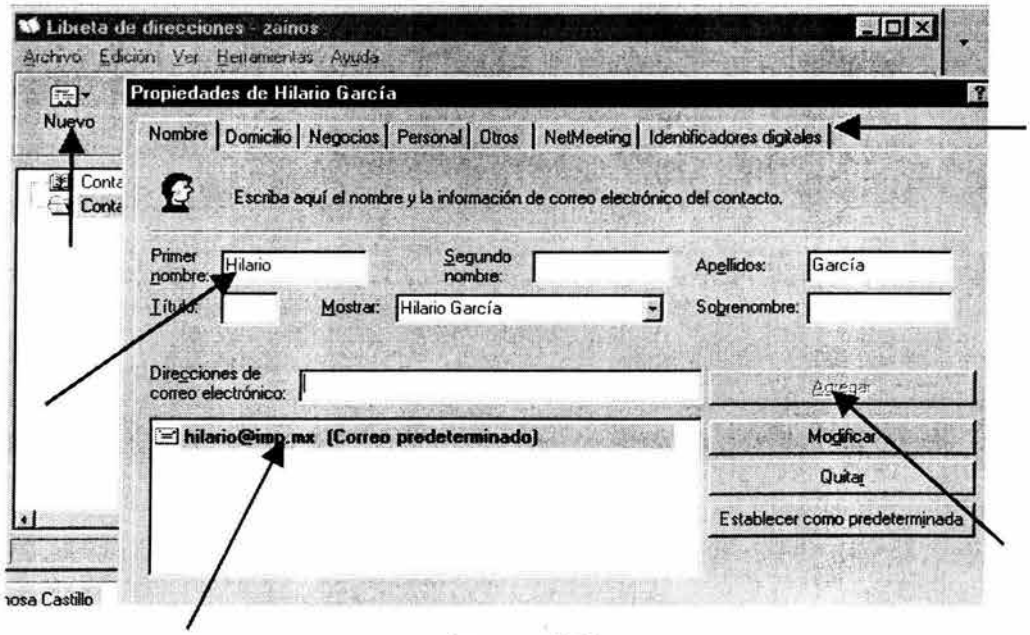


Imagen 2.16

Vaya a la pestaña **"Identificadores Digitales"**

Windows le mostrará la siguiente ventana (Imagen 2.17):

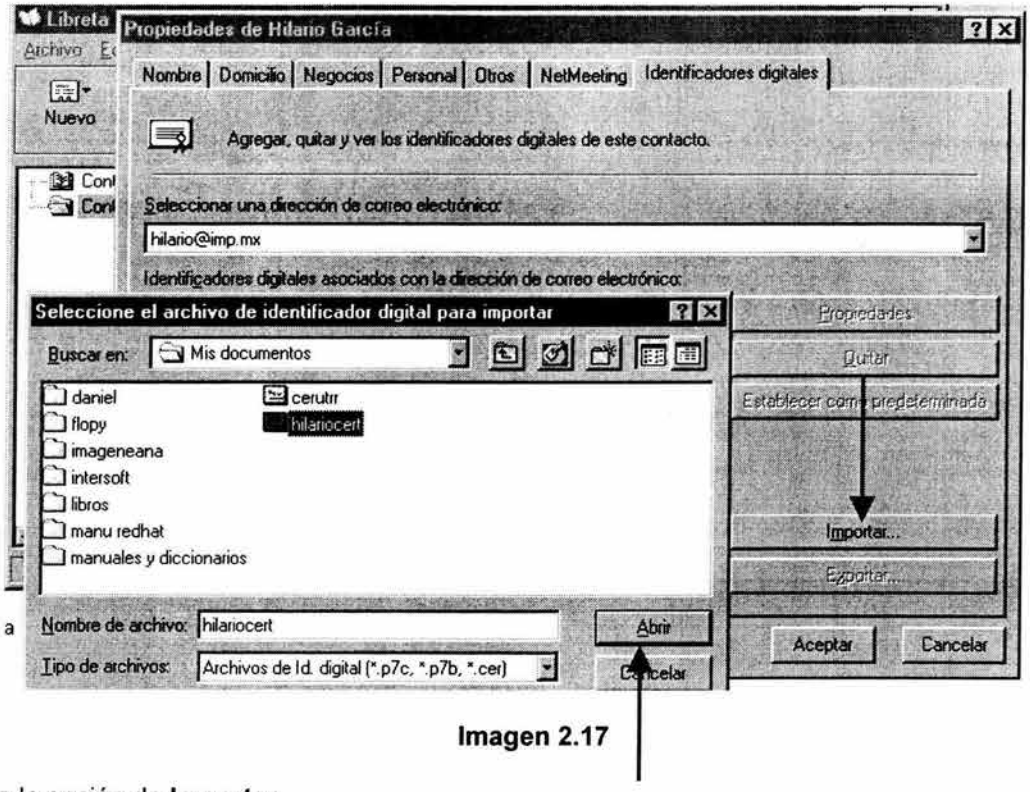


Imagen 2.17

Vaya a la opción de **Importar...**

Seleccione el archivo a importar (Imagen 2.17), el cual debe ser alguno de los que guardó en el paso que ilustra la imagen 2.10 o 2.15

De click en **Abrir**

Windows le mostrará la siguiente ventana, indicando con esto que la importación ha sido satisfactoria (Imagen 2.18):

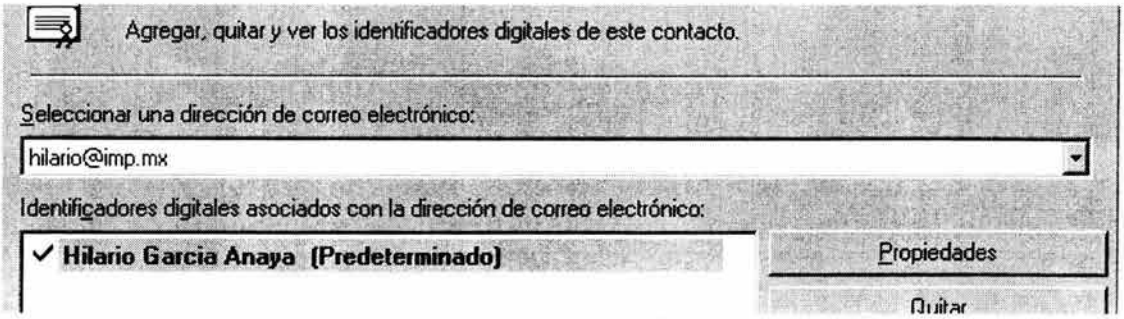


Imagen 2.18

De click en **Aceptar** y después en **Cerrar.**

Repita este procedimiento para todos aquellos contactos con los que desee intercambiar correo seguro.

NOTA: Recuerde que adicionalmente al procedimiento descrito anteriormente, usted puede hacer uso del servicio de directorio LDAP-IMP y utilizar este para descargar él o los certificados deseados.

