Capítulo 3

Desarrollo de un sistema avanzado de gestión de certificados X.509

Este capítulo detalla el diseño y la implementación de aquellas cuestiones presentes en la PKI que ha servido como base de nuestra investigación. En concreto, se presenta la infraestructura desarrollada en el Proyecto PISCIS. En primer lugar, se introducirá el diseño general de la PKI, es decir, sus elementos constituyentes y la relación entre los mismos. A continuación, se describirán las operaciones básicas de gestión realizadas por dicha infraestructura. Por último, se detallarán las propuestas innovadoras que incorpora este sistema, más concretamente el mecanismo de definición de políticas de certificación y los sistemas de validación y autorrevocación de certificados.

3.1 Objetivos a cumplir por la PKI desarrollada en el marco del Proyecto PISCIS

El Proyecto PISCIS (Piloto de definición de una Infraestructura de Seguridad para el Comercio Inteligente de Servicios) [53] tenía como objetivo fundamental diseñar e implementar una infraestructura de seguridad sobre la cual tener la posibilidad de crear y poner en marcha un sistema de comercio electrónico caracterizado por hacer uso de los últimos avances de investigación en lo que a seguridad en las comunicaciones se refiere. Este objetivo dio lugar al desarrollo de una infraestructura de certificación avanzada, la adaptación de sistemas de tarjeta inteligente a los modelos de seguridad definidos por los principales clientes Web y el desarrollo de un modelo de pagos [174] adaptado a los requisitos impuestos por el propio entorno real de aplicación.

Como se comentó en la sección 2.5.2, la Universidad de Murcia, y más concretamente el grupo de investigación ANTS, viene trabajando desde hace varios años en el ámbito de la especificación e implementación de PKIs. La PKI del Proyecto PISCIS constituye la evolución de la infraestructura del Proyecto SSL, e incorpora nuevas características y servicios en materia de certificación propuestos recientemente, así como algunas ideas propias de investigación más innovadoras. A priori, se determinó una serie de requisitos

que debería cumplir la nueva PKI con el fin de satisfacer los objetivos del proyecto. Entre dichos requisitos encontramos:

- El diseño de la infraestructura debía especificar todo el sistema basándose en desarrollos propios, sin hacer uso de soluciones comerciales ajenas que condicionaran la evolución de la misma.
- La infraestructura debía ser versátil a la hora de ofrecer los distintos servicios básicos de gestión del ciclo de vida de los certificados. En concreto, se debían contemplar varias alternativas en lo que al proceso de creación, renovación y revocación de certificados se refiere.
- De igual modo, el soporte para tarjetas inteligentes debía ampliarse para adoptar otros tipos además del utilizado en la Universidad, por ejemplo las tarjetas con capacidades criptográficas RSA [96] y las tarjetas JavaCard [144].
- La configuración de la PKI, así como el cumplimiento de las prácticas de certificación, debía ser tratado de forma concisa, permitiendo a la vez que la PKI pudiera adaptarse de forma sencilla a escenarios con particularidades y requisitos muy diversos. En general, la administración de la infraestructura debía ser sencilla, estructurada y completamente distribuida.
- Debían incorporarse los nuevos servicios de valor añadido surgidos en materia de PKI, como los mecanismos de consulta en línea del estado de los certificados o servicios de sellado digital de tiempo.

Las soluciones a una parte de dichos objetivos, que serán presentadas a continuación, fueron satisfechas como parte de esta tesis. El resto forma parte de trabajos de investigación desarrollados por otros miembros del grupo de investigación ligados a dicho proyecto. Las aportaciones propias pueden agruparse en cuatro grandes bloques: diseño general del sistema, mecanismo de políticas, extensión de las operaciones básicas de gestión de certificados y servicios adicionales de PKI enfocados principalmente a la validación de certificados.

