Infraestructura de clave pública

PKI

Helena Rifà Pous

PID_00214914
Módulo 7

UOC
Universitat Oberta de Catalunya

Índice

Introducción 5

Objetivos 6

1. Conceptos básicos 7
1.1. Problemática de la distribución de claves 7
1.2. Propósitos de una infraestructura de clave pública (PKI) 8
1.3. Estándares de infraestructura de clave pública 8

2. Componentes de una infraestructura de clave pública 10
2.1. Autoridad de certificación 10
2.2. Autoridad de registro 11
2.3. Suscriptores y entidades finales 11
2.4. Usuarios 12
2.5. Repositorios 12
2.6. Autoridad de validación 13
2.6.1. *On line Certificate Status Protocol (OCSP)* 14
2.6.2. *Simple Certificate Validation Protocol (SCVP)* 14
2.7. Autoridad de sellado de tiempo 15

3. Modelos de confianza 16
3.1. El modelo distribuido 16
3.2. El modelo plano 18
3.3. El modelo jerárquico 18
3.4. El modelo de navegación por lista de confianza 19
3.5. El modelo de certificados cruzados 20
3.6. El modelo Bridge-CA 21

4. Ciclo de vida de claves y certificados digitales 23
4.1. Generación del par de claves 23
4.2. Registro 24
4.3. Certificación 24
4.4. Recuperación de claves 25
4.5. Revocación de certificados 25
4.6. Renovación de certificados 26

5. Estructuras de datos básicos de PKIX 28
5.1. Certificados X.509 28
5.1.1. Nombre de los componentes de un certificado 29
5.1.2. Estructura de un certificado 29
5.2. Listas de revocación (CRL) 32
5.2.1. Tipo de listas de revocación de certificados 33

5.2.2. Puntos de distribución de listas de revocación de certificados 34
5.2.3. Formato de las listas de revocación de certificados 34

6. Formato y procedimientos para generar mensajes: PKCS 37
6.1. El PKCS#7 38
6.1.1. Los *data* 38
6.1.2. Los *signedData* 38
6.1.3. Los *envelopedData* 40
6.1.4. Los *signedAndEnvelopedData* 42
6.1.5. Los *digestedData* 43
6.1.6. Los *encryptedData* 43
6.2. El PKCS#10 44
6.3. El PKCS#12 45

7. Protocolos que utilizan infraestructura de clave pública 47
7.1. IPsec 47
7.2. SSL 48
7.3. S/MIME 49
7.4. Seguridad en XML 49

Resumen 51

Actividades 53

Ejercicios de autoevaluación 53

Solucionario 54

Glosario 55

Bibliografía 56

Introducción

En este módulo vamos a estudiar de qué manera es posible implantar sistemas criptográficos basados en clave pública que permitan el uso generalizado de servicios de integridad, confidencialidad, autenticidad y no repudio. En primer lugar, veremos cuál es la problemática asociada a la implementación de la criptografía asimétrica, y presentaremos la solución de la infraestructura de clave pública (o PKI, *public key infrastructure*).

La tecnología de infraestructura de clave pública está en fase de desarrollo. Participan muchas instituciones y grupos de usuarios, y el proceso de estandarización no es fácil ni rápido. En este módulo no pretendemos ver todas las posibilidades de una infraestructura de clave pública, sino sólo dar una idea general de la tecnología. Estudiaremos sus componentes esenciales, en especial el certificado digital X509, y veremos cómo se interrelacionan para crear los diferentes modelos de PKI. De esta manera, obtendremos una visión global de los pasos necesarios para ofrecer servicios de criptografía de clave pública a un conjunto de usuarios.

Una vez conocidos los principios de una infraestructura de clave pública, veremos de qué manera es posible generar mensajes seguros entre diferentes aplicaciones de acuerdo con los estándares actuales.

Finalmente, veremos algunos de los protocolos y servicios de seguridad actuales más conocidos, que se fundamentan en infraestructuras de clave pública para su correcto funcionamiento.

Objetivos

Los materiales didácticos de este módulo os permitirán alcanzar los objetivos siguientes:

1. Entender los principios básicos de una infraestructura de clave pública: los problemas que resuelve, sus componentes y sus funciones.

2. Asimilar las características de las estructuras básicas que se utilizan dentro de un entorno de infraestructura de clave pública, qué tipo de información contienen y para qué sirven.

3. Comprender el formato de las estructuras de datos PKCS más importantes utilizadas en criptografía de clave pública.

4. Conocer el funcionamiento de diversos protocolos de seguridad que se basan en la tecnología de infraestructura de clave pública.

1. Conceptos básicos

En este apartado vamos a ver cuáles son las motivaciones del nacimiento de las infraestructuras de clave pública.

1.1. Problemática de la distribución de claves

La criptografía de clave pública permite el intercambio de mensajes confidenciales e íntegros de manera ágil, siempre que dispongamos de la clave pública del interlocutor con el que nos comunicamos. El problema es cómo obtener esta clave pública y cómo estar seguros de que pertenece a quien creemos.

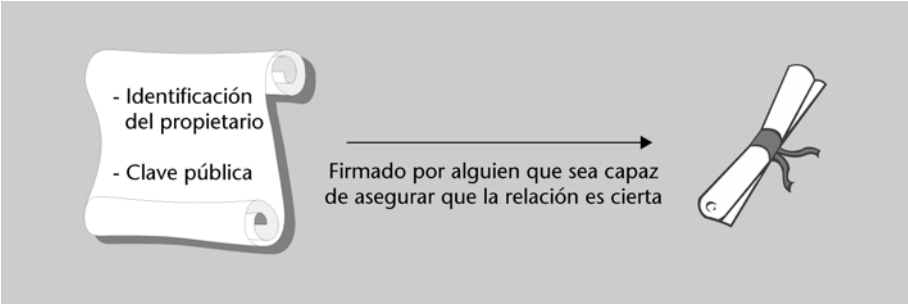
Las claves públicas son justamente eso: públicas; y un intruso lo bastante hábil podría tener acceso al directorio donde están ubicadas y sustituir la clave pública de algún usuario por la suya propia. De esta manera, los interlocutores que utilizaran la clave pensarían que se están comunicando con aquel usuario cuando, en realidad, lo harían con el intruso. Es lo que se conoce como *ataque del hombre a medio camino*.

Los criptosistemas de clave pública se tratan en el módulo "Criptosistemas de clave pública" de esta asignatura.

Diffie y Hellman (1976) pensaron que el problema de la distribución de claves se resolvería con un directorio seguro en línea en el cual se estableciera de manera unívoca el vínculo entre un nombre distintivo de usuario y una clave pública. El directorio público tendría que firmar todas sus transacciones de manera que nadie pudiera suplantar su identidad. El problema radicaba en el bajo rendimiento que el sistema ofrecía en poblaciones de usuarios medias o mayores. L. Kohnfelder (1978) se basó en la idea de una autoridad central de confianza introducida por Diffie y Hellman y propuso crear unos registros de datos firmados –los certificados– que permitirían que la distribución de claves se hiciera desde directorios públicos que no requirieran confianza.

Un **certificado digital** es una estructura de datos que contiene información del propietario de las claves criptográficas, la clave pública en sí y una firma digital de los dos campos anteriores que le da validez.

La firma, realizada por un usuario o entidad externa leal, asegura la integridad contra una posible modificación no deseada de los datos. La confianza en el signatario se extiende al sujeto del certificado.



Desde entonces, los certificados se han convertido en el principal medio de distribución de las claves públicas de los sistemas de clave asimétrica. De todos modos, el uso de certificados no resuelve totalmente el problema de la distribución de claves, sino que traslada esa cuestión a un nivel superior. ¿Quién tiene que firmar el certificado? ¿Qué mecanismos son necesarios para que dos usuarios que no se conocen puedan asegurar su identidad en una comunicación virtual? En definitiva, ¿cuál es el modelo de confianza?

En el apartado 3 de este módulo veremos los modelos de confianza más comunes y qué ofrece cada uno.

1.2. Propósitos de una infraestructura de clave pública (PKI)

El objetivo de una infraestructura de clave pública es la gestión eficiente y digna de confianza de las claves criptográficas y los certificados para que se puedan utilizar para funciones de autenticación, integridad, no repudio y confidencialidad. La infraestructura de clave pública crea un marco seguro de intercambio de datos en un entorno típicamente inseguro como es Internet.

Una **infraestructura de clave pública*** (**PKI**) es el conjunto de *hardware*, *software*, personas, políticas y procedimientos necesarios para crear y gestionar certificados digitales basados en criptografía de clave pública.

*** En inglés, *public key infrastructure.***

El certificado es el elemento central de la infraestructura de clave pública en torno al cual se crea esta infraestructura de soporte que abarca servicios como el registro de usuarios, la emisión de certificados, su distribución desde directorios públicos, su renovación y revocación, la recuperación de claves, etc.

1.3. Estándares de infraestructura de clave pública

Para poder tener aplicaciones interoperables, hay que establecer unos estándares en materia de infraestructura de clave pública. El hecho de que diversos fabricantes se pongan de acuerdo en la sintaxis y la estructura de los datos atrae tanto a desarrolladores como a usuarios finales, y hace que el mercado crezca y mejore.

El mercado de la seguridad se encuentra a medio camino de la transición entre soluciones propietarias y estándares abiertos. Algunos fabricantes no quieren dar

a conocer sus soluciones porque basan la seguridad en la oscuridad y el secreto. Ésta no es una buena política, ya que no cumple la suposición de Kerckhoffs.

La suposición de Kerckhoffs se explica en el subapartado 4.1 del módulo "Fundamentos de criptografía" de esta asignatura.

Los principales organismos estandardizadores en materia de infraestructura de clave pública en Internet son los siguientes:

a) Internet Engineering Task Force (**IETF**). La IETF, como foro global para el desarrollo de estándares de redes, es el lugar preferente para el desarrollo de especificaciones y estándares que ofrezcan soporte al crecimiento de las aplicaciones de servicios de confianza. El trabajo realizado en el campo de las infraestructuras de clave pública dentro del grupo llamado PKIX es un ejemplo de esta actividad, ya que constituye el modelo más sobradamente aceptado y en él nos basaremos a lo largo de este módulo. El medio de publicación de los estándares y la información relacionada es la serie de documentos denominada ***requests for comments*** (**RFC**), de los cuales hay dos categorías principales:

- Los ***standards track*** son aquellos que siguen el progreso del RFC desde la propuesta o borrador hasta el estándar.

- Los ***non-standards track*** no cuentan con la aprobación y el consenso de toda la comunidad Internet y se clasifican en experimentales, informativos e históricos.

b) International Standards Organisation (**ISO**). El ISO se centra en la ratificación de estándares generados tanto por la IETF como por otros organismos nacionales con el fin de acercar más determinadas normas a la industria.

c) European Telecommunications Standards Institute (**ETSI**). El ETSI es un organismo europeo formado por empresas activas en la industria de telecomunicaciones, principalmente como proveedores de servicios y vendedores.

d) European Committee for Standardisation (**CEN**). El CEN comprende los organismos de estandarización nacionales de Europa.

Por otra parte, los **RSA Laboratories** han realizado importantes aportaciones a los formatos de datos basados en infraestructura de clave pública y han conseguido que muchas de sus especificaciones se hayan convertido en estándares *de facto*, como los PKCS.

Veremos algunos ejemplos de PKCS en el apartado 6 de este módulo.

2. Componentes de una infraestructura de clave pública

En este apartado veremos los componentes de una infraestructura de clave pública típica. Aunque sus componentes esenciales son la autoridad de certificación, los suscriptores y los repositorios, hemos añadido otros, como la autoridad de registro, la autoridad de validación o la autoridad de sellado de tiempos, que dan una visión completa de las funciones que puede llegar a proporcionar una infraestructura de clave pública. Otros componentes opcionales que no detallaremos son el repositorio de claves (que permiten facilitar servicios de recuperación de claves criptográficas) o la autoridad documental.

2.1. Autoridad de certificación

La **autoridad de certificación*** (CA) es la responsable de emitir y revocar certificados. Es la entidad de confianza que proporciona legitimidad a la relación de una clave pública con la identidad de un usuario o servicio.

* En inglés, *certification authority*.

Políticas de certificado y prácticas de certificación

La relación legal y técnica entre una autoridad de certificación y sus suscriptores y usuarios está regida por las políticas de certificados (*certificate policy*, CP) y la declaración de prácticas de certificación (*certification practice statement*, CPS). Una **política de certificado** expone la garantía de seguridad que se puede dar al certificado y los usos para los cuales es adecuado. Es un concepto general, no particular de una organización. Sirve como vehículo para establecer las bases de la interoperabilidad. Las **prácticas de certificación** estipulan la manera en que una autoridad de certificación en particular establece esta garantía de seguridad. Normalmente, la CPS se suele presentar en forma de documento al alcance del público.

Una infraestructura de clave pública puede tener una o más autoridades de certificación. La creación de una autoridad de certificación se inicia con la generación del par de claves (públicas y privadas) que se utilizarán para firmar y validar los certificados digitales que emita la autoridad de certificación. Las claves tienen que ser lo bastante fuertes para que la probabilidad de que un atacante las rompa sea extremadamente baja durante el tiempo de vida de los certificados que se firmarán. Eso dependerá de la combinación de la longitud de la clave y la calidad del algoritmo de generación de claves.

La relación entre las diferentes autoridades de certificación se describe en el apartado 3 de este módulo.

Una vez generado el par de claves, la clave pública se tiene que distribuir de una manera segura entre todas las entidades de confianza potenciales. Esta distribución se puede hacer tanto por medio de un certificado digital emitido por una autoridad de certificación en la que los usuarios ya confían, o por un certificado generado por la propia autoridad de certificación que se acabe de crear (este tipo de certificados reciben el nombre de *certificados autofirmados*). Para garantizar la seguridad de la transmisión, estos certificados digita-

les de las autoridades de certificación se tendrán que distribuir desde un canal externo (será necesario pasar el certificado físicamente, por correo, incorporado al *software*, etc.).

La autoridad de certificación también tiene que proteger su clave privada de un uso no autorizado. La seguridad de una infraestructura de clave pública depende de que se utilicen prácticas adecuadas para crear y gestionar toda la infraestructura de confianza. Es importante que el control de acceso a las claves de la autoridad de certificación sea muy estricto; por eso, generalmente las claves se guardan en una tarjeta o *hardware* criptográfico y se hace uso de la protección física que representa contar con la autoridad de certificación aislada y sin ninguna conexión de red.