La manera de realizar esto se encuentra en la sección cuatro del documento **Manual de Procedimiento PKI-IMP.** Este documento se encuentra disponible en la sección de **Introducción** en la página principal del servidor PKI-IMP (http://raguel.imp.mx/pub).

Ahora trataremos lo referente utilizando el Sistema Operativo Windows XP

2.2- WIN XP

MS Outlook y Outlook Express

Abra una ventana del cliente de correo

En el menú, vaya a la barra de **Herramientas →Opciones → Seguridad**

Windows le mostrará una ventana como la siguiente (Imagen 2.19):

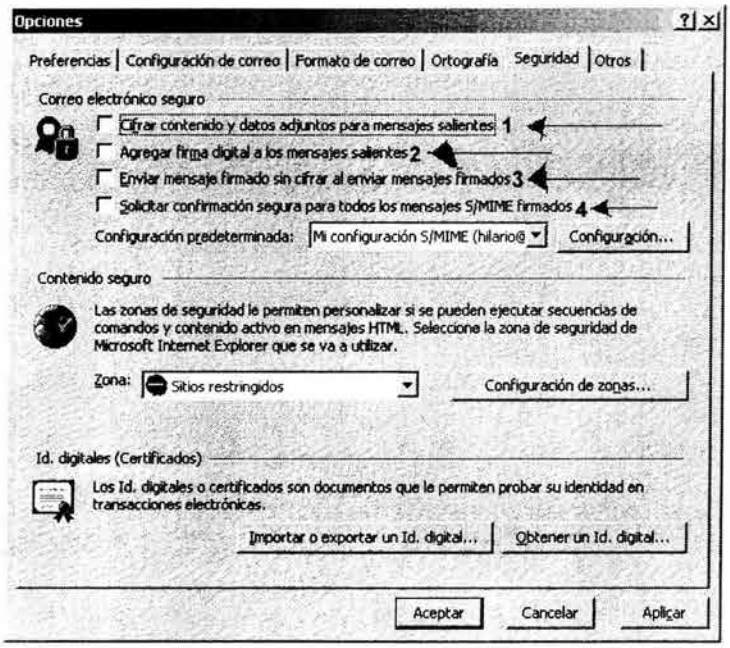


Imagen 2.19

En esta sección se le presentan varias opciones de lo que usted puede realizar con su certificado digital y con su llave privada. Para la parte de **"Correo Electrónico Seguro"**

1- Cifrar Contenido y datos adjuntos para Mensajes salientes.- Seleccione esta casilla si es que usted desea que todos sus mensajes, así como los adjuntos que pudiese contener, sean cifrados, esto es, que solo el destinatario pueda leer el mensaje.
2- Agregar Firma digital a los mensajes salientes.- Seleccione esta casilla si es que desea que todos sus mensajes vayan firmados por usted (firma digital), esto garantiza al destinatario que usted y nadie más crearon y enviaron dichos mensajes. Al seleccionar

esta casilla además de la anterior, el mensaje viajará firmado y cifrado lo cual garantiza una mayor seguridad

3- Enviar Mensajes firmados sin cifrar al enviar mensajes firmados. Esta opción solo firma el mensaje y este viaja en claro (no cifrado)
4- Solicitar confirmación segura.- Seleccione esta casilla a fin de ser notificado al enviar un mensaje seguro.
5- Seleccionar configuración de seguridad (Ver procedimiento de MS Outlook en la Sección b del punto 2.1)

Puede usted seleccionar cualquiera de las opciones anteriormente presentadas, sin embargo por razones prácticas se recomienda no marcar ninguna y hacerlo únicamente en aquellos mensajes que así lo requieran al momento de redactar el mensaje.

2.2.1 Agregando Destinatarios de Correo Seguro

La manera en que se hará esto es utilizando la libreta de direcciones del cliente de correo, ya que es la más sencilla y la que presenta menos complicaciones.

Primero tendrá que verificar que usted posee el certificado digital de dicho destinatario.

En cualquier ventana de su explorador, vaya a la barra de **Herramientas→ Opciones de Internet→ Contenido → Certificados,** Windows le mostrará una ventana como la siguiente (Imagen 2.20):

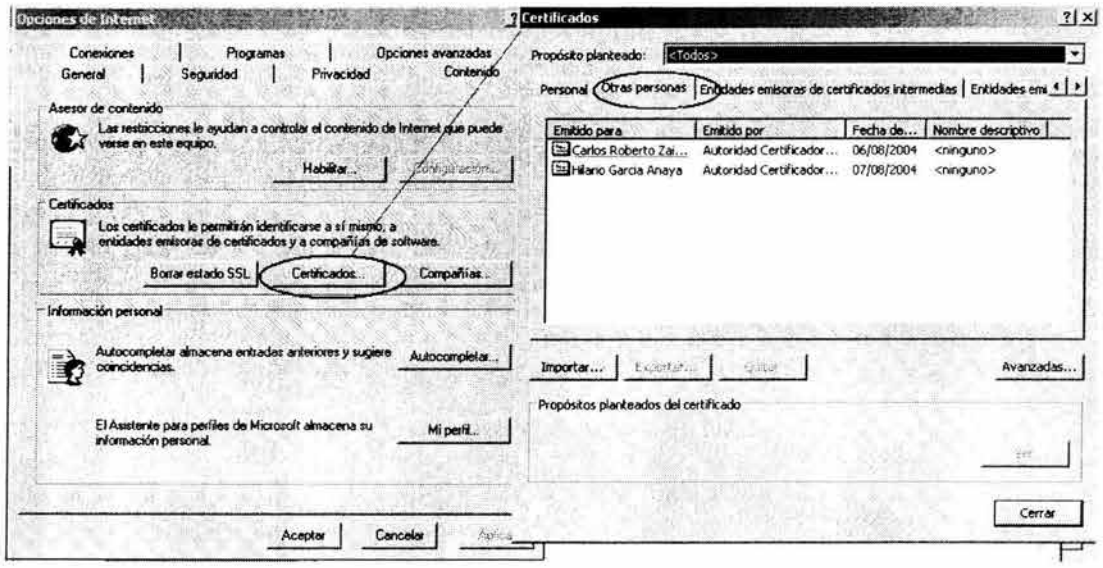


Imagen 2.20

Vaya a la pestaña de **"Otras Personas"** verifique que ahí se encuentre la persona a la que desea enviar correo seguro, si no se encuentra en esa lista, proceda como lo indica la sección: **2.1.2 Descargando certificados de terceros (Otras Personas).**

En el caso de que usted tenga el certificado digital del destinatario en el repositorio, proceda como se indica en la sección **2.1.1 Exportando un certificado del repositorio del explorador**.

