UNIVERSIDAD NACIONAL AUTONOMA DE MEXiCO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON

SISTEMA DE AUTENTICACION EN LA UNAM

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A N:

ARACELI BARCENA MONTIEL

ABELARDO IPARREA HERNANDEZ

ASESOR:

DR. ENRIQUE DALTABUIT GODAS

MÉXICO, D. F. JUNIO 2004

Agradecemos a todas las personas involucradas en este proyecto, en especial y con mucho cariño a nuestros padres, con gran admiración a nuestros maestros y por habernos cobijado y brindado una excelente formación académica, a nuestra querida Universidad Nacional Autónoma de México.

CONTENIDO

ÍNDICE DE TABLAS Y FIGURAS

TABLAS

FIGURAS

CAPÍTULO I *Seguridad Informática*

CAPÍTULO II *Procesos de Autenticación*

CAPÍTULO III *Autenticación con Certificados Digitales*

CAPÍTULO IV *Estándares de Seguridad.*

CAPÍTULO V *Desarrollo de una Autoridad Certificadora*

PALABRAS PRELIMINARES

La idea de desarrollar este proyecto de tesis tiene su origen en el curso llamado "seguridad informática" impartido en la ENEP Aragón, en el que nuestras primeras expectativas fueron conocer de forma general los conceptos básicos acerca del tema, para ampliar nuestros conocimientos como ingenieros en computación. Conforme avanzó el curso, nuestro interés aumentó a tal magnitud que decidimos dedicarle más tiempo al estudio de la materia y de alguna forma hacer algo con lo que pudiéramos retribuir todo lo que nos ha apoyado la universidad, de tal manera nació la idea de desarrollar una propuesta de un sistema de autenticación digital para que funcione en la UNAM.

La estructura de esta tesis se ha diseñado de una manera concreta con los temas necesarios para desarrollar nuestra propuesta, comenzando con conceptos básicos de seguridad, un análisis de la criptografía de llave pública y de sus aplicaciones, estudio de la autenticación con certificados digitales y por último la descripción del desarrollo de nuestro sistema de autenticación, con un estilo simple y directo, de una manera amigable para que cualquier persona que esté o no familiarizada con la materia, comprenda la justificación de esta tesis. Hemos aspirado a lograr una apariencia limpia y ordenada en la que los aspectos importantes destaquen con claridad.

OBJETIVOS

1. Realizar una investigación acerca de las tecnologías de certificación digital que se utilizan actualmente.

2. Realizar una comparación entre los algoritmos criptográficos más importantes y elegir el adecuado para nuestra propuesta.

3. Conformar un producto cien por ciento para la UNAM, basado en librerías criptografías y código realizado por alumnos de esta institución.

4. Desarrollar una herramienta de software que funcione como el corazón del sistema de certificación digital.

5. Llevar al cabo una propuesta para la implementación de un sistema de certificación digital para la UNAM.

INTRODUCCIÓN

Actualmente la información electrónica ha cobrado más valor e importancia de lo que tenía hace apenas algunos años y se ha constituido como la parte vital para la supervivencia de cualquier organización, sin embargo no estamos acostumbrados a cuidarla, dando oportunidad de que puedan manipularla a su libre albedrío entidades ajenas a ella. Por ejemplo, estamos preocupados por tener alarmas y cerraduras para proteger nuestros bienes, pero lamentablemente descuidamos la protección de algo intangible pero muy importante que es nuestra información.

Debido a las graves consecuencias derivadas de tener información desprotegida, los sistemas criptográficos aplicados a la computación, están ganando gran importancia ante el miedo de que una transmisión de datos pueda ser manipulada por algún oportunista y éste pueda tener acceso a información confidencial; hablamos de información importante, por ejemplo una transacción de miles de dólares; determinados temas empresariales que podrían ser benéficos para la competencia; o datos personales como nuestro número personal bancario.

Los sistemas criptográficos son las herramientas que ayudan a conformar sistemas de comunicación, almacenaje y tránsito seguros, sin embargo conviene aclarar que no existe la seguridad absoluta, es decir, siempre está presente el elemento de riesgo. La seguridad informática es un conjunto de técnicas encaminadas a obtener altos niveles de seguridad en los sistemas computarizados para garantizar confianza ante sus usuarios, donde sus principales objetivos son brindar protección a la confidencialidad, integridad, autenticidad y disponibilidad de la información..

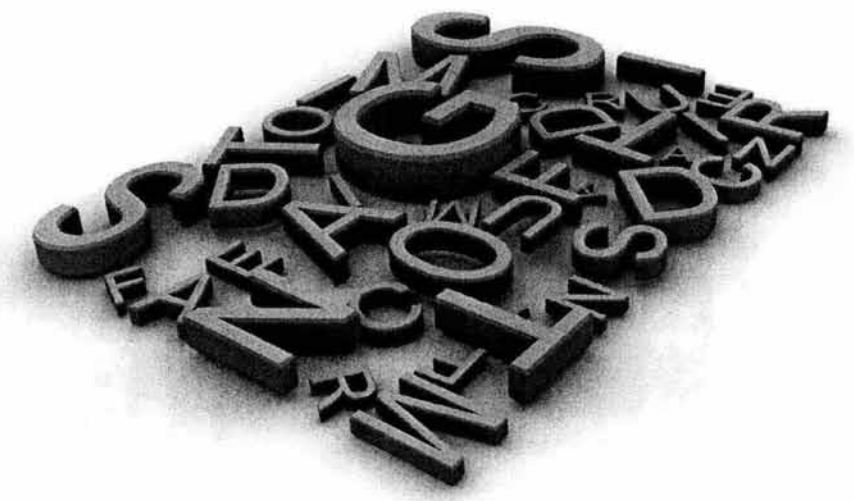
Ante este panorama, nos hemos visto preocupados por la seguridad de las comunicaciones dentro del país y en específico dentro de la universidad ya que actualmente la UNAM no cuenta con servicios de protección de datos disponibles para la comunidad

universitaria que se comunica a través de las redes internas y por internet, porque, si bien es cierto que algunos usuarios obtienen software gratuito para enviar y recibir sus archivos cifrados y de esta manera obtener algo de seguridad, también es cierto que no tienen la garantía de estar tratando con la persona correcta, produciéndose un fenómeno irregular de autenticación de partes. Algunos otros han comprado software para contrarrestar los efectos del uso y abuso de una red, como por ejemplo un firewall o un software detector de intrusos, sin embargo, son esfuerzos aislados y, en la mayoría de las veces, muy costosos que no podrán satisfacer al 100% las exigencias de un sistema inseguro que más bien necesita seguridad bien planeada y ajustada a las necesidades del entorno de implementación.

Este proyecto de tesis tiene la finalidad de cubrir estas necesidades de seguridad en la UNAM, proponiendo y desarrollando un sistema de autenticación y de cifrado de datos que se pueda distribuir entre toda la comunidad universitaria, y que ésta a su vez cuente con un servicio seguro, bien planeado y sin costo. Así pues en las siguientes páginas nos concentraremos en la seguridad dentro de un entorno de red, más que a la seguridad en computadoras aisladas, enfatizando el aspecto de la autenticación digital auxiliados por la tecnología criptográfica y por los avances en certificación electrónica. Debemos hacer notar que consideramos la información, esencialmente, en forma digital y que su protección se asegura utilizando medios lógicos, más que físicos, como el software que pretendemos desarrollar.

CAPÍTULO 1

Seguridad Informática



1.1 Objetivo de la Seguridad Informática

El concepto "Seguridad" por sí mismo, y de manera general implica que podamos tener confianza acerca de algo que esté expuesto a determinado riesgo, es decir, la seguridad debe garantizar que cualquier situación se encuentre fuera de peligro. Hoy nos encontramos invadidos por computadoras y por redes de ellas que controlan muchas, si no es que la mayoría de nuestras actividades diarias tales como hablar por teléfono, viajar por avión o revisar nuestro correo electrónico y otras no tan cotidianas como guardar secretos científicos, políticos o militares; hacer transacciones bancarias millonarias, en fin, las computadoras han propiciado que el mundo se vuelva cada vez más pequeño y que las posibilidades de intercomunicación humana crezcan, pero también se ha hecho presente que, para que el mundo moderno funcione se necesita la menor cantidad de errores en la operación y en el desempeño de los sistemas de cómputo que hoy día nos gobiernan; y esto se traduce en mayor responsabilidad para los administradores y creadores de dichos sistemas quienes en todo caso serían los culpables de una falla en éstos.

Sin embargo no siempre serán los administradores y los desarrolladores de sistemas los culpables de que algún error aparezca, o del mal funcionamiento de una máquina, o de la pérdida de información contenida en archivos; entonces, ¿porqué fallan los sistemas de cómputo? Es posible que para una persona cualquiera le resulte atractiva cierta información y dado que no tiene autorización para poseerla, hará lo posible para conseguirla así tenga que violar ciertas normas y medidas de seguridad. Para otra persona tal vez le resulte divertido destruir el nuevo software que instalaron en su trabajo; por otro lado a lo mejor es de vital importancia modificar información, clasificada como secreta, para boicotear alguna operación que comprometa ciertos intereses; en todo caso se trata de acciones llevadas a cabo por gente que persigue beneficio propio como personas enojadas, curiosos, criminales, personas vengativas, espías y hasta por personas que no necesariamente lo hacen por perjudicar, no obstante lo que tienen en común es que de una u otra forma buscan romper la seguridad que protege a los sistemas de información.

Hablando de computadoras es obvio que lo único que se tiene son a las propias máquinas y a los bits que circulan y son procesados por éstas, sin embargo traducido esto a nuestro mundo moderno hablamos de información. Así pues la seguridad informática tiene como objetivo garantizar y salvaguardar la confidencialidad, la integridad, la disponibilidad y la autenticidad de la información que se tiene almacenada en un sistema de cómputo o bien que viaja a través de ellos.

No hay que perder de vista que la seguridad informática lucha en contra de personas que hacen un mal uso de la tecnología y no en contra de la tecnología misma.

1.2 El Valor de la Información

Las computadoras nos han permitido almacenar grandes cantidades de información, ya no tenemos que utilizar los grandes y estorbosos archiveros llenos de papeles que contienen datos de nosotros o tal vez datos de alguna empresa; sin embargo así como las computadoras han logrado por sí mismas guardar millones de datos, las redes de computadoras han permitido que toda esa información sea potencialmente pública, es decir, manipulable por cualquiera que se encuentre conectado a la red y sin que se identifique necesariamente.

La mayoría de las instituciones de cualquier índole aun utilizan papel para operaciones internas tal como depósitos de dinero en el caso de los bancos, memorandos en oficinas, listas de pacientes en hospitales, etc., pero la información que de verdad vale por ejemplo los datos de un cuenta habiente en una institución bancaria (datos personales, saldo, operaciones mensuales) se encuentran almacenados en grandes bases de datos. Tan solo en los Estados Unidos de América los perfiles de individuos y de compañías se encuentran en bases de datos que controlan instituciones como el FBI (Federal Bureau of Investigation) o la NSA (National Security Agency) entre otras, y potencialmente esta información está al servicio de los atacantes que quieran sacar provecho de ello.

La información acerca de nosotros por supuesto que es muy valiosa, pues ¿a quién le gustaría que utilizaran su identidad para hacer un fraude? Nuestro dinero por si mismo es valioso y generalmente lo tenemos guardado en algún banco, ¿qué pasaría si alguien consigue gastar todo nuestro saldo? Estos son ejemplos en los que se observa cómo ahora la información ha cobrado valor debido a lo que representa dentro del entorno de la globalización computacional, y si esto es alarmante tratándose de un individuo, las implicaciones que tendría un ataque a la información, que pudiera comprometer la seguridad nacional, serían catastróficas.

Por lo anterior es necesario hacer una evaluación entre el valor de la información que deseamos proteger y el costo de hacerlo, para así poder decidir sobre qué mecanismos utilizar y qué políticas hay que seguir; por lo tanto es muy importante poder identificar qué tan valiosa es nuestra información según lo que representa ya que a final de cuentas solo son bits.

1.3 Amenazas

Una de las características que definen a un sistema de cómputo es la de ser inseguro lo cual quiere decir que se encuentra siempre bajo amenazas o que tiene una condición potencial de ser atacado. Hay que aclarar correctamente estos dos conceptos (ataque y amenaza) o por lo menos decir en que se diferencian para no confundirse. Un ataque es la realización de una amenaza mientras que una amenaza es el riesgo de ser atacado o dicho de otra forma es el potencial que existe de sufrir daños.

En un análisis de riesgos es importante conocer o saber categorizar a las amenazas que pudieran resultarle más importantes a un sistema determinado ya que de esta forma se pueden realizar esquemas de seguridad eficientes que beneficien al sistema.

Las dos primeras categorías de amenazas son: las realizadas por el hombre y las propias de un entorno ambiental, ésta se refiere a los riesgos que existen en un lugar físico en el cual se encuentra el sistema de cómputo, tales como temblores, huracanes, tormentas, erupciones volcánicas, etc. La categoría de realizadas por el hombre (que es la que nos interesa) incluye una serie de estilos para llevar a cabo a las mismas, que van desde infectar una máquina hasta tratar de modificar información ajena y secreta. Siguiendo con la clasificación encontramos dos subcategorías: amenazas de actividad hostil y de actividad no hostil. La primera engloba a aquellas amenazas que tengan un fin u objetivo criminal o malicioso; y la segunda se refiere a las amenazas que se dan por accidente, casualidad o por simple curiosidad, por ejemplo que quede desconectado un equipo debido a su mala ubicación. Continuamos y hacemos la siguiente subdivisión: amenaza sofisticada y amenaza no sofisticada. La amenaza sofisticada es la realizada por alguien que es experto y que conoce las debilidades del sistema al cual pone en riesgo, es decir, actúa con seguridad

y con conocimiento pleno de lo que hace; caso contrario es la amenaza no sofisticada en la que el posible atacante no sabe a ciencia cierta lo que hace. La última división es la siguiente: Amenazas externas e internas. Estas amenazas están sujetas a la ubicación del posible atacante. Si el sujeto amenazante se encuentra dentro de la red del equipo al cual quiere atacar entonces se considera como amenaza interna; caso contrario es la amenaza externa en la que el atacante bien puede situarse fuera del sistema al cual quiere atacar para actuar desde allí.

Combinando las seis subcategorías de amenazas realizadas por el hombre resultan ocho que se pueden tomar como definitivas para un estudio o análisis de riesgo.

- Hostil, sofisticada e interna: Las que puede realizar un administrador profesional de sistemas que, por ejemplo, se encuentre disgustado.
- Hostil, sofisticada y externa: En este tipo se puede mencionar el espionaje industrial por ejemplo.
- Hostil, no sofisticada e interna: Actividad de un empleado disgustado que no sea el administrador del sistema.
- Hostil, no sofisticada y externa: Un atacante ajeno al sistema que tiene el deseo de dañar pero que no tiene los conocimientos necesarios.
- No hostil, sofisticada e interna: Actividades propias de la curiosidad del administrador o bien que se hubiera presentado un daño por accidente.
- No hostil, sofisticada y externa: Posible ataque de una persona ajena y curiosa sin deseo de dañar.
- No hostil, no sofisticada e interna: Actividad de un usuario curioso que le parezca divertido adivinar contraseñas o borrar archivos.
- No hostil, no sofisticada y externa: Actividad sin intención que se puede interpretar como maliciosa ya que se efectúa fuera del sistema.

En cualquier caso, quien tome en cuenta esta clasificación para decidir qué sistema de seguridad le conviene, debe tener en cuenta cuál o cuáles de las amenazas son las que pudieran presentarse con más frecuencia o cuáles son las más peligrosas debido a la naturaleza de lo que se quiere proteger. Haciéndose esto se pueden obtener mejores resultados.

Por ejemplo en México como en muchos otros países las instituciones financieras ponen central interés en las amenazas de tipo internas, mientras que la seguridad militar nacional no tiene una estructura bien definida en la cual basarse para desarrollar un sistema de seguridad uniforme; en E.U, para diseñar un sistema de seguridad eficiente, las instituciones militares se preocupan más por las amenazas externas.

1.4 Ataques

Como ya se dijo en la sección pasada, estamos invadidos por las computadoras, lo cual quiere decir que tenemos un mundo tecnológicamente mejor controlado, en constante crecimiento, y con servicios más eficientes, pero también quiere decir que la delincuencia va en aumento. Entre más computadoras hayan conectadas entre sí, más crímenes cibernéticos existirán.

Hoy día se presta especial atención a los delitos informáticos simplemente porque han aumentado cada día más, porque cuestan mucho dinero y porque se vuelven más peligrosos cada vez. Todos estos delitos que se cometen a través de equipos de cómputo, para perjudicar a los mismos equipos y a la información contenida en ellos, son nada menos que amenazas consumadas, o sea ataques informáticos, mismos que se tienen bien catalogados según su naturaleza. Por ejemplo aquí en México hace un par de años se publicó una noticia en la que se ve involucrado un ataque informático, y esta dice así:

> "EFE (15-06-2000).- México, DF.- Un pirata cibernético trabaja actualmente intentando descubrir las claves secretas del disco compacto con las listas de los presuntos defraudadores de la banca mexicana que en 1999 reveló la auditoría del canadiense Michael Mackey, dijo hoy a EFE la diputada Dolores Padierna.
>
> - No sabemos cuándo se podrá conocer el contenido del disco, pero ya se está trabajando en ello, porque nos interesa revelar los nombres que están en esa lista -, afirmó la legisladora del opositor Partido de la Revolución Democrática (PRD).
>
> Padierna dijo que ya se contrató a una persona, aunque declinó precisar su identidad, ya que lo que más interesa es conocer el contenido de la información.
>
> En 1999, el canadiense Michael Mackey entregó los resultados de una auditoría que ordenó la Cámara de Diputados a todas las operaciones del "rescate bancario", que realizó el Gobierno a través del extinto Fondo Bancario de Protección al Ahorro (FOBAPROA)." [1]

Actualmente no sabemos, a ciencia cierta, si el hacker o pirata cibernético logró o no descubrir las llaves que protegen los archivos incriminadores (ojalá y lo hubiera logrado para poner en evidencia a varios actores políticos), lo que sí sabemos, que es lo de relevancia para este documento, es que esta acción constituye un ataque informático que se puede catalogar como "intercepción" pues lo que se hizo fue atentar contra la confidencialidad de la información ganando acceso a un recurso ajeno que fue interceptado, en este caso el disco compacto con la lista de implicados. Cabe mencionar que desde el punto de vista computacional el caso de la lista del FOBAPROA es un delito informático independientemente de criterios políticos y sociales.

Así pues se tienen cuatro categorías de ataques un tanto específicas: interrupción, intercepción, modificación y fabricación; y dos más que clasifican a los ataques de forma general: ataques pasivos y activos. En la *fig. 1.1* se muestra el panorama de una comunicación ideal la cual todos quisiéramos tener sin ningún riesgo.

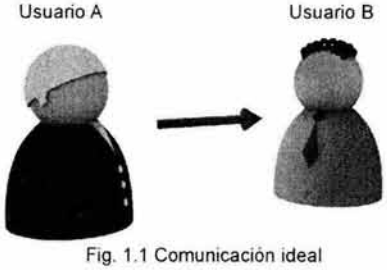


Fig. 1.1 Comunicación ideal

1 Agencia Noticiosa EFE. 15 de Junio de 2000, artículo publicado en www.terra.com.mx/noticias.

- Interrupción o denegación de servicio: Este ataque se produce cuando un recurso del sistema (llámese servidor, archivo, equipo, etc) es destruido o se vuelve inaccesible para los usuarios tal vez porque fue desconectado, modificado o desconfigurado. Este es un ataque en contra de la disponibilidad. Por ejemplo, si alguien corta la línea de conexión entre un servidor y una pc, ambos se convertirán en inaccesibles mutuamente. En otras palabras es el bloqueo total del flujo de información, situación que se muestra en la *fig. 1.2*.

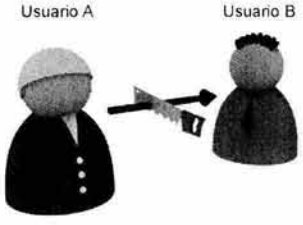


Fig. 1.2 Interrupción o denegación de servicio

- Intercepción: Como en el ejemplo del FOBAPROA, si una entidad no autorizada (persona, equipo o programa) consigue acceso a un recurso ajeno del sistema que circule u opere a través de una red, entonces hablamos de un atentado a la confidencialidad o sea una intercepción. *Fig. 1.3*.

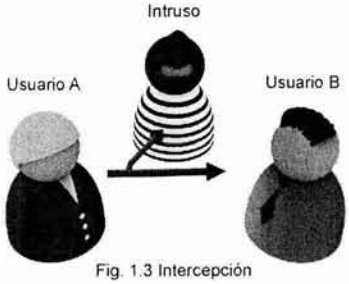


Fig. 1.3 Intercepción

- Modificación: Si una entidad no autorizada realiza un ataque de intercepción y además es capaz de manipular y cambiar la información contenida en el recurso atacado, entonces hablamos de modificación, es decir, un ataque a la integridad. Un ejemplo es el cambiar los datos de un archivo de contraseñas para beneficio propio y/o perjuicio de un tercero, *ver fig. 1.4*.

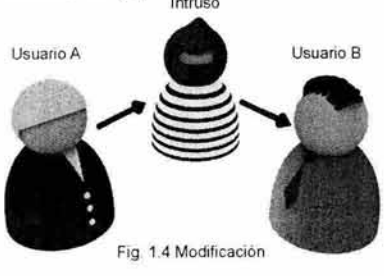


Fig. 1.4 Modificación

- Fabricación: Este ataque se da cuando se genera deliberadamente nueva información por una entidad no autorizada, y la pone en circulación enmascarándola como datos provenientes de una fuente segura. Es un ataque contra la autenticidad, esta situación se muestra en la *fig. 1.5.*

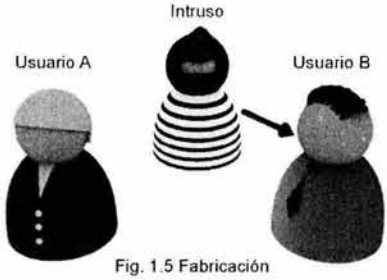


Fig. 1.5 Fabricación

- Ataques pasivos: En este tipo de ataques no se altera la comunicación entre entidades, más bien es una etapa recopilatoria de información para urdir un ataque más fuerte, es decir, un ataque pasivo se caracteriza por monitorear el flujo de información o el actuar de los usuarios para obtener datos que revelen un patrón del comportamiento de su víctima tal vez horarios de conexión, servicios que utiliza, datos sobre el origen y destino del flujo de información, volumen de datos que circula a determinadas horas, etc. Es decir sólo se analiza el tráfico de datos mas no se altera o dicho de manera popular, solo se "escucha" la transmisión.

- Ataques activos: A diferencia de los ataques pasivos, uno activo implica la alteración o modificación del flujo normal de la información, o bien crear nueva información que será un falso flujo de datos. También dentro de los ataques activos podemos encontrar, o más bien están incluidos los cuatro tipos de ataques mencionados anteriormente, sin embargo aún se puede hacer una subclasificación dentro de esta categoría:
    - Suplantación: El nombre o la idea es muy clara; una entidad no autorizada se hace pasar por otra que posea permisos para ganar acceso a ciertos recursos ajenos que le interesan.
    - Reutilización: Es exactamente eso, reutilizar. Si se logra capturar o interceptar un mensaje legítimo, entonces es posible que se pueda utilizar varias veces para producir un efecto no deseado. Por ejemplo, quizá un atacante logre capturar un objeto que le de crédito en una tienda virtual cada vez que lo usa, de esta forma en cada ocasión que el infractor necesite crédito utilizará el paquete robado.
    - Alteración de mensajes: Una vez capturado un mensaje legítimo, un atacante puede cambiar fragmentos del mismo para beneficio propio o simplemente para producir resultados diferentes en una comunicación. Por ejemplo tal vez alguien quiera dar de baja en el sistema al alumno X a través de una red, sin embargo la petición fue modificada y a quien se dio de baja fue al alumno Y.

- Degradación fraudulenta del servicio: El objetivo de este ataque es bloquear o entorpecer de alguna forma el flujo normal de información. Por ejemplo, muchos servicios de internet como el correo electrónico de Yahoo o las compras en Amazon se vieron afectados cuando uno o varios atacantes saturaron la capacidad de respuesta de estos sistemas logrando un ataque de denegación de servicio.

Todos los anteriores, son los ataques de los cuales hay que proteger a la infraestructura informática para que esta funcione bien y para que todo lo involucrado con ella siga sirviéndonos con eficacia.

1.5 Servicios de Seguridad

De manera opuesta a los ataques y a las amenazas existen los llamados servicios de seguridad los cuales son los puntos que se deben tomar en cuenta para defender a la información y a los recursos de una red o sistema informático.

Los servicios de seguridad tienen cuatro objetivos fundamentales diseñados para proteger a los datos y a los elementos de un sistema, estos son:

- *Confidencialidad*: Se debe asegurar que ninguna entidad no autorizada obtenga acceso a información que esté contenida o que circule a través de una red.
- *Integridad*: Hay que garantizar que ninguna entidad no autorizada pueda alterar o modificar a la información de un sistema de cómputo.
- *Autenticidad*: Garantiza que cierta información fue creada o modificada por una entidad autorizada y no por una fraudulenta y que además ciertos recursos fueron utilizados con permisos especiales.
- *Disponibilidad*: Le corresponde asegurar que un usuario autorizado pueda tener acceso a la información o a los recursos del sistema en el momento que los requiera.

Una vez descritos los objetivos para los cuales están diseñados, podemos ahora si listar los seis diferentes servicios de seguridad:

1) Control de acceso: Protege al sistema de cómputo del uso no autorizado de sus recursos. Por ejemplo, controlando el acceso a los equipos de impresión se puede evitar que trabajadores holgazanes impriman todo el día sus chistes favoritos que recibieron en su correo electrónico.
2) Confidencialidad: Protege a la información de ser revelada o publicada por usuarios o entidades no autorizadas.
3) Integridad de datos: Este servicio previene la alteración no autorizada o la destrucción de información contenida en un sistema de cómputo.
4) Integridad del flujo de información: Previene la recaudación de información acerca de la red o el sistema como resultado de la observación de las características del tráfico en la misma red, es decir, se evita que la red sea monitoreada con el objetivo de mitigar la posibilidad de detección de patrones en el flujo.
5) Garantía de servicio: se debe prevenir la denegación del servicio provocado por la degradación fraudulenta del mismo. Por ejemplo debe evitarse la caída de los servicios como correo electrónico, sesiones remotas, etc.

6) Autenticación: Establece de manera precisa la identidad de un usuario (ya sea persona, programa o máquina) y además previene que una operación sea rechazada por el emisor que participa en una transacción; a esto último se le conoce como no repudio. El servicio de autenticación asegura que la entidad que emitió cierta información o que utilizó ciertos recursos se encuentra correcta y confiablemente identificada. Este servicio es el de mayor importancia para esta tesis pues de lo que trata el proyecto es de autenticar a las partes involucradas en una transacción electrónica.

Como se puede vislumbrar cada objetivo o cada servicio de seguridad tiene de manera inherente una preocupación en específico, por ejemplo "la integridad de datos" se preocupa porque no se modifique la información de manera no autorizada, sin embargo un atacante no solo va a atentar contra la integridad sino que tal vez su objetivo es obtener acceso, modificar, suplantar y finalmente bloquear la capacidad de respuesta de un sistema, así que no es recomendable implementar estos servicios por separado pensando en que tal vez uno de ellos sí puede ayudarnos y otro no; lo que se debe hacer es tener la capacidad de analizar el entorno en el cual opera el sistema para así poder organizar un esquema de seguridad que involucre a todos los servicios y entonces sí lograr una red o sistema de cómputo lo más seguro posible. Un ejemplo de la interacción entre servicios de seguridad es aquel que combina el servicio de autenticación con otros servicios como control de acceso para lograr un eficiente sistema restringido.

Hasta aquí ya sabemos qué es lo que se debe tomar en cuenta para la protección eficiente de los recursos de un sistema, ahora necesitamos saber con qué elementos contamos para estar en posibilidad de brindarles defensa a estos.

1.6 Mecanismos de Seguridad

Los elementos con los cuales se pueden combatir los ataques informáticos y así proteger a los sistemas de cómputo son precisamente los mecanismos de seguridad, o entendido de otra forma son las herramientas con las que contamos para poder brindar los servicios explicados anteriormente.

Los mecanismos de seguridad más importantes son:

- *Intercambio de autenticación*: Durante una transacción a través de equipos de cómputo es muy importante saber y estar seguro de que la entidad con la que se está trabajando o comunicando sea la deseada, por ejemplo saber con certeza que se está tratando con Telmex y no con Avantel. De esto se encarga este mecanismo, corrobora que una entidad, ya sea origen o destino de la información, sea la que se piensa que es.

- *Cifrado*: Este mecanismo debe garantizar que cierta información sea ilegible para un atacante. Cifrar quiere decir crear mensajes en escritura secreta y se refiere a generar mensajes que sean incomprensibles para entidades no autorizadas. Generalmente a un mensaje en claro se le aplica cierta transformación que depende de un algoritmo y de una llave (que es el elemento de secrecía) para producir un mensaje cifrado o secreto. La llave, como ya se mencionó, es el elemento secreto que permite cifrar información (además del algoritmo por supuesto) y posteriormente descifrarla; básicamente existen dos tipos de cifrado cuya mayor efectividad depende de la aplicación a la cual está

destinada y al correcto uso. Cuando se utiliza una misma llave para cifrar y descifrar estamos hablando de cifrado simétrico o criptografía tradicional, pero si se usan dos llaves diferentes para cifrar y descifrar nos referimos a criptografía asimétrica en donde cada entidad posee un par de llaves relacionadas matemáticamente y que no pueden deducirse una a partir de la otra. Este par de llaves está conformado por una llave pública y por su contraparte privada; lo que se cifre con una solo puede ser descifrado por la otra (*ver fig. 1.6*), además como sus nombres lo indican, una de estas llaves es de dominio público, es decir, se comparte con varias entidades y la otra es privada, o sea que deberá ser administrada y usada sólo por el propietario y por nadie más para así poder garantizar una buena parte del problema de autenticación. Obviamente el hecho de utilizar una u otra técnica de cifrado depende de la aplicación para la cual se requiera la criptografía, del entorno de trabajo y de las amenazas que puedan existir alrededor del sistema de cómputo que se quiera proteger.

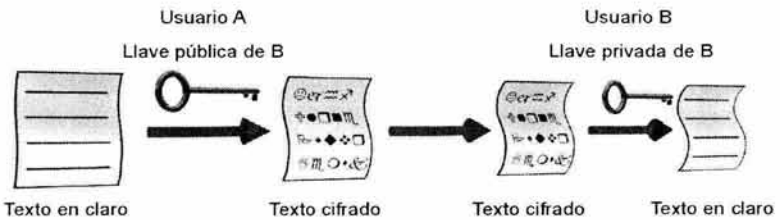


Fig. 1.6 Cifrado asimétrico

- *Integridad de datos*: Para explicar este mecanismo nos valdremos del siguiente ejemplo: imaginemos que deseamos enviar por paquetería un buen trozo de queso fresco de Puebla a Veracrúz, y que éste debe llegar integro a su destino. Para lograrlo lo que hicimos fue pesar el queso y adjuntar al paquete, en un papel, el resultado de la medición; supongamos que fue un kilogramo. El destinatario al recibir el envío en Veracrúz lee el papel adjunto y se apresta a pesar el queso para constatar que llegó completo comparando los resultados (el recibido y el calculado). Si ambos resultan iguales a un kilogramo quiere decir que el paquete no sufrió alteraciones en su recorrido, de lo contrario sabremos que alguien se comió una fracción del queso y que éste obviamente no llegó íntegro. De la misma forma, a la información que viaja a través de una red se le calcula un valor que le sirva como referencia al destinatario para comprobar que efectivamente recibió sin alteraciones sus datos. El valor del que hablamos es el ICV (Integrity Check Value) o valor de comprobación de integridad. Si el ICV que calculó el emisor es igual al que calcule el destinatario, entonces los datos no fueron alterados. Relacionados con el ICV se encuentran técnicas como el checksum y las funciones de resumen de mensajes (message digest functions).

- *Firma Digital*: A través de una firma digital se puede garantizar la autoría de ciertas acciones o de cierta información previniendo la situación del repudio. Una firma digital es análoga a la firma manuscrita ya que cuando firmamos un documento, como un cheque, se podrá comprobar si efectivamente es nuestra firma responsabilizándonos del hecho. En términos de cómputo una firma digital es la transformación de un mensaje utilizando una función de resumen (message digest function) junto con un esquema de

criptografía asimétrica; el resultado de esta transformación es una especie de sello que se le agrega al mensaje para que el destinatario pueda estar seguro de dos cosas: a) Que el sello se generó a partir de la transformación efectuada con la función de resumen y con la llave privada que corresponde a la llave pública del firmante; y b) Que los datos no sufrieron alteraciones ya que la firma se revisó exitosamente. Ambas cosas garantizan autenticación, no repudio e integridad de datos. Para generar una firma digital (nótese que una firma digital depende del mensaje a diferencia de la manuscrita que es independiente a él) se hace lo siguiente: a un mensaje que se pretenda firmar, es necesario aplicarle una función de resumen para obtener un ICV el cual deberá ser un valor único y de longitud fija no importando el tamaño del bloque de información. Cuando el ICV se obtiene con una función de resumen de mensajes (message digest function) se le llama valor de hash. El valor de hash es una versión condensada del mensaje original, también por eso se le llama resumen (muchos le llaman incorrectamente digestión). Este valor hash obtenido se procesa por un algoritmo de firma digital (Digital Signature Algorithm) que utiliza la llave privada del firmante para obtener finalmente la firma digital la cual deberá ir adjunta al mensaje que se desea enviar. Como comentario una función de resumen es un algoritmo matemático que no depende de llaves y que al procesar un mensaje, arroja una secuencia de bits de tamaño fijo.

- *Control de Acceso*: Ya sea que se utilicen contraseñas, permisos, listas de usuarios o técnicas más sofisticadas, con el control de acceso se pretende restringir la utilización de los recursos del sistema o de la red por usuarios no autorizados previniendo el mal uso de los mismos.

- *Tráfico de relleno*: Consiste en engañar o confundir al atacante que esté monitoreando o interceptando información que fluya a través del sistema; esto se logra agregando datos con ningún significado (basura) al flujo de información legítima o normal, de manera que el atacante no sabrá cuál es la información valiosa, dónde se encuentra o qué tanta está circulando.

- *Control de encaminamiento o router*: Permite enviar información a través de rutas elegidas y clasificadas como seguras, además de que se puede solicitar cambiar de camino si es que se detectaron irregularidades por una ruta determinada.

- *Unicidad*: Se refiere a que la información debe contener un elemento que la haga única como por ejemplo un número de serie, un número aleatorio, la fecha y hora, etc., para evitar la reutilización de los datos.

Los mecanismos poseen tres características importantes y fundamentales:

1) Deben tener información secreta como claves y contraseñas.
2) Deben contar con un conjunto de algoritmos que efectúen el cifrado y descifrado de información, que obtengan los valores de hash, y que generen números aleatorios.
3) Debe existir un conjunto de procedimientos que dicten cómo utilizar los algoritmos.

Estos son pues los mecanismos de seguridad; ya sabemos qué es lo que hay que cuidar, con qué lo vamos a hacer pero falta saber cómo lo lograremos.

1.7 Políticas de Seguridad

Según el RFC 1244, una política de seguridad es una declaración de intenciones de alto nivel que cubre la seguridad de los sistemas informáticos y que proporciona las bases para definir y delimitar responsabilidades para las diversas actuaciones técnicas y organizativas que se requieran.

En todo sistema, cualquiera que este sea, se hace imprescindible el trabajar bajo ciertas reglas y normas para poder actuar de manera precisa y coordinada, ya que de no hacerlo así se provocaría un caos en el área operativa. Más aun, en un sistema de seguridad deben estar perfectamente bien delineados los aspectos que se deben tomar en cuenta para un buen desempeño. Pongamos un ejemplo: es posible que en una escuela existan edificios de tres pisos con barandales altos para evitar caídas, personal de vigilancia, circuito cerrado de televisión y otras cosas que se nos pudieran ocurrir para proteger a los estudiantes; sin embargo, pese a todas estas herramientas de protección un alumno sube al tercer piso, trepa el barandal y se cae. Se pueden ver dos fallas rápidamente: una, la irresponsabilidad del estudiante, pues hizo algo indebido que al final lo perjudicó; y dos, la falta de previsión de las autoridades escolares. Los barandales no fallaron, seguramente el circuito de T.V. tampoco, lo que falló es la falta de responsabilidad de los usuarios (en este caso los alumnos) los cuales tienen las herramientas para estar protegidos pero las utilizan equivocadamente; y por otra parte también falló el esquema escolar falto de reglas y penalizaciones que impidan subirse y sentarse en un barandal, mas allá del sentido común. De manera análoga, en un sistema de cómputo también se deben dictar normas para la utilización correcta de la tecnología y de los recursos disponibles en un sistema, es decir, deben ponerse en práctica las políticas de seguridad.

Podemos decir que las políticas de seguridad son un conjunto de reglas, normas y protocolos que definen el camino a seguir para proteger la integridad y el buen funcionamiento de un sistema, además cualquier política debe tener en cuenta los puntos más importantes dentro de la seguridad informática que son: confiabilidad, integridad, autenticidad y disponibilidad.

Hablando de políticas de seguridad, no podemos dejar de mencionar que éstas también se encuentran perfectamente bien estructuradas en recomendaciones internacionales, o dicho de otra forma, existe un estándar en donde se describe la manera de establecer, planear e implementar a las políticas de seguridad; este estándar es el ISO 17799 el cual es un estándar de seguridad muy detallado y comprensible que se compone básicamente de diez partes muy importantes, que engloban a otras áreas de la seguridad, que son las siguientes:

1.- Planeación de continuidad del negocio: La meta de este apartado es el de poder contrarrestar los efectos que pudieran tener eventualidades, como fallas importantes o desastres, sobre el curso normal de las actividades de una organización o en aquellos procesos críticos de la misma.

2.- Control de acceso al sistema: Controlar el acceso a la información; prevenir el acceso no autorizado a los sistemas de información; garantizar la protección de los servicios que se ofrecen en un entorno de red; prevenir el acceso sin autorización a computadoras; detectar actividades no autorizadas; y garantizar la seguridad de la información cuando ésta se transmita por medios computacionales móviles o por medios a distancia.
3.- Desarrollo y mantenimiento de sistemas: Garantizar que los sistemas han sido desarrollados con seguridad incluida; prevenir pérdida, modificación o mal uso de datos de los usuarios en aplicaciones del sistema; proteger la confidencialidad, autenticidad e integridad de la información; asegurar que los proyectos informáticos se hayan realizado de una manera segura; mantener la seguridad en software, sistemas y en la información.

4.- Seguridad física y de entorno: Los objetivos de esta área son: prevenir el acceso no autorizado, daño e interferencia hacia los procesos y la información de alguna organización; prevenir que se comprometan o que se dañen los activos de un negocio y prevenir la interrupción de las actividades de la organización; prevenir el robo de información y que se comprometa la capacidad para procesarla.