3.2 Diseño general de la PKI

Una de las diferencias más importantes respecto a la PKI del Proyecto SSL fue el diseño general de la PKI de PISCIS. Hemos de tener en cuenta que la primera se basaba en el uso de la herramienta Netscape Certificate Server, la cual condicionaba la escalabilidad y la extensibilidad del sistema al ser un producto cerrado. Así pues, desde el principio el nuevo diseño se abordó teniendo en mente que todos los componentes principales del sistema habrían de ser desarrollos propios. Esto hizo que dicho diseño pudiera emprenderse de forma más abierta, sin restricciones externas y buscando, desde un punto de vista de investigación, el mejor esquema posible dadas las necesidades del proyecto. Como consecuencia de ello, se planteó un esquema altamente distribuido, con varios puntos de

acceso al mismo y varias entidades participantes que dotaban a la infraestructura de una mayor escalabilidad y versatilidad.

El sistema está desarrollado completamente en el lenguaje de programación Java y hace uso de software de libre difusión. Todo ha sido implementado haciendo uso de diversas librerías criptográficas que proporcionan soporte para los distintos algoritmos criptográficos, estándares de certificación o protocolos de seguridad (información más detallada acerca de la implementación del sistema puede encontrarse en [131]).

En esta sección vamos a detallar en profundidad cuáles son las distintas entidades participantes así como la relación existente entre las mismas. Las características concretas sobre las operaciones o las políticas de seguridad se verán en secciones posteriores.

3.2.1 Elementos participantes

Por elementos participantes entendemos aquellas entidades que forman parte del núcleo de la gestión del ciclo de vida de los certificados digitales, es decir, creación, distribución, renovación y revocación. Dichas entidades podemos dividirlas en dos grupos bien diferenciados: entidades básicas y entidades de valor añadido.

Entidades básicas

Son aquellas entidades administrativas que forman parte del ciclo de vida básico de un certificado. Digamos que se trata de aquellos elementos que forman parte del conjunto mínimo de participantes en una infraestructura de clave pública, y que están presentes en la mayor parte de las implementaciones.

- *Autoridad de Registro (RA).* Normalmente es la primera entidad de contacto con la infraestructura de certificación. Se trata de un software gestionado por un operador humano que se encargará de realizar todas las validaciones pertinentes que exija cada operación realizada. En líneas generales, la función principal de la RA es la identificación y validación de solicitudes de cualquier tipo. Para realizar sus funciones toma en consideración las opciones determinadas por la política de certificación del sistema (ver sección 3.4).

- *Servidor de solicitudes.* Se trata de un elemento intermedio entre las distintas autoridades de registro y la autoridad de certificación. Su objetivo principal es almacenar todas las solicitudes de servicio realizadas tanto por las RA como por los propios usuarios finales del sistema. Dichas solicitudes quedan almacenadas en este servidor hasta que la autoridad de certificación decida retirarlas y procesarlas como corresponda. Su uso está justificado por dos motivos: el primero de ellos es que de esta forma evitamos que se produzcan comunicaciones directas con la entidad que almacena la clave privada más crítica de todo el sistema, es decir, la autoridad de certificación; el segundo motivo es que de esta forma podemos hacer que la autoridad procese de forma periódica conjuntos de solicitudes, las cuales serán obtenidas mediante una conexión segura o utilizando cualquier otro medio manual, y que en conjunto pueden ser

tramitadas de forma más eficiente que de forma individual (lo cual es particularmente cierto para el caso de las revocaciones basadas en CRLs).

- *Autoridad de Certificación (CA).* Es la entidad principal del sistema, encargada de tramitar todas las solicitudes relacionadas con el ciclo de vida de los certificados. No es posible acceder a ella de forma directa, sino a través de otros elementos intermedios confiables (como el servidor de solicitudes). Periódicamente, emite los certificados digitales asociados a solicitudes pendientes, firma las listas de certificados revocados y las políticas de certificación, y publica la información generada en los repositorios de datos tanto internos como externos.

- *Repositorios de certificados.* El sistema dispone siempre por defecto de un repositorio propio donde se va publicando toda la información generada. Dicho repositorio contiene todas las solicitudes realizadas, los certificados emitidos, las listas de certificados revocados y un histórico de las políticas de certificación. Adicionalmente, es posible publicar información en servidores de directorio externos que sean accesibles mediante el protocolo LDAP [188]. Dicha publicación puede ser controlada de forma que sólo aquella información considerada como pública será volcada en dichos servidores.