Lo siguiente es crear dicho contacto en la libreta de direcciones del cliente de correo correspondiente. Para esto proceda como se indica a continuación.

Dentro de la ventana del cliente de correo, vaya al menú y de click en agregar contacto (en la libreta de direcciones de click en Nuevo Contacto).

Windows le mostrará una ventana como la siguiente, llene la forma con los datos solicitados (Imagen 2.21)

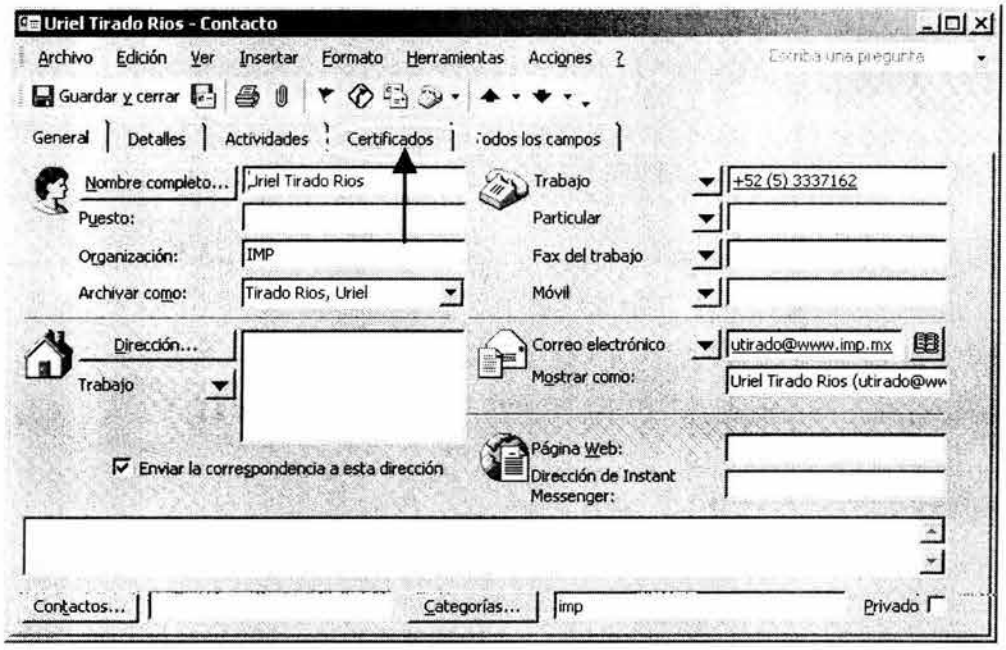


Imagen 2.21

Vaya a la pestaña que dice **Certificados**

Windows le mostrará una ventana como la siguiente (Imagen 2.22)

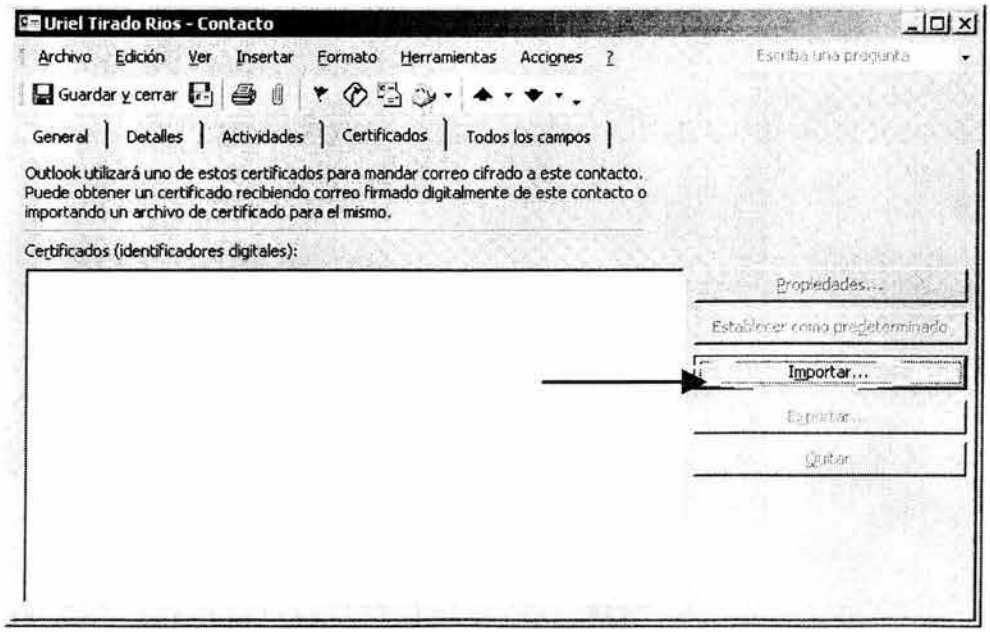


Imagen 2.22

De click en **Importar**.

Explore en busca del certificado anteriormente exportado del repositorio de certificados (Imagen 2.23)

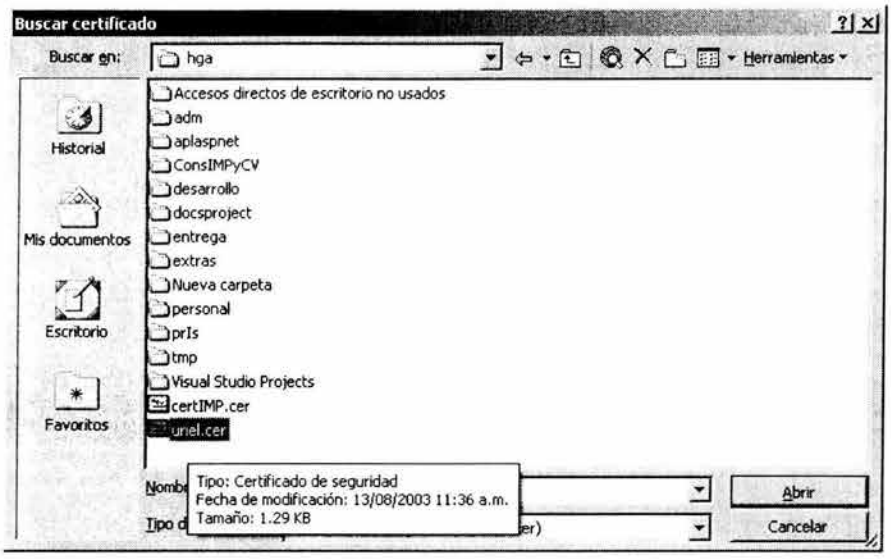


Imagen 2.23

Selecciónelo y de click en abrir.