5.- Acuerdos: Evitar caer en infracciones de cualquier tipo, ya sean criminales, de ley civil, regulatoria, o en general evitar cualquier violación a los requerimientos de seguridad; garantizar la integración de sistemas, políticas organizacionales de seguridad y estándares; maximizar la efectividad y minimizar la interferencia hacia y desde los procesos del sistema de revisión.

6.- Personal de seguridad: Este tipo de medida es para reducir riesgos de errores humanos, robos, fraudes y mal uso de recursos; garantizar que los usuarios estén informados acerca de las políticas y reglas de seguridad que se han implementado en la organización; minimizar los daños que pudieran ocurrir ante un incidente de seguridad y poder aprender de ellos.

7.- Organización de la seguridad: Administrar la seguridad de la información dentro de la organización; mantener la seguridad sobre la información acerca de la organización y sobre los recursos de procesamiento además de vigilar el acceso de terceras personas a activos importantes dentro de la organización; mantener la seguridad de la información cuando ésta es responsabilidad de otra organización.

8.- Administración de computadoras y de operaciones: Garantizar la correcta y segura operación de los recursos de procesamiento de información; minimizar el riesgo de fallas en el sistema; proteger la integridad del software y de la información; mantener disponibles e íntegros los procesos de comunicación y de procesamiento de información; garantizar la salvaguarda de la información en entornos de red y de la infraestructura en sí; prevenir daños a los activos de la organización y evitar interrupciones en las actividades de la misma; prevenir la pérdida, modificación o mal uso de información que se intercambie entre organizaciones.

9.- Clasificación y control de recursos: Mantener una apropiada protección de los recursos corporativos y garantizar que los activos de información reciban un inmejorable nivel de seguridad.

10.- Políticas de seguridad: El objetivo de éstas es proveer las herramientas necesarias para la administración, dirección y soporte de la seguridad de la información.

En el estándar, dentro de cada una de las secciones citadas arriba, se describe detalladamente la manera de lograr los objetivos planteados dando una visión amplia y adaptable a cualquier organización o proceso susceptible de manejar con políticas o normas de seguridad. Por último, hay que tener siempre en cuenta que todos los ataques informáticos son producidos por accidentes o por intenciones humanas, por lo cual la seguridad empieza y termina con las personas.

CAPÍTULO II

Procesos de Autenticación

2.1 Sistemas de Autenticación

En lo cotidiano existen actividades que por su propia naturaleza requieren de un proceso de autenticación para poder funcionar, es decir, necesitan que una entidad que opere como usuario o como participante de éstas, se identifique de manera positiva y con un alto grado de confiabilidad antes de obtener autorización para jugar ciertos roles dentro de dichas actividades; esto es, para poder accesar a un sistema restringido (cualquiera que este sea) es necesario efectuar un proceso de identificación, autenticación y autorización que llamaremos IAA. Por ejemplo, un sistema restringido es un aeropuerto; imaginemos el siguiente escenario: Luis quiere viajar por avión de México D.F. a Ceilán, antes Sri Lanka, por "Aerolíneas Ponchito" y supongamos que ya tiene su boleto personalizado (con su nombre y alguna otra información), así que, como ya es hora de abordar se acerca a la sala de espera. En la puerta de abordaje un oficial le solicita su boleto y le pregunta su nombre; nuestro amigo viajero le contesta que se llama Luis y le presenta su boleto. Acto seguido, el oficial le solicita que compruebe su identidad, así que le pide su pasaporte, su credencial para votar con fotografía y además la contraseña que le asignaron al comprar su boleto (es una línea muy estricta). Luis así lo hace y con sus documentos da validez a su identidad, es decir le da carácter de auténtico. El oficial satisfecho por haber comprobado que Luis es de verdad Luis, le permite el paso para que pueda subir al avión y emprender su largo viaje.

Sintetizando el ejemplo y rescatando las partes de relevancia, tenemos lo siguiente: el sistema restringido, como ya se dijo, es el aeropuerto; el servicio al cual se desea tener acceso es el de viajar en avión al extranjero; el usuario es Luis. Luis comenzó el proceso IAA cuando presentó su boleto y dijo su nombre, es decir, se identificó; cuando presentó ciertos documentos que certificaron su identidad (pasaporte, etc.) y dijo una contraseña, se autenticó; y cuando el oficial le otorgó permiso para abordar al avión Luis fue autorizado.

2.1.1 Proceso IAA

A continuación definiremos de manera formal los conceptos del proceso de identificación, autenticación y autorización.

2.1.1.1 Identificación

La identidad de una persona o cosa está definida por un conjunto de características propias que lo diferencian de sus semejantes; cuando alguien o algo presenta estas características como suyas ante algún sistema que lo requiera, se está llevando a cabo una identificación. Por lo tanto la identificación es el proceso que habilita el reconocimiento de un usuario en un sistema dado; *ver fig. 2.1.*

2.1.1.2 Autenticación

Autenticar se refiere a acreditar, certificar o bien legalizar a alguna entidad. Cuando alguien o algo se ha identificado ante un sistema restringido, es necesario verificar y confirmar que de verdad la identificación pertenezca a quien la presentó. Por lo tanto autenticar significa, a grandes rasgos, comprobar de forma positiva y con certeza la identidad de un individuo (*ver fig. 2.1*). El concepto es claro, sin embargo ¿cómo se logra esto? o ¿de qué manera se lleva a cabo este paso? Un proceso de autenticación se realiza

básicamente de cuatro formas dependiendo de las exigencias del sistema y de los usuarios; estas son: autenticación directa, autenticación indirecta, autenticación unilateral y autenticación mutua.

2.1.1.2.1 Autenticación directa

Cuando Luis, en el ejemplo anterior, tuvo que mostrar su boleto y dar una contraseña como parte de la verificación de su identidad, estaba entablando una autenticación directa ya que los únicos involucrados eran él y el sistema de aviación, mismo que le proporcionó antes su contraseña. Entonces entiéndase como autenticación directa cuando en el proceso de verificación de identidad estén involucradas sólo las dos partes interesadas. Esto se logra entre otras cosas con la implementación de un sistema biométrico, por ejemplo, un dispositivo de reconocimiento de rostro.

2.1.1.2.2 Autenticación indirecta

Luis, al tratar de dar validez a su identidad, además de la contraseña pactada entre ambas partes (la aerolínea y él), presentó también ciertos documentos que lo avalan como auténtico; estos documentos son: su credencial para votar y sus pasaporte. A su vez la credencial está avalada por el IFE (Instituto Federal Electoral) y el pasaporte es responsabilidad de la SRE (Secretaría de Relaciones Exteriores). Por lo cual estos documentos brindan cierta confiabilidad ya que las instituciones que los expiden están facultadas para hacerlo ya que son de tipo gubernamental y en general se puede confiar en ellas. Así pues, cuando uno se autentica usando un documento de esta naturaleza, se habla entonces de autenticación indirecta ya que para comprobar la identidad de algo o alguien se requiere la intervención de un tercero confiable (en este caso las instituciones que extienden los documentos). En un sistema de cómputo lo análogo a los documentos citados son los certificados digitales, los cuales están avalados por la Autoridad Certificadora que los expidió además de que brindan legalidad a la identidad de los individuos. Esta tesis pretende desarrollar una Autoridad Certificadora, es decir, un sistema de autenticación indirecta.

2.1.1.2.3 Autenticación unilateral

Esta forma de autenticación se refiere a que, en una operación, negociación o transacción en la que estén implicadas dos partes, sólo una de ellas será la que tenga que autenticarse. Un esquema de autenticación unilateral es por ejemplo cuando establecemos una sesión remota con un servidor usando la herramienta telnet, en la que sólo es el usuario quien se autentica y no así el servidor. El sistema de aviación de nuestro ejemplo ilustra la autenticación unilateral, ya que de cierta forma Luis confió en que de verdad estaba abordando un avión de aerolíneas ponchito y también aceptó el hecho de autenticarse él ante la empresa y que la empresa no hiciera lo mismo. Este esquema está basado en la confianza, es decir, aceptar que un sistema no se autentique ante los usuarios implica que su credibilidad es evidente o bien que el sistema no requiera de un alto grado de seguridad, sin embargo es deseable que la autenticación sea por ambos lados.

2.1.1.2.4 Autenticación mutua

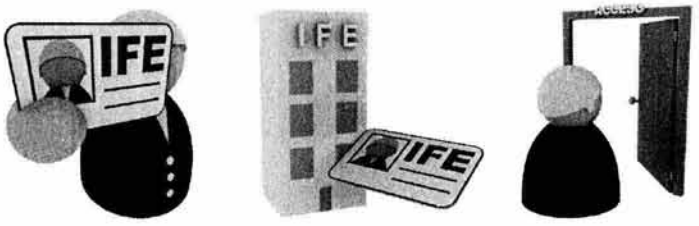
El concepto se presta para entenderse rápida y claramente. La autenticación mutua se refiere a la autenticación de forma recíproca entre las partes involucradas en una transacción, es decir que se realice de manera bilateral. Este esquema es muy importante ya que transacciones tan sensibles como las bancarias en internet, por ejemplo, necesitan confiabilidad en ambos lados, esto es, uno como usuario necesita estar seguro de que se está comunicando con su banco y no con otro diferente, mucho menos con una entidad pirata, y a su vez al banco le resulta imprescindible saber con certeza con cual de sus clientes está tratando. Sólo de esta forma se garantizará en gran parte que la transacción llegue a buen fin y sin percances.

Para concluir con nuestro ejemplo y dejar viajar a Luis en paz podemos decir lo siguiente: como ya sabemos las aerolíneas ponchito no existen, sin embargo Luis tiene un pase para volar con ellos; de hecho ya se identificó e incluso ya se autenticó ante esta aerolínea, presumiblemente fantasma, para poder viajar, pero nunca se preocupó por el hecho de que la empresa no se autenticara ante él. Es posible que ahora mismo se encuentre sentado en un avión de Aeroméxico y con destino a Australia, o en uno de American Airlines y con destino a Texas; en cualquier caso Luis pudo haber sufrido las consecuencias de no comprar su boleto en una línea que empleara un sistema de autenticación mutua. El ejemplo es muy fantasioso, sin embargo, casos en un contexto similar suceden constantemente (o al menos están latentes) en el ciberespacio que es donde hay que poner especial atención en el ámbito de la seguridad. Por cierto Luis si viajó a Ceilán y por aerolíneas ponchito.

2.1.1.3 Autorización

Diariamente somos autorizados para realizar diversas actividades; diariamente conseguimos introducirnos a algún sistema para efectuar alguna operación. Por ejemplo, cuando queremos saber el saldo en nuestra tarjeta de débito acudimos generalmente a un cajero automático en el que nos identificamos con la propia tarjeta al introducirla; nos autenticamos al digitar nuestro PIN (Personal Identification Number); y finalmente somos autorizados para consultar nuestro registro en la base de datos del banco. Esto es precisamente la autorización, es dar permiso o facultar a alguien o algo para tener acceso a un sistema en particular; *ver fig 2.1*.

En el ejemplo aeroportuario, la autorización se presentó en el instante mismo de que Luis fue autenticado, es decir, cuando obtuvo el paso al avión.



Identificación Autenticación Autorización

Fig. 2.1 Proceso IAA (Identificación, Autenticación, Autorización)

De esta forma quedan explicadas las fases del proceso IAA que se llevan a cabo en la mayoría de los sistemas restringidos, sin embargo, hay que notar que el pilar de este proceso o bien el eje de este procedimiento es nada menos que la autenticación, así que podríamos decir en vez de IAA simplemente proceso de autenticación ya que las otras dos fases no tienen sentido sin ésta.

2.1.2 Importancia de la Autenticación

¿Por qué la autenticación es lo más importante para obtener acceso a los sistemas restringidos?, imaginemos lo siguiente: una casa. Cualquier casa común y corriente es un ejemplo de sistema restringido aunque no tan estricto como debiera ser. Se supone que lo ideal es que sólo aquellas personas que habitan un hogar en particular son las únicas autorizadas para entrar en él, y más aun, ellas mismas pueden autorizar a alguien más para que ingrese. Esto implica básicamente dos cosas: una, que los habitantes tengan, cada quien, una llave que les permita ganar acceso a la casa; y dos, que los habitantes le abran la puerta a otras personas o bien que compartan una llave con ellos. Claramente podemos ver que en realidad es muy fácil entrar en esta casa ya que basta con tener una llave para poder accesar, así que, si un extraño encuentra un llavero con el elemento que le permita el acceso a este hogar, podrá entrar sin mayor complicación debido a que, aun sin ser una persona autorizada, posee el objeto clave. Así pues, si se implementara un sistema de autenticación en los hogares, no bastaría con tener la llave, sino que el que la posea deberá probar al sistema que realmente se encuentra autorizado para entrar dándole validez a su identidad.

Eso es precisamente lo que un sistema de autenticación persigue; tiene como objetivo identificar plenamente a las entidades que busquen acceso a un sistema de privilegios restringidos.

Un sistema de autenticación implementado en el mundo de la computación o en cualquier otro permite varias cosas: permite dar confianza a las partes involucradas en una transacción, ya que cada una sabe, con certeza, con quien está tratando; permite realizar registros de la actividad de las entidades que se autentican en un sistema; da lugar al hecho de responsabilizar a usuarios y prestadores de servicios por sus acciones en alguna operación; da pie a que se pueda rastrear a un individuo en particular debido a su historial existente en un sistema. Y de todos estos puntos, uno es el más interesante, el referente a responsabilidad ya que está fuertemente ligado con el concepto de "no repudio" o non-repudiation en inglés.

Repudiar quiere decir rechazar, desentenderse de algo, negar que algo se hizo; por lo tanto no repudio es un concepto que indica que una entidad no podrá negar que realizó

ciertas operaciones en algún lugar en el que fue identificado, es decir, no podrá rechazar que obtuvo acceso a un sistema pues existen ciertos elementos que lo implican. Estos elementos autenticadores que ayudan a evitar el repudio son: el certificado digital y la firma digital cuyas bases y buen funcionamiento están sustentados en la llamada infraestructura de llave pública (Public Key Infraestructure - PKI) que se explicará en el siguiente capítulo.

Un ejemplo básico de no repudio se presenta cuando el alumno le envía la tarea a su maestro por correo electrónico usando un esquema de criptografía de llave pública y de firma digital. De esta manera el alumno queda comprometido, ya que no va a poder negar que fue él quien envió el correo, pues su identidad va implícita en el mensaje debido a la firma digital y al cifrado asimétrico. Obviamente las técnicas y teorías que sustentan a un esquema de firma digital y/o de certificado digital no son, en lo más mínimo, básicas o fáciles de comprender, hay complejos estudios detrás de todo esto que permiten que los sistemas funcionen.

Después de todo, lo que se pretende lograr con un sistema de autenticación en un entorno informático, es reducir la posibilidad de ataques que lleven involucrados en sí un fraude o una mala utilización de los recursos que existen en una infraestructura determinada y también se busca que exista más confianza en los usuarios de sistemas de cómputo que busquen utilizar, por ejemplo, servicios como comercio electrónico o servicios en los que la seguridad lo sea todo para que marchen bien las cosas. En conclusión lo más importante es que los usuarios de sistemas sean cada vez más responsables a la hora de efectuar sus operaciones dentro de un sistema computarizado.

2.2 Metas de la Autenticación

Un buen sistema de seguridad debe ser capaz de ofrecer, entre otras cosas, un proceso de autenticación eficaz y confiable a sus usuarios. Centrándonos en los sistemas de cómputo debemos mencionar que con el desarrollo y apertura de internet y con la revolución digital que vivimos en estos días, la autenticación se ha convertido en uno de los puntos más importantes y sensibles dentro de la seguridad informática, tanto como lo es la secrecía. Básicamente la autenticación busca identificar plenamente (como ya se había mencionado) a cualquier entidad que pretenda introducirse en un sistema restringido, sin embargo en el mundo de la computación pueden mencionarse por lo menos seis metas u objetivos básicos y específicos de la autenticación independientemente del entorno informático en el que se esté trabajando. Estas metas son:

1) Prevenir la divulgación o publicación de información de manera no autorizada
2) Frustrar el análisis de trafico en el flujo de información
3) Obstaculizar ataques enmascarados en los que, información fraudulenta o no genuina, pueda ser insertada en el río de datos que fluye entre los participantes de una transacción.
4) Poder prevenir la modificación, eliminación, revisión o reordenamiento de la información.
5) Prevenir modificaciones referentes al tiempo en que se hace una transacción, es decir, se debe evitar que la información sea retrasada o reutilizada sin autorización, o bien que se modifique su registro de tiempo (timestamping).
6) Procurar que no suceda el repudio de acciones efectuadas por algún usuario.

Estos seis puntos corresponden a situaciones reales que suceden día a día en ambientes informáticos, por lo cual, se necesitan medios y herramientas de la misma naturaleza. Por ejemplo los primeros dos puntos pueden atacarse con medidas y técnicas criptográficas, es decir, con cifrado de información; los puntos 3,4 y 5 se pueden resolver en gran parte utilizando la autenticación de mensajes, con la cual si un mensaje es alterado, entonces perderá su identidad y podrá ser detectado como un mensaje no auténtico. La última meta, la sexta, puede ser alcanzada con la ayuda de un esquema de firma digital, misma que se explicará más adelante.

Tanto la autenticación de mensajes como la técnica de firma digital incluyen un valor de autenticación y un protocolo que habilitan a un usuario a verificar la autenticidad del mensaje o de la información recibida. Estos esquemas, se encuentran sustentados en tres funciones básicas de autenticación, mismas que a su vez soportan gran parte del proyecto de esta tesis; estas funciones de autenticación son:

1)Cifrado de mensajes
2)Checksum criptográfico
3)Funciones de hash

Ninguna de estas tres funciones proveen por sí mismas el servicio de autenticación al cien por ciento, sin embargo si se utilizan correctamente y de forma combinada se podrá obtener un sistema confiable.

2.2.1 Cifrado de mensajes

El cifrado de información ofrece en cierta medida confidencialidad y autenticación si es que se toman ciertas medidas y se actúa bajo ciertas políticas. Hay que tomar siempre en cuenta que se busca mitigar, lo más posible, los huecos o debilidades que se reflejan en los 6 objetivos mencionados antes; con esta referencia podremos calificar de bueno o malo el desempeño de una técnica en particular ante ciertas circunstancias, y también poder esquematizar cuáles funciones en combinación de qué otras, ofrecen los mejores resultados. Veamos la condición del cifrado tradicional.

En el cifrado simétrico o criptografía de llave secreta se utiliza una sola llave para cifrar y descifrar mensajes. Si van a establecer comunicación dos entidades X e Y utilizando cifrado simétrico, primero deben quedar de acuerdo y compartir una sola llave que será el elemento secreto en la comunicación, mismo que brinda cierto grado de confidencialidad ya que un mensaje cifrado con este elemento solo puede provenir de una de las dos entidades, quienes a su vez son los únicos poseedores de la llave pactada (claro esto es lo ideal). Si lo anterior lo consideramos correcto, entonces X, al recibir un mensaje cifrado proveniente de Y, sabrá que de verdad es Y quien se lo envía, pues sólo ellos conocen la llave; con esto se logra un mínimo grado de autenticidad de origen de datos. Sin embargo está comprobado por experiencia, que Y puede repudiar que él envió un mensaje ya que no hay una firma digital involucrada ni algún elemento que lo implique en la transacción; también es posible que un atacante, de forma deliberada, altere de alguna forma el mensaje, e inclusive pueda adivinar o investigar la llave secreta y de esta forma transmitirle a X información diferente de la que Y originalmente envió, sin que el destinatario tenga la posibilidad de saber si lo que descifró es realmente lo que el emisor mandó. Por lo cual el grado de desempeño del cifrado simétrico, como función

autenticadora, es limitado, ya que quien tenga la llave secreta podrá cifrar mensajes y enviarlos sin que necesariamente tenga que autenticarse, por ejemplo:

$$\text{Si } X \rightarrow Y: E_k(m) \text{ y}$$
$$\text{Si } B \rightarrow Y: E_k(m)$$

Suponiendo que m (el mensaje) y k (la llave) son iguales para ambos emisores X y B, Y estará imposibilitado para saber quién fue el que envió el paquete cifrado pues la llave está siendo compartida y de esta forma se ve otro grave problema al utilizar criptografía de llave secreta que es el intercambio de llaves.

El cifrado asimétrico o de llave pública (un tipo de criptografía mucho más fuerte que el tradicional) si se usa correctamente, logra confidencialidad y autenticación con un alto grado de seguridad, ya que entre otras cosas, logra desaparecer el problema antes mencionado de intercambio de llaves, pues del par utilizado en esta técnica, una es la que se comparte con todos y la otra es la que se mantiene secreta y utilizada solo por el propietario. Mencionamos que se debe usar correctamente porque por sí misma no funciona de manera eficiente. El concepto de criptografía asimétrica es fácil de entender (mas no el desarrollo ni todo lo que hay detrás de él); se refiere a que para cifrar información y luego descifrarla se utilizan, además del algoritmo de cifrado, dos llaves diferentes: una pública y otra privada, lo que cifre una descifra la otra. Una entidad X posee dos llaves: una llave pública (K_{UX}) y una llave privada (K_{RX}) y las utilizará para comunicarse de manera segura con otras entidades que a su vez poseen sus propios pares de llaves. Entre ellos comparten sus llaves públicas, no así las privadas que deben ser utilizadas solo por sus dueños. Si Y decide mandarle información a X, que sólo X pueda leer entonces cifrará el mensaje con la K_{UX} , de esta forma la información sólo podrá ser descifrada con la K_{RX} logrando tener confidencialidad. Sin embargo X no va estar seguro de quién fue el autor del mensaje, si fue Y o si fue B, ya que todos tienen su llave pública. Por lo tanto hasta este punto no existe la autenticación. No obstante si Y primero cifra el mensaje con su propia llave privada (K_{RY}) y después lo cifra con K_{UX} estará asegurando la autenticación y la confidencialidad, ya que la información sólo podrá ser leída por X, y a su vez X tendrá la certeza de que fue Y quien se la mandó pues resulta evidente que ésta cifró con su llave privada. Así pues lo ideal es lo siguiente:

$$Y \rightarrow X: E_{KUX}(E_{KRY}(m))$$

En donde K_{UX} brinda la confidencialidad mientras que K_{RY} brinda autenticación. Cabe mencionar que este escenario funciona, si y sólo si, los usuarios son responsables del uso de sus llaves, es decir, no deben exponer su llave privada; no la deben prestar, o en pocas palabras, la deben mantener secreta.

2.2.2 Checksum criptográfico

La segunda técnica de autenticación es el checksum criptográfico, el cual es una función de cifrado que utiliza una llave secreta para producir a partir del mensaje m, un bloque de datos de longitud fija conocido como MAC, siglas que significan "Message Authentication Code" (código de autenticación de mensajes). Este MAC se le agrega al

mensaje m original como si se le pegara con cinta adhesiva para así enviar ambas cosas (m y MAC) al destinatario. Por ejemplo X le envía a Y un mensaje m con su respectivo MAC (por supuesto X e Y previamente acordaron una llave); Y recibe el paquete que contiene el mensaje y el MAC, y procede a calcular el checksum, tal como lo hizo el emisor usando la llave secreta que tiene en común con X; el MAC obtenido por Y deberá ser igual al que le envió X, sólo así se logrará la autenticación del emisor ya que comparten una misma llave secreta, además si ambos códigos son idénticos quiere decir que fueron obtenidos sólo por X o por Y. También se logra identificar la integridad de los datos ya que si en el camino el mensaje es alterado, entonces el MAC obtenido por el destinatario será muy diferente al que recibió del emisor. Cabe mencionar que la probabilidad de que dos códigos, obtenidos por funciones de checksum, sean iguales es muy baja ya que dependen del mensaje, de la llave y del algoritmo utilizado que fue diseñado para evitar problemas de esta naturaleza. Confidencialidad también se logra desde el momento en que una llave secreta es compartida sólo por los involucrados en la comunicación, no obstante caemos en el mismo problema de la criptografía tradicional: el intercambio de llaves. El esquema de utilización de checksum se puede fortalecer utilizando el cifrado de paquetes (m y MAC) con criptografía simétrica, sin embargo es como un pleonasmo criptográfico en el que a final de cuentas se deben compartir llaves y esto implica que alguien las pueda interceptar para suplantar a alguien o simplemente para entrometerse en situaciones ajenas.

2.2.3 Funciones Hash

Funciones Hash o Message Digest. El propósito para el cual fueron diseñadas estas funciones, allá por los años 50s, era el de poder mapear bloques de información de tamaños variables, en bloques de datos más pequeños y de longitud fija que fueran como resúmenes de la información o bien versiones condensadas de la misma. Estos valores o resúmenes obtenidos son llamados valores de hash, mismos que se utilizaban principalmente para la detección de errores en la transmisión de datos. Al igual que el MAC (explicado antes) el valor hash se enviaba junto con el mensaje; al recibirlo, el destinatario recalculaba el valor hash y si era idéntico al recibido, entonces se podía considerar una transmisión exitosa, de lo contrario se sospechaba que un error había ocurrido. Obviamente si alguien intercepta la comunicación, bien podría suplantar el mensaje y recalcular el valor hash para después enviarlo al destinatario, el cual no se dará cuenta de que es un mensaje fraudulento ya que el valor que él calcule será igual al enviado por el atacante. Hasta aquí no se ha obtenido mucho con el valor hash ya que es muy similar al esquema del uso del MAC, sin embargo hay que ver ciertos factores que cambian el panorama. Las funciones hash actúan sobre el mensaje y obtienen un resumen sin necesidad de usar una llave o información secreta alguna lo cual mitiga la necesidad de intercambiar llaves; las funciones de message digest son conocidas como one way hash functions (funciones de una sola dirección) ya que a partir del valor hash no se puede deducir u obtener el mensaje original; y lo más importante, con el desarrollo de la criptografía de llave pública y con la necesidad de usar un esquema de firma digital, las funciones de hash se han convertido en algo muy importante y sustancial para este entorno.

El valor hash se puede utilizar de forma combinada con criptografía tradicional para buscar la autenticación del mensaje mismo y de los involucrados en la comunicación, sin embargo tropezaríamos con los problemas antes mencionados de suplantación, intercambio de llaves, alteración de datos, etc., o al menos existiría un potencial mayor de caer en ellos.

Así que la mejor opción es la que menos huecos presenta: utilizar las funciones de message digest junto con criptografía de llave pública. Al hacerlo así es posible la creación de una firma digital que nos garantizaría la autenticidad de los datos y del emisor, ya que al utilizar una función de hash y las ventajas de la criptografía asimétrica se puede generar una firma digital.

La firma digital para un mensaje m se obtiene aplicando la función, message digest, sobre m, es decir, H(m) lográndose así un valor hash de longitud fija sea cual sea el tamaño de m; este valor será cifrado con la llave privada del emisor (se podría cifrar también con una llave secreta compartida pero es inseguro) lográndose de esta manera la firma digital, la cual nos provee de autenticación de emisor y de datos; nos brinda integridad de información ya que el más ligero cambio en m afecta dramáticamente el valor hash; y también podemos decir que brinda confidencialidad ya que se usa un esquema de criptografía asimétrica. Una comunicación ideal es la siguiente:

$$X \rightarrow Y: E_{KUY}(m \parallel E_{KRX}(H(m)))$$

X le envía un mensaje m a Y, así que calcula el valor hash (H(m)) y lo cifra con su llave privada (K_{RX}) obteniendo la firma digital ($E_{KRX}(H(m))$). Esta firma digital la concatena al mensaje obteniendo un paquete de información que será cifrado con la llave pública del destinatario Y ($E_{KUY}(m \parallel E_{KRX}(H(m)))$). Y, al recibir este bloque de datos se encuentra segura de que la comunicación es confidencial ya que utilizaron su llave pública para cifrar la información y por ende sólo ella puede descifrarla pues es la única poseedora de su llave privada. De esta forma la confidencialidad está dada porque X cifró con K_{UY}. Enseguida encontrará un mensaje m en claro con el cual calculará el valor hash. La firma, como ya se dijo, garantiza la autoría del emisor ya que para cifrar el valor de hash, éste utilizó su llave privada, en este caso K_{RX}. Finalmente Y compara el valor hash obtenido con el recibido; si éstos concuerdan quiere decir que el mensaje es auténtico y que llegó íntegro. Así pues se logró autenticación, confidencialidad e integridad.

Estas tres funciones de autenticación, hay que repetirlo, son la base y los ladrillos con los cuales se puede construir un esquema de autenticación que se acerque bastante a lo óptimo, siempre en combinación armoniosa de otros métodos y técnicas.

2.3 Categorías de Autenticación

Los métodos y técnicas usados para llevar a cabo el proceso de autenticación en el mundo real, cotidiano y físico del ser humano son totalmente paralelos a los métodos empleados en el ciberespacio o campo de la computación, ya que están sustentados por la misma base, es más, la mayoría de las técnicas informáticas son tan solo la adecuación o adaptación de un método físico existente a un entorno nuevo computacional; ejemplo de esto es transladar el concepto de la firma manuscrita a un esquema de firma digital o llevar a medios digitales una carta notarial a través de un certificado digital; ambos casos son utilizados ampliamente en el ciberespacio y obviamente interactúan con el mundo real.

Las categorías de los métodos y técnicas de autenticación están dadas según el tipo de información que se utilice como base para un esquema en particular; existen cuatro categorías de información cuyos conceptos, respectivamente, sirven como plataforma para diseñar un sistema de autenticación. Estas cuatro categorías son:

1)Algo que se sabe
2)Algo que se tiene
3)Algo que se hace
4)Algo que se es

Para explicarlas de forma sencilla construyamos el siguiente ejemplo: supongamos que una pareja de enamorados, provenientes de Ceilán, saldrá a cenar y que el novio quedó en pasar por su amada a las 19:00 hrs. Justo a las siete, la novia muy entusiasmada escucha el timbre de su casa; se apresura a abrir y con lo que se topa no es, exactamente, con su novio sino con la encrucijada de su vida. En la puerta se encuentran dos hombres idénticos (o al menos eso parece) que aseguran ser el verdadero novio. Es una situación difícil, similar a la de alguna película hollywoodense, que de una u otra forma tendrá que ser resuelta rápido, pues las reservaciones están hechas para las 20:00 hrs. La novia se da prisa y lleva a ambos chicos a la sala donde comienza un extraño proceso de autenticación. Lo primero que se le ocurre es preguntarles su nombre, es decir que se identifiquen; ambos responden bien y dicen llamarse Luis (¿será el personaje del ejemplo anterior?). Enseguida, la novia, que por cierto se llama Ximena, tiene una idea y les pide que le cuenten o le digan una frase, una anécdota o algo que sólo ellos dos conozcan ya que de esa manera el impostor sería descubierto. Súbitamente y con una sonrisa de triunfo, el verdadero Luis se levanta y le dice las palabras que sólo ellos conocen; la novia se alegra pues cree haber resuelto el problema y al fin podrían ir a cenar; sin embargo no contaban con que el impostor había robado el diario de Luis y sabía varias palabras e historias que convencerían a la novia; el impostor se las dice y Ximena vuelve a su confusión. Usar la técnica de "algo que se sabe" no le resultó en esta ocasión.

Ximena de pronto recuerda que le obsequió a Luis un anillo exclusivo que sólo se podía comprar por pedido, es decir, en ninguna tienda de joyas o almacén departamental era posible conseguirlo, así que les pide que le muestren dicho regalo. Una vez más con espíritu de victoria Luis se acerca y le muestra el anillo. Lo que Ximena nunca supo es que cayó en las trampas de la mercadotecnia, pues se podían encontrar réplicas idénticas hasta en Bodega Aurrerá. El desenlace es anunciado, el impostor se apresta a mostrarle su copia del anillo dejando la situación igual que al principio. Al igual que su técnica pasada, ésta, la de "algo que se tiene" tampoco le funcionó.

La chica preocupada por que son ya las 19:30 se pone a pensar y se le ocurre que hay algo que Luis puede hacer y que los demás no, algo que lo diferencia de todos, similar al niño de primaria que impresiona a sus amiguitos jugando con su ojo y mostrándoles la parte posterior de su párpado; de esta manera Ximena le pide a Luis que realice su acto de danza clásica sosteniendo una piña sobre su cabeza. A Luis le resulta ridícula la idea pero al final cede y con lujo de maestría baila un fragmento de El Lago de los Cisnes con su piña en lo alto. El impostor se preocupa por la complejidad de la prueba, pero recuerda que llevó clases de ballet en la preparatoria y que cuando viajó a Brasil a un carnaval se le facilitaba eso de cargar fruta en la cabeza. Con un poco de dificultad logra bailar con la piña sobre su aplastado cabello, sin embargo no lo hizo con la fluidez y facilidad con las que lo hacía el verdadero Luis, así que deja dudas sobre su identidad. Esta vez la técnica de "algo que se hace" despertó sospechas en Ximena quien está casi segura de quién es su amado.

Finalmente y para rematar el caso, Ximena piensa en que, si bien Luis y el impostor tienen el mismo rostro y comparten un aspecto corporal similar, no necesariamente tendrían que ser idénticos, así que haciendo memoria, la confundida novia busca recordar algún

rasgo físico característico de Luis que no pudiera ser imitado por nadie más; algo que de verdad haga única a su pareja. Afortunadamente su novio, el verdadero, posee en la planta de su pie derecho un lunar en forma de triángulo que a su vez está atravesado por una cicatriz que se hiciera de pequeño. Sería el colmo si los dos tuvieran esta señal y lo peor, la cena se perdería, sin embargo afortunadamente Luis, el auténtico, le muestra dicho rasgo personal mientras que el impostor queda aturdido pues fue descubierto. Este método usado de manera efectiva fue el de "algo que se es". Finalmente la pareja fue a cenar y el impostor se quedó en casa tomando un café.

De esta forma quedan explicadas las cuatro categorías que sin embargo precisaremos a continuación:

- Algo que se sabe: Es información secreta y compartida por los participantes en una transacción, por ejemplo los passwords y passphrases (contraseñas) son elementos secretos. En computación esto es muy usado, por ejemplo, al iniciar una sesión de correo electrónico en donde se solicita un nombre de usuario y una contraseña para ser autenticado.

- Algo que se hace: Se refiere a información obtenida a partir de los patrones de comportamiento de un individuo (los cuales se consideran únicos), por ejemplo, a veces se logra identificar a un asesino en serie debido a su estilo de delinquir. En computación hay sistemas que utilizan esta información autenticadora, por ejemplo, existe un programa que logra obtener el ritmo y la frecuencia de escritura de una persona frente a un teclado. Si alguien se sienta a escribir en esa misma máquina y su perfil de escritura no coincide con el autorizado, entonces se considerará a esa persona como impostor y se le negará el acceso al sistema.

- Algo que se es: Los seres humanos tenemos rasgos característicos y únicos que nos hacen diferentes a todos los demás. Estos rasgos constituyen la información más genuina que un individuo pueda tener para ser autenticado. Hoy en día se utilizan cada vez más los dispositivos biométricos, los cuales echan mano de esta información para autenticar a los usuarios y de esta manera tener sistemas más confiables. Ejemplo de esto son las técnicas de reconocimiento de rostro, reconocimiento de huella digital, y el que al parecer es más seguro: el método de reconocimiento de retina.

- Algo que se tiene: Es información basada en posesiones que nos sirvan como objetos autenticadores tal como lo es una llave, una credencial o alguna cosa que nos logre autenticar frente a un sistema. El objeto más común y utilizado por todos los mexicanos para autenticarse es la credencial para votar pues constituye un documento que nos señala como auténticos, o bien, acredita nuestra identidad. Esta categoría es de relevante importancia pues en el comercio electrónico o en cualquier transacción a través de medios digitales, el certificado digital (una posesión virtual) juega un papel crucial a la hora de autenticar personas u otras entidades; además esta tesis como ya se ha mencionado busca generar certificados digitales para efectos de autenticación en comunicaciones importantes.

2.4 Técnicas de Autenticación

A partir de las cuatro categorías explicadas antes, se derivan varias técnicas específicas cuyo principal objetivo es autenticar entidades dentro del ciberespacio o del entorno de los sistemas de cómputo. A continuación se explican algunas de ellas:

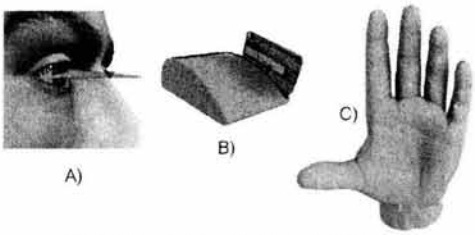


Fig. 2.2 Algunas técnicas de autenticación
A) Impresión de iris, B) Tarjetas magnéticas, C) Geometría de la mano

2.4.1 Passwords o contraseñas

Uno de los esquemas de autenticación típicamente usados, utiliza contraseñas. Un usuario necesitará brindar su nombre de usuario y su respectiva contraseña para ganar acceso a un sistema; el sistema deberá verificar estos datos y sólo dará acceso si es que recibe la contraseña correcta que es la que se espera. Una contraseña puede consistir de una sola palabra (password) o en un conjunto de ellas formando un enunciado o un párrafo (passphrase), en cualquier caso es recomendable crear una contraseña que sea fácil de recordar pero difícil de adivinar pues es un esquema muy sencillo de romper cuando se utilizan passwords típicos como nombres personales, fechas de cumpleaños, o nombres de mascotas. Lo ideal es tener una contraseña que sea sencilla para nosotros pero que no nos la adivinen fácilmente, por ejemplo, para construir un password fuerte podemos hacer lo siguiente: debemos crear un enunciado que nos resulte familiar y sencillo de recordar, algo como "los tres cochinitos se van a la cama" y a partir de él tomar las iniciales de las palabras que lo forman, inclusive combinarlas con números y obtendríamos lo siguiente: "l3csvalc" el cual es un password un tanto complejo para adivinar pero fácil para nosotros de recordar.

Un password, en los sistemas de cómputo actuales, está compuesto de, por lo menos, ocho caracteres, así que un atacante que trate de adivinar nuestra bien diseñada contraseña, tendrá que probar hasta 37^8 posibilidades para encontrar la correcta, considerando que hay 27 letras y 10 dígitos numéricos (sin tomar en cuenta caracteres especiales). De otra forma si escogiéramos nuestra fecha de cumpleaños como password, seguro sería adivinada en unos cuantos minutos.

El diseñar un buen password es importante ya que de no hacerlo se presentarían algunos inconvenientes como el tener que escribirlo en un papel para no olvidarlo y de esa manera exponer nuestra privacidad, así que, para no hacer del dominio público nuestra contraseña podemos seguir las siguientes reglas:

1) Nunca escribir en papel u otro lado visible nuestra contraseña.
2) Nunca revelar nuestra contraseña a terceros.
3) Cambiar regularmente la contraseña.

2.4.2 Cuestionarios

La mayoría de los proveedores de correo electrónico piden para autenticarnos un nombre de usuario y una contraseña, no obstante hay ocasiones en que olvidamos ésta; es entonces cuando aparece en el escenario una técnica auxiliar que es la pregunta secreta. Esta pregunta sereta con su respectiva respuesta constituye la esencia del método de cuestionarios. Para obtener acceso a un sistema, éste nos hará una serie de preguntas cuyas respuestas, presumiblemente, sólo conocemos él y nosotros, así que, al contestarlas correctamente obtendremos acceso. Las preguntas y respuestas tendrán que ser registradas en el sistema en el momento mismo de darnos de alta en él y además estos cuestionarios deberán ser creados por nosotros. Los cuestionarios, para que funcionen, deberán ser actualizados periódicamente y por cierto no son recomendables para usarse en entornos de alta seguridad.