- *Administradores del sistema.* Son las entidades encargadas de la implantación y configuración de la infraestructura. Hay dos tipos de administradores: por un lado, está el administrador principal, encargado de generar los certificados básicos para la autoridad de certificación, las autoridades de registro y el servidor de solicitudes, de configurar la comunicación entre las entidades, y de poner en marcha o desactivar el sistema; por otro lado, están los administradores de la política de certificación, aquellos que establecen cuál debe ser el comportamiento dinámico del sistema, es decir, cómo debe asegurarse el correcto cumplimiento de las prácticas de certificación.

- *Usuarios finales.* Por último, encontramos a los usuarios finales o entidades a certificar. Puede tratarse tanto de seres humanos como de procesos, máquinas u otros dispositivos. Dichas entidades tendrán contacto con la infraestructura mediante los puntos habilitados para tal efecto, es decir, las autoridades de registro, el servidor de solicitudes y los repositorios públicos.

Entidades de valor añadido

Tal y como se comentó en la sección 2.4.2, en los últimos años se ha ido proponiendo un conjunto de servicios adicionales que complementan la gestión básica del ciclo de vida de los certificados. De entre todos ellos, y en vista de las exigencias del proyecto PISCIS, se estimó oportuna la inclusión de dos mecanismos de valor añadido: el mecanismo de verificación en línea del estado de los certificados y el mecanismo de sellado digital de tiempo. El primero de ellos se creyó conveniente teniendo en cuenta el escenario de comercio electrónico al cual iba dirigido el proyecto y, por tanto, la necesidad que se tiene de

proporcionar información muy precisa acerca de la validez de los certificados implicados en las transacciones. El servicio de sellado temporal es otra necesidad derivada de los esquemas de pago electrónico, ya que tanto las facturas como los recibos generados durante las transacciones deben contener información temporal confiable. Además, como veremos en la sección 3.5.2, estos dos servicios adicionales pueden ser combinados para proporcionar nuevos modelos de validación.

Así pues, las entidades de valor añadido del sistema son:

- *Servidor OCSP.* Hoy en día, el servicio de verificación de certificados en línea más aceptado es el protocolo en línea de estado de los certificados (OCSP, Online Certificate Status Protocol) [150]. La infraestructura está dotada de un servidor OCSP que informa de manera inmediata acerca del estado de los certificados consultados. Para ello, la comprobación de la validez del certificado se realiza frente al repositorio interno de certificados, el cual siempre contendrá la información más actualizada.

- *Servidor TSP.* El servicio de sellado digital de documentos asocia una marca temporal confiable a cualquier tipo de documento que reciba como entrada. Según el protocolo escogido (TSP, Time Stamp Protocol) [6], el servidor recibe un resumen digital de los documentos a sellar y genera una sentencia, firmada digitalmente, que establece una vinculación temporal entre el documento y el instante en el cual el servidor recibió el documento.

3.2.2 Relación entre los elementos

El esquema general de colaboración entre las entidades básicas del sistema puede apreciarse en la figura 3.1.