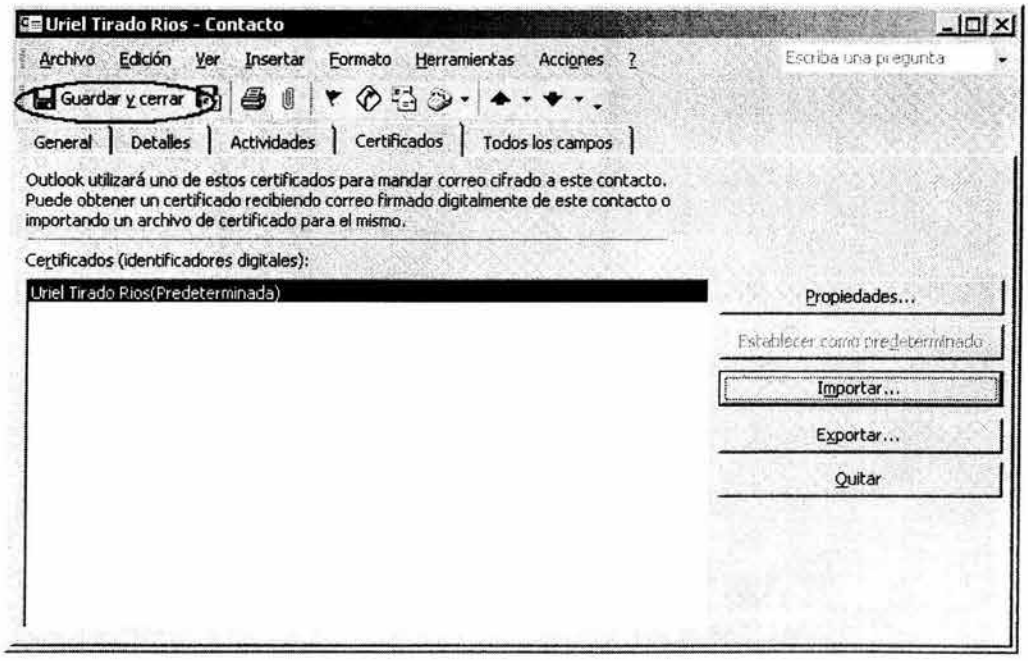


Imagen 2.24

Finalmente de click en **Guardar y Cerrar** (Imagen 2.24)

Con lo realizado hasta el momento, estamos listos para poder enviar y recibir correo seguro. A continuación se indica el procedimiento.

3- ENVIANDO UN CORREO SEGURO

En la ventana principal del cliente de correo deberá ver los elementos agregados así como el indicador de que posee un certificado digital de ese contacto (Imagen 3.0).

Imagen 3.0

Ya que tenemos a nuestro usuario, damos doble click sobre su Nombre y Windows nos abre una ventana de **Redacción.**

Escribimos nuestro mensaje, agregamos los archivos adjuntos correspondientes (si es el caso) y procedemos como se indica:

En el menú de la ventana vamos a **Herramientas**

Seleccionamos las opciones de **Firmar y Cifrar,** estas dos opciones son indispensables para garantizar las propiedades de autenticación, confidencialidad e integridad del mensaje. Sin embargo usted puede o solo **Firmar** o solo **Cifrar,** dependiendo de lo que desee (Lea la sección de Comentarios, al final de este documento, para una mayor referencia).

De manera alternativa a lo anterior, puede solo dar click en los botones de **Firmar** y **Cifrar** en la barra de herramientas de la ventana del mensaje (si están disponibles).

Cuando haya realizado cualquiera de las dos opciones anteriores, verá unos símbolos que indican que el mensaje será Firmado y/o Cifrado digitalmente.

Lo anterior se muestra, en la ventana del mensaje, mas o menos de la siguiente manera (Imagen 3.1):

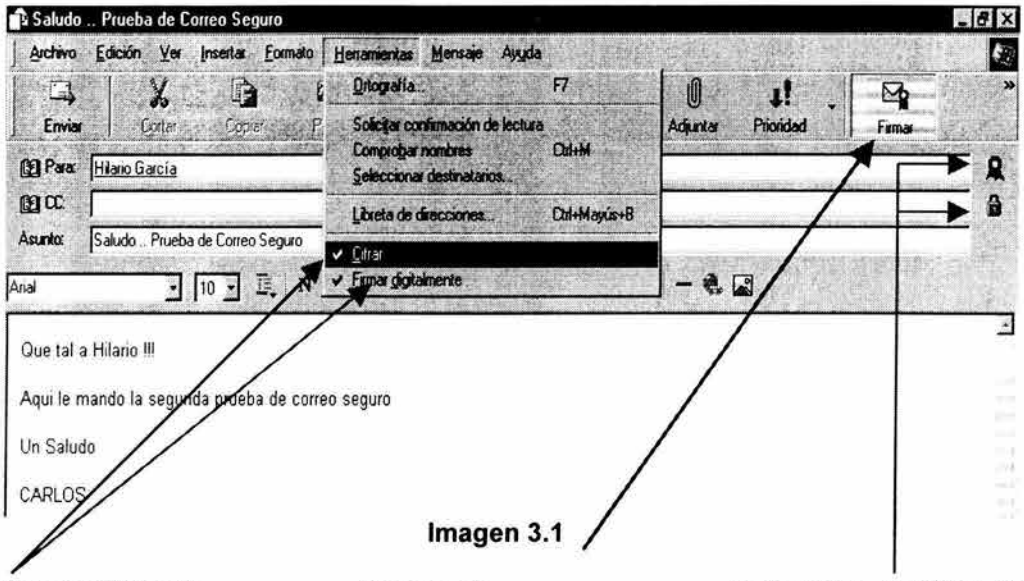


Imagen 3.1

Herramientas de Cifrar y/o firmar Digitalmente

Botones de Firmar y/o Cifrar

Indicadores de Firma y/o Cifrado

De click en **Enviar**

Se inicia el proceso de firma y/o cifrado del mensaje, según lo especificado, utilizando su clave privada y/o el certificado del destinatario.

Nota: Cada vez que quiera utilizar su clave privada para firmar un mensaje, Windows en su momento le solicitará el password que protege a la misma (el mismo que le pidió cuando estaba generando su clave privada y la solicitud de certificación).

Dependiendo de su configuración, Windows puede o no enviarle el siguiente mensaje de Alerta (Imagen 3.2) :

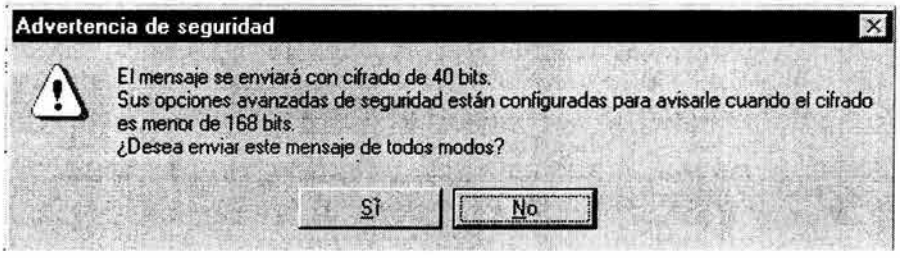


Imagen 3.2

De click en Sí

El contenedor de clave privada le solicitará el password para poder utilizar la misma y firmar digitalmente el mensaje.