Los dispositivos más utilizados

Los dispositivos criptográficos más utilizados son las tarjetas inteligentes (*smartcards*), HSM (*hardware security modules*) y *tokens* USB.

El núcleo básico de una autoridad de certificación es el par de claves criptográficas. Pero además de las claves, las autoridades de certificación suelen estar formadas por diversos subcomponentes o servicios, como el repositorio de certificados, el repositorio de claves, servicios seguros de *logs*, etc.

2.2. Autoridad de registro

La **autoridad de registro*** (**RA**) es la encargada de verificar el vínculo entre las claves públicas y la identidad de sus titulares.

* En inglés, *registration authority*.

En España

En España, las delegaciones y administraciones que la Agencia Tributaria tiene en cada localidad y las oficinas consulares actúan como autoridades de registro de la Fábrica Nacional de Moneda y Timbre.

Las autoridades de registro son un componente opcional de una infraestructura de clave pública que se utiliza para descargar la autoridad de certificación de muchas de las funciones administrativas. En particular, son especialmente útiles en organizaciones grandes y geográficamente dispersas.

Poca seguridad

Cuando el modelo de infraestructura de clave pública no requiere un nivel de seguridad muy elevado, la autoridad de registro se puede automatizar de manera que la validación sea más laxa y se efectúe más mecánicamente.

2.3. Suscriptores y entidades finales

Los **suscriptores** y las **entidades finales** son aquellos que poseen un par de claves (pública y privada) y un certificado asociado a la clave pública.

Con este par de claves podrán efectuar firmas digitales y cifrar y descifrar documentos. Una entidad final representa un organismo, mientras que un suscriptor es una persona.

2.4. Usuarios

Los **usuarios** son los agentes que validan firmas digitales y su ruta de certificación a partir de claves públicas emitidas por autoridades de certificación de confianza. También pueden cifrar documentos para suscriptores y entidades finales.

Observad que, a diferencia de los suscriptores y las entidades finales de una infraestructura de clave pública, los usuarios no tienen por qué disponer de ningún par de claves ni de certificado alguno. Evidentemente, suscriptores y entidades finales son, en particular, usuarios.

2.5. Repositorios

Los **repositorios** son las estructuras encargadas de almacenar la información relativa a la infraestructura de clave pública. Los dos repositorios más importantes en una infraestructura de clave pública son el repositorio de certificados y el repositorio de listas de revocación de certificados.

Una **lista de revocación de certificados*** (**CRL**) incluye todos aquellos certificados que por diversos motivos son inválidos antes de la fecha de caducidad establecida en el mismo certificado.

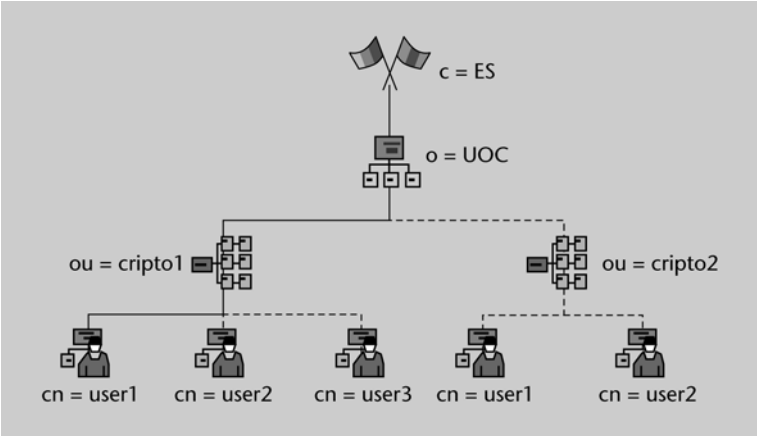
*** En inglés,** ***certification revocation list.***

El tipo de repositorios más utilizados en infraestructura de clave pública son los directorios. Un **directorio** es una base de datos especializada en la que se almacena información tipificada y organizada sobre objetos. Está optimizado para los accesos de lectura, la navegación y las grandes buscas, y su objetivo es dar respuestas rápidas a un alto volumen de peticiones.

El X.500 es un directorio basado en la arquitectura OSI y diseñado bajo los auspicios del ISI y otras organizaciones de estandarización internacionales. En el modelo X.500, la información está organizada de manera jerárquica (estructura típica de árbol) y cada nodo contiene objetos de una clase determinada con diferentes atributos que son los que realmente contienen la información.

Las entradas al directorio están dispuestas en un árbol y utilizan el subconjunto de atributos denominado **nombre distinguido*** (**DN**). La situación de la entrada anterior en el árbol X.500 sería:

* En inglés, *distinguished name.*



Para acceder a los directorios X.500 se definió el **protocolo de acceso al directorio*** (**DAP**). Pero el DAP era un protocolo del legado ISO y tenía el defecto de que no era compatible con el protocolo de Internet TCP/IP. La IETF (1995) añadió a su línea de trabajo el desarrollo de un protocolo que utilizara un modelo de datos X.500, pero que funcionara sobre TCP/IP y que fuera más fácil de implementar y de poner en marcha. Se denominó LDAP (*lightweight directory access protocol*) y en diciembre de 1997 salió a la luz la versión 2 como propuesta de estándar Internet.

* En inglés, *directory access protocol.*

LDAP

El LDAP fue desarrollado originariamente en la Universidad de Michigan.

2.6. Autoridad de validación

La **autoridad de validación*** (**VA**) es la encargada de comprobar la validez de los certificados digitales.

* En inglés, *validation authority.*

Esta autoridad puede ser la propia autoridad de certificación o una entidad externa.

Los protocolos de validación de certificados son mecanismos que proporcionan información sobre el estado actual del certificado (revocado, suspendido, válido o estado desconocido) o relativa a la cadena de certificación necesaria para validar la autenticidad del certificado.

Una de las principales deficiencias de las listas de revocación de certificados es que la periodicidad con que se publican las renovaciones no está bajo el control de las aplicaciones que tienen que validar los certificados. Idealmente, el conocimiento sobre si un certificado está revocado o no tendría que obtenerse en el mismo momento en que se quiere utilizar. Existen diversos protocolos

de verificación en línea para obtener respuestas instantáneas sobre el estado de los certificados implicados.

A continuación analizaremos dos protocolos para validar certificados: el OCSP (*online certificate status protocol*) y el SCVP (*simple certificate validation protocol*).

2.6.1. *On line Certificate Status Protocol* (OCSP)

El ***On line Certificate Status Protocol*** (**OCSP**) es un protocolo cliente/servidor que valida el estado de los certificados solicitados y devuelve una respuesta firmada digitalmente.

El servidor, implementado en la autoridad de validación, conecta con las autoridades de certificación apropiadas o por alguno otro mecanismo de confianza, y obtiene los datos del estado de los certificados a partir de listas de revocación de certificados. La autoridad de validación se hace responsable de las respuestas que envía a sus clientes (por eso las firma), y los clientes confían en la autoridad de validación y aceptan la sentencia.

El OCSP ofrece dos mejoras con respecto a la validación de certificados a partir de listas de revocación de certificados:

1) **Gestión eficiente de los riesgos**: la autoridad de validación puede responder al usuario con información en tiempo real.

2) **Reducción del tráfico generado**: los usuarios no reciben grandes listas de certificados de las que sólo les interesa a una entrada, sino únicamente la información que necesitan. Para asegurar la máxima compatibilidad entre redes, se utiliza el HTTP como protocolo de transporte de las peticiones OCSP.

2.6.2. *Simple Certificate Validation Protocol* (SCVP)

SCVP es una línea actual de trabajo de PKIX para desarrollar un protocolo de validación de certificados que se adapte a las necesidades de dispositivos clientes que no tienen mucha capacidad de proceso ni de almacenaje. Además de las funciones aportadas por el protocolo OCSP, el SCVP ofrece servicios relacionados con la investigación y el procesamiento de cadenas de certificación. Así, cuando un cliente quiere verificar una firma digital, no tiene que construirse toda la ruta de certificación, desde el certificado utilizado para generar la firma hasta la autoridad de certificación raíz, y validar el estado de todos estos certificados. Simplemente puede pedir el estado de los certificados asociados al certificado de la firma que quiere verificar y el servidor se encargará de crear toda la cadena de certificación y validar cada uno de los componentes.

2.7. Autoridad de sellado de tiempo

La **autoridad de sellado de tiempo*** (**TSA**) es la encargada de firmar un mensaje con la finalidad de demostrar que existía antes de un determinado instante de tiempo.

*** En inglés, *time stamping authority*.**

En el documento de definición del servicio, se especifica el protocolo basado en mensajes de petición y respuesta que permiten asociar marcas temporales de confianza a los documentos.

La necesidad de una autoridad de sellado de tiempo es importante para la propiedad de no repudio. Los servicios de no repudio tienen que poder establecer la existencia de unos datos antes de determinados momentos. El papel de la TSA consiste en sellar estos datos con el fin de establecer su evidencia. Algunos ejemplos en que la TSA tiene un papel importante son:

a) Verificación de la firma digital de un documento: si el certificado correspondiente ha sido revocado, el sello de tiempo nos permitirá establecer si en el momento en que se efectuó la firma del documento, el certificado todavía era válido o no.

b) Entrega de documentos antes de una fecha límite: el sello de tiempo nos permite indicar el momento en qué se hizo la entrega de determinada información.

c) Auditorias: el sello de tiempo permite tener fechadas todas las entradas de los históricos.

Para que el sellado de tiempo sea válido, la estructura de datos que lo contiene tiene que estar protegida de forma criptográfica y, además, la marca de tiempo que lleva se tiene que haber obtenido de una fuente de confianza. Eso se puede asegurar con la utilización de proveedores oficiales de valores temporales basados en el tiempo coordinado universal (UTC, *Universal Time Coordinated*), que garantizan una alta precisión en los datos temporales que suministran.

Los pasos para poner un sello de tiempo en un mensaje son los siguientes:

1) Se calcula un resumen digital de los datos que se quieren sellar.

2) Se envía el resumen a una TSA, que añade la fecha y la hora, firma todo el bloque con su clave privada y devuelve el resultado al cliente.

3) El cliente verifica la firma y comprueba que la TSA ha firmado el hash de sus datos y ha añadido un sello de tiempo actual.

3. Modelos de confianza

Hemos visto que para trabajar con criptografía de clave pública es necesario el uso de unos certificados que nos aseguren la correspondencia entre la clave pública y algún atributo, típicamente la identidad de la entidad o la persona propietaria del certificado. Pero los métodos de certificación absolutos son imposibles, ya que un certificado no se puede certificar él mismo. Por esta razón se han propuesto diferentes métodos que pretenden tratar esta situación: el modelo distribuido de red de confianza, el modelo plano, el modelo jerárquico, el modelo de navegación por lista de confianza, el modelo de certificados cruzados y el modelo Bridge-CA.

3.1. El modelo distribuido

El sistema distribuido de red de confianza es el modelo más sencillo de utilizar, adecuado para un grupo pequeño de usuarios que ya tenían tratos antes de la implantación de la infraestructura de clave pública.

En el **modelo distribuido** cada usuario crea y firma certificados para la gente que conoce. No se tiene que desarrollar ninguna infraestructura central con una tercera entidad de confianza en que dé fe de la identidad de los usuarios.

Un ejemplo de este sistema es el *software* PGP, que corresponde a las siglas de *Pretty Good Privacy**. Se trata de un proyecto iniciado a principios de los años noventa por Phill Zimmerman. Con el paso del tiempo, el PGP se ha convertido en uno de los mecanismos más populares y fiables para proporcionar servicios criptográficos en las comunicaciones, sobre todo por correo electrónico.

*** En inglés, *privacidad bastante buena.***

El *software* PGP asume que sólo los usuarios individuales tienen competencias para decidir en quién confían y en qué grado lo hacen. Los mismos usuarios firman o dan autenticidad a los certificados. Si un usuario A se quiere comunicar con un usuario B, se tendrán que intercambiar las claves públicas por un canal inseguro. Estas claves estarán empaquetadas en un formato de certificado autofirmado que, además, podrá ser firmado por otras personas. Con el fin de asegurar que las claves públicas pertenecen efectivamente a cada usuario, tanto A como B calculan el valor del *hash* de sus claves y se lo intercambian por un canal inseguro diferente al que han utilizado para intercambiarse la clave (por ejemplo, por teléfono o correo regular). Los usuarios calculan el *hash* de la clave del certificado que han recibido y lo comparan con el que les han enviado. Si coinciden, asumen que la clave es auténtica.

En este modelo, tenemos que distinguir dos tipos de confianza:

- La que nos permite creer en la validez de una clave, es decir, la que nos permite creer que esa clave pertenece a esa persona.

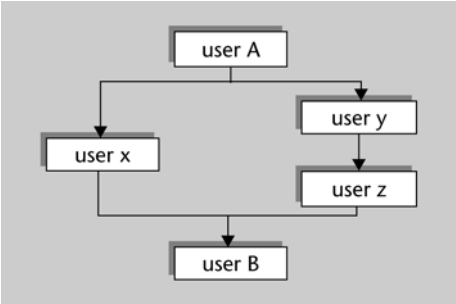
- La que nos permite confiar en una persona como certificadora de claves (esta confianza se basa en una decisión personal, y no en algoritmos matemáticos).

No tenemos que olvidar que el hecho de que un certificado sea auténtico no nos dice nada del propietario. Se puede tener la seguridad de que una clave pertenece a una persona, pero esta persona se podría dedicar a firmar todos los certificados que le llegaran sin asegurarse de su autenticidad, de manera que en ningún caso merecería nuestra confianza.

En el caso del PGP, cada usuario guarda las claves públicas en unas estructuras denominadas **anillos de claves***. Una clave es válida si está firmada (certificada) por el mismo usuario propietario del anillo o por bastantes personas de fiar. Cada clave tiene asociado un nivel de confianza donde establece hasta qué punto se confía en los certificados emitidos por esta clave. Cuanto más bajo es el nivel de confianza, más certificados hacen falta para validar la clave.