2.4.3 Tarjetas magnéticas

Las tarjetas de crédito son un ejemplo de esta técnica. Son dispositivos fabricados con un material aislante y con cinta magnética para grabación pagada a ellos. Para la construcción de estas tarjetas hay un estándar en el que se especifica el tamaño, el peso, medidas de la cinta, áreas de grabación, etc., y fue desarrollado por el American National Standards Institute o ANSI (instituto americano de normas nacionales) en conjunción con el International Standards Organization o ISO (organización internacional de estándares). El documento en cuestión donde están detalladas todas las especificaciones es el ANSI / ISO – 7810 – 1985. Por ejemplo la tarjeta debe medir 8.56 cm de largo por 5.39 cm de ancho, y la banda magnética de grabación debe ser de aproximadamente 1 cm de ancho en donde se registrará información que ayude al proceso de autenticación. Obviamente un sistema que opere con tarjetas magnéticas debe poseer lectores de estas para que el circuito esté completo (*ver fig. 2.2 B*). El proceso de validación en un sistema de tarjetas magnéticas ocurre en cuatro etapas:

1) Entrada: El proceso comienza cuando un usuario introduce su tarjeta en un lector y posiblemente también un número de acceso, esto para verificar que el poseedor de la tarjeta sea realmente el dueño.
2) Cifrado: Para los efectos propios de la autenticación, en el lector (que es una computadora al fin y al cabo) se realizan ciertas operaciones con el número de acceso y con la información grabada en la propia tarjeta para producir un password cifrado.
3) Comparación: Si el password almacenado en la computadora (el lector) corresponde al generado en el paso anterior, entonces se procede a brindar acceso al usuario, de lo contrario se le negará.
4) Crear registros: El simple hecho de comenzar un proceso de autenticación con tarjetas magnéticas, es razón suficiente para comenzar un registro de actividades en el sistema, es decir, se logre o no el acceso, quedarán registradas las ocasiones en que se intentó obtenerlo, con el propósito de detectar comportamientos extraños en el uso de la tarjeta y también de rastrear a aquellas personas que la han usado. De forma complementaria, hoy en día en los cajeros automáticos (que son sistemas basados en el uso de tarjetas magnéticas) se instalan cámaras de video que graban a

aquellas personas que utilizan el sistema creando registros más fuertes y contundentes.

2.4.4 Smartcards o tarjetas inteligentes

El concepto de tarjeta inteligente es muy similar al de las tarjetas magnéticas; estas tarjetas también guardan información del dueño al igual que las otras pero hay algo que las diferencia. Las tarjetas inteligentes son creadas bajo la norma ANSI / ISO – 78106 / 1&2 y allí se especifica que en vez de utilizarse cintas de grabación, se utilizarán microchips, que además de almacenar información acerca de la identidad del usuario, podrán realizar ciertas operaciones que impidan la alteración de información, la duplicación y hasta comprobar la autenticidad de la misma tarjeta. Esto se logra debido a que en el chip se graban programas que puedan hacer cifrado de datos, revisar integridad o hacer cualquier tipo de cálculo que el fabricante desee implementar. Con las tarjetas inteligentes se puede lograr, por ejemplo, el uso de contraseñas "desechables", las cuales se utilizan sólo una vez, esto quiere decir que un lector o sistema auxiliar le proporciona un número aleatorio a la tarjeta, y ésta lo procesará y obtendrá una contraseña desechable la cual se compara con la calculada por el lector, si ambas concuerdan se otorga acceso de lo contrario se niega. La tarjeta inteligente utiliza para generar la contraseña, un algoritmo, el nombre de usuario del propietario, un código de acceso adicional (que tendrá que teclear el usuario), y un dispositivo de autenticación (el lector o computadora del sistema) constituyendo así un sistema de autenticación casi inviolable pues las contraseñas cambian cada vez. Claro está que la tarjeta deberá ser utilizada sólo por el usuario autorizado.

2.4.5 Calculadoras

Las calculadoras de seguridad, son dispositivos pequeños que asemejan a una calculadora convencional con la diferencia de que las primeras realizan cálculos diseñados para interactuar con un sistema de autenticación. Esta técnica consta de dos partes: la calculadora y, su complemento, el sistema de seguridad. Cuando un usuario pretende autenticarse utilizando este dispositivo, lo que debe hacer es identificarse ante el sistema, a lo cual el sistema le responderá con una clave generada en ese momento; el usuario deberá introducir dicha clave en su calculadora y ésta le arrojará una nueva clave de acceso, misma que deberá ser presentada ante el sistema de seguridad. Si esta clave coincide con la calculada por el sistema maestro entonces el usuario obtiene el permiso, si no concuerdan será expulsado. Este esquema es parecido al de contraseñas desechables de las tarjetas inteligentes.

2.4.6 Métodos biométricos

La biometría es la aplicación de métodos estadísticos y de cálculo de probabilidades al estudio biológico de los seres vivos; se encarga de examinar detenidamente y de tomar mediciones de las características biológicas de un individuo. Con los avances en la tecnología se ha vuelto factible la posibilidad de usar medios biométricos en sistemas de autenticación de alta seguridad. Los métodos más conocidos son por ejemplo la utilización de patrones de voz, reconocimiento de huella digital, geometría de la mano, el reconocimiento de rostro y el reconocimiento de retina. Estas técnicas están por encima de

otras por tres razones fundamentales: 1) Los patrones biométricos son únicos; 2) Son muy difíciles de falsificar o duplicar; 3) Siempre están en posesión del usuario, no es algo que se pueda extraviar.

2.4.6.1 Impresiones de voz

Las impresiones de voz son representaciones matemáticas de los modelos obtenidos a partir de la forma de hablar de un individuo; estos patrones están determinados por la frecuencia, tono y amplitud de la señal producida por la voz. En una computadora maestra se registran los datos personales del usuario más el vector que describe su voz para poder utilizarlos más tarde al momento de querer autenticarse. El proceso de verificación de identidad con esta técnica se lleva a cabo en cuatro fases:

1) Pre-procesamiento: En este paso el usuario se identifica y le solicita al sistema permiso de acceso. El sistema por su parte selecciona el patrón de voz que usará para autenticar al usuario en concordancia a la identidad que éste aseguró.
2) Muestreo: El sistema le solicita al usuario que le diga con voz clara su nombre y su código de acceso para poder grabar y procesar muestras de su voz. Si se obtiene una muestra clara, entonces se procesará y se obtendrá un patrón de voz.
3) Probar: El sistema de seguridad usará la información que el usuario le dijo además del patrón de voz obtenido recientemente, para compararla con la que tiene almacenada. De esto depende el acceso.
4) Respuesta del sistema: Si los registros son iguales se otorga el acceso, de lo contrario será negado. Además de esto el sistema registrará todos los intentos por autenticarse en él, ya sea que hayan sido exitosos o fallidos.

2.4.6.2 Huellas digitales

El proceso de autenticación con huellas digitales se refiere a otorgar o negar acceso a un usuario debido al resultado en la coincidencia, de la forma, de su impresión digital. El proceso es el siguiente; debe existir un sistema de seguridad que tenga almacenadas las representaciones matemáticas de las huellas digitales de sus usuarios, para entonces seguir los siguientes pasos:

1) Obtener información: esto implica que el individuo brinde algunos datos de su identidad, como lo son su nombre de usuario y tal vez un código de acceso además de otorgar una muestra de su huella digital que será obtenida por un explorador de imágenes (scanner) especial que deberá tener el sistema maestro. El sistema sabrá con cual registro trabajar.
2) Limpiar la huella digital: Una vez que el sistema tiene la muestra de la impresión digital, la convertirá a una forma matemática que pueda manejar y de esa forma obtener sólo la información relevante que se necesita de una huella.
3) Extracción de rasgos: En este paso se construye una representación matemática que exprese los rasgos característicos y únicos de la huella digital.
4) Comparación: Si la huella obtenida en el paso anterior coincide con la almacenada y además los datos que presentó el usuario son validados, entonces se estará en una posición de otorgar o negar el acceso al sistema.

5) Respuesta del sistema: Según el resultado de la comparación se da o se niega el acceso, además de que se registran todos los intentos, fallidos y exitosos, de autenticación en el sistema. Por cierto las huellas digitales son únicas e imposibles de falsificar.

2.4.6.3 Impresiones de retina

Esta técnica está basada en el modelo obtenido a partir de los vasos sanguíneos que se encuentran en la retina de un individuo. Al igual que las huellas digitales, los patrones encontrados en una retina son imposibles de falsificar y resulta imposible encontrar duplicados, así que también constituyen una técnica fuerte de autenticación.

Para el proceso de autenticación con impresiones retinales se necesita, al igual que en las técnicas anteriores, tener almacenada previamente una representación matemática de éstas, en un sistema central de seguridad. De esta forma cuando un usuario desee obtener acceso al sistema, se procederá a tomar lectura de su retina, obtener una representación digital de ésta y finalmente compararla con la almacenada para así ganar acceso o ser rechazado, cualquiera que sea el caso.

2.4.6.4 Impresiones de iris

En la biometría del iris se analizan las características encontradas en la membrana de color que rodea a la pupila. La exploración y la obtención de los patrones del iris se realizan mediante una cámara digital convencional y no requiere contacto intrusivo del aparato lector con el individuo (*ver Fig. 2.2 A*). Por el momento la fortaleza de estos sistemas, a nivel comercial, no está en la facilidad de uso ni en la integración con otros sistemas, aunque se espera que el desarrollo de esta tecnología siga evolucionando y muy pronto se integre al equipo de sistemas biométricos usados ya, casi de forma habitual.

2.4.6.5 Geometría de la mano

Este también es un método biométrico en el que se obtiene acceso a un sistema si es que concuerdan las mediciones obtenidas al quererse autenticar con las mediciones almacenadas en el sistema. Esta técnica utiliza como datos, el tamaño de la mano, espacio entre los dedos, pliegues de la piel, líneas en los dedos y en la palma, entre otras cosas (*ver fig. 2.2 C*); sin embargo su uso es poco práctico ya que existe una gran variedad en los tamaños de las manos de las personas y eso se traduce en un problema a la hora de fabricar un dispositivo de lectura de mano que pueda ser estándar para toda la gente.

En un futuro no muy lejano se podrá autenticar a las personas con dispositivos biométricos que utilicen, como información de autenticación, a los patrones únicos e inigualables que se encuentran en el ADN de las personas, tal como sucedía en una película futurista llamada GATTACA. Ésta habla de un futuro en el que la discriminación no se basa en el color de la piel, la religión o la educación. Con el avance de la genética la discriminación radica ahora en la perfección genética. De esta manera existen seres "inválidos", los concebidos naturalmente y los "válidos", quienes fueron perfeccionados de forma artificial en su código genético. Entre otras tantas situaciones que se presentan en la película, existe la nueva forma de identificarse cuando, por ejemplo, los seres perfectos

van a su trabajo y se autentican a partir de muestras de sangre en las que se analiza el genoma de los trabajadores del futuro. Así pues GATTACA es una cruda película en la que se refleja el deseo permanente de perfección en la raza humana. [2]

2.4.7 Autenticación por posicionamiento

Con la llegada y desarrollo de los sistemas de posicionamiento global o GPS (Global Positioning System), se ha creado un novedoso criterio que sirve para la autenticación, este criterio se refiere a "dónde se está". Un GPS está conformado por un dispositivo móvil y por una red satelital con la cual interactúa. Los satélites pueden determinar la posición casi exacta en el planeta de los dispositivos GPS debido a señales que rebotan entre sí y que además son corregidas por complejos algoritmos y sistemas de información.

¿Pero, cómo usar esta técnica como una de autenticación? Para que este método funcione, es necesario que el usuario o la entidad a ser autenticada tenga una posición fija y predeterminada en el planeta, de esta forma si se detecta una posición global no autorizada, entonces se sospechará que existe un ataque; si la posición es la correcta entonces el usuario se puede considerar como autenticado.

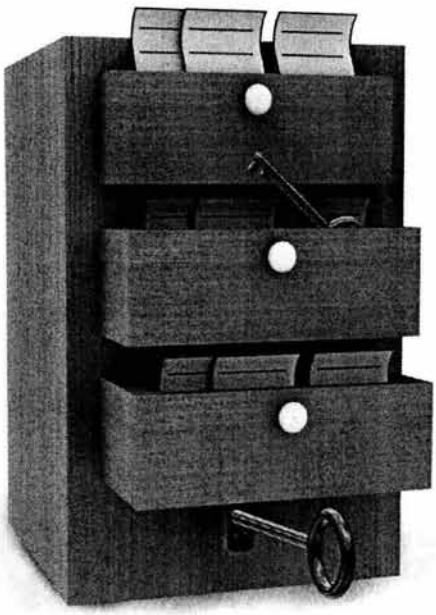
2.4.8 Autenticación con certificados digitales

Este método es el más importante para esta tesis, así que será explicado extensamente en el capítulo siguiente.

2 GATTACA. E.U. 1997. Director Andrew Niccol con Ethan Allen y Uma Thurman.

CAPÍTULO III

Autenticación con Certificados Digitales



Introducción

En el mundo físico y tangible, las personas y las instituciones necesitan conocerse antes de realizar cualquier tipo de negocio o transacción; éstas confían en credenciales físicas para identificarse y asegurarse que la otra parte es de confianza y así realizar el negocio. Similarmente, en el mundo digital se requieren de métodos con la misma funcionalidad que los tradicionales para llevar al cabo cualquier tipo de transacción con seguridad.

Independientemente de que las transacciones se realicen en línea o en el mundo físico, las partes involucradas deben contestar a estas preguntas: ¿Quién eres? y ¿Cómo puedes probar tu identidad?, cuestionamientos que trataremos de responder nosotros, de manera que quede claro todo el proceso de autenticación digital y que se comprenda la importancia que tiene, el mismo, en este proyecto.

Antes de realizar una transacción, debe establecerse la identidad de las partes involucradas, es decir, debe existir un proceso que verifique que una empresa, individuo o entidad cualquiera, exista, para ello están las terceras partes o entidades de confianza (como las autoridades de certificación o AC), que tienen el papel de respaldar la identidad de las personas que participan en una negociación. Una vez que la identificación se establece y verifica, se puede emitir una credencial que podrá usarse para probar una identidad; en el mundo físico estas credenciales podrían ser un pasaporte o una licencia de conducir, en el mundo digital, la credencial más completa es el certificado digital. Podemos decir entonces que cualquier persona que desee realizar una transacción electrónica debe establecer su identidad y presentar credenciales que la avalen frente a cualquier participante del intercambio.

En este capítulo se trata el tema de la autenticación de entidades, tomando como principal herramienta al certificado digital; se describirán los fundamentos necesarios para establecer y crear relaciones de confianza en el mundo informático; también veremos detalladamente cómo funciona cada uno de los elementos que nos permiten lograr la autenticación digital (certificados digitales, autoridades certificadoras, etc.); y finalmente recordaremos, como punto importante, que todo este proceso de autenticación esta basado en la criptografía de llave pública que ya conocimos en los capítulos anteriores. Así pues como primer propósito se explicará y discutirá el concepto de infraestructura de llave pública así como los elementos y esquemas básicos que la forman.

3.1 PKI - Infraestructura de llave pública (Public Key Infraestructure)

El uso cada vez mayor de las redes computacionales para efectos de comunicación, requiere de mecanismos de seguridad que permitan brindar, a cualquier persona ó institución (como puede ser la UNAM), servicios a distancia sin comprometer datos sensibles. Conscientes de ello, pretendemos desarrollar para la UNAM una infraestructura de llave pública basada en los estándares actuales (X.509, PKCS, etc.) para proporcionar certificados digitales a todos los miembros de su comunidad de forma fácil y eficaz, obteniendo así comunicaciones seguras, íntegras y confidenciales.

Típicamente las PKI's están constituidas (entre otras cosas) por certificados digitales, autoridades de certificación (AC) y autoridades de registro (AR), y su seguridad está basada en la criptografía de llave pública. En la PKI que pretendemos desarrollar, se

tienen que definir y establecer todos los métodos necesarios para generar los certificados digitales de forma óptima y de esta forma garantizar la confidencialidad de la información, la integridad del documento, la verificación de la identidad de las partes y la procedencia de la información.

3.1.1 ¿Qué es una PKI?

El conjunto de hardware, software, usuarios, administradores, políticas y procedimientos necesarios para crear, administrar, almacenar, distribuir y revocar certificados digitales basados en criptografía de llave pública, define y constituye a una PKI (Public Key Infraestructura); este concepto o definición a simple vista es muy claro y se percibe relativamente sencillo, sin embargo implementar una PKI supone tomar ciertos retos complejos que bien pueden parecer desalentadores.

La idea básica de una PKI es proteger información a través del cifrado de datos dentro de un entorno de telecomunicaciones, en donde cada usuario que participe en el intercambio de mensajes, posea dispositivos de software que le permitan el cifrado y descifrado de información además de contar con un par de llaves asimétricas: una pública que podrá intercambiar y repartir libremente entre su comunidad, y otra privada que deberá estar al resguardo y exclusiva utilización del legítimo dueño.

Un usuario para comunicarse de forma segura con otra entidad, cifra la información con la llave pública del destinatario y cuando éste recibe el mensaje cifrado, lo puede descifrar únicamente con su llave privada. De esta forma es muy posible y usual que los usuarios posean más de un par de llaves, tal vez uno por cada grupo de comunicación en el que participen, para establecer comunicaciones discretas con varias entidades. Lo anterior, a mediano plazo, supone un problema de administración de llaves para los usuarios ya que cuando su grupo de contactos crezca, también crecerá en la misma medida el número de llaves que tengan que manejar, por tal motivo aparece la necesidad de utilizar algún método para administrar las llaves y el uso de las mismas, es aquí donde hace su aparición la PKI, la cual habilita la posibilidad de crear, distribuir y administrar, de manera centralizada y segura las llaves de los usuarios a través de canales inseguros. Esto se logra con un sistema que es capaz de distribuir, verificar e intercambiar documentos electrónicos firmados, expedidos por una entidad distribuidora conocida como Tercero Confiable (TTP – Trusted Third Party), así pues, un usuario puede obtener y verificar la validez de una determinada llave debido a que la TTP la está avalando vía un certificado, el cual es el documento firmado del que hemos hablado, y es la base de funcionamiento de toda PKI.

De manera formal y retomando puntos anteriores, al documento firmado cuyo objetivo es distribuir las llaves públicas de los usuarios, se le conoce como certificado digital, mismo que es firmado y emitido por un tercero confiable o entidad conocida como Autoridad Certificadora (AC), la cual liga ciertas propiedades con un sujeto que es identificado de determinada manera según el contexto y el tipo de certificado utilizados. Por ejemplo, en una PKIX (PKI basada en certificados X.509) el sujeto es identificado por una estructura de datos llamada "Nombre Distinguido" (Distinguished Name), y éste es ligado a la llave pública del mismo sujeto, estableciéndose así un vínculo inseparable entre el usuario y su propia llave, todo esto avalado, mejor dicho certificado por la AC.

En el presente proyecto de tesis pretendemos establecer un esquema de PKI usando la tecnología X.509.

3.1.2 Características y elementos que forman una PKI básica

Varios factores de la inminente revolución digital han propiciado que, de pronto, la gran mayoría de los esfuerzos tecnológicos se centren en el desarrollo de técnicas que permitan el correcto funcionamiento de "nuevos" esquemas de trabajo y de negocios tales como comercio electrónico, correo electrónico seguro (S/MIME), conectividad de extranets (VPN), autenticación para acceso remoto a recursos, etc., en los que la tecnología PKI constituye una solución elegante a los problemas de autenticación, confidencialidad e integridad que se encuentran siempre presentes en entornos de información electrónica, ya que una PKI hace que la seguridad basada en criptografía asimétrica sea fácil de administrar y de usar, aunque pueda parecer compleja de implementar.

La "I" de las siglas PKI que significa infraestructura, es un concepto un tanto difuso e inmaduro ya que éste depende en gran medida de las necesidades presentes en un entorno específico de implementación, es decir, en un mundo ideal podríamos tener sólo un modelo estándar que sirviera en todo lugar, no obstante existe una gran cantidad de estándares (algunos aun en evolución) que recomiendan ciertos elementos que deberían estar presentes en cualquier PKI. Sin duda estos estándares son cruciales en la elaboración de una PKI pero en la mayoría de las veces es más práctico y funcional hacer caso a las demandas específicas del entorno de desarrollo; es por ello que a continuación explicaremos brevemente los elementos que a nuestro parecer existen en toda PKI; estos son:

- Sistema de autenticación.
- Directorio central.
- Modelo de confianza y rutas de certificación.
- Políticas de uso y administración.
- Administrador u oficial de seguridad.
- Usuarios

De los cuales podemos construir un modelo básico de una PKI genérica que sea compatible con todos los tipos de PKI. *Fig 3.1*

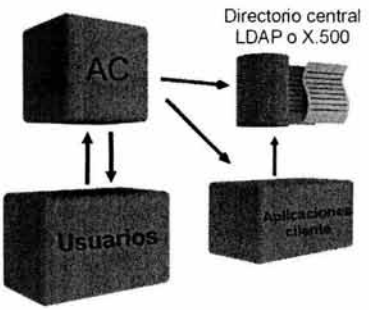


Fig. 3.1 Modelo básico de una PKI

3.1.2.1 Sistema de Autenticación

El primer y más importante paso dentro de una PKI es el establecimiento de un sistema de autenticación que permita a los usuarios identificarse positivamente antes de obtener algún derecho o permiso dentro de su entorno de trabajo además de poder autenticar a otros usuarios que deseen establecer comunicación. Hoy día aun se utiliza la identificación basada en contraseñas, sin embargo en una PKI se utiliza un método más seguro y eficiente llamado autenticación con certificados digitales. Cada certificado contiene información precisa y contundente acerca de la identidad de cierto usuario además de que dicha identidad está ligada a una llave pública y a una firma digital. Para obtener un certificado digital, un usuario presenta un requerimiento ante la autoridad registradora autorizada, la cual verificará la autenticidad de los datos y le solicitará a la AC la expedición de un certificado.

Un certificado es un archivo digital con el cual se puede verificar la validez de una llave y además este certificado puede ser revocado y puesto en una CRL, es decir hacerlo inválido debido, por ejemplo, a que la llave privada del usuario fue comprometida. Un certificado también tiene una fecha de vencimiento, después de la cual el mismo queda inservible. Así pues podemos decir que tanto el certificado digital como las llaves asimétricas son los elementos más importantes dentro de toda PKI.

3.1.2.2 Directorio Central

El certificado por sí mismo es un documento digital que generalmente se almacena y administra en un directorio central; si hablamos de un usuario que trabaja desde casa, almacenará sus certificados en un lugar específico dentro de su computadora, después los usará por demanda, sin embargo cuando se implementa una PKI se debe establecer como parte de ella un directorio en donde se puedan almacenar y recuperar los certificados, por ejemplo si existiese ya un directorio que cumpla con las especificaciones X.500 o con el modelo LDAP bien podría servir para las actividades de la PKI. El problema que existe aun, hablando de directorios, es que en ocasiones el conjunto de aplicaciones cliente que interactúan dentro de la PKI no podrán hacerlo con los directorios por cuestiones de compatibilidad y esto le resta funcionalidad a la PKI, sin embargo está en marcha el llamado foro de interoperabilidad de directorios (DIF) que pretende encontrar una solución a este problema.

3.1.2.3 Modelo de confianza y rutas de certificación

En términos generales, la confianza es un sentimiento de seguridad o una actitud de fe hacia una persona en la cual creemos por conocerla bien o por que es muy cercana a nosotros; no obstante tratándose de sistemas de cómputo no podemos ir por la vida confiando a ciegas en una aplicación o en un equipo sólo por que nos "cae bien" o porque llevemos años junto a ellos, no funciona así. Una definición más precisa y profunda de confianza que se puede aplicar dentro del contexto tecnológico es la siguiente: confiar en un dispositivo o procedimiento automatizado es la seguridad de que, cuando se pretenda solicitar la ejecución de una tarea, ésta se efectuará acorde a una serie de reglas predefinidas y con estricto apego al funcionamiento del sistema, garantizando de esta forma la obtención de los resultados esperados. Así pues el modelo de confianza de una PKI, en

un contexto tecnológico es transitivo desde el momento en que puede confiar y delegar confianza con varias AC's, mismas que podrán entonces interactuar entre ellas para validar certificados. Por ejemplo si manejamos un esquema de confianza directa quiere decir que en toda la PKI sólo existe una AC en la cual los usuarios confían y utilizan para verificar la validez de los certificados estableciéndose una ruta de certificación AC raíz – usuario que es la ruta más corta que existe y que se puede implementar siempre y cuando la comunidad de usuarios no sea muy grande *ver fig. 3.2.*

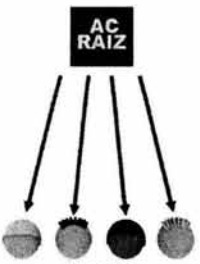


Fig. 3.2 Autoridad Certificadora raiz

Cuando el lugar de implementación es muy grande, por ejemplo una red MAN en donde obviamente habrá un número muy alto de usuarios, se vuelve impráctico tener una sola AC que administre y brinde servicio a tantas entidades, es entonces cuando se transfiere la confianza a otras AC's que coexisten e interactúan en una misma PKI organizadas jerárquicamente (*ver fig. 3.3*); de tal forma que una AC de mayor nivel pueda certificar a otras en niveles inferiores, y a su vez ésta estar certificada por una de mayor rango, quizás la de más alto nivel, la AC raíz. De esta manera se establecen rutas de certificación y de transferencia de confianza, por ejemplo, supongamos que Luis confía y está certificado por una AC-X y que Ximena confía y está certificada por una AC-G, entonces si ellos quisieran establecer comunicación segura tendrían que seguir una ruta de certificación que sea de confianza para ambos, dicho de otra forma, para que Luis verifique la validez del certificado de Ximena debe circular por todas las AC's que se encuentran entre ellos para lograrlo, en este caso debe seguir la ruta XBACG.

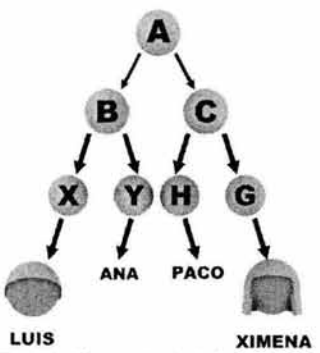


Fig. 3.3 AC organizada jerárquicamente

Un caso más sencillo es cuando se establece la certificación cruzada, lo cual quiere decir que se crean certificados de AC que se encuentran en un punto medio de la ruta de certificación entre los dos participantes, es decir, se puede incluir un certificado que sirva como enlace entre las rutas de ambos, en nuestro ejemplo se podría hacer esto con las AC's B y C acortando así la ruta de certificación y de confianza entre Luis y Ximena, así podrían seguir una ruta XBCG en vez de la ruta completa XBACG lo cual hace más sencilla la labor de certificación y verificación y establece además una red de confianza, misma que al crecer demasiado podría tornarse ambigua y poco funcional. *Fig. 3.4.*

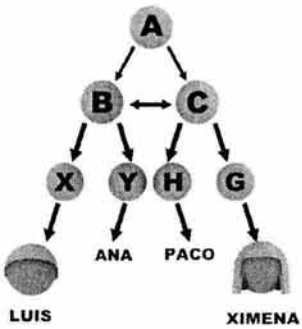


Fig. 3.4 AC organizada jerárquicamente y con certificación cruzada

3.1.2.4 Políticas de uso y administración

Otro elemento de toda PKI es el conjunto de políticas que comprende todos los lineamientos y reglas involucrados en su funcionamiento. Por ejemplo si un usuario extravía su llave privada o la comparte con un tercero, tendrá la obligación de notificar tal situación al equipo de seguridad de la AC para que ésta tome las medidas necesarias, tal vez la revocación del certificado.

Las políticas son en conjunto una parte crucial dentro de la PKI ya que de ellas depende el rumbo que se deba seguir ante un percance o determinada acción, por lo tanto son fundamentales en situaciones críticas en las que se deban tomar decisiones. Generalmente las políticas se conforman estudiando el entorno de implementación y en presencia de personal técnico, administrativo y de preferencia legal. Grosso modo las políticas proveen una explicación detallada de cómo la AC debe administrar los certificados que expide además de otros servicios como administración de llaves.

3.1.2.5 Administrador u oficial de seguridad

Tal y como si se tratara de cualquier otro sistema de información, una PKI necesita de personal que administre todas las funciones y servicios que ésta ofrece. Por la naturaleza tan específica de una PKI puede ser difícil encontrar personas con un perfil adecuado, así que un buen comienzo puede ser el nombrar un oficial de seguridad quien será el responsable de configurar, administrar e implementar la PKI así como llevar al cabo las políticas de funcionamiento de la misma infraestructura como se muestra en la *figura 3.5.* Esta persona puede fungir como líder de proyecto y será uno de los componentes

principales para que las funciones de la PKI vayan adelante. Adicionalmente puede ser que se necesite también un administrador de directorio y por supuesto el personal necesario para establecer la autoridad registradora.

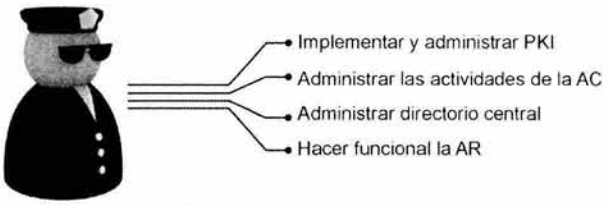


Fig. 3.5 Administrador u oficial de seguridad

3.1.2.6 Usuarios

Los usuarios son la motivación principal para realizar todos los esfuerzos tecnológicos que permitan llevar por buen camino a las actividades digitales de la actualidad, es por ello que las entidades finales son también parte fundamental de cualquier PKI (de hecho de cualquier sistema de cómputo) ya que sin ellas simplemente este trabajo carecería de sentido.

3.2 PKIX – PKI basada en certificados X.509

Una PKIX es una PKI basada en el estándar X.509 que pertenece a la familia de estándares ISO/IEC de ITU-T referentes a infraestructura de llaves asimétricas. El documento X.509 es el más comprensible y mayormente aceptado como modelo estándar aunque, por otra parte, también es poco usado debido en gran medida a la incompatibilidad con muchas aplicaciones. Una PKIX está formada básicamente por los siguientes elementos:

- Autoridades Certificadoras
- Autoridades Registradoras
- Usuarios
- Almacenamiento de certificados

En donde la interacción de todos ellos constituye la PKI. A continuación se explicarán los elementos involucrados en una PKIX.

3.2.1 Certificados Digitales

El certificado digital es el corazón de nuestra propuesta de PKI para la UNAM, nuestro propósito es construir un certificado adecuado para los servicios de autenticación dentro de la institución. Para ello es necesario comprender como funciona y para que nos va a servir.

3.2.1.1 ¿Qué es un certificado digital?

Cuando deseamos enviar un mensaje confidencial a otra persona, basta con crear nuestro par de llaves (una privada y una pública), cifrar el mensaje con su llave pública y

así estar seguros de que sólo el destinatario correcto podrá leer el mensaje en claro, por que contiene la correspondiente llave privada. Pero quién o qué nos garantiza que la llave pública que se nos envió es efectivamente de la persona correcta y no de un impostor. Para tal situación necesitamos a los certificados digitales.

Los certificados digitales son como una especie de documento de identidad electrónica, basados en la criptografía de llave pública y en el sistema de la firma digital. La misión principal de un certificado digital es garantizar con toda confianza el vínculo que existe entre una persona y una pareja de llaves.

Un certificado digital es un documento electrónico que contiene datos característicos de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada **Autoridad Certificadora** (AC) (*ver fig 3.6*). Además los certificados tienen la característica de que no se pueden falsificar ya que van firmados digitalmente por la AC, y no pueden modificarse porque la firma no correspondería con el resumen (**hash**) que se obtendría de los datos modificados. Por tanto al utilizarlo, el software que los gestiona daría un menaje de invalidez.

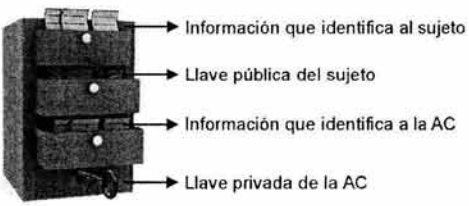


Fig. 3.6 Certificado básico

Pero surge otra incógnita, ¿quién avala a la autoridad certificadora? Para resolver este problema se crea la jerarquía de certificación, donde cada certificado emitido por una AC debe estar firmado por una AC de mayor grado, hasta llegar a la AC superior o raíz, que se avala a sí misma como Verisign.

Los certificados por ser un mecanismo confiable y seguro de identificación electrónica, podemos utilizarlos principalmente en la verificación de la firma digital, para asegurar una sesión de intercambio de llaves y en la autenticación de la identidad de un usuario. Así como para todas aquellas aplicaciones donde se necesite garantizar la identidad de las partes como comercio electrónico, transferencias bancarias, pagos de impuestos, etc.

Hay que recordar que estamos planteando cómo funciona un certificado, pero hay que tener en cuenta que un certificado digital es tan sólo un archivo almacenado en nuestra PC, y que para obtener uno y que nos sea de utilidad, es necesario contar con entidades que los expidan, tener software que los administre y que los comprenda, y que éstos estén codificados en un formato compatible como lo es el X.509.

3.2.1.2 Estructura del Certificado X.509

Un certificado es una estructura de datos firmada digitalmente por una AC. Entonces para definir esta estructura necesitamos utilizar el formato X.509.

El estándar X.509 no es más que la definición de la sintaxis para los certificados y para las listas de certificados revocados (CRL). Es decir cómo se define cada componente

del certificado o de la CRL. Esta sintaxis se encuentra originalmente en notación ASN.1, de la cual se hablará más adelante, lo que importa es mostrar de una manera general, todos los elementos de esta estructura.

El formato de los certificados digitales es estándar, siendo el X.509 v3 el recomendado por la Unión Internacional de Telecomunicaciones (ITU) y el que está en vigor en la actualidad. Este estándar se ha desarrollado en tres etapas, certificados versión 1, 2 y 3 para su completa definición y refinamiento.

Certificado versión 1

La primera versión del X.509 aparece en 1988, siendo la primera propuesta para una PKI en todo el mundo. A continuación se muestran los campos básicos del certificado X.509 versión 1.

- **Versión.** Indica a que versión pertenece el certificado, en este caso a la 1.
- **Número de serie.** Es el número de identificación único para el certificado, asignado por la AC.
- **Firma.** Identifica al algoritmo utilizado por la AC para firmar el certificado.
- **Emisor.** El nombre del emisor; identifica a la entidad que ha firmado el certificado y sigue la nomenclatura de nombres distinguidos (DN, *Distinguished Names*) de X.500.
- **Validez.** Indica el período de tiempo en el que el certificado es válido.
- **Usuario o sujeto.** Es un nombre distinguido X.500 que identifica de forma única al poseedor del certificado.
- **Información de llave pública del usuario.** Contiene la llave pública del usuario junto con el identificador del algoritmo que se utilizó para generarla.

Certificado versión 2

Este certificado fue como un borrador y nunca fue oficialmente publicado como un estándar. Aparece en 1993 y sirve como base para el certificado versión 3; la mayor contribución de esta versión fue la definición del CRL (Lista de Certificados Revocados) y la adición de dos campos más:

- **identificadores únicos de emisor.** Es campo es opcional y se usa para identificar a la AC emisora del certificado
- **identificadores únicos de usuario.** Este campo es opcional y se utiliza para identificar de manera única al propietario del certificado

Certificado Versión 3

La versión 3 fue establecida en el año 1994-95. Los campos básicos de la versión 3 son exactamente los mismos que los de la versión 2, sólo que se agregaron las denominadas *extensiones X.509* para permitir que se incluyera información adicional sobre el uso al que están destinadas las llaves y sobre las políticas de certificación asociadas a un certificado *ver fig. 3.7.*

Un certificado X.509 permite realizar varias acciones sobre él. Puede usarse para ver la información sobre los certificados, convertir los certificados a otros tipos, firmar peticiones de certificado, o editar las opciones de confianza o aceptabilidad de un certificado.

El formato X.509 es válido tanto en enfoques basados en la seguridad a nivel del interfaz de transporte (como SSL, usado frecuentemente en Internet), como a nivel de aplicación (SHTTP, S/MIME, etc.).

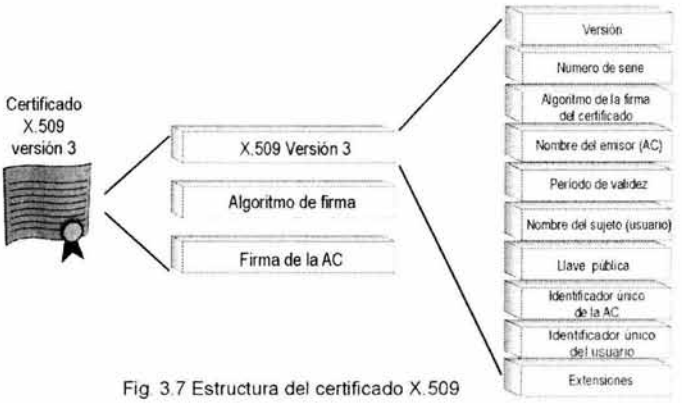


Fig. 3.7 Estructura del certificado X.509

3.2.1.3 ¿Cómo trabaja un certificado digital?

En este apartado vamos a conocer el esquema general de un certificado digital, para ello veremos como se obtiene un certificado y como se valida.

3.2.1.4 Emisión del certificado digital

Como ya se mencionó los certificados digitales son emitidos por las autoridades de certificación, entidades de confianza como Verisign, Thawte y como pretendemos que sea la AC de la UNAM, por lo cual para obtener uno, lo debemos solicitar a una autoridad certificadora, para nosotros la UNAM. La persona o entidad interesada debe cumplir con ciertos requisitos; como primer paso debe generar sus llaves pública y privada en la comodidad de su computadora, después debe llenar un documento llamado requerimiento de certificación (PKCS 10) que es como una solicitud que contiene datos representativos del solicitante (nombre personal o de la empresa, domicilio personal o social, correo electrónico, puesto, etc.) y su llave pública. El requerimiento es un documento autofirmado o firmado digitalmente por el usuario, esto significa que está conciente que acepta la llave pública y ser el legítimo dueño de la llave privada.

Como se verá mas adelante una autoridad registradora tiene la responsabilidad de verificar los datos que se presentan en el requerimiento y por tanto, obtener prueba de que el sujeto es propietario de la correspondiente llave privada.

El segundo paso, es que el usuario debe presentar ante una autoridad registradora su requerimiento de certificación y documentación suficiente que lo identifique. La autoridad registradora, extrae la llave pública, y autentica el requerimiento para demostrar que fue firmado con la correspondiente llave privada. Adicionalmente, verifica que la documentación es suficiente para acreditar que los datos que se presenten en el certificado

sean verdaderos. La AR firma el requerimiento con su llave privada y genera un precertificado que es enviado a la AC correspondiente.