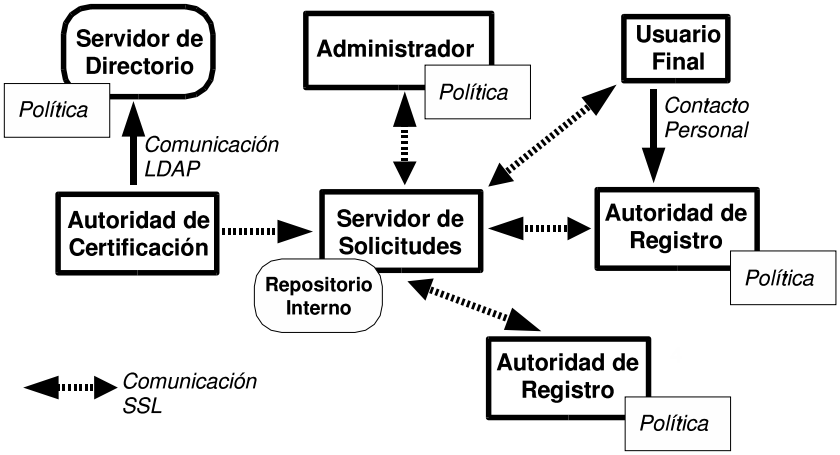

Figura 3.1: Colaboración entre las entidades de la PKI

Con el fin de distinguir claramente las diversas relaciones existentes, se irán analizando de forma individual las conexiones directas que se pueden producir como consecuencia de la tramitación de alguna de las operaciones ofrecidas.

En primer lugar, se observa que las autoridades de registro se conectan mediante SSL al servidor de solicitudes para dejar almacenadas las peticiones realizadas por los usuarios del sistema. Dicha conexión requiere una autenticación de los dos extremos con el fin de asegurar que sólo las autoridades de registro son capaces de depositar solicitudes y que dichas solicitudes sólo son almacenadas por servidores válidos. Debido a que puede haber (y de hecho será lo más corriente) varias autoridades de registro, el servidor de solicitudes tiene que tener constancia de las distintas autoridades que forman parte del sistema.

La relación entre los usuarios finales y el sistema puede ser de dos tipos. Por un lado, pueden establecer conexiones directas con el servidor de solicitudes para pedir la creación, modificación o revocación de un certificado. En dichas conexiones, protegidas por el protocolo SSL, se realiza siempre una autenticación del servidor. La autenticación del usuario final será necesaria sólo durante las operaciones de modificación del estado de un certificado existente. Por otro lado, los usuarios finales también pueden dirigirse personalmente a las autoridades de registro con el fin de utilizar algunos de los servicios proporcionados por estas entidades.

El administrador de la política de seguridad establece siempre conexiones seguras totalmente autenticadas con el servidor de solicitudes. Mediante estas conexiones, son capaces de proporcionar las nuevas políticas de seguridad del sistema, las cuales serán posteriormente distribuidas a cada una de las autoridades de registro y a la propia autoridad de certificación.

Por otro lado, la comunicación entre la autoridad de certificación y el servidor de solicitudes puede realizarse de varias formas posibles. De hecho, la forma mediante la cual una autoridad de certificación obtiene las solicitudes ha sido siempre un campo de gran controversia [67]. Las tendencias comprenden desde enfoques muy conservadores, como la total desconexión de la autoridad de certificación con el mundo exterior, hasta alternativas donde la máquina que contiene la autoridad de certificación está directamente accesible por parte de cualquier usuario. En esta PKI se tomó la decisión de que la autoridad de certificación estaría ubicada en una máquina que no acepta ningún tipo de conexión entrante, y que puede ser configurada para que de forma periódica sea ella la que establezca una conexión con el servidor de solicitudes para poder obtener todas las peticiones pendientes. La introducción de un elemento intermedio hace que el nivel de seguridad se vea incrementado, pero sin llegar al esquema de la desconexión total, el cual puede llegar a ser muy ineficiente y del que se pueden derivar varios problemas relacionados con la actualización inmediata de incidentes como las revocaciones. Así pues, la autoridad de certificación de esta PKI establece de forma periódica, con una periodicidad que puede ser distinta para cada tipo de solicitud pendiente, una conexión segura y totalmente autenticada con el servidor para retirar las peticiones y proceder a su tramitación.

La última colaboración del sistema es la llevada a cabo entre la autoridad de certificación y el repositorio público de datos. Para ello se establece una comunicación LDAP punto a punto, o una comunicación LDAPS, con el fin de asegurar que sólo la autoridad de certificación es capaz de introducir o modificar datos en las entradas del directorio.