*** En inglés, *key ring*.**

Cuando queremos validar el certificado de un usuario cualquiera, utilizamos aplicaciones de busca de rutas de certificación para crear una cadena de confianza adaptada a nuestras necesidades. Damos la aplicación al identificador de clave de nuestro certificado y el identificador del certificado para validar. El servidor de rutas devuelve un resultado como el siguiente:



El usuario *user x* confirma con la emisión de una firma que la clave *B* pertenece realmente al *user B*. Por otra parte, como nosotros (*user A*) hemos firmado la clave del *user x*, podemos confirmar que la clave B pertenece al usuario esperado. Además, hay otro camino de validación que también va desde nuestra clave hasta la del *user B*.

Es importante que todo el mundo firme claves para que otros usuarios se puedan beneficiar de las firmas. Todas estas firmas forman una especie de red, y por eso se afirma que el PGP es un modelo de red de confianza.

El PGP utiliza dos formas diferentes de clave pública: RSA (con función de *hash* MD5) y Diffie-Hellman (con función de *hash* SHA-1), con longitud de clave de

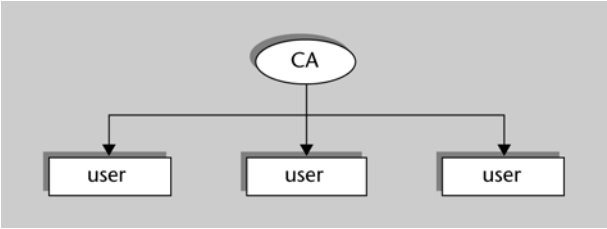
hasta 2.048 (RSA) o 2.096 (DH). Los algoritmos de clave simétrica que puede utilizar son CAST (predeterminado), IDEA y Triple DES.

3.2. El modelo plano

El modelo plano es el sistema más sencillo de infraestructura de clave pública que incluye una única autoridad de certificación como tercera parte de confianza encargada de la emisión y la gestión de los certificados de los suscriptores. Los usuarios pueden validar la identidad de los suscriptores a partir del certificado de la autoridad de certificación. La autoridad de certificación posee un certificado que ella misma ha generado y en la cual los usuarios depositan su confianza.

En un **certificado autofirmado**, la clave pública que se certifica corresponde a la clave privada que se utiliza para firmar el certificado. El nombre del emisor y el titular del certificado son el mismo.

El modelo plano suele usarse en el ámbito de las intranets. Para entornos más extensos y abiertos, este modelo se ha ampliado en el modelo jerárquico.



3.3. El modelo jerárquico

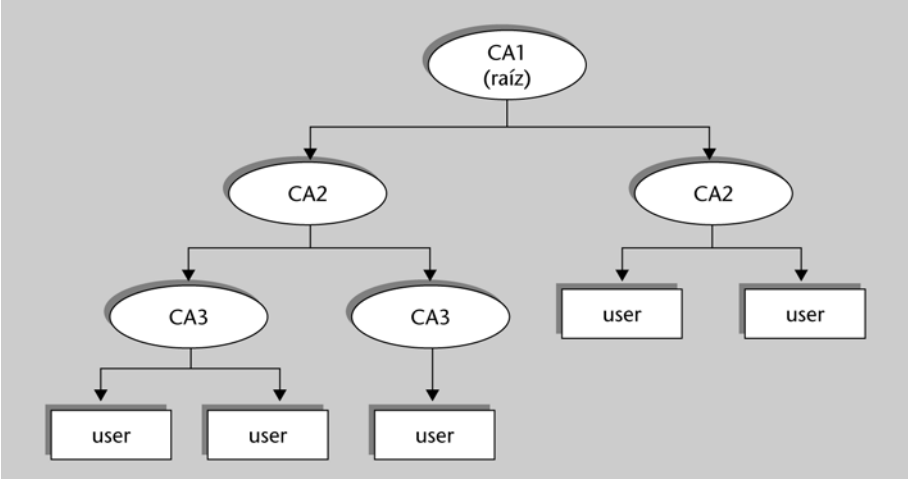
El modelo jerárquico representa la implementación más típica de una infraestructura de clave pública.

En el **modelo jerárquico**, los certificados de los suscriptores y las entidades finales están firmados por una entidad externa que también se identifica con certificados que emitirá una autoridad de certificación de jerarquía superior.

Los certificados de la autoridad de certificación de jerarquía superior pueden estar a la vez certificados por otras autoridades de certificación, y así sucesivamente, hasta llegar a una autoridad de certificación que tiene un certificado autofirmado. Esta autoridad de certificación se denomina **autoridad de certificación raíz***, y las que dependen de ella son las **autoridades de certi-**

* En inglés, *root CA*.

ficación subordinadas. Si los usuarios de la infraestructura de clave pública confían en la autoridad de certificación raíz, este voto de confianza se extenderá por toda la jerarquía.



Los certificados se estructuran de manera jerárquica. Para verificar la autenticidad de un certificado un usuario puede comprobar la firma de la autoridad que lo emitió, que a la vez tendrá otro certificado expedido por otra autoridad de nivel superior. Y vamos subiendo por la jerarquía hasta llegar al nivel más alto, que estará ocupado por un certificado que cuente con la plena confianza de toda la comunidad. Las claves públicas de los certificados de más alto nivel incluso se publican en soporte papel para que cualquiera las pueda verificar.

El PKIX

El PKIX es la plataforma de infraestructura de clave pública más importante que utiliza el modelo jerárquico de confianza.

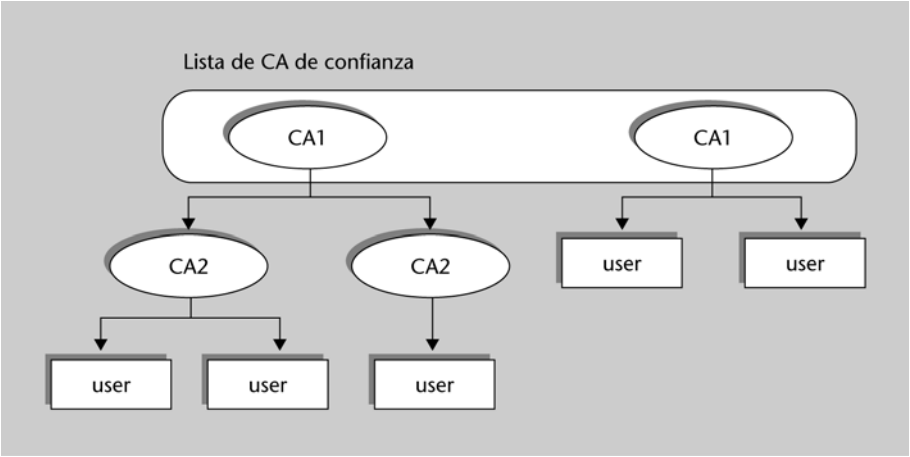
3.4. El modelo de navegación por lista de confianza

Hemos visto que el modelo jerárquico requiere que se establezca una única autoridad de certificación raíz en la que todos los usuarios depositen su confianza. Idealmente, esta autoridad de certificación debería ser de alcance mundial y la organización que la gestionara tendría que ser lo bastante reconocida y prestigiosa como para que cualquier usuario pudiera fiarse de ella.

En la práctica, no sucede así. Hay muchas y muy diferentes autoridades de certificación raíces, y cada una ofrece servicio a pequeñas comunidades de usuarios. Para conectar entre sí a estas islas de infraestructuras de clave pública han surgido formas híbridas del modelo jerárquico que permiten la interfuncionalidad entre diferentes grupos de usuarios controlados por una autoridad de certificación raíz.

La solución más común para interconectar infraestructuras de clave pública jerárquicas es el **modelo de navegación por lista de confianza**, también conocido como **modelo centrado en el usuario**, en el que cada aplicación final tiene una lista de las claves públicas de todas las autoridades de certificación en que confía.

Este modelo, implementado en la mayoría de navegadores web (Netscape, Mozilla, Explorer), permite al usuario un gran nivel de flexibilidad para añadir y borrar autoridades de certificación de su lista de confianza. El acceso a esta lista de confianza tendría que estar totalmente protegido contra modificaciones externas, ya que determina en qué autoridades de certificación confiamos y, por extensión, en qué otros usuarios (certificados por estas autoridades de certificación).



El principal problema de este sistema es que no hay ninguna diferencia entre una infraestructura de clave pública fuerte y una débil. Por ejemplo, la mayoría de usuarios de certificados no miran qué política tiene un certificado específico antes de confiar en él. Así, la mayoría de infraestructuras de clave pública comerciales incluidas en la lista de los navegadores no incluyen muchos de los controles que los diferentes gobiernos exigen a las autoridades de certificación.

Una ventaja de este modelo es que cada aplicación puede optar por confiar en un conjunto diferente de autoridades de certificación.

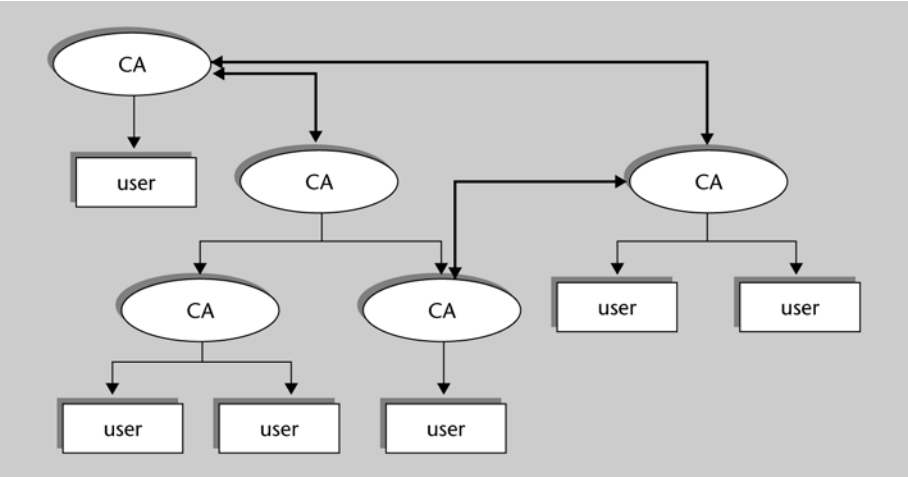
3.5. El modelo de certificados cruzados

En el modelo de certificados cruzados, las autoridades de certificación raíz de cada comunidad de usuarios emiten certificados para otras autoridades de certificación que tienen funciones y políticas equivalentes a las suyas. Como en el modelo jerárquico, cada usuario sólo confía en una autoridad de certificación raíz, pero en este modelo, la autoridad de certificación raíz es de ámbito local, y no central, como en el anterior. La interconexión entre diferentes islas de infraestructuras de clave pública se realiza por el certificado cruzado de una autoridad de certificación a otra.

Podéis ver el modelo jerárquico en el subapartado 3.3 de este módulo.

En un **certificado cruzado**, el titular y el emisor son autoridades de certificación diferentes. Este tipo de certificados se utiliza para que una autoridad de certificación pueda certificar la identidad de otra autoridad de certificación.

Los usuarios finales pueden validar los certificados de usuarios de otras comunidades con el certificado que ha emitido la autoridad de certificación externa en que confían. La estructura que el usuario final ve es la de una infraestructura de clave pública jerárquica, pero cada usuario tendrá una vista diferente.



El problema básico de este modelo es la dificultad de implementar aplicaciones cliente capaces de llevarlo a la práctica. Saber si hay una cadena de certificación que enlace un usuario X con un usuario Y requiere un esfuerzo de busca importante.

Otro inconveniente es el gran volumen de certificados con que se trabaja. Cada autoridad de certificación raíz tiene que emitir un certificado a todas las otras autoridades de certificación que quiere reconocer. Por ejemplo, para establecer un dominio cruzado bidireccional entre diferentes infraestructuras de clave pública haría falta:

a) Para la interconexión de tres infraestructuras de clave pública:

$$C_3^2 = \binom{3}{2} = 6 \text{ certificados creados}$$

b) Para la interconexión de cien infraestructuras de clave pública:

$$C_{100}^2 = \binom{100}{2} = \frac{100!}{98!} = 9.900 \text{ certificados cruzados}$$

El modelo de certificados cruzados permite organizar las autoridades de certificación en estructuras horizontales en vez de estructuras jerárquicas, idea similar a la que fundamenta el modelo distribuido.

3.6. El modelo Bridge-CA

La interconexión de diversas infraestructuras de clave pública por certificados cruzados es muy práctica porque no requiere el establecimiento de ninguna

autoridad de certificación raíz común, pero presenta el inconveniente de que es un modelo poco reescalable. Para resolver este inconveniente surgió el modelo Bridge-CA, que incorpora una autoridad externa que actuará de puente entre las infraestructuras de clave pública que se quieren unir. Todas las autoridades de certificación establecen una certificación cruzada con el punto central de referencia, el Bridge-CA, y de esta manera se crea una comunidad de confianza más amplia.

El US Federal Bridge-CA

El modelo de autoridad de certificación OS Federal Bridge es uno de los proyectos pioneros en este campo. Permitirá interconectar las autoridades de certificación raíz de los diferentes estados y universidades de Estados Unidos.

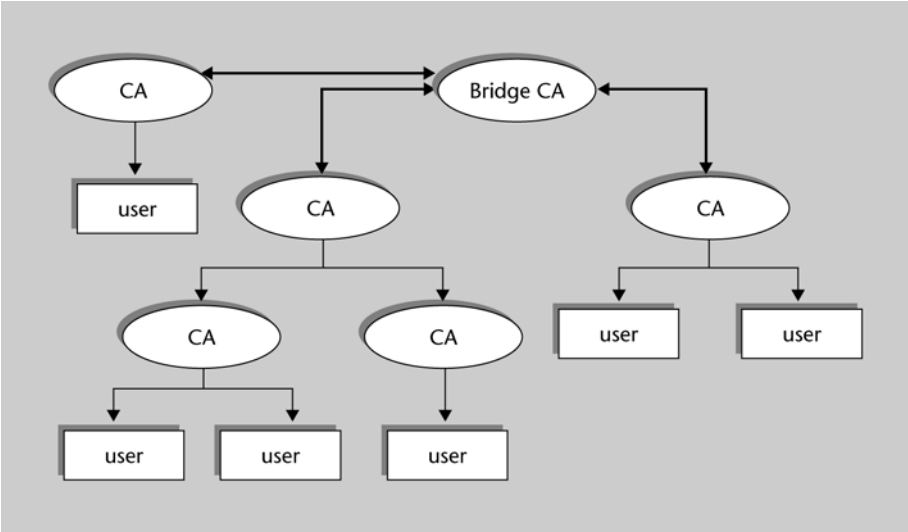
Para establecer un dominio cruzado bidireccional entre diferentes infraestructuras de clave pública hacen falta:

a) Para la interconexión de tres infraestructuras de clave pública:

$$2 \cdot 3 = 6 \text{ certificados cruzados}$$

b) Para la interconexión de cien infraestructuras de clave pública:

$$2 \cdot 100 = 200 \text{ certificados cruzados}$$



La estructura del modelo Bridge-CA es similar a la del modelo jerárquico. La diferencia básica radica en el hecho de que el Bridge-CA no es un punto central de confianza, sino una entidad que facilita la certificación cruzada entre todas las autoridades de certificación participantes. Un usuario que confíe en la autoridad de certificación de su jerarquía podrá validar certificados de otras jerarquías que estén unidas por el Bridge-CA, pero no se verá obligado a confiar en ellas explícitamente.