La AC autentica que el precertificado provenga de una de las AR con las que ha establecido confianza, si es así entonces la AC genera el correspondiente certificado y lo firma digitalmente con su llave privada, después será mandado al usuario. Esta firma, permite que aquellas entidades que deseen realizar comunicaciones con la persona poseedora del certificado, puedan comprobar que la información que éste contiene es auténtica (suponiendo que confíen en la AC emisora).

Una vez que los certificados han sido firmados, se pueden almacenar en servidores de directorios o transmitidos por cualquier medio (seguro o no) para que estén disponibles públicamente. Para que se pueda obtener con facilidad el certificado digital de cualquier persona o entidad las autoridades de certificación disponen de servidores de acceso público que realizan la función de depósito de certificados, en los que se puede buscar el deseado y descargarlo a nuestro ordenador. Es ésta una forma más segura que la de usar directamente un certificado recibido por correo o descargado de una página web, ya que la autoridad de certificación responsable del servidor es la encargada de verificar constantemente la validez y autenticidad de los certificados que distribuye.

3.2.1.5 Validez de un Certificado Digital

La validación de un certificado se realiza verificando la firma digital en él incluida mediante el empleo de la llave pública de su signatario; esta llave pública es a su vez validada usando el certificado correspondiente y así sucesivamente hasta llegar a la raíz de la jerarquía de certificación. Para entender un poco más este proceso veamos el siguiente ejemplo:

Ximena, la chica del ejemplo del capítulo anterior, desea enviarle un mensaje confidencial a Luis; lo primero que hace Ximena es obtener el certificado de Luis, y el certificado raíz del emisor, es decir de la AC que emitió el certificado.

Del certificado del emisor extrae su llave pública y utiliza el procedimiento de verificación de firma digital que se describió anteriormente para verificar la firma de la AC, y comprueba si el certificado de Luis es válido. Si este resulta válido, Ximena extraerá la llave pública de Luis, así con toda confianza cifrará su mensaje y estará completamente segura de que sólo é lo podrá leer (*ver fig. 3.8*). Hay que recordar que Ximena puede confiar en la llave pública de Luis sólo si confía en la AC que lo emitió.

Además la validación también implica comprobar la fecha de expiración de cada certificado en el campo de período de validez de este y que ninguno de los certificados haya sido revocado (no válido), es decir mediante la comprobación de la CRL (Lista de Certificados Revocados) la cual se explica mas adelante; si nuestro certificado se encuentra en la CRL, significa que no es válido. Las AC anulan los certificados y los publican en la CRL por motivos tales como si el usuario hubiese comprometido su llave privada; porque la persona o entidad cometió un delito; o simplemente cuando el certificado caduca.

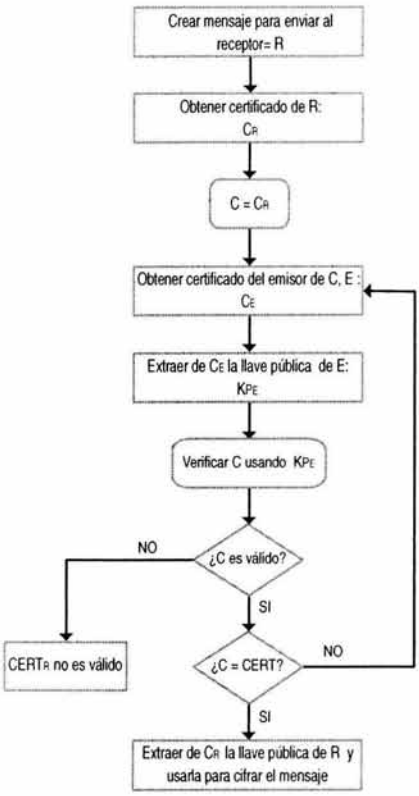


Fig. 3.8 Diagrama de flujo para validar un certificado

3.2.1.6 Tipos de Certificados Digitales

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.

Los certificados de identificación simplemente identifican y conectan un nombre a una llave pública. Los certificados de autorización proveen otro tipo de información correspondiente al usuario, como dirección comercial, antecedentes, catálogos de productos, etc. Otros certificados colocan a la autoridad certificadora en el rol de notario, pudiendo ser utilizados para dar validez de un determinado hecho o que un evento efectivamente ha ocurrido. Otros certificados permiten determinar día y hora en que el documento fue digitalmente firmado (digital time-stamp certificates).

Del tipo de certificado a emitir van a depender las medidas de comprobación de los datos y el precio del mismo. Desde el punto de vista de la finalidad, los certificados electrónicos se dividen en:

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

2. *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 o del servicio que éste va a ofrecer, vinculando también el dominio por el que se debe acceder al servidor. La presencia de este certificado es condición imprescindible para establecer comunicaciones seguras SSL.
3. *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.
4. *Certificados de firma de objetos*: usados para identificar al autor de ficheros 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, ó si por el contrario no confía en él, con lo que el código será rechazado.
5. *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.
6. *Certificados para IPSec-VPN* (Virtual Private Network): los Certificados para VPN son los elementos necesarios para que la empresa aproveche las cualidades y ventajas de la utilización de las VPN de un modo plenamente seguro.

Las VPN surgen como consecuencia de la creciente demanda de seguridad en las comunicaciones ya sea entre router-router o cliente-servidor, y en las actividades dentro de las redes corporativas en donde la comunicación con empleados remotos (con gran importancia en el caso del teletrabajo), sucursales, business partners o clientes es muy importante.

Toda persona o entidad que deseé obtener un certificado debe pagar una cuota a las autoridades de certificación, cuota que irá en función de la clase del certificado y del uso que se le vaya a dar al mismo (ambas están relacionadas). A mayor nivel de comprobación de datos (clase mayor), más costará el certificado. Sin embargo también podemos obtener certificados personales, que se encuentran disponibles en la red, sin ningún costo, la desventaja es que tienen una validez muy pequeña, y que no son muy confiables.

Desafortunadamente existe una desventaja en los certificados digitales, y esta es la incompatibilidad tanto para aplicaciones como para navegadores de internet (browsers). Por ejemplo: si nos enviaron un mensaje firmado en PGP, y lo queremos ver con S/MIME, no se puede por que no es compatible lo cual supone un gran problema ya que se debe tener software de administración de certificados para cada formato de éstos.

Un panorama de lo anterior es el siguiente: certificados SSL para servidor utilizan Netscape, Opera, Internet Explorer; los certificados de firma de objetos los usan Microsoft, RSA Authenticode, Netscape RSA X.509 y SUN Java plugin DSA/RSA X.509; para correo seguro está S/MIMES, X.509 por Netscape y PGP por Eudora.

3.2.2 Autoridades Certificadoras

Las autoridades certificadoras están definidas como las entidades electrónicas facultadas para emitir certificados digitales, análogo a un notario público. Estas pueden ser personas físicas, empresas o instituciones públicas o privadas; por ejemplo una compañía puede emitir certificados a sus empleados, o una universidad a sus estudiantes, tal como pretendemos que lo haga la UNAM. Las AC asocian la identidad del propietario con una llave pública. Es decir, gracias a la AC se garantiza que cuando un usuario B recibe un certificado de A, este puede estar seguro de que ese certificado es de A.

La AC utiliza su propia llave privada para firmar los certificados digitales que emite, lo que es otra aplicación de la firma digital. Para evitar que se falsifiquen los certificados, la llave pública de la AC debe ser confiable: una AC debe publicar su llave pública o proporcionar un certificado de una autoridad mayor que garantice la validez de su llave.

Para que la autoridad certificadora funcione, se necesita que los usuarios confíen plenamente en ella. Por esta razón la AC debe contar con un entorno de seguridad muy fuerte, en particular en la protección de su llave privada; si esta llegara a comprometerse, toda la infraestructura de llave pública seria un fracaso.

La AC como prestadora de servicios de certificación proporciona servicios como la generación de certificados, publicación, renovación y revocación de certificados, publicación de listas de certificados revocados (CRL), generación de pares de llaves, comprobación de validez de certificado, servicio de fechado seguro, entre otros.

3.2.3 Autoridad Registradora

La autoridad registradora actúa como intermediario entre la AC y el usuario. Cualquier usuario que requiera el servicio de un certificado puede acudir directamente con la AR.

La autoridad registradora tiene la responsabilidad de verificar que la llave pública, la identidad y los atributos del posible propietario de un certificado sean totalmente verdaderos. Si obtiene prueba de esto la AR firma con su llave privada el requerimiento de certificación como prueba de que ha verificado correctamente los datos y se lo envía a la AC. La AC tiene una relación de confianza con la AR, y la AC puede contar con la AR para que haga la verificación del ID del requerimiento.

Algunas funciones que realiza la AR son: la recepción de las peticiones de certificado, proceso de la identificación de usuarios, generación de llaves, respaldo de llaves, reporte de revocaciones, etc.

Las autoridades registradoras juegan un papel muy importante en la autenticación con certificados digitales, ya que le quitan un poco de responsabilidad a la AC, siendo muy conveniente que otras personas se encarguen de verificar todos los datos de una persona, para agilizar la obtención de un certificado. En este proyecto sería muy prudente contar con una AR, tal vez un departamento dentro de cada plantel de la UNAM, así todos los posibles candidatos a un certificado, no tendrían que ir hasta las instalaciones donde se ubique la AC.

3.2.4 Revocación de Certificados

Los certificados tienen una duración específica, al igual que el pasaporte o el permiso de conducir. Y las AC también pueden reducir este tiempo por medio del proceso de revocación de certificados.

Estos certificados invalidados son publicados por la AC en una lista de certificados revocados (CRL) donde también se puede incluir la razón por la que se revocó el certificado. Las razones más comunes son:

1) Que esté comprometida o en riesgo la llave privada.
2) Riesgos para la entidad emisora de certificados.
3) Modificaciones en el certificado (por ejemplo si el usuario cambió de teléfono)
4) Si la persona que lo solicitó cometió un delito o fraude.
5) Si el certificado caduca.

3.2.4.1 Listas de Certificados Revocados (CRL)

Para llevar un control de los certificados revocados (no válidos) las autoridades de certificación han implementado unos servidores especiales que contienen bases de datos en las que figuran los certificados anulados, que se conocen con el nombre de Lista de Certificados Revocados o CRL.

Estas listas son publicadas cada cierto período de tiempo por las AC, firmadas digitalmente por ellas e indican cuales de los certificados que ellas han emitido han sido revocados, en la lista viene el número de serie y la fecha en que el certificado fue revocado. Las CRL's, por estar firmadas digitalmente por la AC emisora, podrán comprobar su validez con un proceso análogo al de los certificados.

3.2.5 Registro de tiempo (Time Stamping)

Como se mencionó anteriormente, cuando recibimos un certificado de otra persona, hay que ver si es válido, verificando que no se encuentre en la lista más actualizada de certificados revocados. Pongamos el siguiente ejemplo:

Ximena recibió un certificado de Luis, ella revisó que no estaba revocado en la última CRL, lo considero válido y envió su mensaje a Luis, pero resulta que al siguiente día aparece como revocado en la CRL nueva. Ximena de alguna manera tiene que demostrar que recibió el certificado de Luis antes de que apareciera la nueva lista.

Para solucionar este problema Ximena puede adjuntar una marca temporal en el documento, así Luis podrá utilizar la llave pública que ha caducado o que ha sido anulada para verificar la vieja firma y podrá confiar en la autenticidad del documento. El registro temporal no sólo puede garantizar la validez de los documentos firmados para un uso futuro sino que es crucial para documentos legales y financieros. Por ejemplo, poniendo una marca temporal en el pago electrónico de la factura de la tarjeta de crédito podemos protegernos ante eventuales cargos de intereses al disponer de la prueba de que se pagó la factura a tiempo.

En Noviembre de 1997, algunos investigadores propusieron un protocolo de marca temporal a la IETF (Internet Engineering Task Force) como un borrador de internet (draft-adams-time-stamp-00.txt). Diversas compañías, incluida Microsoft, han adoptado esta tecnología y están utilizándola en sus productos (por ejemplo, Authenticode).

Cuando se necesita poner una marca temporal en un documento, se puede solicitar a una autoridad de registro temporal (TSA, Time Stamp Authority) que nos proporcione la marca temporal, la TSA debe ser una organización de confianza. Las TSA's pueden ser un proveedor de servicios de registro temporal público, como VeriSign o un servidor de registro temporal interno en una intranet. Cuando el servidor de registro temporal recibe la petición, el servidor pone una marca temporal en el documento con su llave privada y nos lo devuelve. Si se envía el documento con la marca temporal a Luis, éste puede utilizar la llave pública de la TSA (incluida en el certificado de la TSA) para verificar la marca temporal. El proceso de registro temporal y de verificación del registro temporal es similar al proceso de firma digital y de verificación de firma.

3.3 La firma digital

La firma convencional o manuscrita tiene como propósitos principales, la identificación del firmante y la aprobación del contenido del documento firmado. Dado que esta firma tiene valor legal, le da al poseedor y firmante del escrito, la seguridad de que si no se realiza lo acordado en dicho documento, tiene el derecho de tomar un camino legal ante cualquier percance, pero también tiene la responsabilidad de brindar cuentas sobre algo no cumplido y que, sin embargo, está avalado por su firma. La firma digital está planeada para que cumpla con los mismos propósitos en un entorno informático; la diferencia es que en el país aun no se cuenta, del todo, con alguna legislación que dote de seriedad y legalidad a los medios electrónicos de comunicación.

La criptografía de llave pública permite que cada mensaje enviado por cualquier usuario lleve su firma digital, análoga, a la firma tradicional. El hecho de poder firmar digitalmente un mensaje hace posible que el destinatario se asegure de que el mensaje que recibe, es enviado de verdad por el supuesto remitente.

3.3.1 ¿Qué es la firma digital?

La firma digital es un mecanismo utilizado para comprobar la integridad de un mensaje y la identidad del emisor que la genera. Por definición, la firma digital es la transformación de un mensaje empleando un criptosistema asimétrico, tal que, una persona que posee el mensaje inicial y la llave pública del firmante pueda comprobar que la transformación se creó usando la llave privada que corresponde a ésta y además que el mensaje no ha sufrido ninguna modificación.

Cuando nosotros realicemos una transacción electrónica, vamos a hacer uso de un sistema de llaves, el cual nos permitirá saber con certeza que la persona con la que estamos tratando es quien dice ser y no un impostor (autenticación), además de que nuestra información viaje cifrada por la red, esto para evitar que si alguien llega a interceptarla no la pueda entender, y si lo hiciera, poder detectar si nuestra información fue modificada o no (integridad de los datos); y por último el emisor no podrá negar que envió el mensaje o que realizó dicha transacción (no repudio).

Las firmas digitales como habíamos mencionado se crean usando un sistema de criptografía asimétrica o de llave pública. Estos criptosistemas de llave pública están basados en el uso de un par de llaves relacionadas matemáticamente; una llave privada conocida sólo por el firmante, usada para crear la firma digital, y otra llave pública

accesible por cualquier persona o entidad para verificar una firma digital o bien para descifrar un mensaje. Esto significa que el proceso de autenticación consta básicamente de dos pasos, uno ejecutado por el firmante y otro por el receptor de la firma digital.

1) Creación y envío de la firma digital
2) Recepción y verificación de la firma digital

De la misma forma, un emisor puede cifrar un mensaje con la llave pública del receptor, de manera que éste último sea la única persona que lo pueda descifrar, ya que sólo él posee la llave privada que permite hacerlo.

Algunos beneficios y propiedades de una firma digital son: es única, fácil de generar, no puede ser fácilmente repudiada, no puede ser imitada por otra persona, no es costosa y puede ser automáticamente sellada con un registro de tiempo. Las utilizamos como auxiliar en el proceso de autenticar mensajes; de autenticar entidades que pretendan realizar transferencias de fondos bancarios; como herramienta de autenticación en compras por internet y para otras transacciones de negocios.

Las firmas digitales se pueden clasificar como:

- **Implícitas**: si están contenidas en el propio mensaje.
- **Explícitas**: si son añadidas como una marca inseparable del mensaje.
- **Privadas:** legible solo para quien comparte cierto secreto con el emisor.
- **Públicas (o verdaderas):** si permite identificar al remitente ante cualquier persona a partir de información públicamente disponible.
- **Revocables:** si el remitente puede, posteriormente negar que la firma en cuestión le pertenece.
- **Irrevocables:** si el receptor puede probar que el remitente escribió el mensaje.

3.3.2 ¿Cómo firmamos un documento?

Firmar un documento es un proceso, básicamente, de dos pasos: como se muestra en la *figura 3.9.*

El primero, es la generación de la firma digital como ya se ha mencionado antes; para ello el firmante tendrá que aplicar una función hash conocida sobre el mensaje a firmar, obteniendo un extracto de longitud fija llamado valor hash, absolutamente específico para ese mensaje, un mínimo cambio en el mensaje produciría un valor hash completamente diferente. Los algoritmos más utilizados para esta función hash son el MD5 y SHA-1, cuya longitud del resumen obtenido oscila entre los 128 y 160 bits. A continuación el valor hash es transformado o cifrado mediante la aplicación de la llave privada del firmante, obteniendo un extracto final (la firma digital) que se añadirá al final del documento.

El autor enviará el documento original en claro y el valor hash cifrado o firma digital, para que la persona que posee la llave pública del firmante pueda verificar la autoría y la integridad de los datos. Hay que recordar que, para realizar esta comunicación de manera confiable, previamente ha existido un acuerdo con una tercera parte de confianza llamada AC la cual dará fe de la relación existente entre una llave pública y su legítimo dueño.

Segundo, es necesario comprobar la validez de dicho documento para lo cual necesitamos la llave pública del autor. El receptor recibe el mensaje firmado, descifrará el valor hash con la respectiva llave pública del firmante, seguido calculará el resumen hash del mensaje original con la misma función que se usó el firmante; si el resultado coincide con el resultado antes descifrado, entonces se considera válido el documento, en caso contrario, indica que el mensaje sufrió alguna modificación posterior por lo que se deberá considerar inválido.

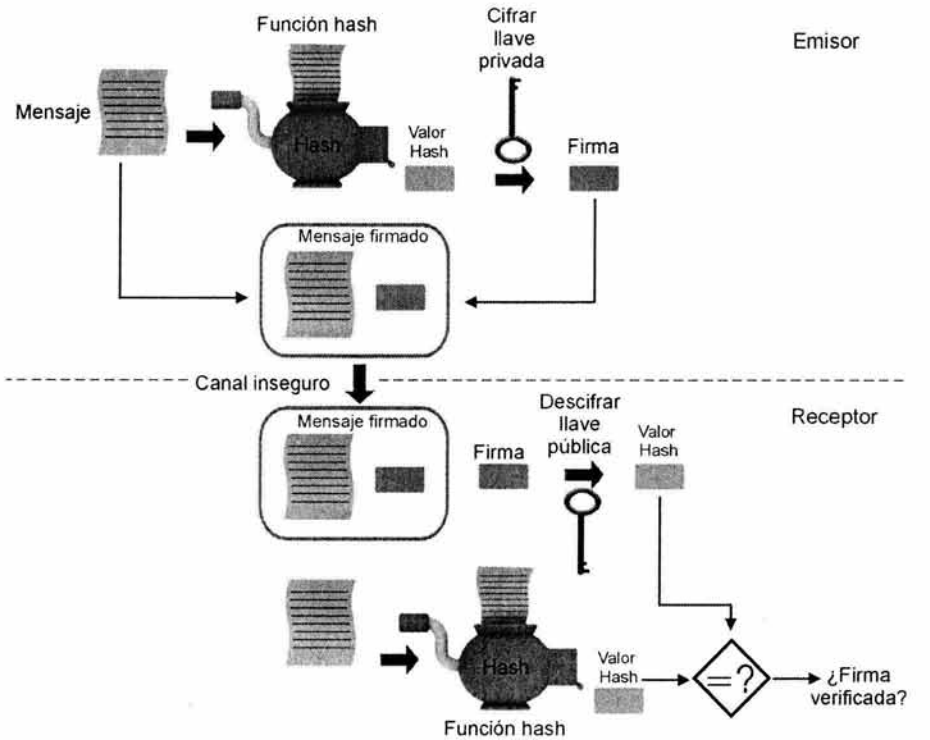


Fig. 3.9 Diagrama para firmar un documento

3.3.3 Modelo de confianza de la llave pública

Antes de verificar la firma de un mensaje con una llave pública que recibimos, es necesario que comprobemos que esa llave es verdaderamente de quien se supone que es.

Existen dos modelos de confianza para la llave pública que nos permiten hacer eso: confianza directa y confianza con terceros.

En el modelo de confianza directa, si conocemos al emisor de un mensaje, por ejemplo Luis, y confiamos en él, podemos intercambiar las llaves públicas personalmente y de forma segura; este modelo funciona en grupos pequeños y con personas que se conocen o que se pueden contactar fácilmente.

En el modelo de confianza con terceros, nosotros probablemente no conozcamos directamente a Luis, pero ambos confiamos en un tercero o intermediario para intercambiar las llaves. Vamos a suponer que el tercero confiable se llama Pedro. Cuando Pedro da nuestra llave pública a Luis, éste puede estar seguro de que la llave es realmente nuestra y viceversa. El modelo de confianza con terceros sirve y debe utilizarse en grandes compañías o instituciones en donde la gente podría no conocerse o bien no pueda contactar el uno con el otro directamente. Sin embargo, la mejor forma para confiar en que una llave pública es de determinada persona, es por medio de un certificado digital emitido por una autoridad certificadora; así no tendremos la mínima duda de que realmente se trata de Luis cuando nos envíe algún documento.

3.3.4 Algoritmos involucrados en la firma digital y en el cifrado de datos

3.3.4.1 Funciones Hash

Los criptosistemas de llave pública, generalmente cifran de manera más lenta que los criptosistemas de llave secreta. También los esquemas de firma digital suelen ser demasiado lentos, y en ocasiones la longitud de la firma suele ser similar o mayor que el propio mensaje que se firma, entonces si queremos firmar un mensaje muy grande, el problema se hace considerable, sin embargo la necesidad de firmar mensajes y el hecho no deseable de que la longitud de la firma sea extensa hace pensar en la búsqueda de una solución a este problema. Esta solución consiste en utilizar las llamadas funciones hash que nos sirvan de herramienta para firmar un mensaje.

Una función hash es una función matemática que realiza un resumen del mensaje a firmar. Comprime el documento en un único bloque de longitud fija, y lo somete varias veces a una sencilla función de desordenación, alterando sus elementos sin tener ningún sentido real y haciendo su contenido ilegible. Ejemplo *figura 3.10* que muestra la función hash MD5.

Las funciones hash son de una sola dirección, es decir son irreversibles, ya que a partir del resumen hash no se puede deducir u obtener el mensaje original. Además estas funciones son de dominio público, fáciles de usar e implementar.

Las funciones hash más conocidas son:

MD2, MD4, MD5 propuestas por Rivest.

- *MD2 (Message Digest 2)* - Utilizada para correo, aunque ya casi no se usa.
- *MD4 (Message Digest 4)* - Aunque se considera un sistema inseguro, es importante porque ha servido de base para la creación de otras funciones hash.
- *MD5 (Message Digest 5)* - Se creó como una versión mejorada de MD4; se utiliza para correo, como autenticador de mensajes en el protocolo SSL (Secure Socket Layer) y para firmar mensajes en el programa de cifrado PGP (Pretty Good Privacy). MD5 utiliza una técnica para crear resúmenes de 128 bits.

El único ataque que se conoce contra estas funciones es la investigación exhaustiva, a pesar de todo esto, MD5 en la actualidad sigue siendo el estándar.

Fig. 3.10 Función hash MD5

SHA-1 (Secure Hash Algorithm versión 1)

Fue desarrollado por el NIST (National Institute of Standars and Technology) y por el NSA, publicado como FIPS PUB 180 en 1993, para ser usado con el DSS para garantizar la seguridad del algoritmo de firma digital DSA; SHA-1 esta basado en el MD4 y MD5.

La versión actual se considera segura (por lo menos hasta el momento de escribir este documento) y es muy utilizada para el algoritmo de firma en el programa PGP con sus nuevas llaves DH/DSS (Diffie-Hellman/Digital Signature Standar).

RIPEMD-160

Desarrollada por un grupo de investigadores europeos entre los que se encuentra Hans Dobbertin (quien rompió MD4-MD5) y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión tenía las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja llaves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

Las funciones hash y la firma digital son elementos indispensables para el establecimiento de canales seguros de comunicación basados en los certificados digitales; estas funciones también se utilizan en todos los procesos de nuestra autenticación digital como veremos a continuación:

- Para firmar un mensaje
- Dentro del requerimiento de certificación
- Dentro del certificado
- Y la función hash que se usó en el CRL.

3.3.4.2 Esquema de firma DSA (Digital Signature Algorithm)

En agosto de 1991 el Instituto Nacional de Estándares y Tecnología de los estados Unidos (National Institute of Standard and Technology, NIST) propuso un estándar para la firma digital (Digital Signature Standard) DSS y su algoritmo correspondiente DSA, solicitando comentarios públicos para la adopción del estándar propuesto. El objetivo era proporcionar a las oficinas gubernamentales de los Estados Unidos una forma estándar de firmar las comunicaciones cuando ello fuera necesario.

El algoritmo DSA esta basado en el SHA (Secure Hashing Algorithm) que es descrito en el FIPS PUB 180 "Secure Hash Standard" y es también una variante de la firma digital de ElGamal, cuyo algoritmo disminuye el tamaño de la firma.

Al igual que RSA, el algoritmo DSA basa también su seguridad en un problema matemático, en este caso en el problema del logaritmo discreto (DLP – Discrete Logarithm Problem), que es el problema de encontrar el número de veces que hay que multiplicar un número conocido para obtener como resultado, otro también conocido.

El algoritmo DSA tiene los siguientes procedimientos: generación de la llave, generación de la firma y la verificación de la firma.

Generación de la llave DSA
Cada usuario elige los siguientes parámetros:

1. p: un número primo con $2^{511} < p < 2^{512}$
2. q: un divisor primo de p-1, tal que $2^{159} < q < 2^{160}$
3. g: un generador del único subgrupo cíclico de Z_p de orden q
4. Para su llave privada, selecciona un entero aleatorio x en el intervalo $0 < x < q$
5. Para su llave pública y, $y = g^x \pmod p$

De esta manera la llave pública es (p,q,g,y) y la llave privada es x.

Generación de la firma DSA
Para firmar un mensaje m, se elabora la firma como sigue:

1. Se selecciona un entero aleatorio k en el intervalo [1, q-1]
2. Se calcula el valor de $r = (g^k \bmod p) \bmod q$.
3. Calculamos $k^{-1} \bmod q$
4. Calculamos para $s = k^{-1} \{H(m) + xr\} \bmod q$, donde H es el algoritmo SHA-1.
5. Si s =0, regresamos al paso 1, (si $s = 0$, entonces $s^{-1} \bmod q$, no existe para s^{-1} se requiere el paso 2 de la verificación de la firma)

La firma digital para el mensaje m es la pareja de (r,s).

Verificación de la firma DSA.
Para verificar la firma (r,s) en m, **B** debe hacer lo siguiente.

1. Obtenemos una copia auténtica de la llave pública de **A** (p,q,g,y)
2. Se calcula $w = s^{-1} \pmod q$ y $H(m)$
3. Se calcula $u_1 = H(m)\, w \pmod q$ y $u_2 = r\, w \bmod q$
4. Se calcula $v = [(g^{u1}\, y^{u2}) \bmod p] \bmod q$
5. Es válida la firma, si y solo si $v = r$

3.3.4.3 Cifrado asimétrico RSA

Fue desarrollado en 1977 y nombrado así por sus inventores Ron Rivest, Adi Shamir, y Leonard Adleman, el sistema de llave pública RSA ha demostrado ser uno de los algoritmos más confiables, utilizado para cifrar y firmar datos.

La seguridad del RSA se basa en el problema de factorización de enteros (IFP – Integer Factorization Problem) y se refiere al hecho de que no existe una forma eficiente de factorizar un número que sea producto de dos números primos grandes, lo cual constituye un problema matemático difícil de resolver.

El algoritmo RSA es usado esencialmente en la generación de llaves asimétricas, cifrado de texto original y descifrado del texto cifrado.

Llaves

Las llaves que se utilizan son calculadas a partir de un par de números primos grandes p y q (actualmente se recomienda que tales números primos tengan más de 200 dígitos).

La llave pública consiste de un par de números n y e, donde $n = p \cdot q$; se elige un entero positivo e, $1 \leq e < \Phi(n)$, de modo que sea primo con el orden del grupo, es decir de modo que mcd (e, $\Phi(n)$) = 1.

Calculamos d como $d = e^{-1}$ (mod $(p\text{-}1) \cdot (q\text{-}1)$).

Definimos que la llave pública del usuario es la pareja (n,e) mientras que su llave privada es el número d.

Cifrado y descifrado.

Para cifrar un mensaje M usando el algoritmo RSA, es necesario previamente codificar el mensaje en un sistema numérico, bien decimal o bien binario, y dividir en bloques, cada bloque es un valor binario menor que algún valor n.

El proceso de cifrado para un bloque de texto en claro M, y el proceso de descifrado para un bloque de texto cifrado C son de la siguiente manera:

$$C = \text{Cifra}(m) = m^{e} \pmod{(n)}$$

$$M = \text{Descifra}(c) = c^{d} \pmod{(n)}$$

Si un usuario A desea enviarle un mensaje a B sólo tiene que calcular lo siguiente:

1) **A** calcula $C = m^{e} \pmod{(n)}$ con la llave pública (n,e)
2) **A** envía el mensaje cifrado C al destinatario **B**
3) **B** recibe el mensaje y descifra el mensaje calculando $M = c^{d} \pmod{(n)}$ con la llave privada d.

La longitud de las llaves, en el sistema RSA actualmente es hasta de 1024 bits. El cálculo de estas llaves se debe realizar en secreto en la máquina en la que se va a guardar la llave privada, y una vez generada es conviene protegerla mediante un algoritmo criptográfico simétrico.

RSA también cuenta con un algoritmo para la firma digital que consiste en lo siguiente:

1) Si el usuario **B** quiere firmar el mensaje m, procede a calcular $s = m^{d} \pmod{(n)}$, donde d es la llave privada de **B.**
2) Para verificar la firma de **B** al mensaje m, se procede como sigue s es la firma de **B** $\Leftrightarrow$ $s^{e} = m \pmod{(n)}$.

Pongamos un ejemplo para comprender los procesos de cifrado y descifrado RSA.

Si $p = 31$ y $q = 53$, entonces $n = 1643$, e es un número primo que no tiene factores en común con $(p\text{-}1) \cdot (q\text{-}1) = 30 \cdot 52 = 1560$. Nosotros podemos seleccionar aleatoriamente e en este caso será 71.

La llave pública entonces es el par (e,n) o sea {71,1643}.

Ahora vamos a calcular la llave privada d. Recordemos $d = e^{-1} (\text{mod } (p\text{-}1) \cdot (q\text{-}1))$.

$$d = 71^{-1} (\text{mod } (1560)) = 791$$

Recuerde que la función inversa es el módulo inverso, lo que significa que:

$$71 \times 791 = 1 \ (\text{mod } 1560)$$

Para cifrar

$$m = 34511228919$$

Al mensaje lo vamos a descomponer en bloques menores que n, bloques de tres, esto porque n = 1643, tiene 4 dígitos. Los correspondientes bloques m_i serán cifrados de la siguiente manera.

$m_1 = 345$ $\quad c_1 = 345^{71} (\text{mod } 1643) = 190$
$m_2 = 112$ $\quad c_2 = 112^{71} (\text{mod } 1643) = 196$
$m_3 = 289$ $\quad c_3 = 289^{71} (\text{mod } 1643) = 81$
$m_4 = 19$ $\quad c_4 = 19^{71} (\text{mod } 1643) = 475$

El texto cifrado es C = {190 196 81 475}.

El proceso para descifrar el texto cifrado es relativamente sencillo, para ello necesitamos calcular:

$m_1 = 190^{791} (\text{mod } 1643) = 345$
$m_2 = 196^{791} (\text{mod } 1643) = 112$
$m_3 = 81^{791} (\text{mod } 1643) = 289$
$m_4 = 475^{791} (\text{mod } 1643) = 19$

Regresamos al mensaje original.

RSA es de los sistemas de llave pública más conocidos y actualmente usados, y también el más rápido de ellos. Presenta todas las ventajas de los sistemas asimétricos, incluyendo la firma digital. Además tiene la característica de que cada una de las llaves puede cifrar y descifrar. Con estos algoritmos es imposible averiguar una de las llaves a partir de la otra.

3.4 PKIECC - PKI con Criptografía de Curvas Elípticas

Para el establecimiento e implementación de una PKI se encuentran involucrados una gran cantidad de estándares que nos dictan las reglas para la elaboración correcta de cada una de las partes que conforman una infraestructura. Como ya vimos la PKIX está basada principalmente en el estándar X.509 el cual especifica la estructura de los certificados que la AC emitirá y administrará; típicamente este estándar está ligado de manera estrecha con los documentos PKCS sobre todo porque se considera como algoritmo predeterminado, o mejor dicho, ampliamente recomendado al de criptografía asimétrica RSA.

RSA es el algoritmo de llave pública más popular que los desarrolladores de sistemas de seguridad utilizan, es por ello que al decir PKIX de inmediato se entiende que se usará criptografía RSA, sin embargo esto no es una regla, de hecho en el presente proyecto se utilizará un tipo de criptografía diferente y aun así seguirá siendo una PKIX; nos referimos a que en lugar de utilizar RSA emplearemos ECC, un tipo de criptografía asimétrica que, aunque no es nueva, sigue en evolución y en vías de ser aceptado como estándar criptográfico, de hecho el documento PKCS-13 publicado por RSA Security habla sobre aspectos de codificación de llaves que utilizan esta tecnología. Por lo anterior no debe sonar extraño que nosotros pretendamos expedir certificados X.509 basados en criptografía de curvas elípticas.

3.4.1 Criptografía utilizada para el proyecto acUNAM

Al proyecto acUNAM por su diseño y naturaleza se le puede considerar como un criptosistema ya que utiliza fundamentos criptográficos para atacar un problema bien definido que es el de la autenticación. La criptografía, como la definen algunos autores, es una disciplina que reúne a varias ramas de la ciencia con el propósito de desarrollar algoritmos y sistemas que permitan el cifrado de información con un alto grado de complejidad y eficiencia, lo cual necesariamente trae de la mano mayor seguridad. Para nuestro caso, diremos que la criptografía es una rama de las matemáticas cuyas técnicas y algoritmos aplicados a flujos o bloques de información digital, nos harán posible solucionar ciertos problemas que supone el inseguro mundo informático. Finalmente la mayoría de los esquemas criptográficos de autenticación, de confidencialidad, etc., por muy elaborados y complejos que parezcan, tienen como base el simple y llano cifrado y descifrado de información, lo cual constituye la razón de ser de la criptografía.

acUNAM utilizará criptografía de llave pública basada en curvas elípticas, técnica escogida de entre las tres más populares e importantes que son RSA, DH y ECC, debido a sus características de fortaleza, velocidad, desempeño y tamaño de llaves. A continuación se establecerán las bases para conformar una comparativa entre los algoritmos mencionados anteriormente y poder de esta forma justificar nuestra elección.

3.4.2 Bases matemáticas de la criptografía de llave pública

Desde que los esquemas de criptografía de llave pública se inventaron en el año 1976 por Whitfield Diffie y Martin Hellman, numerosos sistemas han sido propuestos, de los cuales muchos de ellos han caído en desuso por probar que son inseguros, poco prácticos o muy débiles ante ataques, y sólo unos pocos subsisten y continúan evolucionando. De una u otra forma todos estos sistemas asimétricos basan su seguridad en problemas matemáticos, en extremo difíciles y complejos.

Como ya mencionamos, muchos sistemas asimétricos han perecido a lo largo de los años; hoy día tan sólo tres tipos de sistemas de llave pública son considerados como seguros y eficientes, los cuales listaremos a continuación a partir del problema matemático en el cual esté basada su seguridad:

- *Problema de Factorización de Enteros (IFP):* RSA y Rabin Williams
- *Problema de logaritmos discretos (DLP):* DSA, esquema de acuerdo de llaves Diffie – Hellman, el Gamal, el esquema de firma Schnorr y el de Nyberg – Rueppel.
- *Problema de Logaritmos Discretos y Curvas Elípticas (ECDLP):* ECDSA, y los esquemas equivalentes a el ElGamal y Diffie – Hellman pero con curvas elípticas.

IFP (Integer Factorization Problem)

El problema de factorización de enteros se define de la siguiente forma: dado un número compuesto n que es el producto de dos números primos muy grandes p y q, encontrar p y q.

Mientras que encontrar o calcular números primos grandes es una tarea, relativamente fácil, encontrar dos de ellos que específicamente su producto nos de un número conocido, es un problema de difícil tratamiento. Basados en la dificultad de este problema, Rivest, Shamir y Adleman desarrollaron el criptosistema de llave pública RSA; por su parte Rabin y Williams desarrollaron su propio esquema de llave pública llamado tal como sus apellidos.

Este problema si bien ha tenido a través de los siglos la atención de matemáticos excepcionales como Gauss y Fermat, es sólo en los últimos 20 o 25 años donde se han podido desarrollar algoritmos más eficaces para el proceso de su solución, debido básicamente a que científicos actuales se sintieron motivados por el desarrollo de RSA y también a que con el paso del tiempo se tienen a disposición máquinas más veloces y eficientes.

DLP (Discrete Logarithm Problem)

El problema del logaritmo discreto contempla lo siguiente: si p es un número primo, entonces Zp denota una serie de enteros de la forma $\{0,1,2,...,p-1\}$ en donde las operaciones de suma y multiplicación se realizan módulo p. Existe además un elemento diferente de cero en Zp llamado alfa ($\alpha \in Zp$), cuya particularidad es que puede expresar a cualquier valor de los demás elementos en Zp, como una potencia de sí mismo. Este elemento α es llamado generador de Zp.

Así pues el problema DLP se define así: dado un número primo p, un generador α de Zp y un elemento no cero $\beta \in Zp$, encontrar el único entero L tal que $0 \leq L \leq p-2$, para el cual $\beta=\alpha^{l} \pmod{p}$. El entero L es llamado el logaritmo discreto de β de base α.

Basados en la dificultad de este problema, Diffie y Hellman propusieron, su bien conocido, esquema de acuerdo de llaves llamado como ellos en 1976. Desde entonces muchos sistemas criptográficos se han propuesto tomando como base este problema matemático, tal es el caso del algoritmo de firma digital desarrollado por el gobierno de los Estados Unidos (DSA); el esquema de cifrado y de firma digital del ElGamal; el esquema Schnorr y el desarrollado por Nyberg y Ruepel.

ECDLP (Elliptic Curve Discrete Logarithm Problem)

Si q es una potencia prima, entonces Fq denota un campo finito que contiene q elementos. Típicamente en aplicaciones, q es una potencia de 2 (2^m) o un número primo p difícil de calcular. Entonces el problema de logaritmos discretos y curvas elípticas es el siguiente: dada una curva elíptica E definida sobre Fq, un punto $P \in E(Fq)$ de orden n, y un punto $Q \in E$ (Fq), determinar el entero L, tal que $0 \leq L \leq n-1$, para el cual Q-LP exista.