Introdúzcalo y de click en Aceptar (Imagen 3.3)

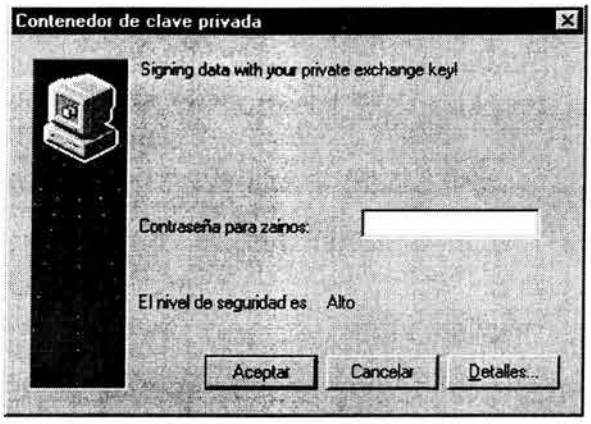


Imagen 3.3

El mensaje se firmará y/o se ensobretará (incluidos los archivos adjuntos) y se enviará en un formato PKCS #7.

El receptor recibirá el mensaje, para poder verlo Windows en su momento le solicitará el password que protege a su clave privada para poder utilizarla, descifrar el mensaje (abrir el sobre digital) y poder leerlo. Adicionalmente el receptor validará la firma y el certificado del emisor, si esto aplica.

De la misma manera, si usted es el receptor del correo seguro deberá introducir el password que protege a su clave privada para poder abrir el sobre digital y poder leer el contenido del mensaje, y/o validar el estado de la firma y del certificado del firmante, si esto aplica. En la sección de comentarios, al final de este documento, se muestran ejemplos ilustrativos de esto.

Esta es quizá la aplicación más sencilla en donde se puede utilizar la infraestructura de PKI-IMP. Al utilizar las opciones de firma y cifrado, garantizamos los cuatro puntos más importantes que debe cubrir una infraestructura de PKI:

- Autenticación de origen del mensaje.
- Integridad del mismo, no sufrió alteraciones de contenido.
- Confidencialidad, nadie mas que el destinatario puede leerlo.
- No repudio de origen, el emisor del mensaje no puede argumentar que no lo envió.

El proyecto PKI-IMP busca poder contar en un futuro próximo con una herramienta que permita cifrar cualquier tipo de archivo y/o aplicación a fin de garantizar la confidencialidad, autenticación, integridad y no repudio no solo de un mensaje de correo electrónico, sino de todo un conjunto de archivos o aplicaciones.

Actualmente el proyecto se encuentra trabajando en eso.

PROYECTO PKI-IMP

URIEL TIRADO RIOS
RESPONSABLE DEL PROYECTO

CARLOS ROBERTO ZAINOS H
ADMINISTRADOR

COMENTARIOS

La presente sección tiene como objetivo aclarar algunas dudas sobre el servicio de correo seguro. En esta sección ampliaremos algunos puntos que son importantes y que usted debe saber.

Como se mencionó al final de la sección anterior, los cuatro servicios más importantes que debe proporcionar una infraestructura de PKI son las siguientes:

- Autenticación de los participantes en transacciones electrónicas (objetivo principal de PKI-IMP).
- Integridad de la información intercambiada.
- Confidencialidad de la información
- No repudio de origen, los participantes no pueden negar su participación en las transacciones realizadas.

Por sus características, y por las facilidades disponibles para ello, el servicio de correo electrónico es la primera aplicación que puede hacer uso de estos cuatro servicios de PKI-IMP (y por el momento la única a nivel institucional).

Partiendo del hecho de que usted cuenta con un certificado digital y un par de claves generadas por medio de la infraestructura PKI-IMP, en primer lugar diremos lo siguiente:

- Tanto el certificado como el par de claves asociadas, son únicas dentro de toda la infraestructura de PKI-IMP. Esto lo verifica y asegura la AC-IMP, quien es la autoridad más alta de esta infraestructura de seguridad. Esto garantiza que no existe la posibilidad de que alguien pueda hacerse pasar por usted.

- Usted y solo usted es el responsable de mantener protegida y en un lugar seguro su clave privada, así como de realizar un respaldo de la misma cuando la recibe. Por políticas de seguridad y operación, AC-IMP no mantiene respaldo de las mismas, solo de los certificados.

- La clave privada, como su nombre lo indica, es "privada" y nadie mas que usted puede utilizarla.

- El uso del certificado y de sus claves, los resultados obtenidos, y las responsabilidades que ello implique fuera del servicio de correo electrónico institucional, es responsabilidad única y exclusivamente de usted.

Los anteriores puntos podemos considerarlos como premisas de la infraestructura PKI-IMP.

A continuación mostraremos de una manera "gráfica" como PKI-IMP proporciona los servicios mencionados al inicio de esta sección en el servicio de correo electrónico.

Cuando usted configuró su cliente de correo electrónico para hacer uso del certificado y de la clave privada correspondiente, automáticamente activó y puso a su disposición mecanismos que le permiten proporcionar facilidades de autenticación, confidencialidad e integridad de los mensajes intercambiados bajo este esquema.

Estos mecanismos son principalmente:

- Firmas digitales
- Verificación de firmas digitales
- Cifrado de mensajes
- Descifrado de mensajes

- Verificación de estado de certificado y
- Acceso a directorio LDAP-IMP

Lo anterior lo podemos verificar si vemos las propiedades de seguridad del cliente de correo (Outlook o Outlook Express) Imagen A-1 :