3.3 Operaciones básicas ofrecidas por la PKI

La PKI se caracteriza por ofrecer un amplio abanico de posibilidades en lo que a accesibilidad a sus servicios se refiere. Con esto se quiere decir que la mayor parte de las operaciones ofrecidas por parte de la misma pueden ser realizadas utilizando medios muy distintos. En este apartado, se consideran como operaciones básicas aquellas que están presentes en la mayoría de las recomendaciones e implementaciones existentes, y que serán por tanto descritas de forma somera al no introducir innovación científica. Analizaremos las distintas posibilidades en materia de certificación, renovación y revocación.

3.3.1 Certificación

El primer paso en el ciclo de vida de un certificado es la creación del mismo a través del procesamiento de una solicitud de certificación. Como ya se ha comentado, un certificado puede estar asociado a una persona o a un proceso software. En el caso de los humanos, podemos encontrarnos con dos posibilidades distintas: que el usuario desee generar su propio par de claves y la solicitud de certificación utilizando alguna herramienta criptográfica, como por ejemplo un navegador; o bien que el usuario solicite que su par de claves sea generado en algún punto confiable de la infraestructura en el cual se construya también la solicitud de certificación. En el caso de los procesos software, lo más común es que las aplicaciones que desean hacer uso de los servicios proporcionados por un certificado X.509 dispongan de su propio software de generación de claves y solicitudes, requiriendo por tanto la colaboración de un operario humano que traslade la solicitud de certificación a algún punto de entrada de la infraestructura. En la figura 3.2 vemos reflejadas las distintas alternativas ofrecidas en esta PKI, las cuales pasamos a describir con más detalle a continuación.

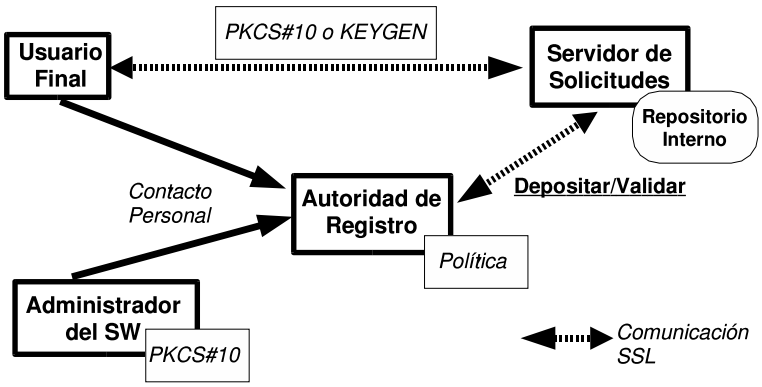

Figura 3.2: Alternativas del proceso de certificación

Creación de solicitudes en las autoridades de registro

La primera alternativa consiste en la creación, por parte de las autoridades de registro, del par de claves y de la solicitud de certificación. El proceso de creación está controlado en todo momento por la política de PKI, la cual impide especificar valores que no conformen con la misma (ver sección 3.4). Además, en el caso de trabajar con tarjetas inteligentes, esta alternativa puede ser la única válida de cara a poder almacenar la clave privada en la tarjeta del usuario, ya que el acceso a los campos (EF, Elemental File) que normalmente almacenan este tipo de información requiere el uso de módulos especiales de seguridad (SAM, Secure Access Modules) [76] que están en posesión exclusiva de entidades confiables como son las autoridades de registro. Una vez generada la solicitud en formato PKCS#10 [120], se transmite mediante una conexión SSL, con autenticación de ambas partes, al servidor de solicitudes para su posterior tramitación.

Creación de solicitudes usando el navegador

Los usuarios son capaces también de solicitar los certificados mediante un navegador instalado en su propio equipo. De esta forma, el control del par de claves y de la creación de la solicitud está siempre en sus manos. Obviamente, estas solicitudes no pueden ser aceptadas tal cual por parte de la autoridad de certificación, ya que no se realiza ningún tipo de verificación de los datos contenidos en la solicitud, lo que podría conllevar la falsificación indiscriminada de los mismos. Por tanto, las solicitudes quedan almacenadas en el servidor de solicitudes a falta de ser validadas por las autoridades de registro, las cuales requerirán la presencia física del usuario para proceder a su identificación y para comprobar si la solicitud realizada cumple con la política de certificación (ver sección 3.4). Como vemos, esta alternativa no está destinada a ahorrarle al usuario su presencia ante una autoridad de registro, sino que trata de proporcionar al usuario un mayor control sobre su información criptográfica y una mayor versatilidad a la hora de solicitar su certificado.