Basados en la dificultad de este problema, Neal Koblitz y Victor Miller de forma independiente propusieron para 1985, usando el grupo de puntos de una curva elíptica definida sobre un campo finito, varios modelos de criptosistemas. Uno de ellos cuya implementación está siendo estandarizada por organizaciones facultadas para ello, es el criptosistema análogo a DSA llamado ECDSA (Elliptic Curve Digital Signature Algorithm). Se puede mencionar que quien implemente criptografía de curvas elípticas a sus sistemas, será pionero en el uso de esta tecnología ya que RSA es actualmente el estándar más utilizado.

3.4.3 Comparación de eficiencia de algoritmos

Basados en las conclusiones a las que han llegado diversas organizaciones como Certicom Corporation, podemos decir que los algoritmos basados en el ECDLP son más robustos de lo que podrían llegar a ser el IFP y el DLP juntos, tan solo basta observar algunos estudios que arrojan información contundente acerca del desempeño de sistemas que emplean algoritmos basados en estos tres problemas matemáticos. Uno de estos estudios, el que se presenta a continuación de forma gráfica muestra un plano en donde se expresa el tiempo para romper el sistema contra el tamaño de las llaves que se deben utilizar (*ver fig. 3.11*). La comparación se llevó al cabo entre ECC y RSA/DSA, estos dos últimos se toman de forma conjunta pues sus tiempos de ruptura son muy similares, además haciendo un paréntesis, en la actualidad se le considera seguro a aquel sistema cuya criptografía no pueda ser rota durante un período de por lo menos 10^{12} años MIPS (Million Instructions Per Second). La gráfica muestra claramente que, para que RSA y DSA puedan lograr un nivel de seguridad aceptable (10^{12} años MIPS) deben usar llaves de por lo menos 1024 bits, llaves muy grandes que traen de la mano sistemas lentos; por otro lado, se observa que ECC logra la misma meta, y aun más, con llaves de 160 bits. La diferencia más notable y dramática es que el uso de una llave ECC de 300 bits (tamaño que sigue siendo muy pequeño) puede brindar mayor seguridad que un sistema RSA/DSA de 2048 bits. Esto significa que ECC es un sistema mucho más seguro, eficiente y de mejor desempeño debido al reducido tamaño de sus llaves y en sí a la dificultad de resolver el problema ECDLP.

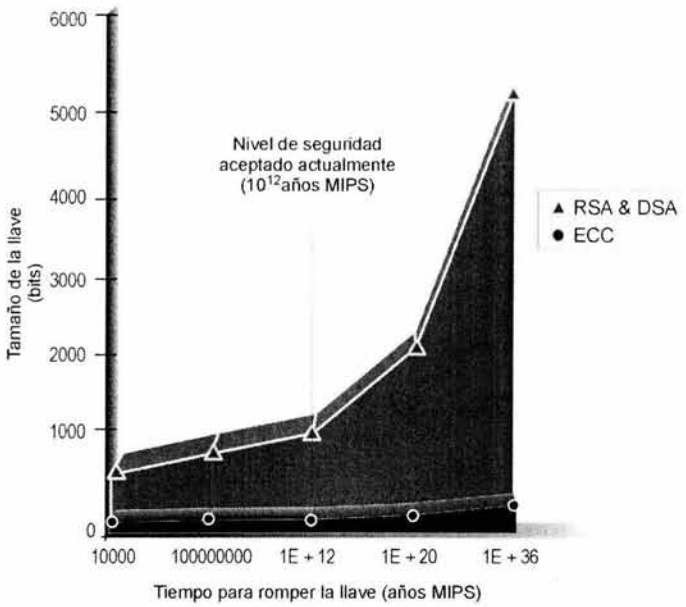


Fig. 3.11 Gráfica de comparación de niveles de seguridad entre ECC y RSA &DSA

acUNAM, proyecto que desarrollaremos, por las razones mencionadas anteriormente, utilizará un algoritmo ECC, contenido en las funciones de la librería lib_233.dll, el cual utiliza llaves de 480 bits garantizando, en este momento, la seguridad durante más de 10^{36} años MIPS según la gráfica revisada. No es casualidad que hayamos escogido este tipo de criptografía, ya que por sus bondades muchas empresas han comenzado a utilizarla como estándar y la han considerado como "un tipo de criptografía madura".

3.4.4 ECC - Criptografía de Curvas Elípticas

En 1985, Neil Koblitz y Victor Miller (independientemente) propusieron el Elliptic Curve Cryptosystem (ECC), o Criptosistema de Curvas Elípticas, cuya seguridad está basada en el mismo problema que los métodos de Diffie-Hellman y DSA, el DLP (Problema de Logaritmo Discreto), pero en vez de usar números enteros como los símbolos del alfabeto del mensaje a cifrar (o firmar), usa puntos en un objeto matemático llamado curva elíptica. ECC puede ser usada tanto para cifrar como para firmar digitalmente.

Hasta el momento no se conoce ataque alguno cuyo tiempo de ejecución esperado sea sub-exponencial para poder romper una llave ECC, esto hace que para obtener el mismo nivel de seguridad que brindan los otros sistemas, el espacio de llaves de ECC sea mucho más pequeño, lo que lo hace una tecnología adecuada para utilizar en ambientes restringidos en recursos (memoria, costo, velocidad, ancho de banda, etc.)

A continuación explicaremos los puntos más importantes del ECC.

Una curva elíptica es una serie de puntos obtenidos a partir de una ecuación de la forma:

$$y^2 = x^3 + ax + b$$

En un caso más general una curva elíptica sobre un campo K, será el conjunto de puntos (x,y) que satisfacen la ecuación:

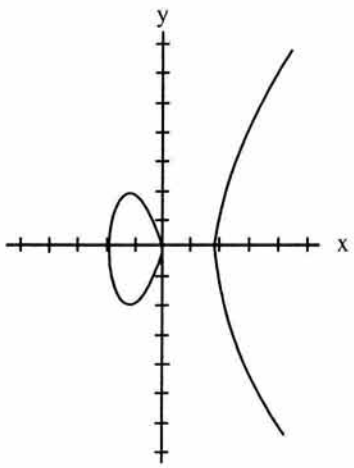
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Donde x e y son las indeterminadas, y los $a_1;\ldots;a_6$ son elementos constantes de un campo.

Para entender de qué estamos hablando, consideremos el campo de los números reales R y veamos un ejemplo con la siguiente curva *ver fig. 3.12* .

$$y^2 = x^3 - 4x + 0.67$$

donde $a_1 = a_2 = a_3 = 0$; $a_4 = -4$; $a_6 = 0.67$



$y^2 = x^3 - 4x + 0.67$

Fig. 3.12 Curva elíptica

La gráfica consiste de dos partes separadas, aunque podrían estar pegadas por un solo punto, o bien consistir solamente de una parte, en forma de "campana".

Los puntos en este tipo de curvas pueden juntarse y formar una estructura llamada grupo. Yendo al caso de los números reales, se podría probar que si $x^3 + ax + b$ no tiene factores repetidos o equivalentes a $4a^3 + 27b^2 \neq 0$, entonces las curvas elípticas de la forma:

$$y^2 = x^3 + ax + b$$

forman un grupo. Lo que se conoce como un grupo abeliano.

Cuando hablamos de grupos lo importante es definir la operación de suma, ver su definición y probar las propiedades de grupo.

Si queremos sumar dos puntos P y Q de la curva, se requiere trazar una línea que pase por ambos puntos. En el caso general esta línea siempre tiene un punto de intersección con la curva. Obtenemos un tercer punto en la intersección de la línea con la curva, y después trazamos una línea vertical. *Fig 3.13*

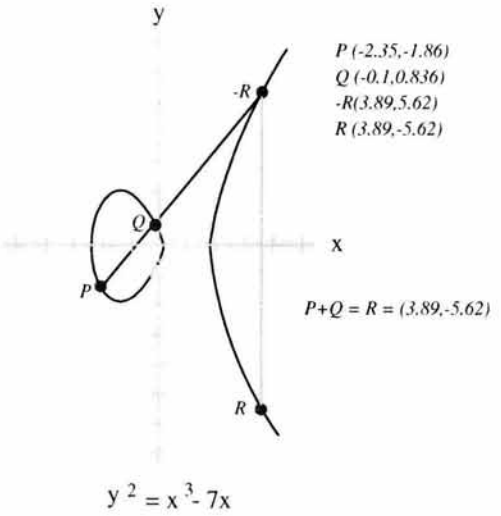


Fig. 3.13 Suma de dos puntos P y Q

El punto R de intersección con la curva de esta línea vertical se define como la suma de P y Q; $R = P + Q$.

También definimos el elemento opuesto $-R$ a un punto R de la curva: simplemente, el punto opuesto a $R = (x, y)$ es el punto $-R = (x, -y)$.

Una parte compleja es definir el elemento identidad del grupo. Si un punto extra es agregado a la curva. Este punto extra se llama *point at infinity* o punto en el infinito, y se lo designa con O. El punto O está en un lugar infinitamente lejos sobre el eje vertical, y es la identidad del grupo de la curva elíptica.

Aunque se dio una definición general de las curvas elípticas, no se considera práctico utilizarlas en forma general, ni tampoco utilizarla en el campo de los R. Por eso, es necesario que se consideren las curvas elípticas definidas sobre Zp donde p es un número primo y sobre los campos finitos de la forma F_{2m} , $m \geq 1$, ya que éstos conjuntos producen las implementaciones más eficientes de la aritmética de curvas elípticas.

3.4.5 ECDSA (Elliptic Curve Digital Signature Algorithm)

Dada la aceptación de los sistemas basados en curvas elípticas, recientemente el gobierno de los Estados Unidos, bajo recomendaciones del NIST estandarizó la segunda versión de su estándar de DSS (Digital Signature Standard), que describe tanto el algoritmo de firma digital DSA como también el ECDSA.

ECDSA es el análogo sobre curvas elípticas al DSA. Esto es, en vez de trabajar sobre el subgrupo de orden q en Z^*p, se trabaja en un grupo de curva elíptica $E(Zp)$. Como dijimos anteriormente, el ECDSA fue estandarizado por el NIST, pero también están en este proceso los comités de estándares ANSI X9F1 e IEEE P1363. La tabla muestra la correspondencia entre algunas notaciones matemáticas usadas en DSA y ECDSA.

DSA	ECDSA
q	n
g	P
x	d
y	Q

Tabla 1. Notaciones matemáticas para DSA y ECDSA

ECDSA es un esquema de firma basado en ECC. Está diseñado para ser existencialmente infalsificable, aún ante la presencia de un adversario capaz de lanzar ataques de mensajes elegidos por él (chosen-message attack). Un esquema de firma es existencialmente infalsificable si no es viable para un adversario falsificar una firma sobre algún mensaje que no ha sido previamente firmado por el usuario legítimo del esquema.

A continuación se dan los procedimientos para generar los pares de llaves, generar una firma y la verificación de una firma usando ECDSA:

Generación de llaves ECDSA. Cada entidad A hace lo siguiente:

1. Seleccionar una curva elíptica E definida sobre Z^*p. El número de puntos en $E(Zp)$ deberá ser divisible por un número primo grande n.
2. Seleccionar un punto $P \in E(Zp)$ de orden n.
3. Seleccionar un entero d estadísticamente único e impredecible en el intervalo $[1, n\text{-}1]$.
4. Se calcula el punto $Q = dP$.
5. La llave pública de A es (E, P, n, Q); la llave privada de A es d.

Generación de una firma ECDSA. Para firmar un mensaje m, A hace lo siguiente:

1. Seleccionar un entero k estadísticamente único e impredecible en el intervalo $[1,n\text{-}1]$.
2. Calcular el punto $kP = (x_1, y_1)$ y $r = x_1 \bmod n$. (Aquí x_1 es tratado como a un número entero, por ejemplo por conversión de su representación binaria).

Si $r = 0$, entonces volver al paso 1. (Esta es una condición de seguridad: si $r = 0$, entonces la ecuación de firma $s = k^{-1} \{h(m) + dr\} \bmod n$ no involucra a la llave privada d!).

3. Calcular $k^{-1} \bmod n$.
4. Calcular $s = k^{-1} \{h(m) + dr\} \bmod n$, donde h es el Secure Hash Algorithm (SHA-1).
5. Si $s = 0$, entonces volver al paso 1. (Si $s = 0$ entonces $s^{-1} \bmod n$ no existe; s^{-1} es requerido en el paso 2 de la verificación de firma).
6. La firma para el mensaje m es el par de enteros (r, s).

Verificación de firma ECDSA. Para verificar la firma (r,s) de A sobre el mensaje m, B deberá hacer:

1. Obtener una copia autentica de la llave pública (E, P, n, Q) de A.
2. Verificar que r y s sean enteros en el intervalo [1, n-1].

Si alguno no esta en el intervalo $[1, n-1]$ retornar "inválida" y parar.

3. Calcular $w = s^{-1} \bmod n$ y $h(m)$.
4. Calcular $u_1 = h(m)w \bmod n$ y $u_2 = rw \bmod n$.
5. Calcular $R = u_1P + u_2Q = (x_0, y_0)$ y $v = x_0 \bmod n$.

Si $R = O$, retornar "inválida" y parar.

6. Aceptar la firma si y solo si $v = r$. Si $v \neq r$ retornar "inválida".

El estándar ANSI X9.62 obliga a que $n > 2^{160}$. Para obtener un nivel de seguridad similar al de DSA (con q de 160 bits y p de 1024 bits), el parámetro n deberá tener alrededor de 160 bits. En este caso, las firmas producidas por DSA y ECDSA tienen la misma longitud de 320 bits.

En vez de que cada entidad genere su propia curva elíptica, las entidades podrían elegir usar la misma curva E sobre Zp, y punto P de orden n; estas cantidades luego son llamadas parámetros del sistema (en DSA, los parámetros de sistema análogos serian p, q y g.) En este caso, la llave pública de una entidad consiste solamente del punto Q. Esto resulta en llaves públicas de menor tamaño.

ECDSA tiene muchas similitudes con DSA, de las cuales las más importantes son las siguientes:

- Ambos algoritmos están basados en el esquema de firmas de ElGamal y usan la misma ecuación de firma: $s = k^{-1} \{h(m) + dr\} \bmod n$.
- En ambos algoritmos, los valores que son difíciles de generar son los parámetros del sistema (p, q y g para DSA; p, E, P y n para ECDSA) los cuales son públicos. Esto ayuda a mostrar que no fueron producidos para reunir ningún criterio secreto (por Ej. proveer a los parámetros con alguna propiedad "trapdoor"). Generar una llave privada, dado un conjunto de parámetros de sistema es relativamente simple y generar la llave pública asociada es directo.

Esto contrasta con el algoritmo RSA, donde los valores que son difíciles de generar (los primos p y q) deben ser mantenidos secretos.

- En su versión actual, tanto DSA como ECDSA usan SHA-1 como única opción para la función de hash. Sin embargo, esta planeado el soporte para SHA-2 a medida que SHA-2 pase por los procesos de estandarización de ANSI y NIST.

También existen diferencias entre ellos:

- ECDSA define que los valores de la llave privada d y el valor de k generado por cada firma sean estadísticamente únicos e impredecibles en vez de sencillamente aleatorio como en DSA.. Ya que existe la posibilidad de que k se repita y entonces un adversario pueda

recuperar d, la llave privada. Aunque, el uso de un valor aleatorio esta permitido, se pretende una mejor definición de los requerimientos de seguridad.
- En ECDSA, un método llamado *compresión de punto* permite a un punto sobre la curva elíptica (por ej., una llave pública Q) ser representado en forma compacta por un elemento del campo y un bit adicional, en vez de dos elementos del campo. Luego, por Ej., si $p \approx 2^{160}$ (de manera que los elementos en Zp son cadenas de 160 bits), las llaves públicas pueden ser representadas como cadenas de 161 bits. Esto puede llevar a una reducción sustancial en el tamaño de un certificado de llave pública en el orden de un 25% cuando es comparado con otros algoritmos asimétricos.
- La prueba de primalidad en DSA es probabilística. Como la tecnología de curva elíptica tiene la capacidad para una prueba de primalidad determinística, tal prueba fue incluida como una opción en el estándar X9.62. Esto le permite a una aplicación con altos requerimientos de seguridad (por Ej., una Autoridad Certificadora) verificar que los valores primos presentados son en realidad primos, aunque a un costo computacional adicional.
- Se han demostrado debilidades teóricas en DSA basadas en el hecho que la función de hash realmente usada en DSA es SHA-1 mod q, no simplemente SHA-1, donde q es el número primo de 160 bits. Esta debilidad permite la falsificación de un mensaje si el adversario puede seleccionar los parámetros del sistema. Esta debilidad no existe en DSA si los parámetros del sistema son seleccionados como se específica en el estándar X9.30. El análogo de q en ECDSA es n, el orden del punto base P. Si $n > 2^{160}$, entonces no existen colisiones y tal ataque no es posible.
- DSA especifica que la integridad del dato firmado es dependiente de la prevención de la revelación no autorizada, modificación, sustitución, inserción y borrado de la llave privada x o el valor k (particular) de una firma. Sin embargo, mientras quizás sea una implicación, el uso no autorizado no es explícitamente prohibido. En ECDSA, el requerimiento para uso autorizado de una llave privada ha sido hecho explícito.

3.5 Computación cuántica

Como mencionamos anteriormente, la criptografía de curvas elípticas está catalogada como un tipo de tecnología madura y fuerte, sin embargo existe la amenaza latente de que en un futuro no muy lejano, las técnicas criptográficas (RSA, ECC, etc.) y toda implementación de una PKI puedan sufrir el peor de los golpes ya que está en camino la herramienta con la cual romper una llave ECC, por ejemplo, sea una operación trivial o bien una tarea de sólo fracciones de segundo. La tecnología a la cual nos referimos es la computación cuántica.

La física cuántica, una rama de la ciencia, complementaria y a veces no tan congruente con la física clásica, trata de explicar la realidad universal a partir del comportamiento y las propiedades de lo muy pequeño, es decir trata de realizar una descripción del universo yendo más a allá de los átomos y de las partículas subatómicas. La computación cuántica es una teoría relativamente joven (la física cuántica cumplió 100 años en 2000 y la computación cuántica se comenzó a desarrollar en los 80's) que trata de aprovechar las propiedades y comportamientos subatómicos para conformar máquinas que puedan ser más eficientes tanto en velocidad y almacenamiento de información como en seguridad.

En una computadora convencional la información está representada por bits, los cuales constituyen la unidad básica de información en sistemas digitales; estos bits se caracterizan por poder tomar en cierto momento el valor 0 ó 1 pero no ambos al mismo tiempo, además estos valores responden a una cierta situación eléctrica, por ejemplo un condensador cargado lo podemos considerar como 1, y descargado como 0. En una computadora cuántica se reemplaza el concepto de bit por el de qubit o bit cuántico, entre ellos la diferencia más radical es que un bit es independiente a los demás en un arreglo convencional, además de que, como ya se había mencionado, un sólo bit puede valer 1 ó 0 en un momento determinado; los qubits por el contrario pueden contener al mismo tiempo los dos valores 0 y 1 de una forma superpuesta y combinada, además de que cada qubit puede ser no independiente de los demás en un arreglo ya que se pueden establecer relaciones entre ellos que incidan sobre el comportamiento de cada uno formando estados colectivamente enredados los cuales traen como consecuencia mayor velocidad en el procesamiento de operaciones. A diferencia de los bits convencionales que se almacenaban según la situación eléctrica de un condensador, los qubits son más complejos en su estructura física ya que en vez de utilizar dispositivos electrónicos para su representación, se utilizan (en algunos prototipos) grupos de iones (átomos a los que se les ha quitado un electrón) que se mantienen juntos y controlados por medio de trampas electromagnéticas, y en otros casos se trata de representar a los qubits aprovechando el spin (propiedad de giro en cualquier átomo) en donde el sentido de giro es el que determina su valor (*ver fig 3.14*). En ambos casos la manipulación de los qubits es a partir de la incidencia de un flujo de energía láser, lo cual es la parte análoga a un lector de una computadora convencional.

Con el desarrollo de la computación cuántica se han logrado hacer algoritmos novedosos que rompen, obviamente, con lo convencional y es aquí justamente donde debemos poner atención ya que el peligro que representa este avance tecnológico a las PKI actuales es de serias implicaciones, más bien de devastadoras implicaciones tan sólo porque toda la criptografía de la actualidad quedaría al descubierto y superada al 100%, o sea inservible.

El IFP (Integer Factorization Problem), problema matemático duro en el que se basa la criptografía de llave pública RSA, es una situación de difícil solución para los algoritmos y equipos de cómputo actuales, ya que factorizar un número de varios cientos de dígitos con precisión, le llevaría a una computadora normal millones de años para lograrlo, y esto se ha comprobado científicamente, en cambio el mismo problema en una computadora cuántica no se llevaría más de un segundo en encontrar los factores primos que satisfagan la solución, de hecho a finales de los 70's el científico Peter Shor de los laboratorios Bell AT&T, propuso ante la comunidad científica un algoritmo basado en la teoría de la computación cuántica que resuelve con eficiencia el IFP, fundamento popular de la criptografía actual. Así pues se ha dicho, de manera un tanto optimista, que en el transcurso de unos 20 o 30 años se estará vendiendo la primer computadora cuántica en versión hogareña, lo cual pone a temblar a cualquier profesional de la seguridad (como lo pretendemos ser nosotros) pues todos los esquemas actuales simplemente se caerían.

Tampoco hay que caer en pánico, si se desarrolla esta supermáquina es obvio que se desarrollen también nuevos algoritmos, esquemas y sistemas de seguridad tal y como lo están haciendo científicos e investigadores como Charles Benett y Gilles Brassard quienes desarrollaron el primer sistema de cifrado cuántico llamado BB84 en 1984. A pesar de todo, confiamos en que la evolución de la tecnología siga su camino de forma coherente al

entorno y esperar de esta forma que no se produzca algún caos informático o desequilibrio en la comunidad investigadora.

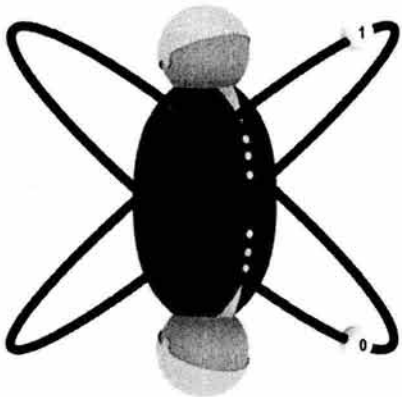


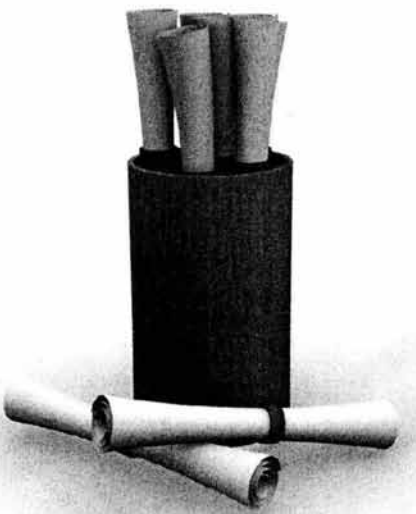
Fig 3.14 Representación de un qubit; éste tiene los dos estados 1 y 0 al mismo tiempo.

Comentario final

La autenticación con certificados digitales es un avance tecnológico que no es tan utilizado como debería ser, es por esto que el presente proyecto está planeado para ofrecer una alternativa y una sólida base con la que se pueda lograr un esquema de comunicación seguro y de autenticación confiable dentro de la UNAM. En el siguiente capítulo se brindará un panorama amplio de los estándares en los cuales hemos basado nuestro trabajo.

CAPÍTULO IV

Estándares de seguridad



Introducción

Los estándares de seguridad son documentos en los cuales podemos basarnos para la creación e implementación de un producto o desarrollo seguro; a partir de ellos se puede determinar la clasificación de la utilidad de un proyecto así como la seguridad que éste ostenta. Los estándares identifican los criterios de seguridad que un producto debe tener para ser considerado como útil dentro de un entorno de trabajo como lo puede ser el gobierno o alguna organización privada. Esta clasificación está basada en las características de seguridad implementadas y descritas en estos documentos.

Las normativas de seguridad empezaron a desarrollarse a finales de los años setenta cuando surgió la necesidad de proteger ciertas comunicaciones que no pertenecían a los restringidos ambientes militares y diplomáticos, hasta entonces únicos usuarios de comunicaciones seguras. Bancos y transnacionales fueron los primeros promotores y los primeros beneficiados de la normalización de los métodos de protección de la información y de las comunicaciones. Las organizaciones internacionales dedicadas a la elaboración de estos estándares han generado varios documentos abarcando a los distintos aspectos de seguridad, y han incrementando el nivel de seguridad de los algoritmos al hacerlos de dominio público.

A continuación mostramos un resumen de las organizaciones y estándares más importantes en el aspecto de seguridad, sin embargo la intención de este capítulo es mostrar un panorama más real de lo que hemos platicado; en capítulos anteriores vimos los requerimientos de certificación, el certificado y las CRL's, ahora es momento de ver por medio de los estándares como se estructuran estos elementos.

Vamos a estudiar algunos estándares de seguridad a detalle como lo son: el PKCS 10, el Directorio X.500 y el más importante para esta tesis, el X.509. También incluimos una breve explicación de la notación ASN.1 para la mejor comprensión de estos estándares.

4.1 Unión internacional de Telecomunicaciones (ITU)

El ITU con sede en las Naciones Unidas, es una organización mundial donde actúan organizaciones gubernamentales y el sector privado para coordinar el establecimiento, operación y servicios de las redes de telecomunicaciones, son los responsables de la regularización, estandarización, y desarrollo de las telecomunicaciones internacionales además de la armonización de las políticas nacionales.

ITU-T

El ITU-T, es uno de los tres sectores del International Telecommunication Union (ITU); éste fue creado el 1 de Marzo del 1993, dentro de la estructura del nuevo ITU reemplazando al antiguo International Telegraph and Telephone Consultative Committee (CCITT) cuyos orígenes van desde 1865.

También fueron creados otros dos sectores el ITU-R Radiocommunication y el ITU-D Telecommunication Development.

El trabajo de estandarización es realizado por 12 grupos de estudio y un grupo especial, en el cual miembros representantes del ITU-T desarrollan recomendaciones para

los varios campos de las telecomunicaciones internacionales, estas son generalmente complicadas porque tienen que garantizar la interconectividad de redes.

Actualmente existen más de 2700 recomendaciones (estándares) en más o menos 70,000 páginas en vigor.

La ITU ha adoptado el modelo OSI con los servicios y protocolos en las recomendaciones de la serie X.2xx. Referentes al tema de la seguridad hay dos recomendaciones concretas cuya última versión data de mediados del año 1994: el estándar X.273, en el que se describe el protocolo de seguridad para la capa de red, y la recomendación X.274, donde se define el protocolo de seguridad para la capa de transporte.

En la capa de aplicación destacan las definiciones del sistema de mensajería que aparecen en la serie X.4xx y el directorio X.500, especialmente la X.509. En ambos casos los documentos definen de forma amplia los procedimientos para proporcionar seguridad a los distintos elementos y funciones, aunque sólo se hace una breve mención sobre los algoritmos.

En la serie X.7xx referente a la gestión del modelo OSI, aparecen algunas recomendaciones relativas a la seguridad como la X.736, la X.737, la X.740 y X.741. Aunque las recomendaciones X.736 y X.740 fueron redactadas en 1992, en 1995 sufrieron algunas modificaciones y se redactaron las otras dos. En ellas se tratan los temas de alarmas, tests, auditorías y atributos para control de acceso.

La recomendación X.800 redactada desde 1991 describe la arquitectura de seguridad para OSI. Durante 1995 se publicaron el resto de publicaciones de la serie X.8xx, hasta un total de doce, y en ellas se tratan de forma genérica los distintos servicios de seguridad (confidencialidad, integridad, control de acceso, autenticación) en cada una de las capas del modelo OSI.

Para los propósitos de esta tesis vamos a estudiar al Directorio X.500 y al estándar X.509. El estándar X.509 es una estructura de autenticación que soporta servicios del directorio X.500, Ambos pertenecen a la serie X de los estándares internacionales propuestos por ISO e ITU. El estándar X.500 está diseñado para proporcionar servicios de directorio en extensas redes de computadoras, como el caso de grandes empresas o universidades.

Antes de comenzar con el estándar X.509 debemos conocer al directorio X.500 para comprender a qué nos referimos cuando hablamos de un nombre distinguido en una petición de certificado o en un certificado digital.

4.1.1 Directorio X.500

El directorio X.500 es muy similar al directorio telefónico que comúnmente utilizamos, en donde, con el nombre de una persona uno puede encontrar información acerca de ésta, como es la dirección o el teléfono. En el directorio X.500 se puede encontrar mucho más que estos simples datos, se puede obtener el nombre de la empresa donde trabaja la persona, el puesto que ocupa ahí, su dirección de e-mail, etc.

El directorio X.500, podría decirse, es la guía telefónica de las redes internacionales (internet) ya que permite buscar direcciones, números de teléfono y direcciones de correo electrónico de las personas que forman las organizaciones que están conectadas a internet.

Pero ¿cómo funciona este directorio?, cada entrada en el directorio describe un objeto (por ejemplo: una persona, un recurso de red, una organización) a este objeto se le asigna un único identificador llamado **Nombre Distinguido (DN**, Distinguished Name).

El directorio X.500 está ordenado de manera jerárquica, llamada árbol de información de directorio (DIT). La entrada consiste en una colección de atributos (por ejemplo una persona podría tener apellido, organización, e-mail). Para encontrar las entradas hay que navegar a través del **DIT**. En la raíz del árbol se encuentra "El Mundo", el cual está subdividido en el siguiente nivel, en países, y en el siguiente en organizaciones. Dentro de las organizaciones se almacena información de gente, recursos, etc. *Fig. 4.1.*

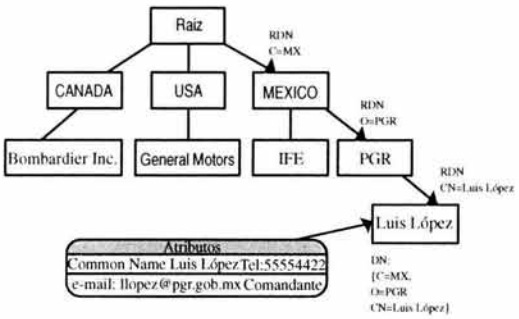


Fig. 4.1 Árbol de información de directorio (DIT)

4.1.1.1 ¿Para qué sirve?

El servicio ofrecido para personas es proporcionar una base de datos distribuida con datos de interés como pueden ser direcciones e-mail, tipo de actividad, dónde se desempeña ésta e incluso fotos y documentos sonoros de los miembros. Actualmente no contempla un ámbito comercial y está restringido a universidades y organizaciones internacionales.

Respecto al servicio para máquinas el más importante es el de proporcionar servicios de **mapeado de UFN (User Friendly Naming)** a direcciones de máquinas o aplicaciones, de tal manera que el usuario pueda utilizar direcciones amigables y fácilmente recordables, haciéndose transparente las complicadas direcciones OSI de presentación.

4.1.2 Estándar X.509

Para que una llave pública se pueda considerar válida y asociada a un usuario determinado debe tener un certificado que así lo demuestre. Esta confianza se obtiene por medio de los certificados de llave pública, que son estructuras de datos que vinculan información de un sujeto con la llave pública. Para ello vamos a estudiar el estándar X.509 que es el documento en el que nos basamos para definir los certificados digitales de esta tesis. Vemos directamente la versión 3 del certificado porque es la que nos interesa que se conozca a fondo.

El estándar X.509 describe el formato y la sintaxis de los certificados y de la CRL, los procedimientos para las rutas de certificación en el ambiente de internet, algoritmos

criptográficos y finalmente módulos ASN.1 para todas las estructuras de datos definidas o referenciadas.

La finalidad de este documento es que se cuente con una descripción que facilite el uso de los certificados X.509 dentro de una aplicación en internet, para aquellos que deseen usar la tecnología X.509, tales como usuarios web, de correo electrónico, autenticación de usuarios, etc. Este documento tiene la intención de apoyar al desarrollo de sistemas que manejen certificados digitales.

El ITU-T X.509 (antes CCITT X.509) o ISO/IEC/ITU 9594-8, el primero publicado en 1988 como parte de las recomendaciones de directorio X.500, define un formato de certificado estándar, el X.509.

El formato X.509 versión 1 publicado en 1988 fue conocido como modelo de certificado estándar; cuando el X.509 fue revisado en 1993, dos campos fueron agregados, resultando el formato versión 2 (v2). Estos dos campos pueden ser usados para apoyar el control de acceso al directorio.

El Internet Privacy Enhanced Mail (PEM) publicado en 1993, incluye datos específicos para una infraestructura de llave pública basada en certificados X.509 v1. La experiencia ganada en pruebas aclaró que los formatos de certificado versión 1 y versión 2 eran deficientes en varios aspectos.

En respuesta a estas nuevas exigencias, ISO/IEC/ITU y ANSI X9 desarrollaron el formato de certificado versión 3, X.509 (v3). El formato versión 3 amplía el formato de la versión 2 añadiendo campos de extensión adicionales.

Con la versión 3, no hace falta aplicar restricciones sobre la estructura de las AC gracias a la definición de las extensiones de certificados. Se permite que una organización pueda definir sus propias extensiones para contener información específica dentro de su entorno de operación. En junio 1996, la estandarización del formato de certificado v3 fue completado.

4.1.2.1 Campos Básicos.

Los certificados X.509 son archivos que contienen distintos campos donde se almacena la llave pública de un usuario, una firma digital, información sobre los algoritmos criptográficos utilizados, datos del usuario y datos de la AC que emitió el certificado. En la última versión (v3) se han incluido extensiones de información adicional sobre el uso al que están destinadas las llaves y sobre las políticas de certificación asociada. *Figura 4.2.*

El certificado X.509 v3 usa notación ASN.1 para la sintaxis del estándar. La definición de la estructura en notación ASN.1 (*Abstract Syntax Notation One*) es la siguiente:

El certificado esta definido como una secuencia (SEQUENCE) de tres campos:

```
Certificate ::= SEQUENCE {
tbsCertificate          TBSCertificate,
signatureAlgorithm      AlgorithmIdentifier,
signatureValue          BIT STRING }
```

Donde el primer campo es:

tbsCertificate. El TBSCertificate también es una secuencia de datos, que contiene información asociada con el sujeto del certificado y la AC quien lo emitió. Cada TBSCertificate contiene los nombres del sujeto y del emisor, una llave pública asociada con el sujeto, un período de validez , un número de versión y un número de serie, también puede incluir extensiones.

```
TBSCertificate ::= SEQUENCE {
    version            [0]  EXPLICIT Version DEFAULT v1,
    serialNumber            CertificateSerialNumber,
    signature               AlgorithmIdentifier,
    issuer                  Name,
    validity                Validity,
    subject                 Name,
    subjectPublicKeyInfo    SubjectPublicKeyInfo,
    issuerUniqueID     [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                            -- If present, version must be v2 or v3

 subjectUniqueID       [2] IMPLICIT UniqueIdentifier OPTIONAL,
                           -- If present, version must be v2 or v3
 extensions            [3] EXPLICIT Extensions OPTIONAL
                           -- If present, version must be v3
}
```

Dentro del TBSCertificate encontramos los siguientes campos, también incluimos su definición:

version (**Versión**): Indica si la versión del certificado X.509 es la 1 (por default), 2 ó 3. Los sistemas que utilizan certificados X.509, deben reconocerlos independientemente de la versión que se trate.

```
Version  :: = INTEGER {v1(0), v2(1), v3(2)}
```

serialNumber (**Número de serie**): Es un entero asignado por la AC para cada certificado, es el número de identificación único para el certificado.

```
CertificateSerialNumber :: = INTEGER
```

signature (**Firma**): Identifica al algoritmo utilizado por la AC para firmar el certificado. Este campo puede contener el mismo identificador de algoritmo que el campo *signatureAlgorithm* en la secuencia del certificado.

issuer (**Emisor**): El nombre del emisor identifica a la entidad que ha firmado el certificado (Autoridad Certificadora) y sigue la nomenclatura de nombres distinguidos (DNs, *Distinguished Names*) del Directorio X.500.

El nombre está definido por la siguiente estructura ASN.1:

```
Name  :: = CHOICE {
      RDNSequence  }

RDNSequence :: = SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName  :: =
      SET OF AttributeTypeAndValue

AttributeTypeAndValue   :: = SEQUENCE {
      type              AttributeType,
      value             AttributeValue    }

AttributeType :: = OBJECT IDENTIFIER
AttributeValue :: = ANY DEFINED BY AttributeType
```

validity **(Validez).** Indica el intervalo de tiempo en el cual el certificado es válido. Este campo esta representado como una secuencia de dos datos, uno indica cuando comienza el período de validez (notBefore) y el otro cuando finaliza (notAfter). Ambos pueden ser codificados como UTCTime o GeneralizedTime.

```
Validity :: =  SEQUENCE  {
      notBefore         Time,
      notAfter          Time  }

Time :: = CHOICE  {
      utcTime           UTCTime,
      generalTime       GeneralizedTime   }
```

Las AC's conforme a esta descripción deben codificar siempre el período de validez antes del año 2049 en tiempo UTCTime; el período de validez después del año 2050 deberá ser codificado en GeneralizedTime.

subject **(Usuario o sujeto).** Es un nombre distinguido X.500 que identifica de forma única al poseedor del certificado. Identifica el nombre del propietario de la llave privada.
El sujeto se define como un Name, al igual que el emisor, su estructura ASN.1 es la misma.

subjecPublicKeyInfo **(Información de la llave pública del usuario).** Contiene la llave pública del usuario junto con el identificador del algoritmo con el que se ha de utilizar.

```
SubjectPublicKeyInfo  :: =  SEQUENCE {
      algorithm              AlgortihmIdentifier,
      subjectPublicKey       BIT STRING        }
```

Unique Identifiers **(Identificadores únicos de emisor y de usuario).** Es una cadena de bits opcional que identifica al emisor o al usuario en el caso de que su DN sea reutilizado con el paso del tiempo. Este campo puede aparecer sólo en la versión 2 o 3.

```
UniqueIdentifier  :: =  BIT STRING
```

Extensions **(Campo extensiones)**. Este campo aparece solo en la versión 3, este campo es una secuencia de una o más extensiones del certificado. Permiten la adición de nuevos

campos a la estructura sin que por ello se tenga que modificar la definición ASN.1 del certificado. Cada uno de estos campos consiste en:

- un identificador de extensión,
- un valor que indica si es o no crítico, y
- una codificación canónica de un valor de un tipo ASN.1 asociado con la extensión identificada

```
Extensions :: = SEQUENSE SIZE  (1..MAX)  OF Extension

Extension  :: =  SEQUENCE {
      extnID            OBJECT IDENTIFIER,
      critical          BOOLEAN DEFAULT  FALSE,
      extnValue         OCTET STRING      }
```

Fig. 4.2 Campos básicos del certificado X.509 version 3

4.1.2.2 Extensiones estándares del certificado

ISO/IEC, ITU y ANSI han definido un conjunto de extensiones estándares para ser utilizadas en los certificados X.509 v3, las cuales proporcionan métodos para asociar atributos adicionales con usuarios o llaves públicas y para gestionar la jerarquía de certificación y la distribución de CRL's.