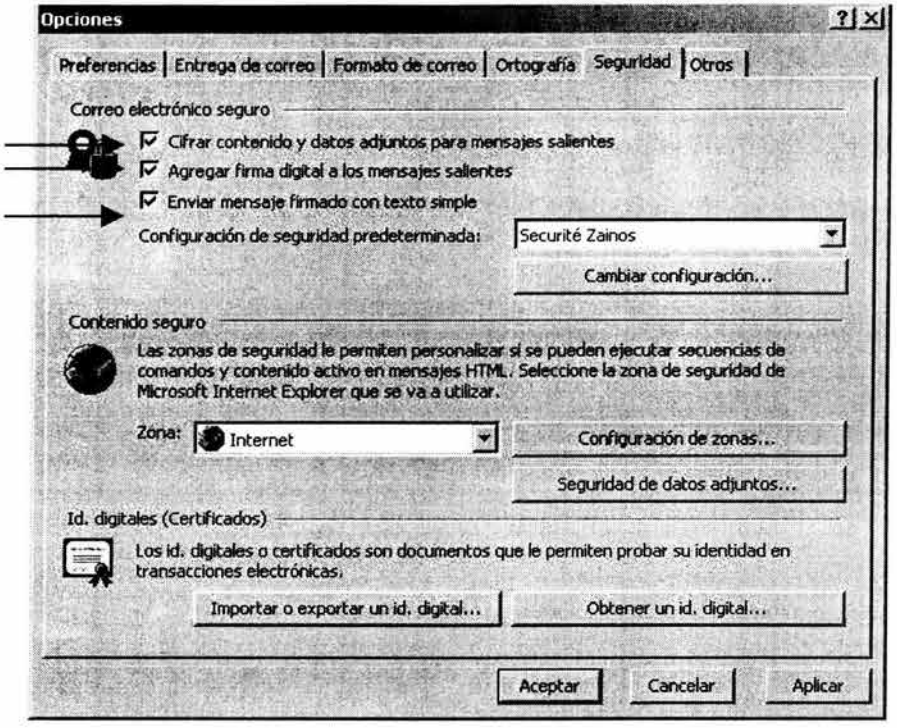


Imagen A-1

Somos libres de especificar que para e el manejo de correo electrónico seguro cifre contenido y datos adjuntos y agregue firma digital a todos los mensajes salientes o solo a aquellos que especifiquemos al momento de redactar los mismos.

Cuando creamos un mensaje y seleccionamos las opciones de firmado y cifrado (Imagen A-2) estamos garantizando lo siguiente:

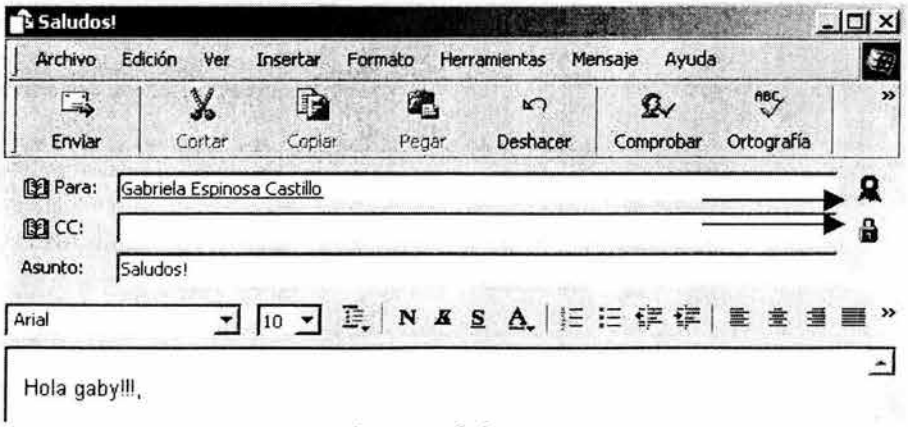


Imagen A-2

- La Autenticación del origen del mensaje (Firma Digital).

  Como el mensaje va firmado con su clave privada, el destinatario tiene la certeza de que solo usted pudo haber generado dicho mensaje ya que al verificar la firma, con ayuda de la clave pública incluida en su certificado digital, esta resultó válida. Recuerde que solo usted puede utilizar su clave privada y su certificado esta disponible para todos los usuarios de PKI-IMP. Esto se "ve"de la siguiente manera (Imagen A-3):

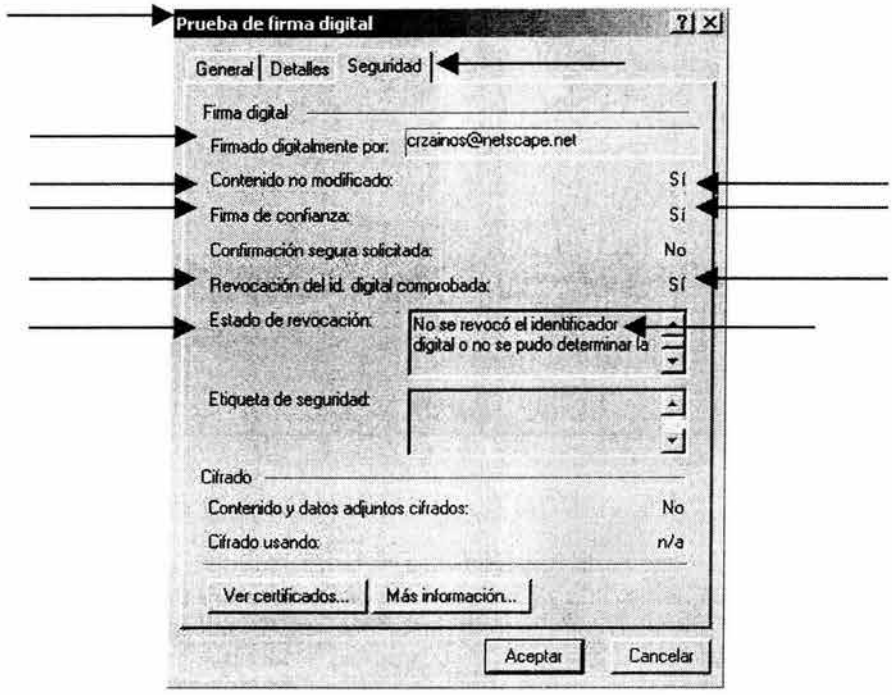


Imagen A-3

Con lo anterior validamos el primer punto

- Integridad del mismo, no sufrió alteraciones de contenido (Funciones Hash).

  Este punto se valida con ayuda de la firma digital. Se puede ver en la Imagen A-3. Si validamos la autenticidad de una firma, es un hecho que por defecto validamos la integridad del mensaje.

- Confidencialidad, nadie mas que el destinatario puede leerlo (Ensobretado digital).

  Respecto a este punto diremos que al generar y antes de enviar el mensaje, el contenido del mismo, así como los archivos adjuntos (si los hubiesen) fueron cifrados utilizando la clave pública del destinatario, contenida en el certificado digital. Lo anterior se conoce también como un sobre digital. Para poder "abrir el sobre" el destinatario deberá utilizar su clave privada.