Procesamiento de solicitudes de entidades software

Ciertas aplicaciones, como los servidores Web seguros o el software de seguridad de nivel de red [50], hacen uso de certificados X.509 para propósitos de confidencialidad y autenticación. Normalmente, estos programas disponen de sus propias herramientas de creación de solicitudes, habitualmente en formato PCKS#10. Con el fin de que dichas solicitudes den lugar a certificados de identidad, deben ser entregadas a una autoridad de registro para que las valide y las inserte en el servidor de solicitudes pendientes. Este proceso lo lleva a cabo el administrador del servicio asociado a la aplicación software, el cual debe identificarse como operario autorizado.

3.3.2 Renovación

Un certificado puede ser renovado cuando está próxima su fecha de expiración y se considera que la información contenida en el mismo puede seguir siendo válida durante un periodo de

tiempo adicional. En general, el principal factor que impide la renovación de un certificado suele ser el tiempo de vida del par de claves, puesto que no suele ser aconsejable usar el mismo par durante un periodo superior a uno o dos años. Sin embargo, en entornos donde se prefiere hacer uso de certificados de corta duración [172], la operación de renovación puede llegar a ser útil siempre que no se hayan producido cambios en la información recogida en el certificado. La PKI ofrece dos formas distintas de solicitar la renovación de un certificado existente.

Renovación basada en las autoridades de registro

Un usuario puede solicitar la renovación de su certificado haciendo uso de una autoridad de registro. Esta entidad, tras recuperar el certificado actual del usuario, comprobará si cumple el elemento de política de PKI que indica el periodo dentro del cual puede solicitarse la renovación de un certificado, así como si el tiempo de vida de la clave contenida en el mismo no excede lo indicado en las prácticas de certificación (ver sección 3.4). Si se cumplen estas condiciones, se crea una nueva solicitud de renovación que será tramitada posteriormente por la autoridad de certificación y que tendrá el efecto de modificar el campo *NotAfter* del certificado actual de forma que refleje el nuevo intervalo. El campo *NotBefore* no se modifica con el fin de conocer en todo momento la edad del par de claves.

Renovación mediante conexión autenticada

La solicitud puede realizarse también haciendo uso de una conexión SSL, con autenticación de ambos extremos, en la cual el usuario pide la renovación del certificado que está empleando en esos instantes. El proceso realizado es el mismo que el explicado en el apartado anterior, siendo la única diferencia el medio de acceso al servicio.

3.3.3 Revocación

Una de las operaciones en las que mayor versatilidad muestra la PKI es en la de revocación. Puesto que algunas de las opciones proporcionadas son ideas originales de este trabajo, la sección 3.5.1 mostrará sus detalles concretos. En este apartado nos centraremos en la tramitación de revocaciones por parte de las autoridades de registro.

A través de estas entidades, los usuarios pueden solicitar la revocación de alguno de sus certificados existentes y especificar además la razón de la revocación así como la fecha a partir de la cual consideran que su certificado dejó de ser válido (conocida como fecha de invalidación). Dicha solicitud es transmitida inmediatamente al servidor de solicitudes en espera de que sea tramitada por la autoridad de certificación. Cuando esto sucede, se realiza un apunte en el repositorio interno de certificados de la PKI y, dependiendo de la configuración del sistema, se emite una nueva lista de certificados revocados para ser publicada en el servidor de directorio correspondiente. Haciendo uso del repositorio interno, el servicio de validación basado en el protocolo OCSP dispondrá de información reciente a la hora de responder a las consultas que se le formulen. La emisión de una nueva

CRL será de utilidad a aquellas aplicaciones que basen la validación de certificados en este mecanismo.