La versión 3 introduce cambios significativos en el estándar X.509, el principal cambio fue hacer al formato del certificado y a las CRL más extensibles, ahora se puede

definir el contenido del certificado como sea conveniente. Cada extensión en un certificado puede estar definida como crítica o no crítica. Un certificado usado en un sistema podría rechazar el certificado si existe una extensión crítica porque no la reconoce, sin embargo si se tiene una extensión no crítica la puede ignorar si no la reconoce.

Las AC's tienen la capacidad para definir como críticas o no críticas las extensiones: *Key usage, Certificate policies, Subject alternative names, Issuer alternative names, Basic constraints, Name constraints y Policy constraints*; todas las demás son siempre no críticas.

Las extensiones estándares pueden dividirse en varios grupos según el aspecto con el que estén relacionadas:

Información de llaves.

- **Authority key identifier**. Debido a que las autoridades certificadoras en ocasiones poseen varios pares de llaves, ya sea por la demanda de estas o por que se han visto comprometidas y se deben de cambiar, esta extensión proporciona las medidas para identificar a la llave pública correspondiente a la llave privada que ha sido usada para firmar el certificado.
  Dicha identificación se realiza buscando: el identificador de la llave (el identificador de la llave del sujeto dentro del certificado emitido) o el nombre y número de serie del emisor.

- **Subject key identifier**. Permite identificar una determinada llave pública utilizada en una aplicación. Esta extensión se utiliza cuando un usuario ha obtenido varios certificados y además de varias AC, el identificador de llave del sujeto proporciona un método rápido para identificar la serie de certificados que contienen una llave pública en particular.

- **Key usage**. Define el uso de la llave contenida en el certificado, por ejemplo puede usarse para cifrar o para firmar certificados. Cuando una llave es utilizada para más de una operación tiene algunas restricciones.

```
KeyUsage : : BIT STRING   {
digitalSignature      (0),
nonRepudation         (1),
keyEncipherment       (2),
dataEncipherment      (3),
keyAgreement          (4),
keyCertSign           (5),
crLSign               (6),
encipherOnly          (7),
decipherOnly          (8)     }
```

Cuando usamos esta extensión debe estar marcada como crítica.

- **Private Key Usage Period.** Especifica el período de validez de la llave privada asociada al certificado. Esta extensión esta dirigida para uso con llaves de firma

digital. La llave privada no puede firmar objetos antes o después de este tiempo establecido.

Información de Políticas. Estas extensiones indican las políticas bajo las cuales el certificado ha sido emitido y los propósitos para el uso del certificado, además facilitan la implementación de las infraestructuras de llave pública. Dentro de este grupo se encuentran:

- **Certificate policies** (Políticas de certificado). Estas extensiones le dan a la AC la habilidad de incluir con el certificado una lista de políticas a seguir cuando se va a crear el certificado. Estas políticas están intencionadas para ayudar al usuario a decidir si el certificado que requiere es conveniente o no para tal objetivo. Por ejemplo una política podría indicar que una llave certificada puede ser usada para correo electrónico pero no para transacciones financieras. En general una política de certificado indica cosas como procedimientos de seguridad para la AC o medidas de identificación para el usuario y otras.

- **Policy mappings**. Este campo, que sólo puede ser empleado en certificados de AC, permite a la AC indicar si una de sus políticas es equivalente a otras empleadas por otra AC.

Atributos de emisor y usuario. Proporcionan medios alternativos para identificar a los emisores o usuarios y también incluyen información adicional sobre el sujeto para facilitar que los usuarios del certificado consideren al sujeto confiable. Este conjunto de extensiones está formado por las siguientes:

- **Subject alternative name** (Nombre alternativo de sujeto). Un certificado versión 3 puede contener uno o varios nombres alternativos para identificar al sujeto. Esto permite que opere sin un directorio X.500, y así dar más diversidad al certificado, un ejemplo de un nombre alternativo puede ser la dirección de correo electrónico, un nombre DNS o una dirección IP, los implementadores pueden definir sus propias formas de nombres alternativos.

  Todas las partes del nombre alternativo deben ser verificadas por la AC para que se considere como seguro para la llave pública.
  Si el sujeto sólo se identifica incluyendo en el certificado una forma de nombre alternativo (por ejemplo correo electrónico), entonces el nombre distinguido del sujeto debe estar vacío, es decir el campo del certificado Subjetc (Sujeto) estará vació, en este caso la extensión de nombre alternativo del sujeto estará marcada como critica.

  Dependiendo del nombre alternativo, ya sea una dirección IP, correo u otra forma, su semántica cambia, cada nombre alternativo se define de manera distinta, el formato de esta extensión presenta una lista de las posibles formas de nombres. También un nombre alternativo puede servir para identificar al emisor de una CRL.

- **Issuer alternative name** (Nombre alternativo del Emisor). De la misma manera que actúan los nombres alternativos de sujeto, los nombres alternativos del emisor proporcionan nombres alternativos para identificar al emisor del certificado o de la CRL.

  Esta extensión si se presenta no debe ser marcada como crítica.

- **Subject directory attributes**. Puede contener cualquier atributo X.500 del sujeto del certificado y proporciona un medio para incluir información adicional a la incluida en los campos de nombre y que puede ser útil para identificar al sujeto.

Restricciones de caminos de certificación. Estos campos permiten a las diferentes entidades de la jerarquía poder enlazar sus estructuras. Las extensiones de este grupo son las siguientes:

- **Basic constraints**. Indica si el sujeto puede actuar como una AC o es sólo una entidad final. Si el sujeto actúa como AC, se define el número máximo de certificados AC que tiene que seguir un certificado en una ruta de certificación. Si este valor se define como cero indica que sólo es una entidad final.

- **Name constraints**. Este campo, que sólo puede incluirse en certificados de AC, restringe el campo de nombres válidos en los subsecuentes certificados emitidos por la AC asociada.

- **Policy constraints**. Esta extensión puede ser utilizada por las AC's para especificar requisitos que pueden necesitar la identificación explícita de políticas de certificación o inhabilitar el mapeo de políticas para el resto del camino de certificación.

4.1.2.3 CRL

Un certificado puede perder su validez (antes de que caduque) por que la llave privada del usuario ha sido comprometida, la AC no certifica al usuario o porque la llave privada de la AC ha dejado de ser secreta. Este hecho lo difunde públicamente la autoridad de certificación mediante las listas de revocación de certificados, cuya representación en notación ASN.1 es:

La sintaxis de la CRL X.509 versión 2 es la siguiente:

```
CertificateList          :: =  SEQUENSE    {
      tbsCertList                TBSCertList,
      signatureAlgorithm         AlgorithmIdentifier,
      signatureValue             BIT STRING  }

TBSCertList :: =  SEQUENCE     {
      Version            Version OPTIONAL ,
                         -- if present, shall be v2
```

```
signature           AlgorithmIdentifier,
issuer              Name,
thisUpdate          Time,
nextUpdate          Time OPTIONAL,
revokedCertificates     SEQUENCE Of SEQUENCE    {
      usercertificate           CertificateSerialNumber,
      revocationDate            Time,
      crlEntryExtensions        Extensions OPTIONAL
                                -- if present, shall be v2
                                } OPTIONAL,
     crlExtensions              [0]  EXPLICIT Extensions OPTIONAL
                                if present, shall be v2
                                     }
```

A grandes rasgos una lista de certificados revocados incluye los datos de la AC que ha emitido tal lista de revocación (issuer), la fecha de emisión de la lista (thisUpdate), la fecha de próxima emisión de otra lista actualizada (nextUpdate) y una secuencia de entradas, donde cada elemento se compone del número de serie del certificado (CertificateSerialNumber) y el instante en que ha dejado de tener validez (Time), opcionalmente es posible un tercer campo de extensiones. Por ejemplo, si el certificado número 26 deja de ser válido, la AC emite una lista donde una de las entradas incluye el valor 26 (número de serie del certificado).

Un certificado deja de ser válido en el momento que figura en una lista de revocación. En contraste, los certificados que han caducado no aparecen en ninguna lista, por tanto, cuando un certificado revocado supera su fecha de validez se elimina de la lista de revocación de certificados.

La recomendación X.509 especifica que cada AC debe mantener dos listas de revocación, una para los certificados que ha emitido y revocado, así como una segunda de certificados que no ha emitido, pero sabe que son revocados. La primera lista se conoce como "lista de certificados revocados" (Certificate Revocation List, **CRL**) y la segunda se denomina "Lista de Revocación Autorizada" (Authority Revocation List, **ARL**). Ambas listas corresponden a los atributos **certificateRevocationList** y **authorityRevocationList**.

4.1.2.3.1 Extensiones CRL

- **CRL Distribution Points**. Puntos de distribución de CRL's. Esta extensión ayuda a reducir el tamaño de las CRL's hechas por la AC. En vez de forzar a los usuarios a aceptar la CRL completa, la AC puede dividir a la CRL en varios caminos y publicar cada parte desde un punto de distribución diferente. Por ejemplo en una compañía la AC podría emitir una CRL diferente para cada departamento de la compañía. Entonces cuando los usuarios deseen verificar algún certificado en particular sólo tendrían que checarlo en la CRL de ese departamento, en vez de checar toda la CRL. Otra manera de fraccionar la CRL sería acordar los motivos de la revocación, de esta forma las listas serían más pequeñas y más específicas, además que son mas fáciles de actualizar.

- **Delta CRL's**. Una Delta CRL es simplemente una lista con los cambios que han ocurrido desde que se emitió la última CRL completa. Este método permite notificaciones de revocación mas frecuentes reduciendo la probabilidad de que un certificado revocado sea falsamente validado. Esta extensión también ayuda a reducir el tamaño de la CRL.

De lo que se trata es de lo siguiente, se publica la CRL completa, y lo único que tiene que hacer la AC es publicar una lista con los cambios que han ocurrido desde la última fecha en que se emitió la CRL. Los usuarios que manejan su propia base de datos de la CRL pueden usar esta extensión para guardar sólo las copias actualizadas, sin tener que bajar de nuevo la CRL completa ahorrando recursos y tiempo.

4.2 FIPS

El gobierno de los Estados Unidos ha redactado las normas denominadas Federal Information Processing Standards (FIPS). Las más destacables relacionadas con el tema de la seguridad son las siguientes:

- FIPS 46-2, FIPS 74 y FIPS 81: algoritmo DES, formas de implementación y modos de operación.
- FIPS 112: utilización de passwords
- FIPS 113: autenticación de datos
- FIPS 180-1: SHA-1
- FIPS 185: Key escrow
- FIPS 186: DSS

4.3 PKCS

Estándares de Criptografía de Llave Pública (Public Key Cryptography Standards) desarrollados por RSA Corporation en forma conjunta con Apple, Microsoft, Digital, Lotus, Sun y Massachussets Institute of Technology (MIT), con el objetivo de acelerar el crecimiento de la criptografía de llave pública, y de facilitar la adopción de estos estándares.

En general los estándares PKCS son en realidad un conjunto de estándares que especifican la "sintaxis de mensajes" y definición de "algoritmos específicos" que nos permiten conocer como guardar la llave privada, la funcionalidad del algunos algoritmos y las reglas de codificación de un requerimiento de certificación o de un certificado.

Los PKCS sólo describen la sintaxis de los mensajes en forma abstracta dando detalles completos acerca de los algoritmos, por esa razón están codificados en notación ASN.1.

A continuación se muestra este conjunto de estándares:

PKCS 1: Este estándar define los mecanismos para cifrar y firmar datos usando el sistema de llave pública RSA.
PKCS 3: Define el Intercambio de llaves Diffie-Hellman
PKCS 5: Cifrado basado en passwords, describe un método para generar una llave privada basada en un password.
PKCS 6: Sintaxis de certificados. Este estándar actualmente ha sido reemplazado por el estándar X.509.
PKCS 7: Sintaxis de mensajes cifrados, describe como se firma un mensaje.
PKCS 8: Sintaxis de información de la llave privada. Este define un método para almacenar información en la llave privada.

PKCS 9: Define una selección de tipos de atributos para usarse en otros estándares PKCS.
PKCS 10: Describe la sintaxis para el requerimiento de certificación.
PKCS 11: Este estándar define una tecnología independiente para la interface de dispositivos criptográficos tales como smartcards.
PKCS 12: Describe un formato portátil para almacenar y transportar llaves privadas, certificados, etc.
PKCS 13: Criptografía con curvas elípticas. Describe los mecanismos para cifrar y firmar datos usando criptografía de curvas elípticas.
PKCS 14: Generación de números pseudo-aleatorios. Este estándar actualmente se encuentra en desarrollo.
PKCS 15: Describe un estándar para el formato de credenciales criptográficas almacenadas en tokens criptográficos.

4.3.1 PKCS 10 Requerimiento de certificación

Es importante detenernos en este estándar ya que es el que nos apoya en la realización del requerimiento de certificación que es el primer paso del proceso de certificación de un usuario. Como se había mencionado, el requerimiento de certificación es una solicitud que contiene datos representativos del solicitante (nombre personal o de la empresa, domicilio personal o social, correo electrónico, puesto, etc.) y su llave pública. El requerimiento es un documento auto-firmado o firmado digitalmente por el usuario, esto significa que está conciente que acepta la llave pública y ser el legítimo dueño de la llave privada.

Un requerimiento de certificación consiste de un nombre distinguido (DN) que contiene los datos del usuario, una llave pública y una serie de atributos opcionales, colectivamente firmados por el solicitante (persona, empresa, terminal, etc).

Este es enviado a la autoridad certificadora, la cual va a transformar el requerimiento en un certificado de llave pública X.509.

A continuación presentamos la sintaxis del requerimiento de certificación.

El requerimiento de certificación se compone de una secuencia de tres partes:

1. Información del requerimiento de certificación
2. Un identificador del algoritmo de firma
3. Y por último una firma digital en la información del requerimiento.

```
CertificationRequest :: =     SEQUENCE {
   certificationRequestInfo   CertificationRequestInfo
   signatureAlgorithm         AlgorithmIdentifier {{SigantureAlgorithms}}
   signature                  BIT STRING
}
```

Como vemos, la información **certificationRequestInfo** está contenida dentro del requerimiento de certificación **(CertificationRequest),** hay que entender que son dos cosas completamente diferentes.

Información del requerimiento de certificación.

Para la información de requerimiento (CertificationRequestInfo) vamos a tener: el nombre distinguido, la llave pública y una serie de atributos que nos proveerán información acerca de la entidad.

```
CertificationRequestInfo ::= SEQUENCE {
      version           INTEGER{v1(0)} (v1,...),
      subject           Name,
      subjectPKInfo     SubjetcPublicKeyInfo{{PKInfoAlgorithms}},
      attributes        [0] Attributes{{CRIAttributes}}
      }

SubjectPublicKeyInfo{ ALGORITHM :IOSet} ::= SEQUENCE{
      algorithm         AlgorithmIdentifier{{IOSet}},
      subjectPublicKey  BIT STRING
      }

PKInfoAlgorithms ALGORITHM ::={
      .... -add any locally defined algorithms here--}

Attributes {ATTRIBUTE:IOSet} ::= SET OF Attribute{{IOSet}}

CRIAttributes ATTRIBUTE ::={
      ... -add any locally defined attributes here--}

Attribute {ATTRIBUTE:IOSet} ::= SEQUENCE{
      type        ATTRIBUTE.&id({IOSet}),
      values      SET SIZE(1...MAX) OF ATTRIBUTE.&Type({IOSet}{@type})
      }
```

La estructura CertificationRequestInfo es una secuencia de datos que contiene:

version: es el número de versión, para la compatibilidad con futuras revisiones de este documento.
subject: es el nombre distinguido del sujeto que quiere ser certificado (la entidad cuya llave pública va a ser certificada)
subjectPublicKeyInfo: Contiene información acerca de la llave pública que va a ser certificada. Esta información identifica al algoritmo de llave pública, ejemplo rsaEncryption, también incluye un bit string donde se representa la llave pública. Los valores de tipo **SubjectPublicKeyInfo{}** definidos por **subjectPKInfo** están restringidos a los valores específicos por la información de objetos PKIAlgorithms que incluyen una marca de extensión (...). Esta marca quiere decir que se puede agregar mas adelante objetos de algoritmos.
attributes: Provee de información adicional acerca del sujeto del certificado.

Los valores de tipo **Attributes{}** definido por **attributes** están restringidos por los valores especificados por la información del objeto **CRIAttributes.**
El valor de certificationRequestInfo debe ir firmado.

Algoritmo de firma (signatureAlgorithm).

Identifica el algoritmo de firma con el cual se ha firmado la información del requerimiento de certificación.

Firma (signature).

Es el resultado de firmar la información del requerimiento de certificación con la llave privada de la entidad.

Entonces de manera general el requerimiento o petición de certificado se construye así:

1. La entidad que necesita una petición de certificación, construye la información del requerimiento CertificateRequestInfo que contiene su nombre distinguido, su llave pública y atributos si lo desea.
2. Después el valor del CertificateRequestInfo es firmado con la llave privada de la entidad.
3. El valor de CertificateRequestInfo, un identificador de algoritmo de firma y la firma de la entidad juntos forman el Requerimiento de Certificación (CertificateRequest).

Sintaxis completa del requerimiento de certificación (CertificationRequest)

```
CertificationRequest ::= SEQUENCE {
  certificationRequestInfo    CertificationRequestInfo,
  signatureAlgorithm          AlgorithmIdentifier{{SignatureAlgorithms}},
  signature                   BIT  STRING
      }

AlgorithmIdentifier {ALGORITHM:IOSet} ::= SEQUENCE{
      algorithm   ALGORITHM.&id({IOSet}),
      parameters  ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
      }
SignatureAlgorithms ALGORITHM ::={
      ... -add any locally defined algorithms here--}
```

4.4 Normativa para Internet y RFC's

Actualmente el mecanismo más rápido para establecer una normativa que afecte a las redes informáticas es la discusión de los resultados de algunos grupos de trabajo a través de internet.

Internet Engineering Task Force (IETF) es una comunidad internacional abierta que conjunta a todas aquellas personas interesadas en la evolución de internet. Para actuar, se forman grupos de trabajo en un área determinada, y los miembros de cada grupo se comunican sus resultados a través de mensajes por correo electrónico.

Algunos grupos de trabajo relacionados con la seguridad son los siguientes:

PKIX: Public Key Infraestructure X.509. El grupo de trabajo PKIX fue establecido a finales de 1995 con el propósito de desarrollar estándares que soporten la PKI basados en el estándar X.509. El PKIX ha producido varios documentos informativos y estándares, entre ellos:

- RFC 2459: Certificate and CRL profile
- RFC 2510: Certificate Menagement Protocols
- RFC 2511: Certificate Request Message Format
- RFC 2527: Certificate Policy and Certification practices Framework
- RFC 2528: Representation of Key exchange algorithm (KEA) Keys in Internet X.509 Public Key Infraestructure Certificates
- RFC 2559: Operational protocols – LDAPv2
- RFC 2560: Online Certificate Status Protocol
- RFC 2585:Operational Protocols – FTP and HTTP
- RFC 2587:LDAPv2 Schema.

S/MIME: Mail Security

- RFC 2311: S/MIME Version 2 Message Specification
- RFC 2312: S/MIME Version 2 Certificate Handling
- RFC 2630: Cryptographic Message Syntax
- RFC 2631: Diffie-Hellman Key Agreement Method
- RFC 2632: S/MIME Version 3 Certificate Handling
- RFC 2633: S/MIME Version 3 Message Specification
- RFC 2634: Enhanced Security Servicies for S/MIME
- RFC 2785: Meted for Avoiding the "Small.Subgroup" attcks on the Diffie-Hellman Key Agreement Method for S/MIME.

4.5 ASN.1 Abstract Syntax Notation One

ASN.1 (Notación de sintaxis abstracta uno) ha llegado a ser un importante lenguaje universal para definir la representación de estructuras de datos y formatos de protocolos. Se requiere un lenguaje de definición de objetos estándar, así como las reglas de codificación. ASN.1 define un conjunto y estructuras de datos que puedan ser reutilizados de forma general por todas las aplicaciones que pretendan implementar estándares. Estos datos se deben poder representar, codificar, transmitir y decodificar.

Consta de dos partes:

- Sintaxis abstracta : Define tipos de datos.
- Sintaxis de transferencia : Define las reglas básicas de codificación.

Se trata de que todas las máquinas utilicen la misma notación sintáctica ASN.1, así, el transmisor convierte a la sintaxis abstracta de transferencia de ASN.1 para la transmisión, y el receptor convierte de ASN.1 a su formato interno.

El nivel de presentación codifica la información según una cierta sintaxis de transferencia. La sintaxis de transferencia define el formato de bits, que es lo que se transmite.

4.5.1 Sintaxis abstracta

La sintaxis abstracta ASN.1 es una notación flexible que permite definir una variedad de tipos de datos, desde tipos simples como enteros y cadenas de bits, hasta tipos de estructuras y juegos de secuencias.
Permite al usuario definir objetos simples y luego combinarlos para formar otros más complejos.

Los tipos de datos simples se denominan primitivos (*ver tabla 2*).

Los tipos de datos compuestos se denominan constructores, y definen tipos más complejos (*ver tabla 3*).

Cuando se establece una sesión, la capa de sesión negocia para que los dos extremos coincidan en la sintaxis abstracta, las reglas de codificación y los protocolos que van a emplear las aplicaciones. Todos estos son objetos y, por lo tanto, son nombrados por identificadores de objetos.

Los tipos que son relevantes para los estándares PKCS son los siguientes:

Tipo simple	Significado
INTEGER	Entero de longitud arbitraria
BOOLEAN	VERDADERO ó FALSO
BIT STRING	Cadena de 0 ó más bits
OCTET STRING	Cadena de 0 ó más octetos
ANY	Unión de todos los datos
NULL	Ningún tipo en absoluto
OBJECT IDENTIFIER	Nombre del objeto

Tabla 2. Tipos primitivos ASN.1

Constructor	Significado
SEQUENCE	Lista ordenada de varios tipos
SEQUENCE OF	Lista ordenada de un solo tipo, como un conjunto
SET	Colección desordenada de varios tipos
SET OF	Colección desordenada de un solo tipo
CHOICE	Cualquier tipo tomado de una lista dada

Tabla 3. Principales tipos constructores ASN.1

4.5.2 Sintaxis de transferencia

La sintaxis de transferencia usada por ASN.1 se llama BER (Basic Encoding Rules-Reglas Básicas de Codificación). Puede representar cualquier valor de ASN.1 como una cadena de octetos.

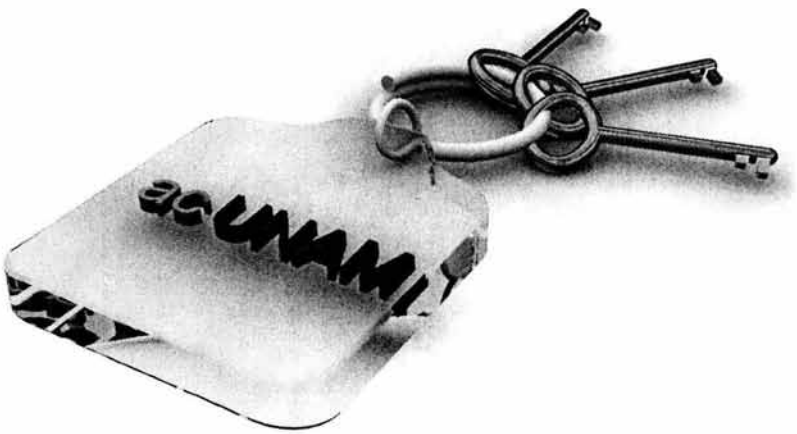
Por lo tanto, la sintaxis de transferencia nos indica cómo se transmiten los datos. El principio que guía las reglas básicas de codificación es que cada valor transmitido, tanto primitivo como constructor, consiste en cuatro campos:

- Identificador (8 bits): tipo o etiqueta.
- Longitud del campo de datos (8 bits): en bytes.
- Campo de datos.
- Bandera de fin de contenido (opcional): si se desconoce la longitud de los datos

De esta forma hemos dado un panorama general sobre los documentos internacionalmente usados, como estándar, que nos servirán a nosotros como base para el desarrollo de lo que queremos sea una autoridad certificadora para la UNAM. Con lo expuesto en este capítulo y en los anteriores estamos ahora en la antesala del desarrollo de este proyecto ya que contamos con las bases suficientes para comprenderlo y llevarlo al cabo. Así pues en el siguiente capítulo se hará la propuesta formal de nuestro sistema.

CAPÍTULO V

Descripción y desarrollo de una AC para la UNAM



Introducción

Una autoridad de certificación, como la planeamos para este proyecto, es tan sólo una parte integrante del conjunto que conforma una infraestructura de llave pública completa (PKI – Public Key Infrastructure), es por ello que deben tomarse en cuenta a los demás elementos de ésta para que el círculo cierre y funcione adecuadamente. Nos referimos a que una PKI está conformada por los siguientes elementos: técnicas criptográficas (asimétricas y simétricas), equipo de cómputo, usuarios, administradores, custodios, autoridades certificadoras, autoridades de registro, políticas de funcionamiento, etc., los cuales interactúan entre sí formando un sistema complejo de seguridad que tiene como base a la criptografía de llave pública, y como objetivo la protección de documentos electrónicos para poder garantizar el libre y seguro ejercicio de las transacciones digitales. Cabe mencionar que si no existieran las llaves pública y privada, una PKI simplemente no podría existir, ya que gracias a ellas y a la tecnología que hay detrás, se pueden establecer esquemas de protección de datos que nos facilitan algunas tareas sensibles y cotidianas.

En el presente capítulo, y después de haber explicado a detalle, en los anteriores, las bases para la construcción de una autoridad certificadora (AC), se describirá a continuación el proceso y las herramientas usadas para el desarrollo de una aplicación que emita y administre certificados digitales (además de requerimientos de certificación y listas de certificados revocados) como parte de una PKI. También se hará una propuesta tentativa para la puesta en marcha del sistema en la UNAM que pueda servir como plataforma de lanzamiento para un desarrollo aún más robusto. Este será pues un capítulo de carácter mas técnico que los anteriores debido a que se tocarán temas y términos referentes a programación, matemáticas, interpretación de estándares, y otros más que tienen que ver con las herramientas usadas, los cuales necesariamente deberán referenciarse de una manera, por decirlo así, menos amigable para el lector.

5.1 Herramientas necesarias para la construcción de acUNAM

Para el desarrollo de la AC, objetivo de esta tesis, las herramientas más poderosas con las que contamos son nuestra mente y nuestro cerebro pues es en ellos donde podemos, libremente, imaginar y crear la solución a un problema determinado después de haber comprendido la esencia del mismo. Nuestro cerebro, lugar de la mente y el pensamiento, es la herramienta que nos permite razonar todo este conocimiento abstracto del que ha sido objeto el presente tratado, sin embargo para hacer una extensión de dicho conocimiento al mundo práctico y funcional debemos salir de la cabeza y tomar la ayuda que nos brindan diversas herramientas externas, las cuales servirán como puente entre el mero pensamiento y la realidad objetiva. Así pues listaremos a continuación los instrumentos que nos han auxiliado en nuestra tarea.

5.1.1 Equipo de cómputo

Sin una computadora no sería posible el desarrollo de un sistema informático, de hecho sería incongruente el término, así pues y como no damos nada por hecho ni consideramos obvios algunos detalles, debemos mencionar que el desarrollo se hará en un equipo convencional con procesador Pentium II y con una plataforma Windows ME.

5.1.2 Software

En el largo camino que supone un desarrollo informático es necesario, en la mayoría de las veces, emplear aplicaciones que nos sirvan como auxiliar para la realización de pruebas y/o código reutilizable o reciclado, que nos permitan llevar a buen fin la construcción de un programa de cómputo. Además se hace indispensable el uso de software que nos ayude a realizar diagramas y documentos escritos, con el objetivo de documentar y conceptualizar un proyecto en particular, es decir, emplear aquel software que hace posible que usted esté leyendo estas líneas. La lista siguiente contempla el software libre, donado y comprado que se utilizó para realizar la AC.

Lib_233.dll

Lib_233.dll es un archivo al que podemos llamar el Kernel del sistema o el núcleo de la AC. Es un archivo de librería desarrollado totalmente en lenguaje C que contiene las funciones criptográficas usadas por la AC para cifrar, firmar y generar la pareja de llaves asimétricas usando algoritmos complejos como lo son DES (Data Encryption Standard), MD5 (Message Digest 5) y ECC (Elliptic Curve Cryptography). Al ser DLL (Dynamic Link Library) se le utiliza de forma activa durante la ejecución del programa-consola de la AC, es decir, es una librería dinámica, lo cual puede ser no muy seguro pues está sujeta a ataques de suplantación. Lo ideal es que se compilara junto con la aplicación que la usa para que quedara integrada a ella y no tener que hacer ligas o llamadas constantes ya que puede comprometer al sistema entero.

Borland Turbo C++ 3.0 y MS Visual C++ 6.0

Del software utilizado, este es el más importante, pues es aquí donde pudimos vaciar y expresar todas nuestras ideas y nuestro razonamiento; los compiladores junto con sus editores son la herramienta en donde escribimos el código de aplicación y de librería que nos permitieron, al final, poder generar certificados y demás archivos. Usamos un solo lenguaje de programación, C++, y dos compiladores Borland Turbo C++ y Visual C++.

El compilador C++ de Borland para DOS (Disk Operating System) es una herramienta que no utiliza muchos recursos de máquina por lo que se puede trabajar de manera más rápida y cómoda; al no tener una interfaz gráfica compleja, como las herramientas visuales, permite programar de forma fluida y sin mayor complicación. Este compilador se usó para escribir las diferentes librerías en formato estándar de C (archivos .h) que contienen las funciones, mejor dicho, las clases necesarias para la construcción de los tres tipos de archivos básicos para esta tesis, es decir, requerimientos de certificación PKCS10, certificados digitales y listas de certificados revocados, ambos basados en el estándar X.509. Cabe mencionar que la mayor parte del código escrito es compatible con otros compiladores C++.

El compilador Visual C++ de Microsoft para windows, se usó por dos razones primordiales: una, por ser una herramienta que permite fabricar interfaces gráficas para usuario basadas en ventanas, de forma rápida y sencilla con toda la funcionalidad de windows; y dos, porque la librería lib_233.dll fue compilada con este software, por lo cual al utilizar una herramienta diferente de compilación, como C++ Builder de Borland,

suponía problemas de compatibilidad. De esta forma y con la ayuda de estos dos compiladores pudimos crear el sistema que llamamos, y llamaremos de aquí en adelante, acUNAM.

SeguriTOOLS

acUNAM es una aplicación pensada y programada para emitir archivos (requerimientos de certificación, certificados digitales y listas de certificados revocados) con una estructura bien definida basada en estándares internacionales y con una sintaxis que sigue al pie de la letra la codificación DER (Distinguished Encoding Rules). Por supuesto para lograr lo anterior utilizamos la mejor técnica que existe para aprender algo, la de "prueba y error" así que para cada archivo generado debíamos verificar su correcta codificación y sintaxis; es aquí donde hace su aparición SeguriTOOLS. Ésta, es una herramienta desarrollada y donada a nosotros por SeguriDATA Privada S.A. de C.V. Entre otras cosas, SeguriTOOLS nos permitió, al ser un intérprete de DER y BER, verificar la consistencia de nuestros archivos generados contra la estructura que en verdad debería ser, es así como logramos acercarnos y finalmente llegar a la correcta codificación de nuestros archivos.

Hex Workshop

HexWorkshop es un editor de disco y de memoria desarrollado por la empresa Break Point Software, que cae en la categoría de shareware o software de libre distribución (aunque se debe pagar para una mejor versión). Con este programa tenemos la libertad de ver "físicamente" como está estructurado un archivo, o un flujo de datos, a nivel de bits y bytes y poder, además, manipularlo y modificarlo libremente con el objetivo de visualizar su comportamiento según el formato que se le de en determinado momento, o bien, según los cambios que sufre por modificaciones en el código de acUNAM. Esta herramienta es importante ya que nos permitió hacer ajustes a tiempo y poder trabajar sin los "ojos vendados". Con este software pudimos manipular bit a bit los archivos generados.

5.1.3 Documentos

Debido al alto grado de especialización en la información necesaria para construir acUNAM, se hizo necesario contar con apoyo impreso que nos ayudara a la difícil tarea de interpretar código "extraño", términos no conocidos, estructuras requeridas, etc., es por ello que usamos varios documentos de los que nombraremos los más relevantes.

A Layman's Guide to a Subset of ASN.1, BER, and DER

Este documento es una herramienta indispensable cuando se trata de leer e implementar estándares informáticos internacionales, en los que la información sustancial está expresada en notación ASN.1 (Abstract Syntax Notation One). Este es precisamente nuestro caso; tanto el estándar X.509 (publicado por ITU-T) como el PKCS10 (publicado por RSA Security), como otros tantos que hemos consultado, están expresados en (al menos las estructuras de datos) en notación ASN.1.

Layman's Guide, es una guía para principiantes (como su nombre lo dice) que además de explicar la lógica de ASN.1, explica paso a paso las codificaciones DER y BER lo cual hace posible, al final, la comprensión de documentos que tratan estándares informáticos.

Estándares internacionales

Estos documentos son, para el desarrollo de acUNAM, el equivalente a la tabla de los 10 mandamientos o a la Biblia de los religiosos, es decir, los estándares dictan las reglas y formatos que deberán seguirse para un desarrollo e implementación de un proyecto en particular. A partir de ellos generamos diagramas, pseudo-código, estructuras de datos, etc., a fin de ajustar nuestros archivos y nuestro sistema, lo más fiel posible, a lo que sugieren las recomendaciones internacionales. En el capítulo IV se trataron a fondo los documentos necesarios para el nacimiento de acUNAM, estableciéndose como herramientas indispensables en la elaboración de toda esta tesis.

5.2 Propuesta de una PKI para la UNAM

5.2.1 ¿Por qué la UNAM?

Existen varias razones para considerar que la UNAM pueda fungir como autoridad certificadora en un entorno computacional. La Universidad Nacional Autónoma de México es por mucho la institución educativa y de investigación mas grande e importante del país; además de ser nuestra alma mater, la calificamos como el laboratorio de pruebas e investigaciones más importante y de mayor seriedad en México con la infraestructura suficiente como para llevar al cabo proyectos como este, que no tienen otro propósito mas que el progreso de la institución y por ende del país.

Al decir que la UNAM tiene la infraestructura necesaria para echar a andar un proyecto grande, nos referimos a que en ella podemos encontrar, por ser una institución entregada y comprometida con la educación profesional, todos aquellos elementos humanos, didácticos y tecnológicos que en conjunto pueden lograr los objetivos de un proyecto como el nuestro o en su defecto detectar las debilidades del mismo para emprender nuevos caminos de desarrollo; lo importante es sembrar la semilla del conocimiento y del progreso y lo demás será ganancia académica.

Por principio de cuentas, debemos mencionar que la UNAM históricamente ha demostrado ser pionera en la investigación e impartición de todos los campos del conocimiento; el caso que a nosotros nos concierne, el de la tecnología informática y en específico la seguridad en cómputo, es también una rama del conocimiento en donde se ha notado esta condición de la UNAM ya que en ella existen y colaboran los mejores y más calificados profesionales en el tema, con los cuales hemos tenido la fortuna y el placer de trabajar. En la ENEP Aragón, en donde se encuentra localizado el laboratorio de seguridad, se puede contar con la experiencia de sus académicos y la disposición de sus alumnos para poder emprender la puesta en marcha de nuestro proyecto; en el CTA (Centro Tecnológico Aragón) lugar donde se ubica el laboratorio de seguridad podemos contar también con el equipo de cómputo adecuado para las pruebas, y ¿por qué no? para la implementación final de acUNAM.

Además de la ENEP Aragón, y siendo más ambiciosos, podemos pensar, como una opción más seria en DGSCA (Dirección General de Servicios de Cómputo Académico), la cual es la parte de la UNAM preocupada por la mayoría de los problemas informáticos de la institución. En esta dirección se encuentra el departamento de seguridad informática, lugar en el que se realizan investigaciones del área para publicarlas y ponerlas a disposición de la comunidad universitaria. También ha participado en proyectos externos colaborando con instituciones como el Instituto Federal Electoral (IFE). En DGSCA se encuentra personal altamente capacitado como para evaluar el funcionamiento de acUNAM y creemos que puede llegar a ser la parte de la institución más confiable en el ámbito del cómputo, convirtiéndose en la AC que la UNAM y el país necesitan. Hay que decir que DGSCA puede ser la solución en lo que a recursos se refiere, si es que el proyecto es aprobado, ya que a través de ella, la UNAM podría en un momento dado abastecernos con el equipo de cómputo e intelectual para darle empuje a este trabajo.

La idea es que acUNAM al principio funcione como una AC local, que sólo de servicio en facultades y escuelas de forma privada e interna como parte de un proyecto de investigación, y en un futuro pueda hacerlo para toda la institución, tiempo en el cual se podría pensar más alto y proponer a la UNAM como pionera y colaboradora líder de un proyecto más grande, en el que se podría pensar inclusive en la participación de otras universidades del país y de esta manera, tal vez, conformar una entidad confiable de certificación que pudiera brindar servicios a instituciones externas para por fin olvidarnos del problema que supone el uso de servicios de certificación que ofrecen empresas extranjeras como Verisign, Thawte o Entrust, es decir, poder ser autosuficientes en nuestra plataforma informática nacional. Tan sólo, como ejemplo, las empresas Telmex y Banamex, en su faceta digital, se encuentran certificadas por Verisign, (empresa norteamericana) y ninguna otra empresa que conozcamos la certifica alguna mexicana lo cual habla de nuestra dependencia tecnológica con el extranjero la cual, consideramos, debe desaparecer. Así pues por lo expuesto anteriormente elegimos a la UNAM como centro de implementación, evolución y desarrollo de acUNAM ya que su infraestructura lo permite.

5.2.2 Modelo de una PKI básica para la UNAM

La propuesta principal de esta tesis es el establecimiento de una PKI, la cual es un esquema de trabajo y de comunicación basado en tecnología de criptografía asimétrica, que tendrá como objetivo operar dentro de la UNAM como una entidad encargada de cumplir las funciones de certificación, revocación de certificados y básicamente brinde protección a documentos electrónicos a través de la utilización de criptografía de llave pública. Para obtener lo anterior se debe analizar y diseñar un modelo de PKI básico que pueda ser adaptable a las necesidades de la universidad (y/o de cualquier otra entidad que adopte el sistema) independientemente de que su implementación se efectúe a nivel de escuela o facultad, o bien que se haga a nivel de institución, lo cual sería cumplir con ganancia extra el objetivo de este trabajo.