  Esto se verifica de la siguiente manera (Imagen A-4):

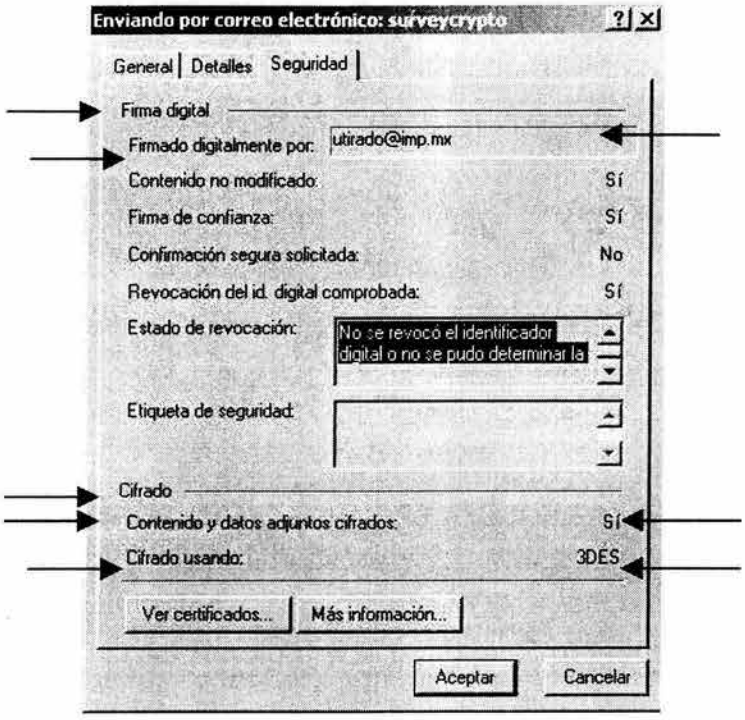


Imagen A-4

Con ayuda de esto validamos el tercer punto

- No repudio de origen, el emisor del mensaje no puede argumentar que no lo envió.

  La manera sencilla de demostrar esto es que como solo existe un par de claves en la infraestructura, lo que se verificó o descifró con una clave, tuvo que ser firmado o cifrado con la otra, no hay más. Este punto para implementarse de manera formal, requiere de trabajo adicional básicamente administrativo. Es necesario implementar mecanismos de auditoria y otro tipo de cosas que PKI-IMP por el momento no ha realizado. Probablemente en el futuro se pueda realizar.

En relación a lo mencionado, agregaremos los siguientes comentarios:

Un mensaje solo firmado digitalmente por usted garantiza al receptor, al momento de validar la firma, que el mensaje es auténtico y que no ha sido modificado el contenido del mismo (propiedades de autenticación e integridad). Un mensaje solo firmado no garantiza la confidencialidad del mismo.

Un mensaje solo cifrado con el certificado del destinatario, le garantiza a usted que solo el receptor del mensaje podrá ver el contenido del mismo, ya que nadie mas que el conoce su clave privada (propiedad de confidencialidad). Un mensaje solo cifrado no garantiza la autenticidad e integridad del mismo.

Adicionalmente, y para concluir, incluimos los siguientes detalles importantes:

Cuando recibimos un mensaje enviado bajo el esquema de correo seguro, nuestro cliente de correo nos mostrará una viso como el siguiente (Imagen A-5):

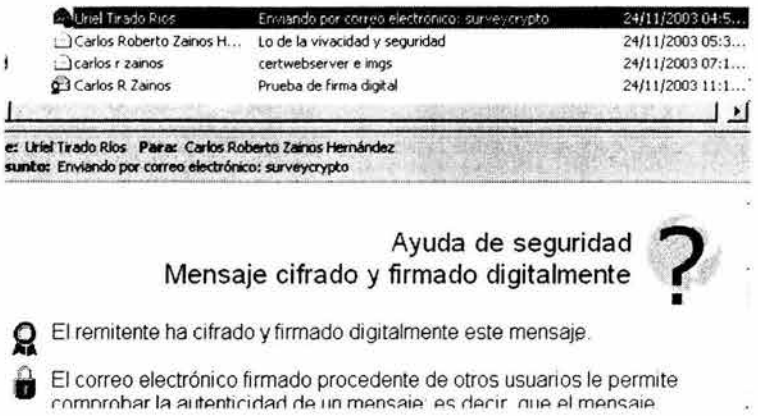


Imagen A-5

Lo anterior indica el estado del mensaje, para poder leerlo el cliente de correo nos solicitará el password que protege nuestra clave privada, la verificación de la firma la realiza el mismo cliente de manera "transparente" para nosotros. Si no conocemos dicho password, ya sea que lo olvidamos o de algún modo hubiésemos podido "interceptar" el mensaje, simplemente no se podrá ver el contenido del mismo.

Ahora, si utilizamos un cliente que no sea capaz de realizar estas funciones, o que pudiésemos haber interceptado de alguna manera el mensaje, lo que veríamos sería algo como lo siguiente (Imagen A-6)

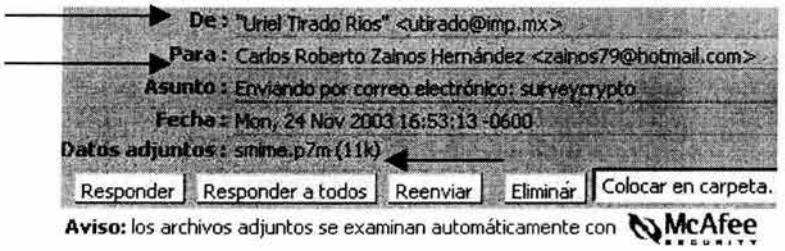


Imagen A-6

El resultado sería el mismo, no podría ver el contenido del mensaje.

El presente documento se hizo basado en los clientes de correo de Microsoft por ser estos los más populares dentro del Instituto, considerando los cambios y actualizaciones de equipo de cómputo. Eudora es otro cliente que tiene varios seguidores dentro del IMP, pero no brinda las facilidades para la implementación del servicio de correo seguro. Este caso aún se encuentra en estudio.

Otro cliente que si soporta estos servicios, pero que no se incluyó en este trabajo por no ser tan popular dentro de la comunidad IMP, es el cliente de correo WebMail de Netsacpe. Esperamos que este cliente, en el futuro, gane preferencia dentro de la comunidad IMP

En esta parte incluimos algunos resultados de pruebas hechas con este cliente solo como referencia para mostrar que tanto el certificado como el par de claves emitidos por PKI-IMP son útiles en cualquier ámbito donde se les quiera utilizar y no solo en las aplicaciones de Microsoft.