Podéis ver el modelo jerárquico en el subapartado 3.3 de este módulo.

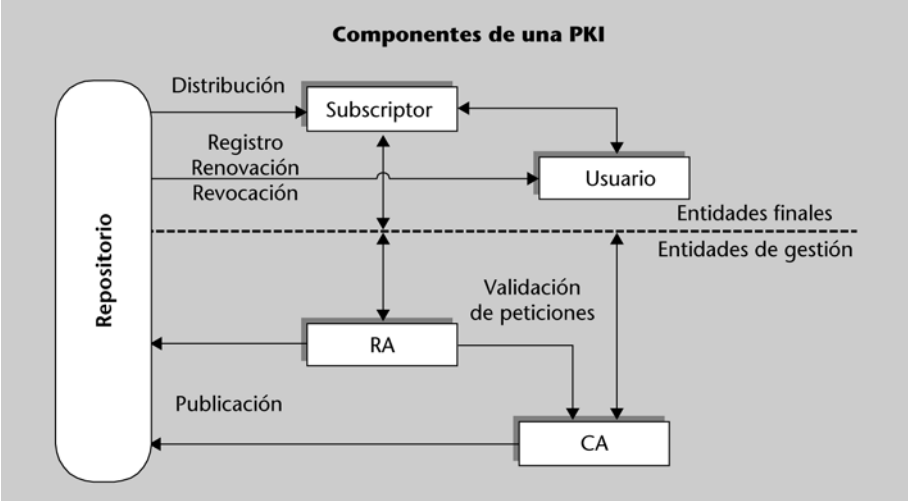
4. Ciclo de vida de claves y certificados digitales

En este apartado veremos el ciclo de vida de las claves y de los certificados digitales y, de esta manera, ilustraremos los procesos básicos comunes a todas las infraestructuras de clave pública. Partiremos del hecho de que ya existe el par de claves de una autoridad de certificación y nos basaremos en un modelo de confianza plano.

La figura (abajo) ilustra la relación entre los componentes básicos de una infraestructura de clave pública.

Los suscriptores generan un par de claves criptográficas y se registran en una autoridad de registro con el fin de poder solicitar un certificado digital. Si la autoridad de registro aprueba la petición de certificación, envía los datos a la autoridad de certificación para que emita el certificado. El certificado se publica en un repositorio al que todos los usuarios tienen acceso y, de esta manera, el suscriptor y el usuario pueden utilizar servicios de seguridad para comunicarse.

En los subapartados siguientes describiremos con más detalle los pasos que se siguen.



4.1. Generación del par de claves

La primera parte del ciclo de vida de un par de claves y del certificado digital relacionado es la generación de las claves. Según la **política de certificación*** (**CPS**) de la autoridad de certificación, tanto el suscriptor en su entorno local como las entidades externas pueden generar el par de claves criptográficas. En

* En inglés, *certification practice statement.*

este último caso, las claves se tienen que distribuir a los usuarios en ficheros cifrados o tarjetas criptográficas.

Los navegadores

Los navegadores disponen de herramientas para generar las parejas de claves criptográficas.

Las claves criptográficas se tienen que generar de acuerdo con las necesidades de las aplicaciones y los servicios que las utilicen. Por ejemplo, la vida útil de los datos que se desea proteger determina la longitud óptima de las claves. Por otra parte, el periodo de tiempo considerado de validez de una clave o certificado dependerá del tipo de operaciones que se quiera efectuar.

Con el fin de proporcionar un medio coherente de discernir qué claves son apropiadas para determinadas aplicaciones y cuáles no, los certificados contienen un campo con la información del tipo de certificado que representan. Básicamente hay dos tipos:

a) En los **certificados de cifrado** puede ser necesario contar con una copia de seguridad de la clave privada con el fin de poder recuperar la información en claro, si la clave de descifrado se pierde o se destruye. Esta práctica es frecuente en el entorno empresarial.

b) Los **certificados de firma** se utilizan para dar servicios de autenticación y de no repudio, y es importante que no haya ninguna copia de la clave privada correspondiente. Cualquier sistema de recuperación de claves para este tipo de certificados invalidaría sus propiedades.

4.2. Registro

El registro es el proceso por el cual un suscriptor o entidad final se da a conocer por primera vez a una autoridad de certificación (directamente o desde una autoridad de registro) con el fin de pedir un certificado.

El suscriptor tiene que proporcionar información en lo referente a su identidad, como su nombre, un identificador único, alguna prueba de identidad y otra información adicional que esté especificada en la CPS de la autoridad de certificación. Si él mismo ha generado el par de claves criptográficas, también tiene que dar la clave pública que pretende certificar. Toda esta información se estructura en una petición de certificado con formato PKCS#10. La verificación de los datos aportados en el proceso de registro se hace siguiendo las directrices establecidas en la CPS.

Veremos el formato PKCS#10 en el subapartado 6.2 de este módulo.

4.3. Certificación

La certificación es el proceso por el cual la autoridad de certificación emite un certificado, es decir, firma digitalmente la clave pública y los datos de un sujeto y, por lo tanto, da fe de esta correspondencia.

El suscriptor proporciona los datos que el certificado tiene que contener por medio de una petición de certificado PKCS#10. Estos datos han sido validados previamente, y se ha comprobado que el suscriptor tiene derecho a contar con un certificado de estas características. En el proceso de certificación también se define el periodo de validez del certificado y el uso que se podrá hacer del mismo.

Una vez generado el certificado, la autoridad de registro (o, si no hay ninguna, la autoridad de certificación) se encargará de hacerlo llegar al suscriptor y colgarlo en el directorio público adecuado.

4.4. Recuperación de claves

En algunas situaciones es importante poder recuperar una clave privada para acceder a información cifrada previamente. Por ejemplo, una empresa puede necesitar acceder a documentos previamente cifrados por un empleado que se haya dado de baja de la empresa.

La autoridad de certificación puede ofrecer sistemas de recuperación de claves mediante un sistema de copias de seguridad al que sólo puedan acceder los usuarios autorizados para obtener los datos. El sistema tiene que garantizar que las claves no se pueden comprometer.

4.5. Revocación de certificados

La revocación de certificados es el proceso por el cual una autoridad de certificación invalida un certificado antes de su periodo de expiración.

Existen diversas razones por las cuales un certificado puede ser revocado:

a) Compromiso de la clave privada asociada al certificado.

b) Cambio del suscriptor por otro proveedor de servicios de infraestructura de clave pública.

c) Cambio en la información que lleva el certificado (nombre del suscriptor, compañía a que pertenece, etc.).

d) Sustitución prematura de la pareja de claves de la autoridad de certificación.

En cualquiera de estas situaciones, el certificado deja de tener utilidad y el suscriptor tiene que notificárselo a la autoridad de certificación/registro. La notificación

se puede realizar en persona, por correo electrónico firmado o por cualquier otro canal autenticado. La clave privada correspondiente al certificado revocado se puede utilizar para autenticar una petición de revocación, pero por razones obvias, si la petición es la causa del compromiso de la clave, ésta no valdrá para la autenticación en la generación de un nuevo certificado.

El método predominante de distribución de la información de los certificados que han sido revocados son las listas de revocación de certificados (CRL).

Una **lista de revocación de certificados** (**CRL**) es una lista de los certificados que han sido revocados, firmada y fechada por la autoridad de certificación correspondiente.

La autoridad de certificación publica las listas de revocación de certificados en un servicio de directorio a intervalos regulares, y es responsabilidad del cliente obtener y comprobar la lista para determinar la validez de los certificados.

El ciclo de publicación de CRL

El ciclo de publicación de las CRL es crítico en una aplicación de autorización de usuarios para acceder a un registro con datos financieros.

El tiempo transcurrido entre la revocación de un certificado y la publicación de la lista de revocación de certificados que contiene la información es crítico, ya que existe una fisura de confianza en el certificado en un momento en que la clave privada puede estar comprometida y durante el cual se pueden tomar decisiones críticas.

4.6. Renovación de certificados

La renovación de un certificado digital puede estar condicionada principalmente por dos motivos:

Los periodos de renovación

El periodo apropiado para una clave RSA de 1.024 bits que se utilice para autenticación es de unos dos años.

- Pérdida de validez del certificado (caducidad del certificado, revocación por cambio de información del mismo, etc.).

- Pérdida de validez de las claves que certifica (revocación del certificado por compromiso de la clave privada asociada).

El certificado digital caduca cuando expira el periodo de validez. El **periodo de validez** de un certificado se establece de acuerdo con las restricciones de una serie de factores: el algoritmo de clave pública y la longitud de la clave, la utilización del certificado, los acuerdos financieros y de mantenimiento con la autoridad de certificación, los requisitos del ciclo de vida de los servicios y el *software* que el certificado necesita.

La renovación del certificado digital por pérdida de validez es simple: la petición de renovación se firma con la clave privada asociada al certificado que se desea renovar, la cual todavía tiene validez. De esta manera, el suscriptor se

puede autenticar ante la autoridad de certificación sin tener que volver a pasar toda la fase de registro. En este caso se puede certificar el mismo par de claves o generar otro par. Normalmente, en caso de caducidad del certificado se genera un nuevo par de claves, ya que se utiliza justamente la fecha de caducidad de los certificados para limitar el tiempo de vida de las claves y reducir los posibles ataques al criptosistema por la fuerza bruta.

En el caso de revocación del certificado por compromiso de la clave privada, la renovación del certificado no es tan transparente. La infraestructura de clave pública tiene que ofrecer medios en los que no intervenga el par de claves para anunciar que el certificado viejo deja de ser válido en favor de un nuevo certificado. A menudo se asocia una contraseña al proceso de generación del certificado con el fin de poder revocarlo. En este caso, la renovación del certificado pasa por la generación de un nuevo par de claves y la emisión de un nuevo certificado para las claves nuevas.

El peor caso se produce cuando la clave comprometida es la de la propia autoridad de certificación. Entonces hay que invalidar tanto el certificado de la autoridad de certificación como el de todos los suscriptores que dependían de él.

El hecho de que la autoridad de certificación tenga las claves comprometidas hace dudar de la confianza en cualquier mensaje electrónico de la autoridad de certificación que notifique este hecho, porque ¿quién firma el mensaje, si las claves no son válidas? Una manera de paliar los efectos de la caída de una autoridad de certificación es generar y distribuir previamente unas claves sustitutorias que se utilizarán en caso de compromiso de las nominales. Además, habrá que reemitir certificados a todos los suscriptores, de manera que estén firmados por una clave no revocada.

5. Estructuras de datos básicos de PKIX

En este apartado nos centraremos en las estructuras de datos definidas por la IETF en sus estándares del grupo PKIX.

5.1. Certificados X.509

El formato de certificados más ampliamente aceptado en infraestructura de clave pública viene definido por el ISO/IEC JTC1 SC21 y se conoce con la denominación X.509v3.

El X.509

El término X.509 viene de la especificación X.500 de servicios de directorio.

El **estándar internacional X.509v3**, que también se publica con el nombre de **ITU-T Recommendation X.509**, es importante por dos razones básicas: define el marco para la provisión de servicios de autenticación y un formato de certificado para las claves públicas.

El formato de los certificados X.509 ha evolucionado a partir de tres versiones en diferentes ediciones del estándar. La primera versión apareció en 1988. En 1993 se revisó y se obtuvo la versión 2, a la que se añadieron dos campos opcionales de identificación única. El uso del X.509 junto con el *Internet privacy enhanced mail* (PEM) hizo surgir la necesidad de dotar al certificado de más flexibilidad. En junio de 1996 apareció la versión 3 del certificado, que incluye la posibilidad de tener campos de ampliación.

El X.509 se diseñó como un marco general de autenticación para dar soporte a los servicios de directorio X.500. Este carácter abierto origina múltiples posibilidades para algunos de los campos del certificado (principalmente en las ampliaciones). Para regular el uso de las ampliaciones y su significado, aparecen los perfiles de certificados, que concretan y limitan las combinaciones de atributos que se pueden definir para cada tipo de certificado. El perfil de certificado más extendido es el definido por el grupo de trabajo PKIX de la IETF. Este grupo se encarga tanto de definir el perfil de certificados y listas de revocación como de desarrollar especificaciones y protocolos relacionados con el entorno de infraestructura de clave pública.

En los subapartados siguientes veremos la manera de designar las entidades participantes en la estructura de clave pública, y posteriormente comentaremos el formato general de un certificado y analizaremos cada uno de los campos.

5.1.1. Nombre de los componentes de un certificado

Los certificados X.509, al igual que el resto de componentes de la infraestructura de clave pública, se designan por un nombre distinguido.

Un **nombre distinguido** (**DN**) es un conjunto de atributos con valores asociados. Este tipo de estructura permite que cada organización decida qué atributos son interesantes para identificar a las entidades dentro de su empresa.

Atributos más usuales de los nombres distinguidos	
CN:	Nombre común
OU:	Unidad organizativa
O:	Organización
C:	País
L:	Localidad
S:	Estado
STREET:	Calle
T:	Título de la persona

Los atributos más habituales se muestran en la tabla del margen. Para algunos usos del certificado, el sistema de nombres por nombres distinguidos puede resultar insuficiente. En estos casos se utilizan unos campos especiales del certificado denominados nombres alternativos que permiten especificar la dirección de correo electrónico, la dirección IP de una web, el DNS, el nombre EDI, etc.

Campos especiales

Las direcciones de correo electrónico se especifican en el campo *Nombre alternativo del titular* (*subjectAltName*) siguiendo el formato rfc822Name definido en [RFC822].