Como ya se mencionó, una PKI está compuesta de varios elementos (todos necesarios) de los cuales, la autoridad certificadora, la autoridad registradora, los certificados digitales y por supuesto los usuarios finales son las piezas básicas e imprescindibles con las que se debe contar para el establecimiento de una PKI. A continuación, de forma gráfica, se presenta un modelo sencillo de una PKI que sirve perfectamente para nuestro propósito. *Fig 5.1*

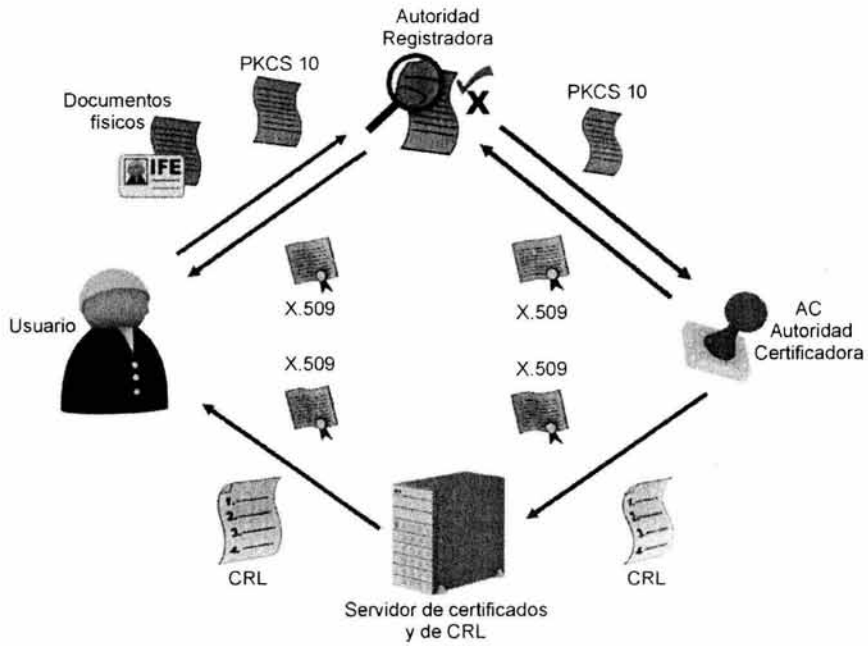


Fig. 5.1 Modelo básico de una PKI para la UNAM

A continuación se explicarán de forma sencilla los elementos de este modelo.

5.2.2.1 Usuarios Finales

Obviamente todo esfuerzo tecnológico está hecho pensando en las necesidades de los usuarios de los sistemas, en este caso la necesidad que se presenta para los usuarios finales es la de tener la certeza de que la información que intercambie con otras entidades estará protegida, y que además estas entidades, como él mismo, se puedan autenticar entre sí para establecer la confianza durante una transacción. Gracias a los elementos de una PKI, la información se puede proteger mediante las técnicas de cifrado de datos además de poder agregar una firma digital, la cual nos servirá como elemento autenticador y verificador de integridad. Todo esto es precisamente el conjunto de actividades que un usuario debe manipular y comprender para poner a funcionar un esquema PKI. Un usuario, dentro de sus obligaciones, debe entender la importancia y el papel que juegan tanto una firma como un certificado digital, dentro de un sistema de llave pública, para poder establecer así patrones seguros de comunicación, ya que de no hacerlo, perdería de vista el significado del sistema y seguramente éste fallaría. El usuario final, será el único responsable de cuidar el buen y exclusivo uso de su llave privada, además de solicitar la certificación ante una AC (como acUNAM) de su llave pública, esto quiere decir que no se debe atener al funcionamiento del sistema mismo, sino que debe integrarse e involucrarse con él, convirtiéndose de esta forma en un elemento de suma importancia dentro de la PKI y dejar de ser solamente un usuario.

5.2.2.2 Aplicación de usuario

Los usuarios, a quienes finalmente va dirigido este proyecto, deben contar con un programa informático que les permita, en la comodidad de su hogar y en la privacía de su computadora, realizar las actividades necesarias que los hagan sentir seguros al manipular y administrar documentos electrónicos.

Esta aplicación de usuario debe ofrecer por lo menos los siguientes servicios:

- Generación de un par de llaves: Para efectos de firma de documentos y para poder solicitar un certificado digital, es absolutamente imprescindible que el usuario cuente con sus llaves asimétricas, de las cuales la privada deberá proteger y guardar celosamente; siendo ambiciosos, en un futuro se podría guardar esta llave en una tarjeta inteligente, por ejemplo.
- Generación de un requerimiento de certificación: Si el usuario desea certificarse ante una AC, en este caso ante acUNAM, debe generar una solicitud, la cual irá firmada por sí mismo declarando que los datos allí expresados son ciertos y que se pueden comprobar en cualquier momento; además este requerimiento o solicitud deberá ser presentado ante una AR junto con los documentos que sustenten la información declarada en el mismo. Dada la naturaleza de este trámite, no es posible aún llevar al cabo esta actividad de forma automática o como se estila decir "en línea".
- Firmar y verificar firmas: Esta aplicación de usuario también deberá contar con las funciones de firma y de verificación ya que de esta manera y con la ayuda de un certificado raíz de acUNAM (certificado autofirmado por la AC que contiene su llave pública) el usuario podrá corroborar la autenticidad de un documento recibido a partir de la firma del documento y de la llave pública del supuesto remitente contenida en un certificado. También con esta funcionalidad podrá firmar documentos electrónicos que serán susceptibles de ser verificados por los destinatarios, quienes deberán contar también con una aplicación de usuario.
- Cifrado de datos: La actividad primordial para garantizar la confidencialidad de la información es precisamente el cifrado de datos, por lo tanto la aplicación de usuario debe contar con la funcionalidad de cifrado y descifrado de información, con la cual podrá generar paquetes de información en texto ilegible y/o recuperar datos en texto claro a partir de información cifrada.
- Esta aplicación será de libre distribución entre los que deseen recibir los servicios de acUNAM.

5.2.2.3 Autoridad Registradora

Ya sea que la implementación se haga en el departamento de cómputo de una sola escuela o facultad, o que se haga para toda la institución en DGSCA, es necesario contar con una entidad física (estática ó móvil) que se encargue de recibir o recuperar solicitudes de certificación (requerimiento digital y documentación en papel) para poder corroborar los datos de los usuarios que desean ser certificados y de esta manera aprobar la expedición de un certificado digital. Si la AR está 100% convencida de que la verificación de datos ha sido un éxito, entonces entrega a la AC los requerimientos para su procesamiento y posterior certificación.

Esta AR puede ser una oficina ubicada en las escuelas o facultades para que los usuarios interesados se dirijan a ella y solicitar su certificado. Si el usuario es de alto rango, es decir, directivos, académicos, investigadores, etc., bien se le podría encargar a un "mensajero" certificado por la AC que recopilara las solicitudes "a domicilio".

5.2.2.4 Autoridad Certificadora

Como ya se mencionó acUNAM es la aplicación que emite certificados digitales y listas de certificados revocados. Esta aplicación deberá existir en una máquina que de preferencia esté desconectada de cualquier red y que además cuente con protección física tal como un perímetro, custodios y vigilancia en general. Los custodios deberán estar presentes en el momento de realizar operaciones de certificación o de revocación de certificados ya que de esta manera se garantiza la no corrupción en el funcionamiento de la AC.

La AC, acUNAM, verificará los PKCS10 que la AR haya aprobado y si es el caso expedirá el correspondiente certificado digital firmado por ella misma, sin embargo hay que resaltar que la entrega del certificado al usuario no se hará desde la misma máquina en donde reside la AC. acUNAM deberá ser capaz de administrar todos aquellos certificados digitales que haya emitido o por lo menos llevar un control de ellos, además en un futuro podrá brindar el servicio de custodia de llaves privadas de los usuarios. acUNAM tentativamente podría instalarse de forma aislada en el departamento de cómputo de la escuela o facultad que requiera los servicios de certificación; la máquina, sobra decirlo, deberá también estar protegida de forma lógica, es decir evitar cualquier tipo de virus o programa que pudiera comprometer el funcionamiento de la AC, por lo cual esta máquina será usada con fines exclusivos de certificación.

acUNAM deberá contar con su propio certificado digital (certificado raíz), el cual será de carácter público para que la infraestructura funcione adecuadamente. Adicionalmente, a través de alguno de los administradores, acUNAM podrá recibir solicitudes o peticiones de revocación de certificados y también podrá decidir, si es que algún usuario no está actuando bajo las políticas dictadas, la revocación de algún otro. Las CRLs que se generen se publicarán con la periodicidad necesaria para que los usuarios puedan estar informados del estado de sus propios certificados y de los de sus contactos.

5.2.2.5 Entrega de certificados

La entrega de certificados digitales X.509 expedidos por acUNAM, se puede hacer de dos formas: física y electrónica.

- Física: de forma física, el usuario que solicitó su certificación podría dirigirse a la oficina de la AR y recoger allí su certificado grabado en algún medio de almacenamiento digital, o bien algún mensajero certificado podría hacer la entrega en propia mano del usuario. También es factible (aunque ya no muy usado ni muy práctico) el servicio de correo convencional.
- Electrónica: de manera digital, un certificado puede ser entregado a un usuario a través de un sitio en internet que ofrezca el servicio de conexión segura. Después de que acUNAM ha expedido un certificado, éste lo puede poner a disposición de los usuarios en un servidor de certificados con el cual se puede hacer contacto vía internet. Para que esto funcione el usuario deberá proporcionar ciertos datos que lo

autentiquen en el sitio como el propietario de un certificado que reside allí; el sitio entonces establecerá una conexión segura basada en el protocolo SSL (Secure Socket Layer) entre el servidor y la máquina del usuario permitiéndose así el seguro tránsito de información, en este caso del certificado. Esta conexión se hará también cuando un usuario requiera el certificado raíz de la AC o la CRL actualizada.

- SSL (Secure Socket Layer): El protocolo de intercambio seguro de información SSL fue desarrollado por Netscape Communications Corporation para ofrecer seguridad y privacidad en las comunicaciones electrónicas basadas en internet; este protocolo soporta y tiene la funcionalidad de autenticación de servidor y autenticación de usuario o cliente. SSL es una aplicación independiente que permite el uso de protocolos como HTTP, FTP y Telnet y pasa como un programa transparente al funcionamiento de éstos. Este protocolo SSL, está facultado para intercambiar llaves de sesión y de cifrado y así poder autenticar al servidor antes de que la aplicación de alto nivel (como acUNAM) intercambie información con un cliente. Finalmente podemos resumir que Secure Socket Layer es un protocolo que establece un canal seguro de comunicación usando cifrado de datos de forma simétrica, autenticación y llaves de sesión. Por todo esto consideramos que el uso de SSL en la tarea de entrega de certificados, es una acertada actividad para garantizarle al usuario la integridad de sus datos.

Así pues quedan explicados los elementos del diagrama básico de una PKI mostrado anteriormente. La parte que hemos desarrollado y en la que estamos situados, como objetivo de esta tesis, es la aplicación acUNAM (*ver fig. 5.2*), la cual realiza la función de autoridad de certificación ya que es software diseñado para emitir certificados digitales y listas de certificados revocados bajo el estándar X.509 además de que se ha implementado también la aplicación de usuario con la que se pueden generar un par de llaves asimétricas ECC y construir requerimientos de certificación en formato PKCS10.

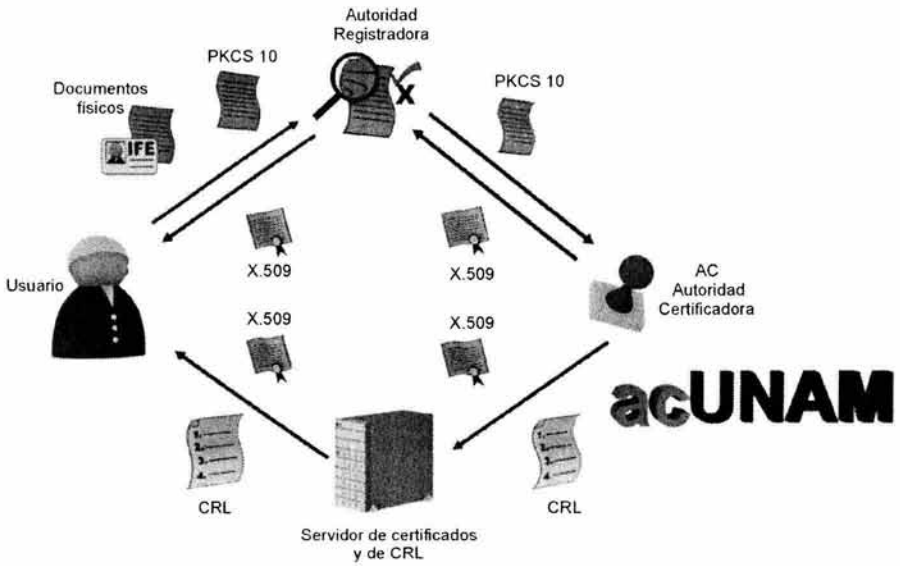


Fig. 5.2 Ubicación de acUNAM dentro de la PKI de la UNAM

5.2.3 Tentativa de funcionamiento

La aplicación acUNAM por sí sola, no es más que un programa que expide documentos digitales lo cual no asombra a nadie, lo importante dentro de todo este proyecto es el concepto que hay detrás; lo notable de este trabajo es el translado de la actividad de certificación habitual a la certificación digital. Lo primero que debemos tener en cuenta, tanto nosotros como desarrolladores del proyecto como los posibles usuarios, es que es necesario cambiar la forma de pensar con respecto a los documentos electrónicos, y confiar en que la tecnología que se está manejando, de verdad nos puede garantizar seguridad en entornos digitales. A lo que deseamos llegar es a que las personas involucradas en este entorno puedan confiar en una AC (como acUNAM) tal como lo hacen con instituciones emisoras de credenciales físicas como es el caso de la Secretaría de Relaciones Exteriores o el Instituto Federal Electoral; si se logra que las personas confíen en acUNAM y la vean como una institución más de certificación, se habrá dado el primer y más importante paso en el camino de este desarrollo. Lo anterior, la confianza en la AC, se puede alcanzar con dos factores importantes: 1) La UNAM considerándola de manera interna es en sí, una institución en la que todos confiamos gracias a su intachable reputación y seriedad que ha demostrado durante décadas, por lo cual si la universidad certifica a acUNAM como entidad de confianza pues se habrá logrado andar la mitad del camino, sin embargo fuera de la institución, no todos perciben de igual manera a la universidad, así que para establecer la confianza a nivel exterior, debemos tomar en cuenta el siguiente punto, 2) acUNAM como entidad de certificación debe contar con la credibilidad suficiente para ganar la confianza de los usuarios, así que es factible que, con las nuevas leyes que se están

cocinando respecto a información electrónica, se certifique vía un notario público facultado para brindarle un carácter más legal que didáctico, estableciéndola como una entidad confiable.

5.3 Procedimiento de desarrollo de la aplicación acUNAM

acUNAM está diseñada de forma didáctica por lo cual se integraron en una misma consola a los cuatro módulos que realizan las operaciones más importantes de una AC *ver fig 5.3.*

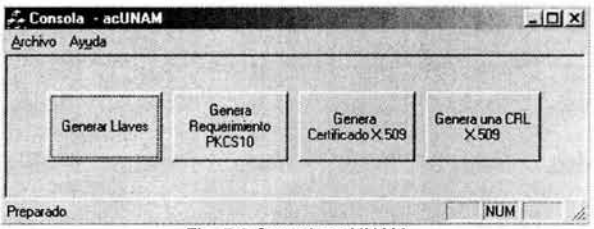


Fig. 5.3 Consola acUNAM

Obviamente en la práctica, que no en la enseñanza, estos módulos deberían estar separados formando, quizás, una aplicación cliente servidor, pero al tener fines de aprendizaje, todos ellos conviven en un mismo espacio y tiempo dentro de la computadora. Los cuatro módulos de acUNAM son:

- Generar un par de llaves asimétricas ECC.
- Generar un requerimiento de certificación PKCS10.
- Expedir un certificado digital X.509.
- Emitir listas de certificados revocados X.509.

Antes de explicar cada módulo se expondrá de forma gráfica el flujo que se produce desde que un usuario genera un par de llaves hasta que la AC publica una CRL (*ver fig. 5.4*); adicionalmente se hará un listado y una breve explicación de las funciones contenidas en las clases, escritas en C++, que dan pie a la construcción de acUNAM. Estas clases están almacenadas en archivos de cabecera o librerías estándar de C.

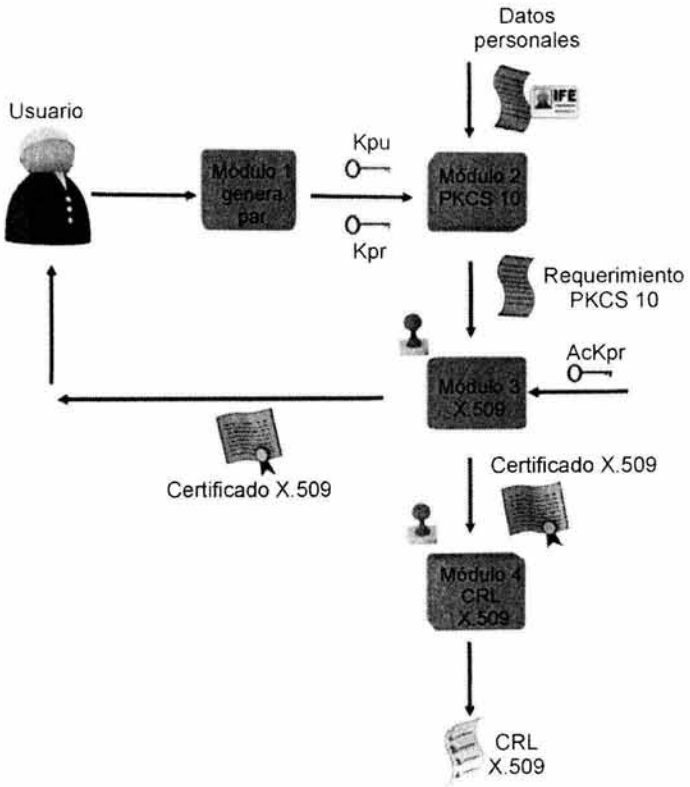


Fig. 5.4 Diagrama de flujo de acUNAM

5.3.1 Clases y librerías desarrolladas para acUNAM

Clase Pkcs10 contenida en la librería crtreq01.h

codificaRDN
Sintaxis: void codificaRDN(int banderaOID, char *valor, char *Rdn);
Descripción: Codifica una estructura individual de las que conforman un Relative Distinguished Name, es decir un AttributeTypeAndValue.

Tipo	Parámetro	Descripción
Int	banderaOID	Número que indica el identificador de objeto a utilizar.
char *	Valor	Arreglo en el que está contenido uno de los datos del usuario, como su nombre, dirección, etc.
char *	Rdn	Arreglo en el que se almacena una estructura AttributeTypeAndValue.

codificaSubject
Sintaxis: void codificaSubject(SubjectStruct si, char *subjectfinal);
Descripción: Codifica una estructura Subject a partir de una secuencia de Relative Distinguished Names.

Tipo	Parámetro	Descripción
SubjectStruct	Si	Estructura en la que se encuentran almacenados todos los datos del usuario como nombre y dirección.
char *	Subjectfinal	Arreglo en el que se almacenará la estructura Subject resultante.

hextodec
Sintaxis: int hextodec(char mas, char menos);
Descripción: Regresa el número decimal equivalente a uno hexadecimal de un byte.

Tipo	Parámetro	Descripción
Char	mas	Dígito más significativo del número hexadecimal.
Char	menos	Dígito menos significativo del número hexadecimal.

longhex
Sintaxis: void longhex(int longi, char *longarray, int *numbytes);
Descripción: Dado un número decimal entero, la función obtiene el hexadecimal equivalente y su tamaño en bytes.

Tipo	Parámetro	Descripción
Int	longi	Número decimal entero del cual se desea su equivalente hexadecimal.
char *	longarray	Arreglo en el cual se almacenará el número hexadecimal.
int *	numbytes	Variable entera en donde se almacenará el tamaño en bytes del número encontrado.

codificaCRInfo
Sintaxis: int codificaCRInfo(char *sujeto, char *llavepub, char *crifinal);
Descripción: Codifica una estructura Certification Request Info (información a ser firmada) a partir de una estructura Subject.

Tipo	Parámetro	Descripción
char *	sujeto	Arreglo que contiene una estructura Subject.
char *	llavepub	Arreglo que contiene la llave pública del usuario.
char *	crifinal	Arreglo en el que se almacenará la estructura Certification Request Info.

Clase X509 contenida en la librería certx509.h

codificaValidity
Sintaxis: void codificaValidity(char *notbfaf, int flagHv);
Descripción: Codifica las fechas de validez en formato UTC formando la estructura Validity de ASN.1.

Tipo	Parámetro	Descripción
char *	notbfaf	Arreglo en donde se almacenará la estructura Validity.
Int	flagHv	Variable que indica si es horario normal u horario de verano. Si es 0 es horario normal; mayor a cero es horario de verano.

recuperaFirma
Sintaxis: void recuperaFirma(char *requerimiento, char *firmarec);
Descripción: De un requerimiento PKCS10 (archivo.req) se extrae la firma y se almacena en un arreglo.

Tipo	Parámetro	Descripción
char *	requerimiento	Arreglo con el nombre y ruta del archivo PKCS10.
char *	firmarec	Arreglo en donde se almacenará la firma recuperada.

recuperaLlavepu
Sintaxis: void recuperaLlavepu(char *requerimiento, char *pkinfo, char *sololLave);
Descripción: Se extrae de un archivo PKCS10 (.req) la estructura SubjectPublicKey y la llave del usuario.

Tipo	Parámetro	Descripción
char *	requerimiento	Arreglo con el nombre y ruta del archivo PKCS10.
char *	pkinfo	Arreglo en donde se almacenará la estructura SubjectPublicKey.
char *	sololLave	Arreglo en donde se almacenará la llave del usuario.

recuperaSubject
Sintaxis: void recuperaSubject(char *requerimiento, char *subject);
Descripción: A partir de un archivo PKCS10, se recupera la estructura Subject que contiene la información personal del usuario.

Tipo	Parámetro	Descripción
char *	requerimiento	Arreglo con el nombre y ruta del archivo PKCS10.
char *	subject	Arreglo en el que se almacenará la estructura Subject.

recuperaCRI
Sintaxis: int recuperaCRI(char *reqerimiento, char *cri);
Descripción: Recupera a partir de un archivo PKCS10 la estructura CertificationRequestInfo, la cual fue firmada al generar el requerimiento; además regresa el tamaño en bytes de dicha estructura.

Tipo	Parámetro	Descripción
char *	requerimiento	Arreglo con el nombre y ruta del archivo PKCS10.
char *	cri	Arreglo en donde se almacenará la estructura recuperada.

codificaTBSCertificate
Sintaxis: int codificaTBSCertificate(char *requerimiento, char *tbsC, int flag);
Descripción: Codifica la parte del certificado que se firma y además regresa el tamaño en bytes de la misma.

Tipo	Parámetro	Descripción
char *	requerimiento	Arreglo con el nombre y ruta del archivo PKCS10.
char *	tbsC	Arreglo en donde se almacenará la estructura TBSCertificate
Int	flag	Bandera que indica si será un certificado de usuario o de AC.

Nota: Esta función necesita que estén presentes los archivos acserie.ac y nameac.ac en el directorio de trabajo.

codificaCertificado
Sintaxis: int codificaCertificado(char *tbsfinal, int sztbsfinal, char *firmaf, char *certfinal);
Descripción: Codifica un certificado digital X.509 a partir de una estructura TBSCertificate y una firma; regresa el tamaño del certificado en bytes.

Tipo	Parámetro	Descripción
char *	tbsfinal	Arreglo que debe contener una estructura TBSCertificate.
Int	sztbsfinal	Tamaño en bytes de la estructura TBSCertificate.
char *	firmaf	Arreglo que contiene una firma digital.
char *	certfinal	Arreglo en donde se almacenará una estructura Certificate.

Clase crlX509 contenida en la librería crlx509.h

codificaCertificadoRevocado
Sintaxis: void codificaCertificadoRevocado(char *acunamcrl, char *certificado, char *crlfinal);
Descripción: Codifica la estructura CertList a partir del número de serie extraído de un certificado digital X.509 y si es el caso de una CRL ya existente.

Tipo	Parámetro	Descripción
char *	acunamcrl	Arreglo con el nombre y ruta de un archivo CRL.
char *	certificado	Arreglo con el nombre y ruta del archivo X.509.
char *	crlfinal	Arreglo en donde se almacenará la estructura CertList.

codificaTBSCertList
Sintaxis: int codificaTBSCertList(char *revokedCrts, char *tbsList);
Descripción: Codifica la estructura TbsCertList, a partir de una estructura CertList y regresa su tamaño en bytes..

Tipo	Parámetro	Descripción
char *	revokedCrts	Arreglo que contiene una estructura CertList.
char *	tbsList	Arreglo en donde se almacenará una estructura TbsCertList.

codificaCRL
Sintaxis: int codificaCRL(char *tbsCrl, int szTbs, char *firmaCrl, char *crlfinal);
Descripción: Codifica una estructura Certificate Revoked List a partir de una estructura TbsCertList y una firma digital; además regresa el tamaño en bytes de la lista.

Tipo	Parámetro	Descripción
char *	tbsCrl	Arreglo que contiene una estructura TbsCertList.
Int	szTbs	Tamaño de la estructura TbsCertList.
char *	firmaCrl	Arreglo que contiene una firma.
char *	crlfinal	Arreglo en donde se almacenará la lista final de certificados revocados.

5.3.2 Módulos

A continuación se expondrán los algoritmos usados que corresponden a cada uno de los cuatro módulos de acUNAM que se encuentran juntos en una consola principal para efectos de aprendizaje.

Módulo 1 Generación de un par de llaves ECC

Tanto el usuario final como la propia autoridad certificadora necesitan un par de llaves asimétricas que les permitan desarrollar las funciones de cifrado, descifrado y firma, lo cual da sustento precisamente a la PKI. Para la generación de este par de llaves se usa un algoritmo ECC y se obtienen llaves de 480 bits en el caso de la llave pública y de 256 bits para el caso de la privada. *Fig. 5.5*

Algoritmo

1. Se le pide al usuario un nombre para sus llaves; por defecto tendrán las extensiones .pub y .pri.cif.
2. Se le solicita una contraseña con la que se protegerá con DES su llave privada.
3. Se usa la función generarllaves(); de la librería lib_233.dll y se le pasan como parámetros lo solicitado anteriormente.
4. Las llaves, pública y privada, han sido creadas.

Ver diagrama del módulo 1 en el apéndice B.

Fig. 5.5 Aplicación para generar un par de llaves ECC

Módulo 2 Generar un requerimiento de certificación PKCS10

El requerimiento de certificación PKCS10 es la solicitud digital que el usuario debe presentar ante una autoridad registradora junto con el sustento documental que de confiabilidad y autenticidad a los datos declarados en el mismo.

Algoritmo

1. El usuario debe llenar un formulario con sus datos además de proporcionar sus llaves pública y privada (*ver fig. 5.6*)
2. Se utilizará un objeto de la clase PKCS10 para acceder a funciones y estructuras.
3. Cada uno de los valores recuperados en el formulario (nombre, dirección, etc.) se almacenará en una estructura Info del tipo SubjectStruct que servirá como parámetro de una función de librería.
4. La estructura Info con los datos personales del usuario se pasará como parámetro de la función codificaSubject();, la cual con la ayuda de la función codificaRDN(); creará un arreglo final que contendrá una estructura Subject.
5. La estructura Subject generada en el paso anterior, junto con la llave pública del usuario servirán como parámetros de la función codificaCRInfo();, la cual producirá una estructura CertificationRequestInfo.
6. La estructura CRI se almacenará en un archivo temporal (crtrqinf.tmp).
7. Con la función firma(); de la librería lib_233.dll y como parámetros, el archivo crtrqinf.tmp, la llave privada del usuario, y la contraseña que protege a dicha llave se obtendrá la firma del documento que tiene una longitud de 58 bytes.
8. Finalmente se integran en un mismo arreglo a la estructura CRI, a la firma digital obtenida y a los datos del algoritmo de firma (en este caso ECC con MD5) y se escribe un archivo que será ya un requerimiento de certificación PKCS10 (archivo.req).

Ver diagrama del módulo 2 en el apéndice B.

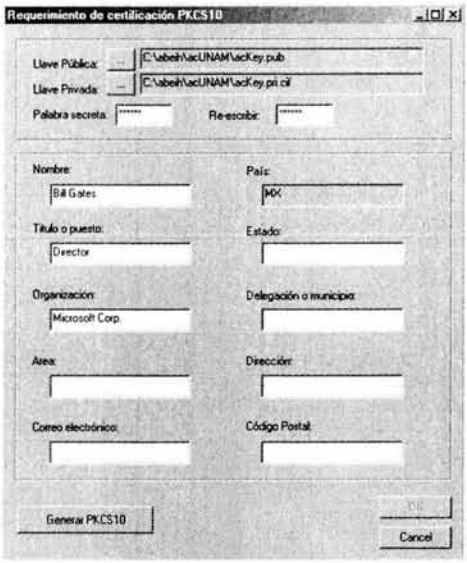


Fig. 5.6 Aplicación para generar un requerimiento de certificación PKCS 10

Código ASN.1 del requerimiento PKCS10

```
BEGIN
CertificationRequest ::= SEQUENCE {
   certificationRequestInfo CertificationRequestInfo,
   signatureAlgorithm       AlgorithmIdentifier{{ SignatureAlgorithms }},
   signature                BIT STRING
}

CertificationRequestInfo ::= SEQUENCE {
   version      INTEGER { v1(0) } (v1,...),
   subject      Name,
   subjectPKInfo SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
   attributes   [0] Attributes{{ CRIAttributes }}
}

SubjectPublicKeyInfo {ALGORITHM: IOSet} ::= SEQUENCE {
   algorithm            AlgorithmIdentifier {{IOSet}},
   subjectPublicKey     BIT STRING
}

PKInfoAlgorithms ALGORITHM ::= {
   ... -- add any locally defined algorithms here -- }

Attributes { ATTRIBUTE:IOSet } ::= SET OF Attribute{{ IOSet }}

CRIAttributes  ATTRIBUTE  ::= {
   ... -- add any locally defined attributes here -- }
```

```
Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
  type   ATTRIBUTE.&id({IOSet}),
  values         SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type})
}

AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {
  algorithm      ALGORITHM.&id({IOSet}),
  parameters     ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL
}

SignatureAlgorithms ALGORITHM ::= {
  ... -- add any locally defined algorithms here -- }
END
```

Módulo 3 Expedir un certificado digital X.509

El certificado digital X.509 es el documento digital que habilita la posibilidad de autenticar a cierta entidad en un contexto electrónico; es la credencial avalada por una autoridad certificadora (en este caso acUNAM) que permite identificarse de manera positiva y con un alto grado de confiabilidad. *Fig. 5.7*

Algoritmo

1. El administrador de la AC deberá proporcionar a la consola de la aplicación un requerimiento de certificación PKCS10 (archivo.req), una dirección en donde se pueda almacenar el nuevo certificado y la llave privada de la AC (ackey.pri.cif).
2. Se usará un objeto de la clase X509 para acceder a las funciones.
3. Del requerimiento proporcionado se extraerá lo siguiente: con la función recuperaFirma(); se extraerá la firma generada por el solicitante; con la función recuperaLlavepu(); se extraerá la llave pública y la estructura que la contiene en arreglos separados; con recuperaCRI(); se extrae la estructura CertificationRequestInfo, de la cual se obtuvo una firma. Cada una de las estructuras recuperadas se almacenarán en archivos temporales llamados: firmarec.fir, pk.pub y crirec.req.
4. Una vez que se tienen la llave pública del solicitante, la firma y la estructura CRI, se procede a verificar la autenticidad del requerimiento con la función verifica(); de la librería lib_233.dll.
5. Si la función verifica(); regresa "ok" podemos continuar, de lo contrario se prohibe la generación del certificado.
6. Con la función codificaTBSCertificate(); se genera una estructura del tipo tbsCertificate a partir de un requerimiento, la cual se almacenará en un archivo temporal tbsfile.tmp. Este archivo contiene la información a ser firmada del certificado en construcción.
7. Con la ayuda de la función firma(); se obtiene, a partir del archivo tbsfile.tmp y de la llave privada de la AC, una firma digital que se almacenará en el archivo tbssign.fir.
8. Finalmente la función codificaCertificado(); con sus parámetros tbssign.fir y una estructura tbsCertificate genera el certificado digital X.509 para el usuario final (archivo.crt).

Ver diagrama del módulo 3 en el apéndice B

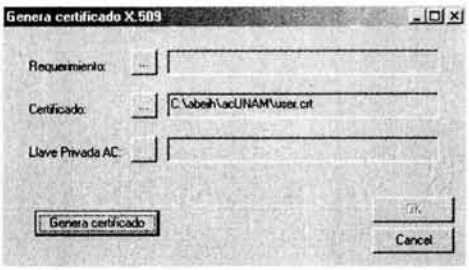


Fig. 5.7 Aplicación para generar un certificado X.509

Código ASN.1 del certificado X.509

```
BEGIN
Certificate ::= SEQUENCE {
tbsCertificate        TBSCertificate,
signatureAlgorithm    AlgorithmIdentifier,
signatureValue        BIT STRING }

TBSCertificate ::= SEQUENCE {
   version              [0]  EXPLICIT Version DEFAULT v1,
   serialNumber         CertificateSerialNumber,
   signature            AlgorithmIdentifier,
   issuer               Name,
   validity             Validity,
   subject              Name,
   subjectPublicKeyInfo   SubjectPublicKeyInfo,
   issuerUniqueID       [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version must be v2 or v3

 subjectUniqueID        [2] IMPLICIT UniqueIdentifier OPTIONAL,
                        -- If present, version must be v2 or v3
 extensions                    [3] EXPLICIT Extensions OPTIONAL
                        -- If present, version must be v3
}
Name : : = CHOICE {
        RDNSequence     }

RDNSequence : : = SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName : : =
        SET OF AttributeTypeAndValue

AttributeTypeAndValue  : : = SEQUENCE {
        type                    AttributeType,
        value                   AttributeValue    }

AttributeType : : = OBJECT IDENTIFIER
AttributeValue : : = ANY DEFINED BY AttributeType

Validity : : =  SEQUENCE  {
notBefore       Time,
notAfter        Time     }
```

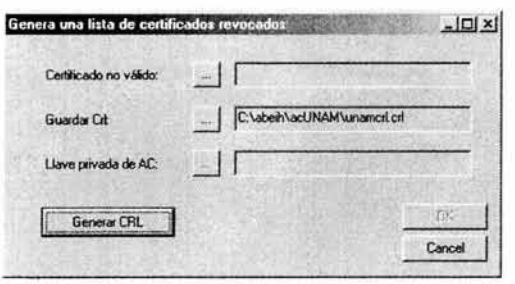
```
Time : : = CHOICE       {
utcTime                 UTCTime,
generalTime             GeneralizedTime }

SubjectPublicKeyInfo : : = SEQUENCE {
algorithm               AlgortihmIdentifier,
subjectPublicKey        BIT STRING          }
END
```

Módulo 4 Emitir listas de certificados revocados X.509

Una lista de certificados revocados o CRL es un archivo firmado por la AC en donde se listan los números de serie y las fechas de revocación de aquellos certificados digitales que de una u otra forma han sido revocados. Esta CRL se deberá publicar periódicamente según las políticas establecidas para el funcionamiento de la PKI.

Algoritmo

1. El administrador debe proporcionar a la consola de la AC un certificado digital (el cual será revocado), la llave privada de la AC y una ruta de destino para almacenar la CRL *ver fig. 5.8.*
2. Se usará un objeto de la clase crlX509.
3. Usando la función codificaCertificadoRevocado(); se genera la lista de certificados, a partir de su número de serie, que serán considerados como revocados a partir de la fecha de generación o actualización de la CRL. Se obtiene un arreglo cerrev que contiene la estructura revokedCertificates.
4. Con la función codificaTBSCertificate(); y como parámetro la estructura revokedCertificates, se genera la estructura a ser firmada tbslista, la cual se almacenará en un archivo temporal tbs.tmp.
5. A partir de tbs.tmp y la llave privada de la AC se obtendrá una firma digital que se almacenará en el archivo tbscrlsign.fir, esto con la función firma(); de la librería lib_233.dll.
6. Con la función codificaCRL(); y con tbs.tmp y tbscrlsign.fir como parámetros generamos finalmente la estructura que nos permitirá escribir la CRL X.509 (archivo.crl). *Ver diagrama del módulo 4 en el apéndice B.*



5.8 Aplicación que genera una CRL

Código ASN.1 de la CRL X.509

```
BEGIN
CertificateList          ::= SEQUENSE{
        tbsCertList              TBSCertList,
        signatureAlgorithm       AlgorithmIdentifier,
        signatureValue   BIT STRING     }

TBSCertList     ::=      SEQUENCE     {
        Version          Version OPTIONAL,
                                 -- if present, shall be v2
        signature                AlgorithmIdentifier,
        issuer                   Name,
        thisUpdate               Time,
        nextUpdate               Time OPTIONAL,
        revokedCertificates      SEQUENCE Of SEQUENCE      {
                usercertificate          CertificateSerialNumber,
                revocationDate   Time,
                crlEntryExtensions       Extensions OPTIONAL
                                         -- if present, shall be v2
                                         } OPTIONAL,
                crlExtensions            [0]  EXPLICIT Extensions OPTIONAL
                                         -- if present, shall be v2}
END
```

Todas las estructuras mencionadas en estos algoritmos (revokedCertificates, CertificationRequestInfo, etc.) fueron descritas anteriormente cuando se explicó la codificación ASN.1 de los tres archivos generados, por lo cual no es necesario explicarlas nuevamente.

Durante todo el proceso, desde que se genera una pareja de llaves hasta que se expide un CRL se generan varios archivos de apoyo, que se listarán a continuación: crtrqinf.tmp, crtrqinf.fir, firmarec.fir, pk.pub, crirec.req, tbsfile.tmp, tbssign.fir, tbs.tmp y tbscrlsign.fir.

Este ha sido pues el proceso de programación de cada módulo, en el que se ve la interacción de las librerías que programamos y la aplicación de las reglas de codificación establecidas por los estándares que hemos utilizado.

5.4 Comentario con respecto a la criptografía utilizada

El utilizar criptografía ECC "hecha en casa" (hay que recordar que el motor criptográfico lib_233.dll fue desarrollado en la UNAM) nos provocó la necesidad de hacer ciertos ajustes a la aplicación, sin que se presentara problema alguno, simplemente hay que tener en cuenta este detalle para futuras modificaciones al proyecto. Lib_233.dll es una librería creada en su totalidad por investigadores de la UNAM, la cual para generar una firma digital utiliza de forma combinada dos técnicas bien conocidas aunque no registradas ni estandarizadas en conjunto (pues si lo están por separado) ante algún organismo internacional como ITU o ISO. Nos referimos a que para obtener una firma a partir de un documento se obtiene un valor hash MD5 que después se cifrará con ECC, estableciéndose así la dupla MD5/ECC misma que aún no se encuentra reconocida internacionalmente como estándar de firma. El único esquema de firma digital ya registrado que utiliza tecnología de curvas elípticas es el algoritmo llamado ECDSA el cual utiliza la dupla SHA1/ECC para generar la firma. Por lo tanto la aclaración que deseamos hacer con

respecto a esta situación es que el identificador de algoritmo de firma que se encuentra plasmado en los certificados emitidos por acUNAM es el de ECDSA ya que no existe uno registrado que identifique al algoritmo que utilizamos. Esto no supone un problema de desempeño ni mucho menos, pero sí es una situación que se debe tomar muy en cuenta cuando el proyecto tenga la posibilidad de expandirse y de seguir evolucionando.

5.5 Requerimientos para la implementación

Tal como si se tratara de una receta de cocina, podemos decir que los elementos necesarios para que la PKI se establezca son: elementos humanos capacitados en seguridad informática, desarrollo de software, redes, etc.; equipo de cómputo para instalar la aplicación acUNAM y para almacenar el sitio web de distribución de certificados; cableado e insumos generales; un lugar físico seguro dentro de las instalaciones de la UNAM para establecer el área operativa, y otro para la instalación de la oficina de la AR.; finalmente lo más importante, la aprobación y certificación de la UNAM para que acUNAM pueda entrar en operaciones.