Un menaje Firmado y cifrado se "ve" de la siguiente manera en el cliente WebMail (Imagen A-7)

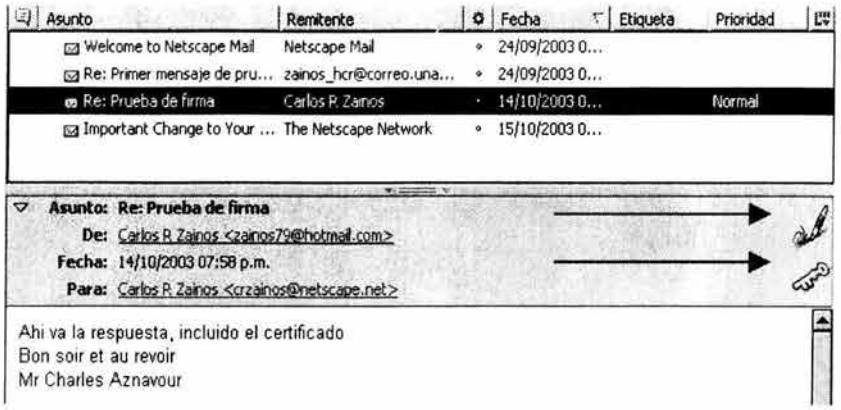


Imagen A-7

Al igual que para los clientes Microsoft, para poder descifrar y validar el contenido del mismo, es necesario introducir el password que protege a la clave privada (Imagen A-8)

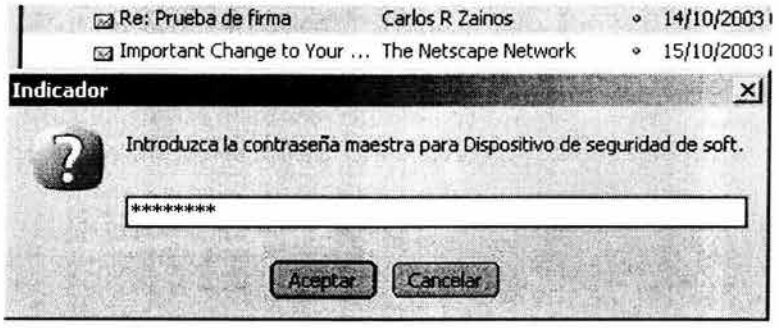


Imagen A-8

Si no sabemos dicho password, no podremos ver el contenido del mismo.

El estado del mensaje, y las propiedades del mismo, se verifican igualmente de manera muy sencilla (Imagen A-9) :

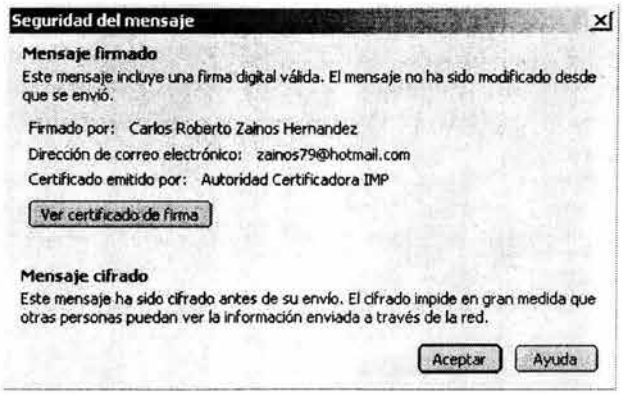


Imagen A-9

Con esto damos por terminado el presente trabajo. Esperamos que este documento, así como los demás documentos de PKI-IMP, sea de gran utilidad para usted al igual que toda la infraestructura y

los servicios de PKI-IMP; ya que fue concebida, diseñada e implementada pensando en la seguridad de la información que el IMP maneja a través de usted.

Para cualquier duda, aclaración, comentarios y/o sugerencias, contacte a los encargados del proyecto.

PROYECTO PKI-IMP

URIEL TIRADO RIOS
RESPONSABLE DEL PROYECTO

CARLOS ROBERTO ZAINOS H
ADMINISTRADOR

ANEXO E-1

Resultado de análisis de vulnerabilidades de red de PKI-IMP

Software de análisis : Nessus

14.11.2003

Network Vulnerability Assessment Report

Sorted by host names

Session name: PKI IMP	**Start Time:** 14.11.2003 16:26:26
Finish Time: 14.11.2003 17:11:32	**Elapsed:** 0 day(s) 00:45:05

Total records generated:	22
high severity:	3
low severity:	9
informational:	10

Summary of scanned hosts

Host	Holes	Warnings	Open ports	State
raguel.imp.mx	3	9	10	Finished

raguel.imp.mx

Service	Severity	Description
snmp (161/udp)	Info	Port is open

sunrpc (111/tcp)	**Info**	Port is open
http (80/tcp)	**Info**	Port is open
ldap (389/tcp)	**Info**	Port is open
sunrpc (111/udp)	**Info**	Port is open
listen (1025/tcp)	**Info**	Port is open
unknown (1024/udp)	**Info**	Port is open
x11 (6000/tcp)	**Info**	Port is open
kdm (1024/tcp)	**Info**	Port is open
ssh (22/tcp)	**Info**	Port is open
ssh (22/tcp)	**High**	You are running a version of OpenSSH which is older than 3.0.2. Versions prior than 3.0.2 are vulnerable to an environment variables export that can allow a local user to execute command with root privileges. This problem affect only versions prior than 3.0.2, and when the UseLogin feature is enabled (usually disabled by default) Solution : Upgrade to OpenSSH 3.0.2 or apply the patch for prior versions. (Available at: ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH) Risk factor : High (If UseLogin is enabled, and locally) CVE : CVE-2001-0872 BID : 3614
snmp (161/udp)	**High**	SNMP Agent responded as expected with community name: public CVE : CAN-1999-0517, CAN-1999-0186, CAN-1999-0254 BID : 177, 7081, 7212, 7317
ldap (389/tcp)	**High**	Improperly configured LDAP servers will allow any user to connect to the server and query for information. The LDAP bind function in Exchange 5.5 has a buffer overflow that allows a remote attacker to conduct a denial of service or execute commands in all version prior to Exchange server sp2 Note: no test was done to see what version of Exchange server is running, nor attempt to verify service pack. Solution: Disable NULL BIND on your LDAP server Also see: http://www.microsoft.com/technet/security/bulletin/ms99-009.asp Risk factor : Medium CVE : CVE-1999-0385 BID : 503
kdm (1024/tcp)	Low	RPC program #100024 version 1 'status' is running on this port
x11 (6000/tcp)	Low	This X server does *not* allow any client to connect to it however it is recommended that you filter incoming connections

		to this port as attacker may send garbage data and slow down your X session or even kill the server. Here is the server version : 11.0 Here is the message we received : Client is not authorized to connect to Server Solution : filter incoming connections to ports 6000-6009 Risk factor : Low CVE : CVE-1999-0526
sunrpc (111/tcp)	Low	The RPC portmapper is running on this port. An attacker may use it to enumerate your list of RPC services. We recommend you filter traffic going to this port. Risk factor : Low CVE : CAN-1999-0632, CVE-1999-0189 BID : 205
snmp (161/udp)	Low	Using SNMP, we could determine that the remote operating system is : Linux raguel.imp.mx 2.4.7-10 #1 Thu Sep 6 17:21:28 EDT 2001 i586
listen (1025/tcp)	Low	RPC program #391002 version 2 'sgi_fam' (fam) is running on this port
unknown (1024/udp)	Low	RPC program #100024 version 1 'status' is running on this port
sunrpc (111/udp)	Low	RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port
http (80/tcp)	Low	The following directories were discovered: /cgi-bin, /icons, /manual
sunrpc (111/tcp)	Low	RPC program #100000 version 2 'portmapper' (portmap sunrpc rpcbind) is running on this port