5.1.2. Estructura de un certificado

En un certificado hay tres tipos principales de campos:

a) Básicos: campos que aportan información sobre la autoridad de certificación que ha emitido el certificado, la entidad/suscriptor a que pertenece la clave pública, la propia clave pública, y el periodo de validez y la identificación del certificado.

b) Necesarios para la firma: campos que utilizará quien reciba el certificado para comprobar que el documento ha sido firmado correctamente.

c) Ampliaciones: campos que han aparecido para cubrir las nuevas necesidades de atributos en un certificado. Algunos de los posibles usos de los campos de ampliación son la mejora de la gestión de la herencia de certificación, las políticas de seguridad o los datos sobre el usuario y su clave. Las ampliaciones pueden estar definidas en estándares o por una organización particular que haga uso de certificados.

Lenguaje de los certificados

Los certificados X.509, al igual que las otras estructuras definidas para PKIX, están especificadas en lenguaje ASN.1 (*Abstract Syntax Notation One*).

Los campos básicos de un certificado X.509v3 son los siguientes:

Campo	Descripción
version	Identifica el número de versión del certificado: v1, v2, v3.
serialNumber	Entero asignado por la autoridad de certificación. Identifica unívocamente el certificado de una autoridad de certificación.
signature	Tiene que coincidir con el algoritmo de firma de los campos necesarios para la firma.
issuer	Nombre distinguido de la autoridad de certificación que ha firmado y emitido el certificado.

Campo	Descripción
validity	Dos campos que identifican el periodo en que el certificado es válido.
subject	Nombre distinguido que identifica al propietario del certificado.
subjectPublicKeyInfo	Campo con información sobre la clave pública por la cual se emite el certificado.
issuerUniqueID	Campo opcional y poco utilizado que permite la reutilización de nombres de autoridad de certificación.
subjectUniqueID	Campo opcional que permite la reutilización de nombres de titulares.
extensions	Campo opcional que contiene los campos de ampliaciones del certificado.

Los campos necesarios para la firma son:

Campo	Descripción
signatureAlgorithm	Contiene el identificador del algoritmo de *hash* (MD2, MD5, SHA-1) y el algoritmo de clave pública (RSA, DSA) que se ha utilizado en la firma del certificado digital.
signature	Contiene el valor de la firma de los campos básicos y las ampliaciones del certificado con el algoritmo de firma descrito en el campo anterior.

Identificación de los certificados

El número de serie del certificado, junto con el nombre de la autoridad de certificación, tiene que servir para identificar unívocamente un certificado en cualquier contexto.

Todas las ampliaciones aparecen dentro del campo *extensions* del certificado, y cada una está formada por tres subcampos:

Campo de una ampliación	Descripción
extnID	Identificador unívoco de la ampliación que representa.
critical	Valor lógico que marca si la ampliación es crítica o no. Las aplicaciones pueden ignorar las ampliaciones no críticas en caso de que no las reconozcan. La presencia de una ampliación crítica no reconocida provoca el rechazo del certificado.
extnValue	Contiene el valor de la ampliación.

Cada ampliación sólo puede aparecer una vez en cada certificado.

A continuación, comentamos algunas de las ampliaciones más destacadas de los certificados e indicamos el contenido del subcampo *extnValue*:

Tipo de ampliación	Descripción
subjectAltName	Nombre alternativo del titular (dirección de correo electrónico, etc.).
issuerAltName	Nombre alternativo de la autoridad de certificación emisora.
keyUsage	Define y limita los posibles usos de la clave certificada.
extKeyUsageSyntax	Complemento del *keyUsage* y el *basicConstraints* para delimitar los posibles usos de la clave dentro de las necesidades particulares de una organización.
basicConstraints	Distingue si el certificado es de una entidad final (falso) o de una autoridad de certificación (verdadero). Permite limitar la longitud de cadena de certificación.

Tipo de ampliación	Descripción
nameConstraints	Limita el espacio de nombres que pueden utilizar los certificados siguientes de la cadena.
certificatePolicies	Información sobre la CPS de la autoridad de certificación.
policyMappings	Identifica las políticas de una autoridad de certificación con las de otra.
policyConstraints	Limita la verificación del camino de certificación según unas políticas.
inhibitAnyPolicy	Limita el uso de la política *anyPolicy* a un número de certificados de la cadena.
privateKeyUsagePeriod	Especifica periodos de validez para la clave privada que son diferentes de los del certificado.
authorityKeyIdentifier	Identificador único de la clave pública necesario para validar el certificado. Útil cuando una autoridad de certificación tiene diversas claves de firma.
subjectKeyIdentifier	Identificador único de certificados que hace el *hash* sobre la clave pública o genera un número único.
crlDistributionPoints	Indica la localización de las listas de revocación de certificados relacionadas con el certificado a partir de unos puntos de distribución.
fraeshestCRL	Indica dónde encontrar información de las listas de revocación de certificados más recientes.
authorityInformationAccess	Ampliación privada definida por PKIX que indica cómo acceder a información y servicios que la autoridad de certificación emisora proporciona.

La ampliación *keyUsage*, que define los propósitos del certificado, es una de las más importantes. El PKIX recomienda que en caso de utilizarla, esta ampliación se marque como crítica. Los valores que puede tomar son una combinación de las opciones siguientes:

Valores de *keyUsage*	Descripción
digitalSignature	El certificado puede verificar firmas de documentos.
nonRepudiation	Se pueden verificar firmas con la propiedad de no repudio.
keyEncipherment	La clave puede ser utilizada para transportar claves.
dataEncipherment	La clave se puede utilizar para cifrar datos de usuario que no son claves.
keyAgreement	Se utiliza la clave para algoritmos de establecimiento de una clave de sesión.
keyCertSign	La clave se utiliza para validar la firma de otros certificados (utilizado en autoridades de certificación).
encipherOnly	Se utiliza conjuntamente con *keyAgreement* para limitar el uso de la clave a cifrar datos durante el proceso de establecimiento de la clave.
decipherOnly	Se utiliza conjuntamente con *keyAgreement* para limitar el uso de la clave a descifrar datos durante el proceso de establecimiento de la misma.
subjectUniqueID	Campo opcional que permite la reutilización de nombres de titulares.

La estructura de datos del certificado indicado es la siguiente:

```
Certificate: SEQUENCE (1006)
   toBeSigned SEQUENCE (726)
     version [0] INTEGER (1) 0x02 (2)
     serialNumber INTEGER (1) 0x24 (36)
     signature SEQUENCE (13)
       algorithm OBJECT IDENTIFIER (9) pkcs1-sha1WithRsaSignature
       parameters TYPE (2) NULL (0)
     issuer SEQUENCE OF (71) OU=Coyote Subordinate CASA (MIR), O=UOC, C=ES
     validity SEQUENCE (30)
       notBefore UTCTime (13) "030929144413Z"
       notAfter UTCTime (13) "040929144251Z"
     subject SEQUENCE OF (65) CN=user, OU=Criptografía, O=UOC, C=ES
     subjectPublicKeyInfo SEQUENCE (159)
       algorithm SEQUENCE (13)
         algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
         parameters TYPE (2) NULL (0)
       subjectPublicKey BIT STRING (141) Encapsulates TYPE (140) with
           rSAPublicKey SEQUENCE (137)
             smodulus INTEGER (129) 0x00D0A60F7EF71D4FCB342982E2333C575392AFF9B..
       exponent INTEGER (3) 0x010001 (65537)
     issuerUId [1] IMPLICIT BIT STRING OPTIONAL NOT PRESENT
     subjectUId [2] IMPLICIT BIT STRING OPTIONAL NOT PRESENT
     extensions [3] SEQUENCE OF (361)
       extension SEQUENCE (14)
         extnId OBJECT IDENTIFIER (3) id-ce-keyUsage
         critical BOOLEAN (1) TRUE
         extnValue OCTET STRING (4) Encapsulates TYPE (4) with
           BIT STRING (2) 03 F8
       extension SEQUENCE (29)
         extnId OBJECT IDENTIFIER (3) id-ce-extKeyUsage
         critical BOOLEAN (1) FALSE DEFAULT
         extnValue OCTET STRING (22) Encapsulates TYPE (22) with
           SEQUENCE OF (20)
             OBJECT IDENTIFIER (8) id-kp-clientAuth
             OBJECT IDENTIFIER (8) id-kp-emailProtection
       extension SEQUENCE (17)
                              ...
signatureAlgorithm SEQUENCE (13)
   algorithm OBJECT IDENTIFIER (9) pkcs1-sha1WithRsaSignature
   parameters TYPE (2) with NULL (0)
signature BIT STRING (257) 00 73E9098140B1DD4BBE66F0487AD1EDC85E4B ...
```

5.2. Listas de revocación (CRL)

Las listas de revocación de certificados sirven para que una autoridad de certificación pueda distribuir información sobre los certificados que ha emitido y han sido revocados.

Una **lista de revocación de certificados** (CRL) es una estructura de datos firmada por la autoridad de certificación emisora que contiene los datos necesarios para determinar el estado de un certificado.

En la figura vemos el tipo de información que muestra MS Internet Explorer de un certificado.

En los certificados X.509 se mantiene información sobre dónde se puede obtener la lista de revocación de certificados de la autoridad de certificación emisora, en concreto en la ampliación *crlDistributionPoint*.

En el mismo estándar X.509 se especifica un formato para las listas de revocación de certificados. La primera versión del estándar se publicó en 1988. Al igual que pasó con los certificados (ved el subapartado 5.1.2), más tarde, en la versión 2, el estándar se modificó con el objetivo de introducir los campos de ampliación.

En los subapartados siguientes veremos los tipos básicos de listas de revocación de certificados, cómo se distribuyen y cuál es el formato de su estructura de datos.

5.2.1. Tipo de listas de revocación de certificados

Hay tres tipos básicos de listas de revocación de certificados: CRL simple, delta CRL y CRL indirecta.

Una **lista de revocación de certificados simple** es un fichero secuencial que contiene una lista de los certificados revocados.

Tasas de revocación

La tasa de revocación de certificados se sitúa en un 10% en un año.

Este fichero crece a medida que el tiempo pasa y se convierte en intratable e ineficiente. Por este motivo, han surgido otras formas de listas de revocación de certificados que proporcionan a la aplicación final sólo el conjunto de información de revocación que puede ser relevante.

Un **delta CRL** es una lista de revocación de certificados que complementa la información contenida en otra lista de revocación de certificados base.

El delta CRL está diseñado para reducir las dimensiones de la lista que se transmite a la entidad final. Inicialmente, y después a intervalos regulares planificados, se publica una lista de revocación de certificados completa, la CRL base, que contiene información de todas las revocaciones conocidas. A partir de este punto se emiten una serie de listas de actualización que contienen los cambios

que se han producido desde la última publicación de un delta CRL o CRL base. Cada actualización se distribuye como una lista de revocación de certificados, y la autoridad de certificación la emite y la firma de la misma manera que una lista de revocación de certificados.

Una **CRL indirecta** es una lista de revocación de certificados que contiene información de revocación de diversas autoridades de certificación.

Estos tipos de listas de revocación de certificados se utilizan para reducir el número de listas de revocación de certificados que hay que obtener para poder comprobar la validez de un certificado. A fin de que una entidad final pueda utilizar una CRL indirecta, se tiene que confiar en ella como se confía en la autoridad de certificación que emite el certificado.

5.2.2. Puntos de distribución de listas de revocación de certificados

Los puntos de distribución de listas de revocación de certificados sirven para fragmentar la lista de revocación de certificados global de una autoridad de certificación en varias menores. Cada subconjunto engloba un número determinado de certificados y/o un subconjunto de razones de revocación. Los certificados X.509 contienen un apuntador dirigido al punto de distribución correspondiente.

Con el uso de puntos de distribución de listas de revocación de certificados disminuye el consumo de recursos de red (ancho de banda y tiempo de transmisión), debido a la menor dimensión de cada lista de revocación de certificados de un punto de distribución respecto a una única lista que incluyera las revocaciones de todos los certificados.

5.2.3. Formato de las listas de revocación de certificados

En una lista de revocación de certificados podemos encontrar cuatro tipos principales de campos:

a) Básicos: campos que aportan información sobre la autoridad de certificación que ha emitido la lista de revocación de certificados, la fecha en que se ha actualizado, la fecha de la siguiente actualización y los certificados que han sido revocados.

b) Necesarios para la firma: campos que utilizará quien reciba la lista de revocación de certificados para comprobar que el documento está firmado correctamente.

c) Ampliaciones de la CRL: ampliaciones que proporcionan métodos para asociar atributos adicionales a la lista de revocación de certificados. Pueden dividirse en estándares o por organizaciones particulares.

d) Ampliaciones de la entrada de la CRL: ampliaciones que proporcionan información adicional sobre las entradas de la lista de revocación de certificados.

Los campos básicos de una lista de revocación de certificados son los siguientes:

Campo	Descripción
version	Identifica el número de versión de la lista de revocación de certificados: v1, v2.
signature	Tiene que coincidir con el algoritmo de firma de los campos necesarios para la firma.
issuer	Nombre distinguido de la autoridad de certificación que ha firmado y emitido la lista de revocación de certificados.
thisUpdate	Fecha de emisión de la lista de revocación de certificados.
nextUpdate	Fecha posterior en que se emitirá la siguiente lista de revocación de certificados.
revokedCertificates	Lista con los certificados revocados.
crlExtensions	Campo opcional que contiene los campos de ampliaciones de la lista con los certificados revocados.

Los campos necesarios para la firma son:

Campo	Descripción
signatureAlgorithm	Contiene el identificador del algoritmo utilizado para firmar la lista de revocación de certificados. Es un campo análogo al que encontramos en un certificado.
signature	Contiene la firma de la lista de revocación de certificados según el algoritmo establecido.

Las ampliaciones, que tienen el mismo formato que las definidas para el estándar X.509, son de los tipos siguientes:

Podéis ver el estándar X.509 en el subapartado 5.1 de este módulo.

Tipo de ampliación	Descripción
authorityKeyIdentifier	Identifica la clave de la autoridad de certificación con la cual se puede validar la lista de revocación de certificados.
issuerAltName	Nombre alternativo de la autoridad de certificación emisora.
crlNumber	Permite determinar cuándo una lista de revocación de certificados actualiza a otra.
deltaCrlIndicator	Indica que se trabaja con lista delta de revocación de certificados e identifica la lista de revocación de certificados base.
issuingDistributionPoint	Indica dónde se encuentra la lista de revocación de certificados y el tipo de certificados que habrá en la lista (de autoridades de certificación, de suscriptores, revocados por un determinado motivo, etc.).