De esta forma queda plasmada la idea de funcionamiento de acUNAM como una pequeña versión de una PKI robusta. Obviamente todos los elementos citados anteriormente deben trabajar de manera coordinada y bajo ciertas políticas dictadas según lo requerido para su óptimo desempeño.

5.6 Jerarquía de Certificación de acUNAM

¿Quién certifica a una AC?. Una PKI permite que una AC pueda certificar otra. La jerarquía de certificación funciona así: cada certificado emitido por una AC puede estar firmado por una AC de mayor grado, hasta llegar a la AC superior, donde se encuentra la llamada raíz de la certificación, que es una entidad que genera y firma sus propios certificados; debe ser muy confiable y ampliamente disponible, un ejemplo de una AC que se encontraría en el nivel más alto sería el caso de Verisign. Además por motivos prácticos es obvio que no puede existir una sola AC que actué como autoridad en un entorno global.

Los costos de distribución de los certificados de entidades raíz son los más elevados, ya que si se cambiara la raíz se tendría que crear de nuevo toda la PKI. Si un certificado raíz cambia, debe revocar el antiguo certificado y agregar el nuevo para todos los clientes de la organización.

En el proyecto de la acUNAM, la ruta de certificación sería la siguiente:

La UNAM por ser una institución pública reconocida en nuestro país, no tendría que solicitar un certificado para que la avalaran como AC interna, lo que necesitaría es registrarse ante un notario público como una AC prestadora de servicios de autenticación, por lo tanto estaría en el nivel más alto de la jerarquía, nuestra raíz de certificación. Misma que certificaría y verificaría directamente la validez de los certificados de los usuarios clientes.

5.7 Medidas de Seguridad para una AC

- Nunca se debe conectar a la red el equipo en donde reside la Autoridad de Certificación, es decir debe estar aislado física y virtualmente.
- No automatizar el proceso de emisión de certificados, para lo que es recomendable la presencia física del usuario ante un AR o en su defecto presentar una adecuada documentación.
- Restringir el acceso de los usuarios locales a la AC.
- Restringir el acceso físico a la AC.
- Proteger el quipo de interferencias de radiofrecuencia.
- No guardar la llave privada de la AC sin cifrar (no hay que olvidar que es la que se utiliza para firmar, y si esta se compromete se viene abajo la PKI)
- Procurar que los usuarios de la AC utilicen llaves de una longitud adecuada según el criptosistema en el que se encuentren.

5.8 Expectativas

Como en todo proyecto, se espera que en el camino de desarrollo e implementación de acUNAM existan evaluaciones de funcionamiento, crisis en el desempeño y finalmente todos aquellos ajustes pertinentes para que quede y opere bien. Por todo esto esperamos que acUNAM pueda servir de base para proyectos futuros que puedan establecer un entorno seguro y confiable en lo que a documentos y transacciones digitales se refiere. Como profesionales emprendedores y optimistas que somos deseamos que acUNAM pueda cubrir cabalmente las necesidades básicas de autenticación que la institución reclama, sin embargo también somos honestos y realistas y sabemos que se necesitará más tiempo y trabajo para que la PKI que proponemos pueda ser lo que hemos soñado.

A quién no le agrada que su trabajo sea reconocido y más aún que sea tomado en cuenta para formar parte de un proyecto en el que confía y cree, ésta es nuestra expectativa más ambiciosa, nos gustaría que este trabajo de tesis alcanzara su realización o formara parte de un proyecto de seguridad en la UNAM. Pero aunque no se lograra esta meta nos daríamos por bien servidos si esta tesis o la herramienta desarrollada por nosotros, acUNAM, le sirve a tan sólo una persona y así saber que nuestro trabajo y esfuerzo no quedó olvidado en un rincón apartado de esa gran biblioteca que nos ayudó a realizar esta tesis.

CONCLUSIÓN

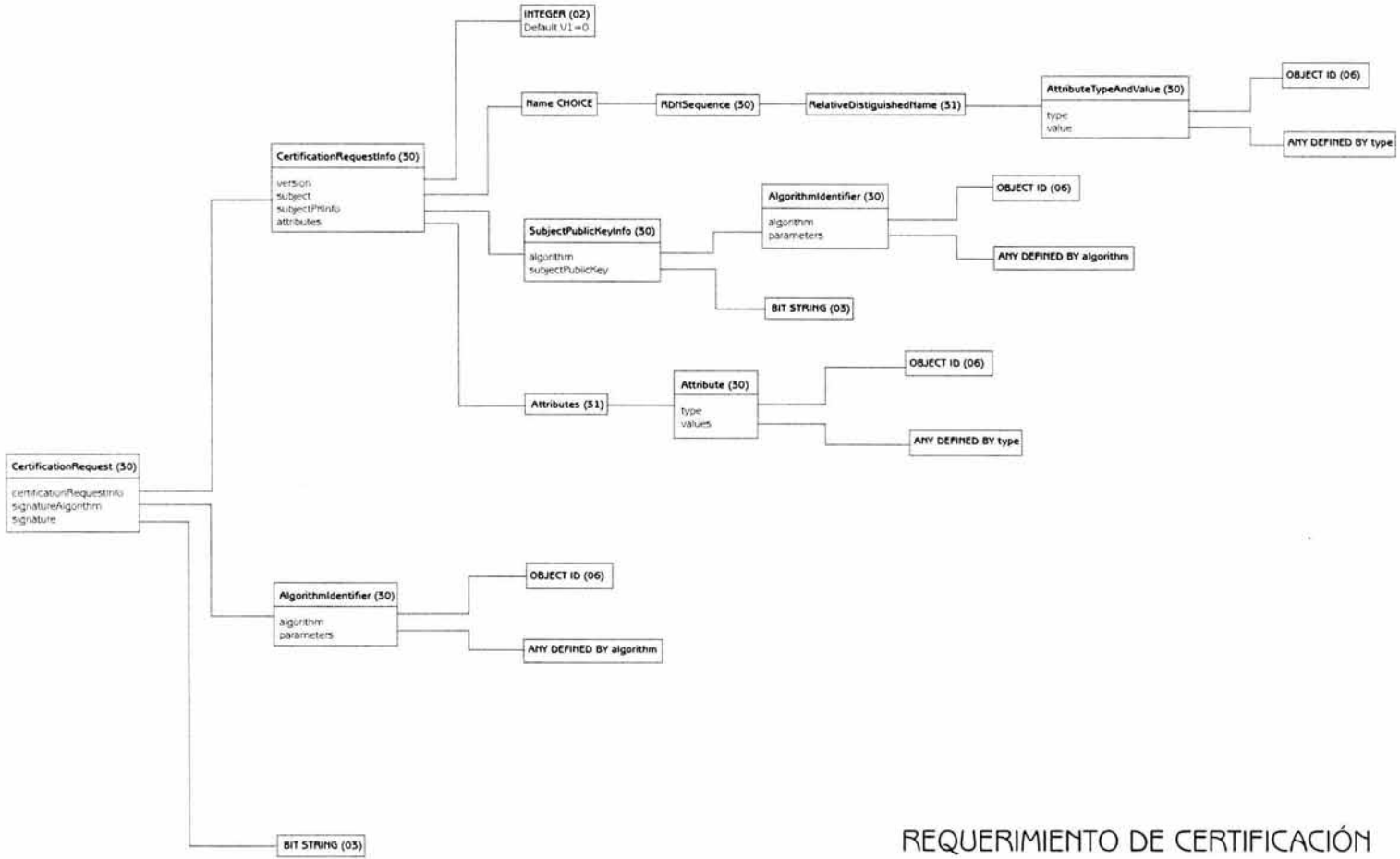
El reto de implementar una PKI es más grande de lo que pensamos. Como cualquier otro sistema informático, acUNAM tiene una tarea muy difícil de cumplir, la de ser funcional para la comunidad, ya que su objetivo es hacer utilizables y funcionales a la criptografía de llave pública, a la tecnología de firma digital y al concepto de certificado digital, lo cual trae inmersos ciertos aspectos que podrían tornar más lento el camino evolutivo que un proyecto de esta naturaleza debe tener. Y es que, para que conceptos como el de criptografía o el de firma digital puedan ser usados de forma habitual en el dominio de la vida cotidiana se debe proporcionar a los usuarios una forma fácilmente comprensible de utilizar dicha tecnología, y no es que acUNAM no haya sido diseñada para tal propósito, sino que desde la existencia de varios criptosistemas, todos ellos compartiendo el mismo concepto y las mismas metas, pero diferentes en detalles de funcionamiento, la compatibilidad entre ellos se pierde, tal como se extravía al mismo tiempo la visión que creemos debe ser unificada en lo que a seguridad y utilización de tecnología se refiere. Por ejemplo, muchos años han pasado desde 1976, año en que se inventó el primer algoritmo de llave pública, y tal parece que aquí en nuestro país apenas se están haciendo intentos para integrar esta tecnología a la vida cotidiana; de cierta forma se comprende tal situación por las condiciones de educación escolar tan duras en México, pero tratándose de un país según ellos "sobre-desarrollado" como E.U. es incongruente enterarse de que la súper-potencia militar y de negocios, se encuentre (con las debidas proporciones) en los albores de la implementación de sistemas de seguridad lo cual se supone constituye su orgullo nacional; el país más seguro del mundo. Por todo lo anterior no nos debe sorprender que las tecnologías concernientes a seguridad aún parezcan divergentes entre sí a los ojos de miles de usuarios, debido a que, por ejemplo, en México tenemos en funcionamiento sistemas de seguridad literalmente parchados con aplicaciones que prometen ser las más seguras (pueden ser, pero ¿serán las apropiadas?); tenemos comercio electrónico que basa su seguridad en AC's extranjeras en las que, a su vez, los usuarios depositan su confianza; estamos invadidos también por empresas de otros países que nos ofrecen servicios de seguridad y nos venden diferentes certificados que usan diferentes tipos de algoritmos y que nos sirven para diferentes usos. Con tantas opiniones, con tantos servicios y con tantos caminos por tomar es fácil gritar a los cuatro vientos: ¿a quién le hago caso? ó ¿en quién confío? No queremos decir con esto que acUNAM sea el producto revolucionario que va a acabar con todos los problemas, de lo que sí estamos seguros es de que la investigación hecha para desarrollar el proyecto, en donde tocamos temas de seguridad y de cómputo en general, podrá servir y motivar a los que vienen detrás de nosotros con ideas, tal vez, más frescas que bien podrán complementar nuestro trabajo. Así pues esperamos que este tratado y las ideas que intentamos plasmar en él, sean bien recibidas por los interesados en la seguridad informática.

Como comentario final, en México hay una enorme necesidad de que exista una legislación aprobada, robusta, funcional y bien fundamentada que se refiera a la responsabilidad de utilizar medios electrónicos para la realización de trámites u operaciones importantes como los de carácter legal y/o financiero; mientras esto no suceda, proyectos como el de nosotros quedarán en el olvido como un intento fallido más que pretendía hacer autosuficiente algún aspecto sensible en el país, en nuestro caso la autenticación de personas en el sub-mundo informático.

Así pues podemos concluir que todos los esfuerzos tecnológicos que se hacen pensando en la mejora de las actividades humanas deben ir de la mano de la responsabilidad y del trabajo que hagamos todos, es decir, ni la tecnología, ni el mejor de los algoritmos criptográficos, ni la más impresionante supercomputadora, por sí mismas harán más seguro el entorno en el que nos movemos, más bien seremos nosotros como usuarios y profesionales responsables, con ayuda de las herramientas tecnológicas, los que hagamos y conformemos sistemas más confiables y seguros.

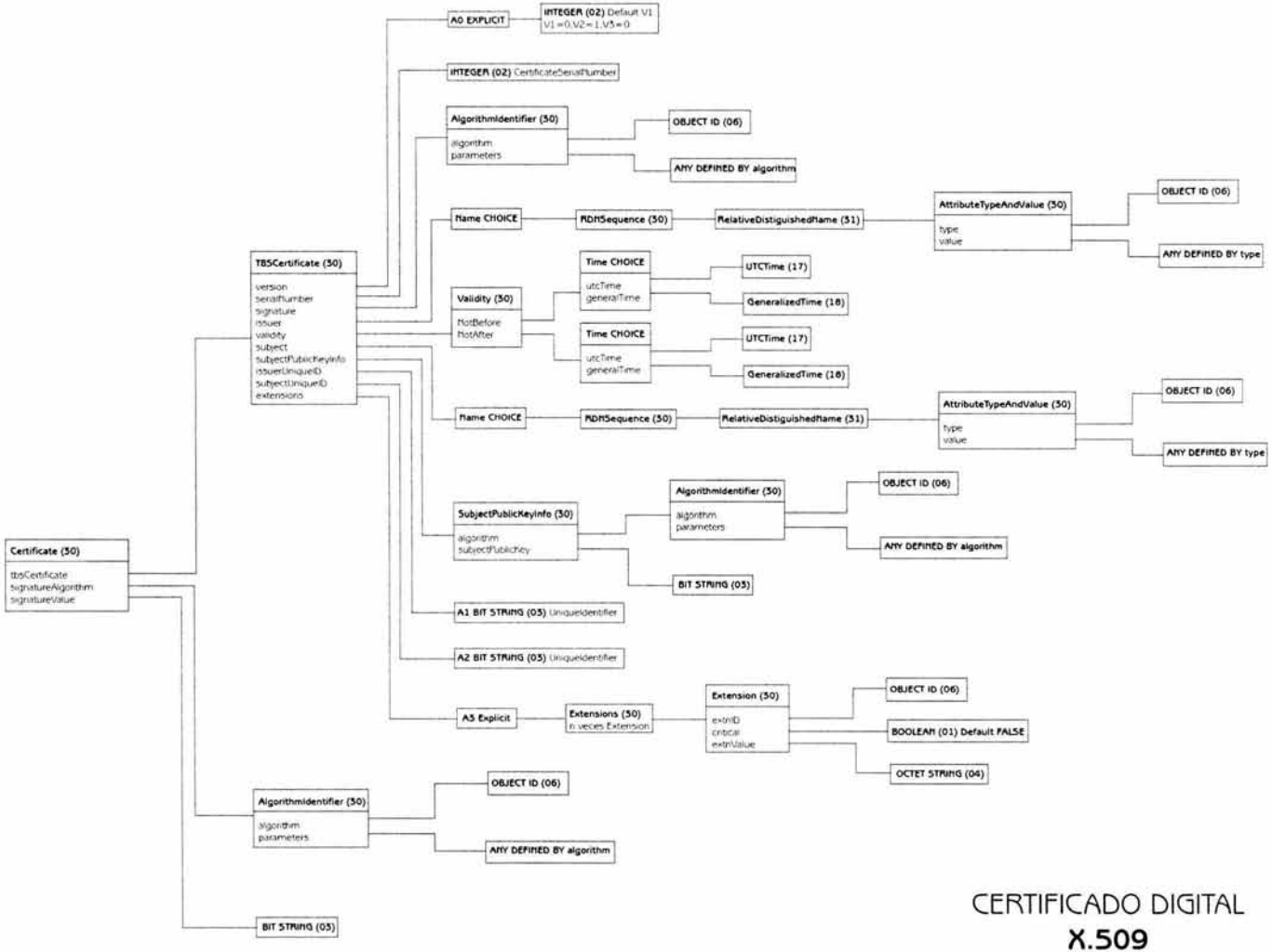
Como ya se había mencionado antes "la seguridad comienza y termina con las personas".

APÉNDICE A
DIAGRAMAS PKCS10 Y X.509



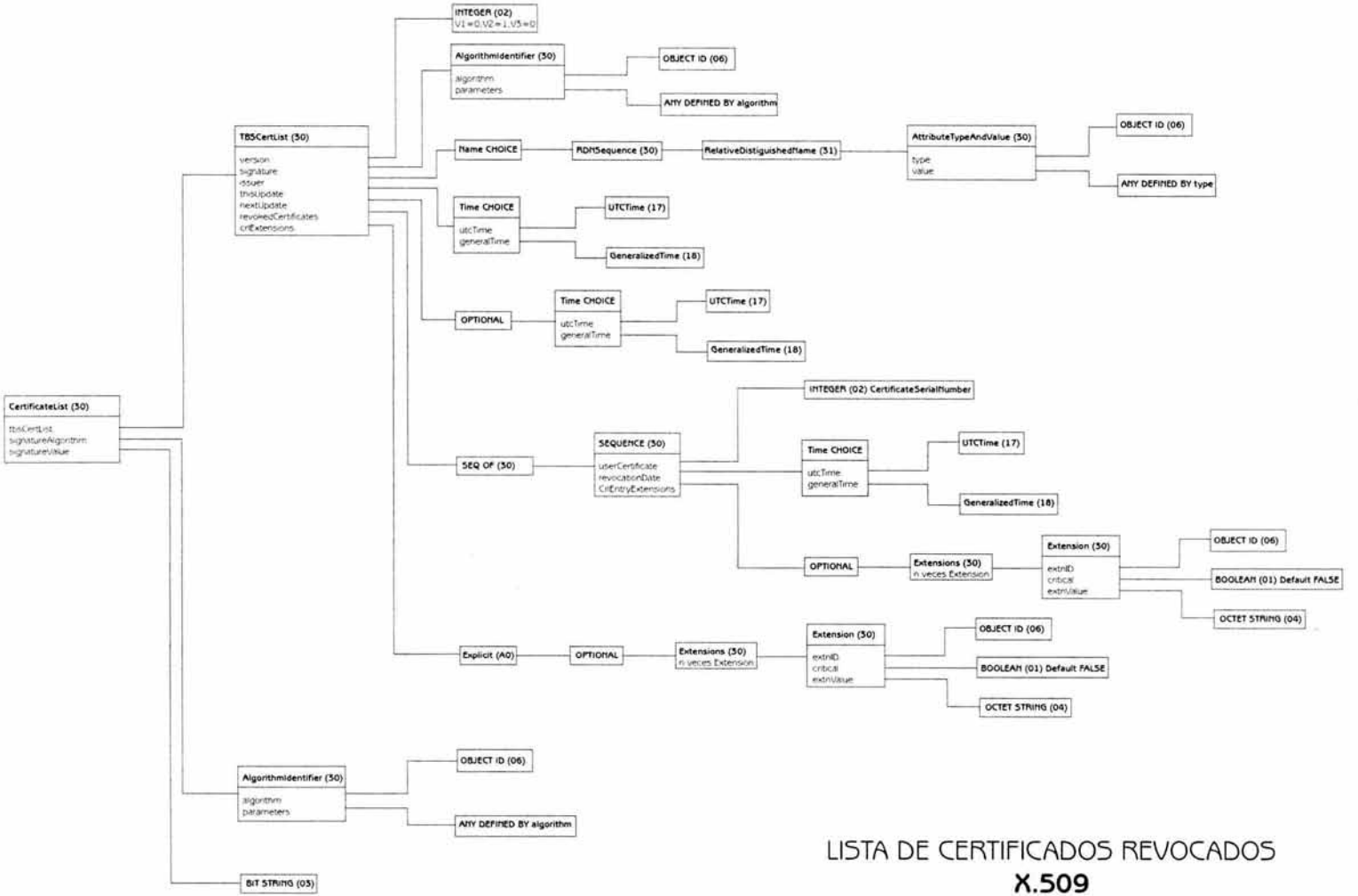
REQUERIMIENTO DE CERTIFICACIÓN

PKCS 10



CERTIFICADO DIGITAL

X.509

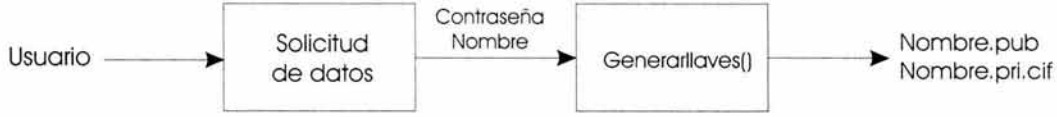


LISTA DE CERTIFICADOS REVOCADOS

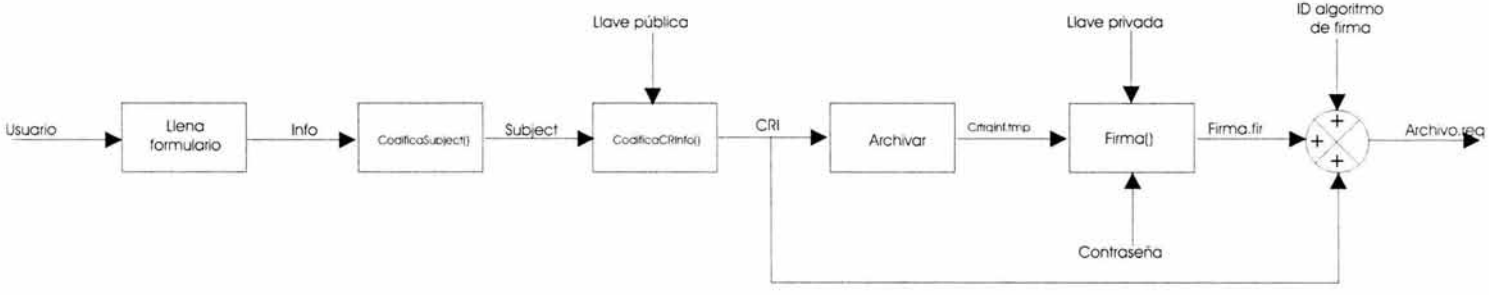
X.509

APÉNDICE B
DIAGRAMAS A BLOQUES

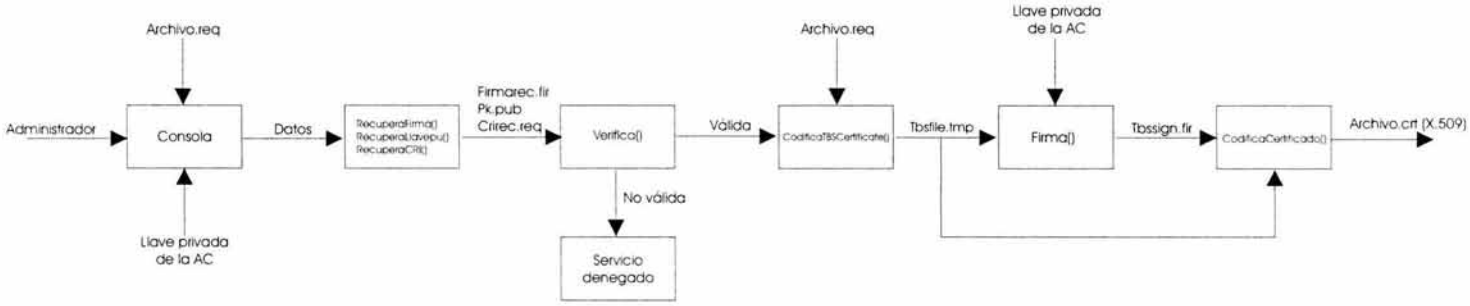
Generación de llaves asimétricas



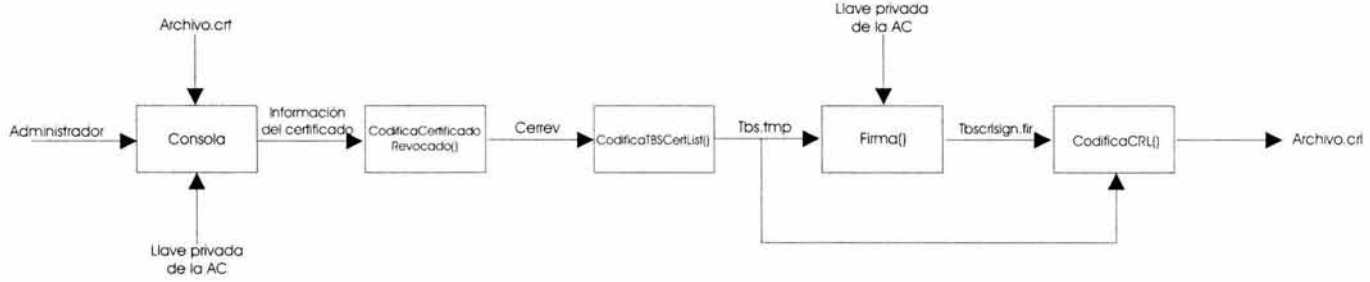
Generación de requerimiento de certificación



Generación de un certificado



Generación de una lista de certificados revocados



GLOSARIO

GLOSARIO

AC: Autoridad Certificadora. Tercera entidad de confianza que crea, asigna y distribuye certificados digitales.

Algoritmo: Es una descripción de los pasos básicos a seguir para completar determinada tarea.

Amenaza: Es un evento que tiene el potencial de causar daños a los recursos del sistema, es el riesgo de ser atacado.

ANSI: American National Standard Institute (Instituto Nacional Americano de Estándar)

AR: Autoridad Registradora. Actúa como intermediario entre la AC y el usuario, tiene la responsabilidad de verificar que la llave pública, la identidad y los atributos del usuario sean verdaderos para obtener un certificado digital.

ARL: Authority Revocation List (Lista de revocación autorizada).

ASCII: American Standard Code for Information Interchange. Estándar usado mundialmente para intercambio de información. La tabla básica de caracteres ASCII esta compuesta por 128 caracteres incluyendo símbolos y caracteres de control. Existe una versión extendida de 256 caracteres.

ASN.1: Abstract Syntax Notation 1 (Notación de sintaxis abstracta 1). Lenguaje de definición de notaciones que describe la estructura de datos que se utilizan en muchos estándares.

Ataque: Es la actividad hostil o ingenua que de una u otra forma tiene como objetivo romper la seguridad de un sistema determinado y eventualmente causar daños.

Autenticación: Proceso que verifica que la entidad que emitió cierta información o que utilizó recursos se encuentra correcta y confiablemente identificada.
Identifica plenamente a las entidades que busquen acceso a un sistema de privilegios restringidos.

Autorización: Proceso de concesión de derechos. Es dar permiso o facultar a alguien ó a algo para tener acceso a un sistema en particular.

BER: Basic Encoding Rules (Reglas básicas de codificación)

Certificado Digital: Es una estructura de datos firmada digitalmente por una AC.
Es un documento electrónico que contiene datos característicos de una persona o entidad y la llave pública de la misma haciéndose responsable de la autenticidad de los datos que figuran en el certificado una entidad de confianza denominada Autoridad Certificadora.

Checksum: Valor numérico utilizado para verificar la integridad de un bloque de datos.

Cifrado: Genera mensajes que son incomprensibles para entidades no autorizadas.
Generalmente a un mensaje en claro se le aplica cierta transformación que depende de un algoritmo y de una llave para producir un mensaje cifrado o secreto.

Cifrado asimétrico: Esta basado en el uso de un par de llaves diferentes (pero relacionadas entre sí) para cifrar y descifrar, conformado por una llave publica y una privada; lo que se cifre con una solo puede descifrado la otra.

Cifrado simétrico: Se utiliza la misma llave para cifrar y descifrar

Cliente: Proceso que solicita y obtiene un servicio de red.

Comercio electrónico: Es cualquier forma de transacción comercial en la que las partes se comunican entre sí por medio de ordenadores. Es un servicio de la tecnología que permite la realización de operaciones de negocios y la compraventa de bienes y servicios mediante la utilización de sistemas electrónicos.

Confidencialidad: Propiedad de que la información no sea revelada ni puesta a disposición de partes no autorizadas.

Control de acceso: Protege al sistema de cómputo del uso no autorizado de sus recursos.

Criptoanálisis: Se ocupa de romper los procedimientos de cifrado para así recuperar la información original.
Técnicas de ataque orientadas a la recuperación de llaves criptográficas o texto en claro sin conocer el sistema de cifrado o la clave. En particular se aplica a la obtención de textos en claro a partir de la intercepción de textos cifrados.

Criptografía: Es la ciencia que mediante el tratamiento de la información, protege a la misma de modificaciones y utilización no autorizada. Utiliza algoritmos matemáticos complejos para la transformación de la información en un extremo y la realización del proceso inverso en el otro extremo.

CRL: Certificate Revocation List (Lista de certificados revocados)

Denegación de servicio: Cuando un recurso del sistema se vuelve inaccesible.

Descifrado: Operación inversa del cifrado. Transforma una cadena ilegible de datos a una cadena legible por medio de la llave privada.

DH: Diffie-Hellman. El primer algoritmo de criptografía de llave pública inventado en 1976 que usa el fundamento matemático del logaritmo discreto.

Disponibilidad: Propiedad de ser accesible, asegura que un usuario autorizado pueda tener acceso a la información o a los recursos del sistema en el momento que los requiera.

DIT: Directory Information Tree (Árbol de información de directorios)

DLP: Discrete Logarithm Problem (Problema de Logaritmo Discreto)

DNS: Domain Name System. Sistema de nombres de Dominio. Base de datos distribuida que gestiona la conversión de direcciones de Internet expresadas en lenguaje natural a una dirección numérica IP. Ejemplo: 121.120.10.1

DSA: Digital Signature Algorithm (Algoritmo de Firma Digital)

DSS: Digital Signature Standar (Estándar de Firma digital)

ECC: Elliptic Curve Cryptography (Criptosistema de Curva Elíptica)

ECDSA: Elliptic Curve Digital Signature Algorithm (Algoritmo de Firma digital con Curvas Elípticas)

Estándar de seguridad: Documento que contiene especificaciones técnicas, reglas o definiciones de características para asegurar que los productos, procesos y servicios son adecuados para su propósito.

FBI: Federal Bureau of Investigation

FIPS: Federal Information Processing Standar (Estándar Federal de Procesado de Información)

Firewall: Es un sistema de defensa que se basa en la instalación de una "barrera" entre la PC y la Red, por la que circulan todos los datos. El tráfico entre la red y la PC es autorizado o denegado por el firewall, siguiendo las instrucciones que le hayamos configurado.

Firma digital: Es la transformación de un mensaje utilizando una función de resumen junto con un criptosistema asimétrico, este se añade al mensaje para que el destinatario pueda comprobar la integridad del mensaje y la identidad del emisor.

FTP: File Transfer Protocol (Protocolo de transferencia de archivos)

Hacker: Experto en informática capaz de entrar en sistemas cuyo acceso es restringido. No necesariamente con malas intenciones.

Hash: La función Hash es una función matemática que selecciona, de manera aparentemente aleatoria, unos valores dentro de un amplio rango, obteniendo un resumen de la información (valor hash) que se utiliza para la detección de errores en la transmisión de datos.

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

ICV: Valor de comprobación de integridad. Valor que sirve de referencia para el destinatario para comprobar que recibió sin alteraciones sus datos.

Identificación: Es el proceso que habilita el reconocimiento de un usuario en un sistema dado.

IFP: Integer Factorization Problem (Problema de Factorización de enteros)

Integridad: Este servicio previene la alteración no autorizada o la destrucción de información transmitida o almacenada.

Intruso: Persona o entidad que gana acceso a un sistema restringido de forma ilegal.

Ipsec: IP Security protocol (Protocolo IP de seguridad) Protocolo cifrado de red para proteger paquetes IP utilizando servicios criptográficos de seguridad que permiten la autenticación, integridad, control de acceso y confidencialidad.

ISO: International Organization for Standardization (Organización Internacional para la Estandarización)

ITU-T: International Telecommunication Union (Unión Internacional de Telecomunicaciones)

LAN: Local Area Network (Red de área local)

LDAP: Lightweight Directory Access Protocol. Un protocolo de acceso a directorios más simple que X.500 para acceso a directorios.

Llave: Secuencia de símbolos que controla las operaciones de cifrado y descifrado

Llave privada: Es una llave alfanumérica creada por medio de algoritmos matemáticos que sirve para descifrar mensajes que han sido creados por la llave pública que forma la pareja. Es conocida sólo por el titular y se debe mantener en secreto.

Llave pública: Es una llave alfanumérica creada por medio de algoritmos matemáticos que sirve para cifrar mensajes, es de dominio público y suele ser creada junto con una llave privada.

MAN: Metropolitan area Network (Red de área Metropolitana)

MD: Message Digest (resumen de mensajes)

Mensaje: Información enviada de una entidad a otra en una red.

Mensaje cifrado: Texto que es ilegible, para poder leerlo es necesario aplicarle una operación, es decir decodificarlo.

Mensaje claro: Texto que esta escrito en claro, es decir que puede ser leído sin tener que realizar ninguna operación, no esta codificado.

Métodos Biométricos: Los sistemas biométricos se basan en alguna característica física del usuario para verificar su identidad.

MIT: Massachusetts Institute of Technology (Instutito de Tecnología de Massachusetts)

NIST: National Institute of Standard and Technology (Instituto Nacional de Estándares y Tecnología)

No repudio: Propiedad que se consigue por medios criptográficos, que impide a una persona o entidad negar haber realizado una acción en particular referente a datos.

NSA: National Security Agency (Agencia de Seguridad Nacional)

OSI: Open Systems Interconnection (Interconexión de sistemas abiertos)

Passphrases: Conjunto de palabras secretas (passwords) que forman un enunciado o un párrafo.

Password: Palabra o cadena de caracteres secretos que un medio automático reconoce con el fin de autenticar a un usuario para permitirle acceso a información restringida.

PEM: Privacy Enhancend Mail (Correo con privacidad mejorada). Se trata de un borrador para una norma Internet que persigue dar seguridad a los servicios de correo electrónico.

PGP: Pretty Good Privacy (Privacidad muy buena). Es un programa que proporciona autenticación y confidencialidad tanto para el envío de correo electrónico como la protección de archivos almacenados.

PKCS: Public Key Cryptography Standard (Estándar de criptografía de llave pública). Conjunto de estándares sobre la implementación de algoritmos de llaves públicas en una forma fiable, segura e ínter operable.

PKI: Public key Infraestructure (Infraestructura de Llave Pública)

PKIX: Internet X.509 Public Key Infrastructure. El nombre del grupo de trabajo de la IETF que crea los estándares para PKI en Internet.

PIN: Personal identification number (Número de identificación personal)

Políticas de seguridad: Es un conjunto de reglas, normas y protocolos que definen el camino a seguir para proteger la integridad y el buen funcionamiento de un sistema.

Protocolo: Conjunto de normas que los equipos utilizan para comunicarse entre sí a través de una red y poder hablar el mismo idioma.

Repudio: Negación o rechazo de alguna de las entidades participantes en una comunicación de haber participado totalmente o en parte.

RFC: Request for Comment (Solicitud de comentarios). Serie de documentos iniciada en 1967 que describe el conjunto de protocolos de Internet. Los RFC son elaborados por la comunidad Internet.

Riesgo: Situación en la que hay peligro.

Router: Permite enviar información a través de rutas elegidas y clasificadas como seguras, además permite cambiar de ruta si se detecta alguna irregularidad.

RSA: Rivest, Shamir and Adleman. Algoritmo utilizado para cifrar y firmar datos, nombrado así por sus inventores.

Seguridad informática: Protege a los recursos de cómputo contra usos y abusos no autorizados.

Servidor: Equipo que controla el acceso de los usuarios a una red y les da servicio e información.

SET: Secure Electronic Transactions (Transacciones Electrónicas Seguras). Protocolo creado para proporcionar mayor seguridad a los pagos on-line con tarjetas de crédito verificando la identidad de los titulares de las tarjetas con "certificados digitales" y cifrando los números de las tarjetas durante todo el trayecto, desde el usuario, el vendedor y el centro de proceso de datos. Este estándar ha sido creado por VISA y Master Card y tiene un amplio apoyo de la comunidad bancaria mundial.

SHA: Secure Hash Algorithm Algoritmo criptográfico para la firma digital, proporcionar un resumen del mensaje o un valor hash.

SHTTP: Secure Hypertext Transfer Protocol (HTTP Seguro). Mejora de HTTP con funciones de seguridad que protege de forma independiente cada transacción realizada.

S/MIME: Secure Multipart Internet Message Extensions. Literalmente, MIME Seguro. Protocolo abierto para la transmisión de mensajes por e-mail, que permite firmar digitalmente y cifrar los mensajes y documentos adjuntos del e-mail, como garantía de seguridad.

SSL: Secure Socket Layer. Protocolo, creado por Netscape, para crear conexiones seguras al servidor, de tal modo que la información viaja cifrada a través de Internet.

Telnet: Programa que permite abrir una sesión (login) en un ordenador remoto conectado a la red. Se puede acceder a cualquiera de los servicios que la máquina remota ofrezca a sus terminales locales.

TI: Tecnología de la información

Timestamping: Registro de tiempo. Es una marca temporal en el documento que garantiza la validez de la firma digital del documento.

Transacción electrónica: Acuerdo basado en concesiones recíprocas mediante un dispositivo electrónico.

TSA: Time Stamp Authority

VPN: Virtual Private Network (Red privada virtual) Red privada construida sobre una red pública que no obstante proporciona privacidad y/o autenticación a los usuarios de dicha red.

Vulnerabilidad: Debilidad que puede ser explotada para violar un sistema o la información que contiene.

WAN: Wide Area Network (Red de área amplia)

X.500: Norma desarrollada conjuntamente por ISO/IEC e ITU-T dentro de la arquitectura OSI, para la creación y mantenimiento de servicios de directorio de forma distribuida.

X.509: Especificación de certificado de llave pública como parte de X.500 y a menudo utilizado en sistemas de llave pública.

BIBLIOGRAFIA

BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN

Fuentes Bibliográficas.

FÚSTER, Sabater A, De la Guía Martínez D.,Hernández Encinas L., *"Técnicas Criptográficas de protección de datos"*. Alfa Omega & Ra-Ma. 2ª ed. México 2001.

FISH, Eric A., White Gregory B., *"Secure Computers and Networks, Analysis, Design and Implementation"*. CRC Press. USA. 2000.

BAUER, Friedrich L., *"Decrypted Secrets: Methods and Maxims of Cryptology"*. Springer-Verlag. 2ª ed. Alemania 2000.

NICHOLS, Randall K., *"ICSA Guide to Cryptography"*. McGrawHill. 1a ed. USA 1999.

FREEDMAN, Alan, "Diccionario de Computación". McGrawHill. 5ª ed. México 1993.

Fuentes De Internet.

http://www.htmlweb.net/seguridad/varios/firma_certificados.html

http://www.htmlweb.net/seguridad/cripto/cripto_1.html

http://www.abanet.org/scitech/ec/isc/dsg-tutorial.html

http://www.it-cenit.org.ar/Publicac/BancaMD/BanCom5

http://tejo.usal.es/~nines/d.alumnos/criptografia2/documentos/doc4_4.htm#[X509]

http://www.seguridata.com/nf/rsa/rsa1.htm

http://rinconquevedo.iespana.es/rinconquevedo/Criptografia/autenticacion.htm

http://www.carsoft.com.ar/crip_asim.htm

http://revista.robotiker.com/articulos/articulo51/pagina1.jsp

http://www.w2000mag.com/atrasados/1999/30_abr99/articulos/firmas11.htm

http://www.delitosinformaticos.com/especial/seguridad/algoritmos.shtml

http://www.um.es/atica/buscadores/directorio.html#intro

http://www.isi.salford.ac.uk/staff/dwc/Version.Web/Contents.htm

http://www.faqs.org/rfcs/rfc2459/.html

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

http://www.rediris.es/rediris/boletin/39/enfoque3.html

http://www.qubit.org

http://www.criptonomicon.com