Las ampliaciones de entrada en la lista de revocación de certificados aparecen en el subcampo *crlEntryExtensions* de cada entrada de la lista de certificados revocados contenidos en el campo *revokedCertificates*. Pueden tener los valores siguientes:

Ampliación de entrada	Descripción
reasonCode	Identifica la causa por la cual se ha revocado un certificado.
holdInstructionCode	Indica la acción que hay que llevar a cabo cuando se encuentra un certificado que ha sido suspendido.
invalidityDate	Contiene la fecha en que se sabe o se sospecha que el certificado se invalidó o la clave fue comprometida. Puede ser anterior a la fecha de revocación.
certificateIssuer	Identifica al emisor del certificado en una entrada de una lista de revocación de certificados indirecta.
issuingDistributionPoint	Indica dónde se encuentra la lista de revocación de certificados y el tipo de certificados que habrá en la lista (de autoridades de certificación, de suscriptores, revocados por un determinado motivo, etc.).

6. Formato y procedimientos para generar mensajes: PKCS

Las **normas PKCS**, o ***public-key cryptography standards***, son un conjunto de especificaciones desarrolladas en los laboratorios RSA con el objetivo de establecer una norma común a toda la industria sobre los formatos de los datos utilizados en criptografía de clave pública. En realidad no son estándares, sino una aproximación creada *ad hoc* por una única empresa con el soporte y la colaboración de otros fabricantes (Apple, Microsoft, DEC, Lotus, Sun, MIT, etc.).

La primera publicación de PKCS se realizó en 1991 y, desde entonces, desarrolladores de todo el mundo los han referenciado e implementado sobradamente. Algunas contribuciones de las series PKCS han sido adoptadas como estándares formales y *de facto*, como PKIX, SET; S/MIME y SSL. Actualmente hay diez normas PKCS: PKCS #1, #3, #5, #7, #8, #9, #10, #11, #12 y #15. Los PKCS #13 y #14 todavía no han sido publicados, los PKCS #2 y #4 han sido incorporados al PKCS #1, y el PKCS #6 se ha retirado en favor de la versión 3 del estándar X.509.

Los PKCS incluyen normas que definen tanto la sintaxis de una estructura de datos independientemente del algoritmo como formatos específicos para un algoritmo concreto.

PKCS	Descripción
1	Define los mecanismos para cifrar y firmar datos con el criptosistema de clave pública RSA. También define una sintaxis, idéntica a X.509, para las claves públicas y privadas.
3	Define un protocolo Diffie-Hellman para el intercambio de claves.
5	Describe un método para cifrar una cadena de texto con una clave secreta derivada de una frase de paso. Su objetivo principal es permitir la transmisión cifrada de claves privadas entre ordenadores, como se describe en el PKCS#8, aunque puede ser utilizada para cifrar mensajes. Utiliza MD2 o MD5 para producir una clave a partir de una frase de paso. Esta clave se utiliza para cifrar con DES (en modo CBC) el mensaje en cuestión.
7	Define una sintaxis general para los mensajes que incluyen mejoras criptográficas, como firmas digitales o cifrado.
8	Describe el formato de la información de la clave privada. Esta información incluye una clave privada para algún algoritmo de clave pública y, opcionalmente, un conjunto de atributos.
9	Define los tipos de atributos seleccionados para utilizar en otros estándares PKCS.
10	Describe una sintaxis para peticiones de certificación.
11	Define una interfaz de programación independiente de la tecnología, llamada Cryptoki, para dispositivos criptográficos como tarjetas inteligentes y tarjetas PCMCIA.
12	Especifica un formato portátil para almacenar y transportar claves privadas de usuario, certificados, secretos diversos, etc.
13	Definirá los mecanismos para cifrar y firmar datos utilizando criptografía basada en curvas elípticas.
14	Cubre la generación de números pseudoaleatorios.
15	Complemento del PKCS #11 que define el formato de las credenciales criptográficas almacenadas en dispositivos criptográficos.

6.1. El PKCS#7

El PKCS#7 describe una sintaxis general para datos que pueden llevar incorporada criptografía, como firmas digitales o sobres digitales. La sintaxis admite recursividad, de manera que, por ejemplo, un sobre digital puede estar dentro de otro y una entidad puede firmar datos que anteriormente han sido introducidos en un sobre digital.

Los diferentes tipos de objetos PKCS#7 que nos encontramos son*: data*, *signedData*, *envelopedData*, *signedAndEnvelopedData*, *digestedData* y *encryptedData*. A continuación los veremos con más detalle.

6.1.1. Los *data*

Se trata de información en claro que no ha sufrido ningún tipo de manipulación criptográfica.

6.1.2. Los *signedData*

El objeto *signedData* engloba información a la que se ha aplicado una firma digital. La información que lleva esta estructura de datos es la siguiente:

signedData	Descripción
version	Versión del estándar que se utiliza.
digestAlgorithms	Conjunto de identificadores de los algoritmos de *hash* que se han utilizado para crear las firmas digitales.
contentInfo	Objeto que queremos firmar.
certificates	Cadena de certificación desde los certificados utilizados por la firma hasta los certificados de las autoridades de certificación de nivel más alto.
crls	Conjunto de listas de certificados revocados.
signerInfo	Conjunto de información sobre los firmantes.
version	Versión del estándar que se utiliza.
issuerAndSerialNumber	Identifica el certificado del firmante a partir del nombre de la autoridad de certificación que lo ha emitido y el número de serie del certificado.
digestAlgorithm	Algoritmo utilizado para resumir el contenido del mensaje.
authenticateAttributes	Campo opcional que contiene un conjunto de atributos que han sido firmados. Útil para incluir la fecha en que se genera la firma.
unauthenticatedAttributes	Conjunto de atributos que no han sido firmados.

El contenido *signedData* de PKCS#7

El tipo de contenido *signedData* de PKCS#7 proporciona un caso de uso degenerado en el que no hay ningún signatario del contenido. Este formato se utiliza para distribuir cadenas de certificados y listas de revocación.

El objeto *signedData* puede llevar firmas generadas por diversos usuarios. Para cada firmante habrá que generar el resumen de los datos para firmar con el al-

goritmo que éste especifique, y después éstos se tendrán que firmar con la clave privada del usuario que emite la firma. Como se ve en la tabla, el campo *signerInfo* incluye otros campos que proporcionan información más detallada sobre cómo se genera la firma para cada usuario.

Para crear una estructura de firma digital se tienen que seguir los pasos siguientes:

1) Para cada firmante, se calcula el resumen de los datos con el algoritmo indicado por el usuario que firma. Si el firmante define atributos autenticados, el resumen será el de estos atributos. Si no, sólo se tendrá en cuenta el mensaje contenido en *contentInfo*.

2) Para cada firmante, se firma el resumen calculado en el paso anterior utilizando la clave privada.

3) Para cada firmante, la firma digital y otra información específica del remitente se agrupan en el campo denominado *signerInfo*.

4) Los algoritmos que se han utilizado para resumir el mensaje de todos los remitentes, los diferentes campos *signerInfo* y el mensaje original se agrupan en un campo *signedData*.

Los pasos que tendrá que seguir el destinatario son:

a) Verificar los resúmenes de cada firmante con la clave pública de éste.

b) Observar si los resúmenes obtenidos en el paso anterior coinciden con los que se obtienen al hacer los cálculos con los datos que se autentican.

Una firma digital puede representarse con un PKCS#7 que contenga el objeto *signedData*.

Ejemplo de la estructura de un PKCS#7 *signedData*

```
PKCS#7: SEQUENCE (2417)
   contentType OBJECT IDENTIFIER (9) pkcs7-signedData
   content [0] TYPE (2402) with SEQUENCE (2398)
     version INTEGER (1) 0x01 (1)
     digestAlgorithms SET OF (11) SEQUENCE (9)
       algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
       parameters TYPE (2) with NULL (0)
     contentInfo SEQUENCE (26)
       contentType OBJECT IDENTIFIER (9) pkcs7-data
       content [0] TYPE (13) with OCTET STRING (11) "Hello World"
     certificates [0] IMPLICIT SET OF (2074) SEQUENCE (997)
       toBeSigned SEQUENCE (717)
         version [0] INTEGER (1) 0x02 (2)
         serialNumber INTEGER (1) 0x13 (19)
```

```
crls [1] IMPLICIT SET OF OPTIONAL NOT PRESENT
signerInfos SET OF (272) SEQUENCE (268)
  version INTEGER (1) 0x01 (1)
  issuerAndSerialNumber SEQUENCE (76)
    issuer SEQUENCE OF (71) OU=Coyote Subordinate CA (MIR), O=UOC, C=ES
    serialNumber INTEGER (1) 0x13 (19)
  digestAlgorithm SEQUENCE (9)
    algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
    parameters TYPE (2) with NULL (0)
  authenticatedAttributes [0] IMPLICIT SET OF (93)
    SEQUENCE (24)
      attrType OBJECT IDENTIFIER (9) pkcs9-contentType
      attrValue TYPE (13) with SET OF (11) OBJECT IDENTIFIER (9) pkcs7-data
    SEQUENCE (28)
      attrType OBJECT IDENTIFIER (9) pkcs9-signingTime
      attrValue TYPE (17) with SET OF (15) UTCTime (13) "030929150419Z"
    SEQUENCE (35)
      attrType OBJECT IDENTIFIER (9) pkcs9-messageDigest
      attrValue TYPE (24) with SET OF (22) OCTET STRING (20) 0A4D55A8D ...
  digestEncryptionAlgorithm SEQUENCE (13)
    algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
    parameters TYPE (2) with NULL (0)
  encryptedDigest OCTET STRING (64) 63103EC00F6001BC760C7996 ...
  unauthenticatedAttributes [1] IMPLICIT SET OF OPTIONAL NOT PRESENT
```

6.1.3. Los *envelopedData*

El objeto *envelopedData* consiste en un sobre digital. Los datos están cifrados con una clave simétrica de sesión y esta clave está, a su vez, cifrada con las claves públicas de los destinatarios (el sobre digital puede ir dirigido a unos o más usuarios). Los datos que se cifran pueden ser de cualquier tipo; normalmente son objetos *data*, *digestedData* o *signedData*.

La información que lleva una estructura de datos *envelopedData* es la siguiente:

envelopedData	**Descripción**
version	Versión del estándar que se utiliza.
recipientInfo	Conjunto de la información sobre los destinatarios. Tiene que haber como mínimo uno.
Version	Versión de la sintaxis que se utiliza.
issuerAndSerialNumber	Identifica el certificado del destinatario con el nombre de la autoridad de certificación que lo ha emitido y el número de serie del certificado.
keyEncryptionAlgorithm	Algoritmo de clave pública utilizado para cifrar la clave de sesión.
encryptedKey	Clave de sesión cifrada.
encryptedContentInfo	Contenido del mensaje cifrado e información sobre el cifrado: tipo de contenido del mensaje original y algoritmo de cifrado utilizado.

Como se ve, la información para cada destinatario se agrupa en una estructura de tipo *recipientInfo*.

Para crear un sobre digital se tienen que seguir los pasos siguientes:

1) Se genera de forma aleatoria una clave de sesión para un algoritmo de cifrado simétrico concreto.

2) El contenido se cifra con esta clave de sesión.

3) Para cada destinatario, la clave de sesión se cifra con la clave pública del destinatario.

4) Para cada destinatario, la clave de sesión cifrada y alguna información propia del destinatario se agrupan en un valor del tipo *recipientInfo*.

5) Los valores del tipo *recipientInfo* (uno para cada destinatario) se juntan con el contenido del mensaje original para crear el sobre digital.

El proceso que el receptor ha de seguir es:

a) Descifrar la clave de sesión que va dirigida al sobre digital y abrirlo.

b) Descifrar el contenido del mensaje con la clave de sesión obtenida en el paso anterior.

El cifrado de una información con la técnica del sobre digital se puede representar con un PKCS#7 que contenga el objeto *envelopedData*.

Ejemplo de la estructura de un PKCS#7 envelopedData

```
PKCS#7: SEQUENCE (308)
   contentType OBJECT IDENTIFIER (9) pkcs7-envelopedData
   content [0] TYPE (293) with SEQUENCE (289)
     version INTEGER (1) 0x00 (0)
     recipientInfos SET OF (230) SEQUENCE (227)
       version INTEGER (1) 0x00 (0)
       issuerAndSerialNumber SEQUENCE (76)
         issuer SEQUENCE OF (71) OU=Coyote Subordinate CA (MIR), O=UOC, C=ES
         serialNumber INTEGER (1) 0x24 (36)
       keyEncryptionAlgorithm SEQUENCE (13)
         algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
         parameters TYPE (2) with NULL (0)
       encryptedKey OCTET STRING (128) 1892CA739A7DE62DCC ...
     encryptedContentInfo SEQUENCE (51)
       contentType OBJECT IDENTIFIER (9) pkcs7-data
       contentEncryptionAlgorithm SEQUENCE (20)
         algorithm OBJECT IDENTIFIER (8) rsadsi-encryptionAlgorithm-des-ede3-cbc
         parameters TYPE (10) with OCTET STRING (8) A76D20B015138DF0
       encryptedContent [0] IMPLICIT OCTETO STRING (16) 7269AFD6A5DD ...
```

6.1.4. Los *signedAndEnvelopedData*

El objeto *signedAndEnvelopedData* consiste en un sobre digital autenticado, es decir, la información está cifrada con la técnica del sobre digital y, además, está firmada. Los datos pueden estar firmados por un número cualquiera de firmantes en paralelo y, al mismo tiempo, todo el contenido puede estar cifrado por unos o más destinatarios.

La información que lleva una estructura de datos *signedAndEnvelopedData* es la siguiente:

signedAndEnvelopedData	Descripción
version	Versión de la sintaxis que se utiliza.
recipientInfos	Conjunto de la información sobre los destinatarios. Tiene que haber como mínimo uno.
digestAlgorithms	Conjunto de identificadores de los algoritmos de *hash* que se han utilizado para crear las firmas digitales.
encryptedContentInfo	Contenido del mensaje cifrado e información sobre el cifrado: tipo de contenido del mensaje original y algoritmo de cifrado utilizado.
certificates	Cadena de certificación desde los certificados utilizados por la firma hasta los certificados de las autoridades de certificación de nivel más alto.
crls	Conjunto de listas de certificados revocados.
signerInfo	Conjunto de información sobre los firmantes.

Para crear un sobre digital autenticado, hay que seguir los pasos siguientes:

1) Se genera una clave de sesión de manera aleatoria con algún algoritmo de cifrado simétrico.

2) El contenido del mensaje se cifra con esta clave de sesión.

3) Para cada destinatario, la clave de sesión se cifra con la clave pública del destinatario.

4) Para cada destinatario, la clave de sesión cifrada y alguna información propia del destinatario se agrupan en un valor del tipo *recipientInfo*.

5) Para cada firmante, se calcula el resumen del mensaje con el algoritmo que el firmante desee; si dos firmantes utilizan el mismo algoritmo, sólo se calcula una vez.

6) Para cada firmante, el resumen y otra información adicional se cifra con la clave privada propia de cada firmante, y después se cifra de nuevo con la clave de sesión establecida.

7) Para cada firmante, el resumen doblemente cifrado se añade a otra información adicional del firmante en una estructura de datos *signerInfo*.

8) Los resúmenes del mensaje de todos los firmantes, las estructuras *signerInfo* y *recipientInfo* de todos los destinatarios y firmantes se juntan con el mensaje cifrado en una estructura *signedAndEnvelopedData*.

El proceso que el receptor tiene que seguir para procesar la información es:

a) Descifrar la clave de sesión con la clave privada propia.

b) Descifrar el contenido del mensaje, es decir, la información de los firmantes y el propio mensaje con la clave de sesión.

c) Descifrar los resúmenes, primero con la clave de sesión y después con la clave pública de cada firmante.

d) Confrontar los resúmenes obtenidos con los calculados por el propio destinatario utilizando el mismo algoritmo de *hash*.

e) Leer el mensaje descifrado.

6.1.5. Los *digestedData*

El objeto *digestedData* consiste en cualquier tipo de contenido y su resumen correspondiente. La finalidad de este objeto es garantizar la integridad del contenido y ser la entrada de procesos de creación de objetos de tipo *envelopedData*.

La información que lleva una estructura de datos *digestedData* es la siguiente:

digestedData	Descripción
version	Versión de la sintaxis que se está utilizando.
digestAlgorithmId	Identificador del algoritmo de *hash*.
contentInfo	Contenido que se va a resumir.
digest	Resumen del contenido.

El proceso de creación es el siguiente:

1) Se calcula un resumen del contenido con el algoritmo que se desee.
2) El contenido y el resumen se agrupan en una estructura de tipo *digestedData*.

6.1.6. Los *encryptedData*

Este objeto consiste en cualquier tipo de datos cifrados. No contiene ninguna referencia al destinatario (puede ser que no haya) ni ninguna referencia a la clave que se ha utilizado para cifrar, que se asume que se gestiona desde otros medios. La finalidad es cifrar contenido localmente; por eso, la clave de cifrado puede ser una simple contraseña.

La información que lleva una estructura de datos *encryptedData* es la siguiente:

encryptedData	Descripción
version	Versión de la sintaxis que se utiliza.
encryptedContentInfo	Contenido cifrado.

6.2. El PKCS#10

El PKCS#10 describe el formato de una petición de certificado. Una **petición de certificado** consiste en un nombre, una clave pública y un conjunto opcional de atributos referentes al suscriptor. Uno de los atributos más utilizados es el de una contraseña con la que el suscriptor podrá revocar el certificado.

Una petición está dividida en tres partes: la información de la petición, el identificador del algoritmo de firma y la firma digital de la información.

La información de la petición se estructura en un campo *certificationRequestInfo*:

certificationRequestInfo	Descripción
version	Versión de la sintaxis que se utiliza.
subject	Nombre distinguido del propietario de la clave pública.
subjectPublicKeyInfo	Información de la clave pública.
attributes	Conjunto de atributos con información adicional del suscriptor.

El proceso que sigue la autoridad de certificación al llegar una petición consiste en verificar la firma y, en caso de que sea válida, construir un certificado X.509 a partir del nombre y la clave pública indicados en la petición. Los otros datos que incluirá el certificado (nombre del emisor, número de serie, periodo de validez, identificador del algoritmo de firma, etc.) serán determinados por la autoridad de certificación.

Ejemplo de la estructura de un PKCS#10

```
PKCS#10: SEQUENCE (478)
   toBeSigned SEQUENCE (327)
     version INTEGER (1) 0x00 (0)
     subject SEQUENCE OF (132) CN=user, OU=Criptografía, O=UOC, C=ES
     subjectPublicKeyInfo SEQUENCE (159)
       algorithm SEQUENCE (13)
         algorithm OBJECT IDENTIFIER (9) pkcs1-rsaEncryption
         parameters TYPE (2) with NULL (0)
       subjectPublicKey BIT STRING (141) Encapsulates TYPE (140) with
           rSAPublicKey SEQUENCE (137)
             modulus INTEGER (129) 0x00A1BA3CA9B379D12BD8C34F99D01A3AD3418A2 ...
             exponent INTEGER (3) 0x010001 (65537)
     attributes [0] IMPLICIT SET OF (25) SEQUENCE (23)
       attrType OBJECT IDENTIFIER (9) pkcs9-challengePassword
       attrValue TYPE (12) with SET OF (10) CHOICE (10) PrintableString (8) "contraseña"
   signatureAlgorithm SEQUENCE (13)
     algorithm OBJECT IDENTIFIER (9) pkcs1-md5WithRSAEncryption
     parameters TYPE (2) with NULL (0)
   signature BIT STRING (129) 00 4936BDAD2A39E99EFC2125EDD99FCF15D3F9 ...
```

6.3. El PKCS#12

El PKCS#12 describe una sintaxis para la transferencia de información de identidad personal, que puede incluir claves privadas, certificados, ampliaciones u otros secretos. Las aplicaciones que admite éste formato permiten el intercambio seguro de claves entre diversos repositorios.

El PKCS#12 permite la transferencia directa de información personal bajo diversos modos de confidencialidad e integridad. Hay cuatro combinaciones de modos de confidencialidad y de integridad según se utilicen políticas de clave pública o de contraseñas. Los modos privados cifrarán la información personal para protegerla de la exposición pública y los modos de integridad garantizan que la información no sea falsificada.

Dentro de los modos privados, hallamos:

a) Modo privado de clave pública: la información personal se empaqueta y se cifra en la plataforma de origen con la clave pública del destino.

b) Modo privado con contraseña: la información personal se cifra con un nombre de usuario y contraseña.

Dentro de los modos de integridad hay:

a) Modo de integridad de clave pública: la integridad se garantiza mediante la firma digital del contenido con la clave privada de la plataforma de origen. Esta forma se verificará en el destino.

b) Modo de integridad con contraseña: la integridad se garantiza con un código de autenticación del mensaje (MAC) derivado de la contraseña.

El estándar permite la utilización de los cuatro modos, aunque desde el punto de vista de la seguridad, lo más aconsejable es utilizar los modos de clave pública, tanto para proporcionar privacidad como para garantizar la integridad. El par de claves criptográficas de la plataforma utilizadas para el transporte del PKCS#12 no deben confundirse con las claves personales de usuario que se quieren transportar de un dispositivo a otro.

Por otra parte, no siempre es posible utilizar los modos de clave pública para el transporte seguro de datos personales. Se puede dar el caso de que en el momento de la exportación de unos datos no se conozca la plataforma de destinación, y por tanto, se ignore también la clave pública. En estos casos utilizaremos los modos de contraseña.

Uso de los modos de contraseña

Los navegadores web importan y exportan certificados y claves privadas con un PKCS#12 de modo contraseña.

La información del PKCS#12 se estructura de la manera siguiente:

PKCS#12	Descripción
version	Versión de la sintaxis que se utiliza.
authSafe	Estructura de datos PKCS#7 que tiene el formato de *signedData* si se utiliza modo de integridad de clave pública o *data* si se utiliza el modo de integridad con contraseña.
macData	Campo opcional que está presente si se utiliza modo de integridad con contraseña.
mac	Estructura PKCS#7 *digestData*.
macSalt	Semilla aleatoria para hacer el resumen.
iterations	Este campo no se utiliza; sólo aparece por razones históricas.

El campo *authSafe* del PKCS#12 contiene el envoltorio de la información que se quiere transferir. Está formado por estructuras de datos PKCS#7 que garantizan la integridad y contienen, a su vez, otra estructura PKCS#7 anidada con la información cifrada de los datos para transportar. Los tipos de datos posibles para el PKCS#7 confidencial son:

- *data*: si no utilizamos el cifrado para esconder los datos del PKCS#12.
- *encryptedData*: si utilizamos el cifrado por contraseña.
- *envelopedData*: si utilizamos el cifrado por clave pública.

La información cifrada se crea a partir de la codificación (*data*), el cifrado (*encryptedData*) o el empaquetamiento (*envelopedData*) de los datos en claro dispuestos en una estructura denominada *safeContents*. El *safeContents* está formado por un conjunto de bolsas seguras (*safeBags*) que pueden contener elementos simples de información (certificado, listas de revocación de certificados, clave privada, etc.). Las bolsas seguras disponen de atributos opcionales para que los usuarios puedan asignar un alias a sus objetos seguros y, de esta manera, reconocerlos cuando la información se importa de un entorno a otro.

El estándar PKCS#12 se ha implementado tanto en *software* como en *hardware* y es el formato más utilizado para la transferencia de claves privadas entre aplicaciones.

Ejemplo de la estructura de un PKCS#12

```
PKCS#12: SEQUENCE (4403)
   version INTEGER (1) 0x03 (3)
   authSafes SEQUENCE (4337)
      contentType OBJECT IDENTIFIER (9) pkcs7-data
      content [0] TYPE (4322) with
        OCTET STRING (4318) 308210DA3082035406092A864886F70D010 ...
   macData SEQUENCE (57)
      mac SEQUENCE (33)
        digestAlgorithm SEQUENCE (9)
          algorithm OBJECT IDENTIFIER (5) secsig-algorithm-sha1
          parameters TYPE (2) with NULL (0)
        digest OCTET STRING (20) 15A597AC8B967C00FFF8F846389C0 ...
      macSalt OCTET STRING (20) DC6D414C4732EDBD12FC10B4B69B ...
      macIterationCount INTEGER (1) 0x01 (1) DEFAULT
```

7. Protocolos que utilizan infraestructura de clave pública

Diversos protocolos utilizan la criptografía de clave pública para dar servicios de seguridad y, por lo tanto, tienen que ser capaces de gestionar grandes listas de claves públicas en sistemas distribuidos.

Las infraestructuras de clave pública son la plataforma que los servicios de seguridad de estos protocolos utilizan para hacer el intercambio de claves.

En este apartado veremos diferentes protocolos utilizados en la actualidad que protegen los intercambios de datos. Las aproximaciones varían según el nivel en el que se quiere dar el servicio: el nivel de aplicación (S/MIME, XML), el nivel de transporte (TLS/SSL, SSH), el nivel de red (IPsec) o el nivel físico (cajas negras que cifran todos los datos que salen por un enlace).

7.1. IPsec

El término *IPsec* (*IP Security Protocol*) se refiere a un conjunto de mecanismos diseñados para proteger el tráfico en el nivel de red IP (IPv4 o IPv6).

El **IPsec** ofrece los servicios de seguridad siguientes: integridad, autenticación de los datos de origen, protección contra reenvíos y confidencialidad (confidencialidad de los datos y protección parcial contra el análisis de tráfico).

El IPsec utiliza los mecanismos AH y ESP para proporcionar sus servicios de seguridad:

a) AH (***authentication header***) ha sido concebido para asegurar la integridad y la autenticación de los datagramas IP sin tener en cuenta el cifrado en los datos. Consiste en añadir un campo adicional a los datagramas IP, tradicionales de manera que el receptor pueda utilizarlo para verificar los datos incluidos en el paquete.

b) ESP (***encapsulation security payload***) está diseñado para asegurar la confidencialidad, pero también puede proporcionar autenticidad a los datos. El principio de funcionamiento del ESP consiste en generar un nuevo datagrama a partir del paquete IP original en el que los datos, y eventualmente la cabecera, estén cifrados.

El **IKE** (***Internet Key Exchange***) es un protocolo desarrollado específicamente para IPsec con el objetivo de proporcionar mecanismos de intercambio de claves y autenticación. El IKE permite el intercambio de claves de Diffie-Hellman y utiliza el modelo de PKI para identificar y validar las claves criptográficas. Además, también se utiliza la infraestructura de clave pública en los mecanismos de autenticación que incluyen firmas digitales.

Como el IPsec actúa en el nivel de red, toda la capa de seguridad que proporciona es completamente transparente para todas las aplicaciones que lo utilizan. Los principales usos de IPsec son:

- Conexión de dos redes privadas distantes a través de una red insegura como Internet. En este caso, se establece una red privada virtual (VPN) entre dos pasarelas seguras.

- Protección del acceso a Internet para usuarios externos, como usuarios móviles. La conexión IPsec se realiza entre el usuario móvil y la pasarela de acceso a Internet.

- Conexión extremo a extremo entre dos usuarios que no confían en la red que los separa.

Por cada mecanismo de seguridad de IPsec, hay dos modos de funcionamiento:

1) En el **modo transporte**, sólo se protegen los datos procedentes del protocolo del nivel superior y transportados por datagramas IP. Este modo sólo lo pueden utilizar los equipos finales.

2) En el **modo túnel**, la cabecera IP también se protege (autenticación, integridad y/o confidencialidad). La cabecera nueva se reemplaza otra vez al final del túnel por la autoridad de la cabecera original. El modo túnel es utilizado por equipos finales y pasarelas seguras, y permite proteger el enlace del análisis de tráfico.

7.2. SSL

El SSL (*Secure Sockets Layer*) es un protocolo, originalmente desarrollado por Netscape y sobradamente aceptado por la comunidad de Internet, que ofrece comunicaciones confidenciales y auténticas entre un cliente y un servidor.

El SSL corre sobre el nivel de transporte TCP/IP y por debajo de protocolos como HTTP o IMAP. Está formado por dos subprotocolos:

HTTP

Se denomina HTTPS la utilización de HTTP sobre SSL.

- **Protocolo de registro**: define el formato utilizado para la transmisión de datos.

- **Protocolo de establecimiento**: define el intercambio de mensajes entre cliente y servidor para que establezcan un contexto seguro de comunicación.

El protocolo de establecimiento facilita las acciones siguientes:

- Autenticación mutua de cliente y/o servidor.
- Selección de los algoritmos criptográficos y cifras que tanto cliente como servidor soportan.
- Generación de secretos compartidos.
- Establecimiento de una conexión SSL cifrada.

El SSL utiliza la infraestructura de clave pública fundamentalmente para autenticar a las entidades durante el establecimiento de la conexión. Tanto el cliente como el servidor disponen de un par de claves criptográficas que utilizan para firmar unos datos aleatorios. Los datos, la firma y el certificado correspondiente a la clave privada utilizada (y, opcionalmente, toda la cadena de certificados hasta el certificado de la autoridad de certificación raíz) se envían a la otra entidad para que verifique la firma. La verificación de la firma tiene que tener en cuenta que la cadena de certificados sea válida y que ninguno de estos certificados esté contenido en una lista de revocación. Si la verificación es correcta, la entidad puede asegurar la identidad del interlocutor.

7.3. S/MIME

El S/MIME (*Secure Multipurpose Internet Mail Extensions*) es una especificación de correo electrónico seguro diseñada para añadir servicios de autenticación, integridad, no repudio de origen y confidencialidad a los mensajes electrónicos basados en el formato estándar de Internet MIME.

El S/MIME adopta las normas y formatos definidos en el PKCS#7, de manera que su integración en las aplicaciones de correo electrónico es relativamente sencilla. Los datos que se quieren proteger presentan la estructura de una entidad MIME; los objetos seguros PKCS que resultan después del proceso criptográfico se incrustan en el mensaje de correo electrónico por medio de un envoltorio MIME. S/MIME ofrece formatos para sobres digitales, firmas digitales y sobres digitales autenticados.

7.4. Seguridad en XML

El XML (*Extensible Mark-up Language*) es una especificación desarrollada por el World Wide Web Consortium (W3C) que define una sintaxis y unas reglas sobre el uso de etiquetas para estructurar la información.

Los **estándares de seguridad XML** definen el vocabulario y las reglas de procesamiento necesarios para ofrecer servicios de seguridad a las nuevas tecnologías que utilizan este lenguaje.

Las características de los estándares XML son:

- Están diseñados para ofrecer la flexibilidad y amplitud propias de XML. Permiten que la seguridad se aplique a documentos XML, a elementos XML, sólo al contenido de un elemento, o bien a documentos binarios arbitrarios.

- La seguridad se aplica extremo a extremo, ya que está asociada al contenido y no al canal de transporte.

- Se reutilizan las tecnologías de seguridad existentes en todas partes allá donde sea posible. Por ejemplo, se utilizan certificados X.509 que, en este caso, se codifican en formato texto.

Los estándares básicos de la seguridad XML son los siguientes:

WEB

Para obtener más información sobre los estándares XML, podéis consultar el portal de World Wide Web Consortium: http://www.w3c.org

- **XML Signature** (**XMLDsig**): define mecanismos para soportar la creación y verificación de firmas digitales sobre documentos XML.

- **XML Encryption** (**XMLEnc**): ofrece soluciones de confidencialidad. Es una especificación análoga al objeto *envelopedData* del PKCS#7, con la particularidad de que la confidencialidad se puede aplicar con gran nivel de granularidad sobre el documento XML. Esta característica permite cifrar porciones de mensajes de protocolos XML que tienen que ser encaminados por procesadores intermedios.

- **XML Key Management** (**XKMS**): define protocolos para la gestión de claves públicas (registro, revocación, localización y validación). Permite que los usuarios administren sus certificados de infraestructura de clave pública sin conocer necesariamente su tecnología.

- **Security Assertion Markup Language** (**SAML**): especifica un vocabulario para compartir aseveraciones de autenticación y autorización entre diferentes entidades independientes y, de esta manera, permite los servicios de registro único (*Single Sign-On*, SSO).

- **XML Access Control Markup Language** (**XACML**): define un vocabulario para expresar condiciones y reglas de autorización.

Resumen

En este módulo hemos descrito el funcionamiento de una infraestructura de clave pública y hemos visto las estructuras de datos y los protocolos que utilizan esta tecnología. Los aspectos que hemos tratado son los siguientes:

1) Hemos estudiado los componentes que forman una infraestructura de clave pública con el fin de entender qué servicios nos ofrece.

2) Hemos hecho referencia a los modelos de confianza y hemos visto cómo se establecían las relaciones de confianza entre entidades de un mismo grupo y entre diferentes organizaciones.

3) Hemos descrito los procesos asociados al ciclo de vida de las claves criptográficas y de los certificados.

4) Hemos detallado las características del elemento principal de la infraestructura de clave pública: el certificado digital. Hemos visto cómo están estructurados los datos del certificado y cómo se puede ampliar la información que se contiene con el fin de adaptarlo a futuras necesidades.

5) Hemos explicado el formato de los mensajes PKCS que llevan información asociada a servicios de seguridad.

6) Finalmente, hemos hecho referencia a protocolos de red, transporte y presentación que utilizan la infraestructura de clave pública para ofrecer servicios de seguridad.

Actividades

1. FESTE es una autoridad de certificación española en que participan, entre otros, notarios y corredores de comercio. Buscad cuáles son las políticas de certificación (CP) de los certificados que emite y cuál es la declaración de prácticos de certificación (CPS).

2. Visitad la página web http://www.openvalidation.org y probad la funcionalidad de los servicios de OCSP con vuestro navegador.

3. Ceres es una autoridad de certificación que ofrece sus servicios para que la FNMT (Fábrica Nacional de Moneda y Timbre) pueda proporcionar certificados digitales a todos los ciudadanos españoles para la resolución vía Internet de algunas gestiones con la Agencia Tributaria (por ejemplo, la declaración de renta). A partir de su página principal, http://www.cert.fnmt.es, seguid todos los pasos necesarios para la obtención de un certificado digital.

4. Según hemos visto, los navegadores web utilizan un modelo por lista de confianza. Buscad dentro de los navegadores que utilizáis habitualmente dónde está esta lista y en qué autoridades de certificación raíz confiáis cuando navegáis por Internet. Comprobad si es posible añadir o eliminar autoridades de certificación de esta lista.

Ved el modelo por lista de confianza en el subapartado 3.4 de este módulo.

Ejercicios de autoevaluación

1. ¿Hay que comprobar las listas de revocación de certificados para saber si un certificado ha caducado?

2. Queremos recuperar la clave de firma de un usuario de nuestra empresa que se ha jubilado. ¿Es posible que la CPS lo permita?

3. Indicad los pasos que hay que seguir para verificar la firma digital de una estructura PKCS#7.

4. Los PKCS son estructuras de datos que llevan información criptográfica. Haced una tabla que indique en qué PKCS podéis encontrar una clave pública, una clave privada o una clave simétrica.

5. El perfil de certificado X.509 especificado por PKIX define una ampliación *extKeyUsage* para indicar las aplicaciones a las que se dirige el uso del certificado. Exponed el valor que esta ampliación puede tomar y la relación que tiene con la ampliación *keyUsage*.

6. El certificado de una autoridad de certificación expira a finales del 2005. Esta autoridad de certificación emite certificados de firma digital con una validez de dos años. ¿Hasta qué fecha podrá emitir certificados? ¿Hasta cuándo tiene que seguir dando servicios la autoridad de certificación?

7. ¿Qué operaciones se pueden hacer con un par de claves criptográficas cuando el certificado de cifrado correspondiente ha expirado?

8. Enumerad todas las autoridades de certificación posibles por las que una aplicación pueda considerar inválido un certificado.

Solucionario

1. No es necesario buscar la información sobre la fecha de expiración de un certificado en las listas de revocación de certificados, ya que se puede encontrar en el mismo certificado, dentro del campo *validity*.

2. Las CPS no permiten la recuperación de claves de firma por dos motivos. Por una parte, carece de sentido tener copias de seguridad, ya que la pérdida de una clave de firma no supone ninguno inconveniente para interactuar con datos firmados con anterioridad. Por otra parte, el hecho de tener una copia de la clave privada impediría que se pudiera utilizar en servicios de no repudio.

3. Los pasos que hay que seguir para verificar la firma de un PKCS#7 son:
   a) Descifrar los resúmenes de cada firmante con la clave pública de éste.
   b) Comparar que los resúmenes coinciden con los que se obtienen al hacer los cálculos con los datos que se autentican.

4. PKCS en que se puede encontrar una clave pública, una clave privada o una clave simétrica:

PKCS	Clave pública	Clave privada	Clave simétrica
7	x		x
8		x	
10	x		
12	x	x	

5. Los usos definidos en el perfil de PKIX para la ampliación *extKeyUsage* son los siguientes:

extKeyUsage	**Descripción**	**Valores de *keyUsage***
serverAuth Autenticación de servidor	Se utiliza para la autenticación del servidor en TLS	*digitalSignature*, *keyEncipherment* o *keyAgreement*
clientAuth Autenticación de cliente	Se utiliza para la autenticación del cliente en TLS	*digitalSignature* y/o *keyAgreement*
codeSigning Firma de código	Se utiliza para firmar código descargable. Es necesaria para utilizar *authenticode* (ActiveX, *applets* de Java, *javascripts*, etc.)	*digitalSignature*
emailProtection Protección de correo electrónico	Permite el uso para proteger mensajes de correo electrónico	*digitalSignature*, *nonRepudation* y/o *keyEncipherment* o *keyAgreement*.
timeStamping Sellado de tiempo	Permite el uso para enlazar el *hash* de un objeto a un tiempo procedente de una fuente de tiempo acordada.	*digitalSignature* y *nonRepudiation*

6. Los certificados que emite una autoridad de certificación sólo tienen validez dentro del periodo de validez del mismo certificado de la autoridad de certificación. Como la autoridad de certificación expira al final del 2005 y emite certificados con una validez de dos años, la autoridad de certificación tendrá que dejar de expedir certificados al final del 2003.
   Los certificados de la autoridad de certificación son válidos hasta finales del 2005, y hasta esta fecha la autoridad de certificación tendrá que continuar ofreciendo servicios a sus clientes, no para la expedición de nuevos certificados, sino con el fin de poder tratar casos como la revocación de certificados emitidos.

7. Si el certificado de cifrado expira, la clave pública no se puede utilizar para cifrar más datos. En este caso, el periodo de expiración del certificado limita la vida útil de cifrado de la clave pública. Por otra parte, la clave privada sí que se podrá utilizar más allá de la fecha de caducidad del certificado: el titular de la clave podrá usarla para descifrar información cifrada cuando el certificado todavía era válido.

8. Las condiciones que pueden hacer que un certificado se considere inválido (y por lo tanto, que habría que comprobar si hiciéramos una aplicación basada en certificados) son:
   - El certificado ha caducado.
   - La firma digital del certificado es inválida de forma criptográfica.
   - La autoridad de certificación que ha generado el certificado no es de nuestra confianza.
   - El certificado ha sido revocado.
   - Hay una ampliación crítica no reconocida por la aplicación.

Glosario

autoridad de certificación *f* Autoridad de confianza que emite certificados.

autoridad de registro *f* Entidad que se encarga de las tareas más administrativas de la autoridad de certificación, como la confirmación de la identidad de los usuarios, la verificación de que el usuario posee la clave privada asociada a la clave pública que se quiere certificar, la publicación de los certificados en directorios, etc.

autoridad de sellado de tiempo *f* Autoridad que proporciona una prueba de la existencia de ciertos datos antes de un instante de tiempo determinado.

autoridad de validación *f* Autoridad que proporciona información sobre el estado de revocación de los certificados.

CA *Véase* autoridad de certificación.

certificado digital *m* Estructura de datos que vincula una identidad con una clave pública. El certificado es emitido por una autoridad de certificación.

CP *Véase* política de certificados.

CPS *Véase* declaración de las prácticas de certificación.

CRL *Véase* lista de revocación de certificados.

declaración de las prácticas de certificación *f* Declaración de las prácticas que una autoridad de certificación utiliza para emitir certificados.

entidad final *f* Propietario de un certificado que no es una autoridad de certificación.

Extensible Mark-up Language *m* Especificación que define una sintaxis y unas reglas sobre el uso de etiquetas para estructurar la información.

LDAP *Véase* light weight directory access protocol.

Light Weight Directory Access Protocol *m* Protocolo de acceso a un directorio.

lista de revocación de certificados *f* Lista emitida por las autoridades de certificación en la que se publican todos aquellos certificados que han sido revocados.

OCSP *Véase* online certificate status protocol.

On line Certificate Status Protocol *m* Protocolo para determinar el estado de revocación de un certificado.

PKCS *Véase* public-key cryptography standards.

PKI *Véase* public key infrastructure.

política de certificados *f* Conjunto de normas que indican la aplicabilidad de un certificado a una aplicación o comunidad.

Public Key Cryptography Standards *m* Conjunto de especificaciones desarrolladas en los laboratorios RSA con el fin de establecer una norma común en la industria sobre los formatos de los datos utilizados en criptografía de clave pública.

Public Key Infrastructure *f* Conjunto de *hardware, software*, personas, políticas y procedimientos necesarios para crear y gestionar certificados digitales basados en criptografía de clave pública.

RA *Véase* autoridad de registro.

SCVP *Véase* simple certificate validation protocol.

Simple Certificate Validation Protocol *m* Protocolo para determinar la ruta de certificación de un certificado y su estado de revocación.

TSA *Véase* autoridad de sellado de tiempo.

VA *Véase* autoridad de validación.

XML *Véase* Extensible Mark-up Language.

Bibliografía

Adams, C.; Lloyd, S. (2003). *Understanding PKI* (2.ª ed.). Boston: Addison-Wesley.

IETF-PKIX Working Group. "Public-key infrastructure (X.509)".
En línea: http://www.ietf.org/html.charters/pkix-charter.html

Nash, A.; Duancem, J.; Brink, James E. (2001). *PKI implementing and managing e-security*.
Nueva York (etc.): Osborne / McGraw-Hill.

RSA Security. "PKCS: Public key criptography standards".
En línea: http://www.rsasecurity.com/rsalabs/pkcs/

World Wide Web Consortium (W3C). "XML encryption syntax and processing".
En línea: http://www.w3.org