3.4 Una propuesta de política de seguridad para PKI

Hoy en día, podemos constatar que hay un interés cada vez más creciente en lo que se refiere a la especificación y uso de políticas de seguridad para distintos escenarios de aplicación. Así pues, podemos encontrar propuestas que van desde el control de acceso distribuido [23] hasta la protección de comunicaciones en redes activas [166].

El uso de políticas de seguridad para gestionar sistemas distribuidos es un aspecto que debe ser tratado correctamente. Hemos de tener en cuenta que las políticas pueden llegar a controlar el funcionamiento completo de un sistema, por lo que debemos asegurar que éstas son generadas de forma segura y que su información no ha sido modificada o alterada intencionadamente por terceras partes. En relación con esto, las infraestructuras de clave pública juegan un papel muy importante a la hora de obtener un nivel de seguridad que satisfaga los requisitos de este tipo de servicios.

Lo que resulta realmente interesante es que estas políticas pueden incluso usarse para gestionar no sólo servicios construidos sobre una PKI, sino incluso para controlar la propia PKI [85]. Como veremos en esta sección, las políticas de seguridad son un elemento clave en el funcionamiento de la infraestructura de clave pública aquí presentada, y es uno de los mecanismos más innovadores e importantes que incorpora.

3.4.1 Motivación

Toda PKI lleva asociada una serie de prácticas de certificación [43] que especifican las operaciones que ofrece la infraestructura, los requisitos a cumplir por parte de los solicitantes, las garantías ofrecidas, las responsabilidades derivadas y otros aspectos legales y funcionales. El cumplimiento de dichas prácticas es una tarea que involucra a muchas entidades que forman parte de la infraestructura, entidades tanto humanas como software o incluso hardware. El correcto seguimiento de las prácticas es un punto crucial, ya que simboliza el respeto del contrato que se establece entre los usuarios de la PKI y el proveedor de servicios de certificación.

La idea principal es ofrecer un mecanismo mediante el cual algunos aspectos contenidos en dichas prácticas puedan ser comprobados de forma automática por parte de la infraestructura. El servicio debería ser lo suficientemente flexible como para adaptar el comportamiento de la PKI a prácticas de certificación con requisitos muy diversos. Como consecuencia se determinó que el uso de políticas de seguridad constituía la alternativa más acertada a la hora de intentar abordar este problema.

En consecuencia, se considera una política de PKI a la implementación digital de algunos aspectos contenidos en las prácticas de certificación. Se trata de un documento firmado digitalmente que especifica cómo deben comprobarse algunas cuestiones relacionadas con

los mecanismos básicos de una PKI, tales como la certificación, publicación, renovación o revocación.

Esta propuesta constituye una idea original e innovadora en lo que respecta a la gestión de infraestructuras de clave pública. Si bien el uso de políticas se ha extendido ampliamente a lo largo de los últimos años para gestionar cierto tipo de sistemas distribuidos, no encontramos en la literatura ninguna iniciativa relacionada con su incorporación en el campo de las PKIs.

3.4.2 Ciclo de vida de una política de PKI

Hay tres operaciones básicas en relación con una política de PKI. La primera de ellas es la generación de una política inicial o política base, la segunda es la actualización de dicha política, y la tercera es la validación y la aplicación de la misma.

Respecto a la creación y la modificación, dos son las cuestiones a resolver a la hora de poner en marcha el servicio. La primera de ellas es la autenticación de los usuarios autorizados para emitir las políticas. La segunda está relacionada con el modo de distribución de la misma. En relación con la autenticación de los usuarios, se ha seguido un esquema totalmente centralizado basado en el uso del servidor de solicitudes como punto de acceso al servicio de creación/modificación de políticas. Los usuarios considerados como administradores se encuentran reflejados en una lista de control de acceso que es comprobada cada vez que alguien solicita realizar alguna de estas acciones. Respecto a la distribución de la política, se optó por un sistema en demanda donde cada elemento afectado por ella descarga la nueva versión cada vez que comprueba que no posee la instancia más reciente. De esta forma, no es necesario difundir a todos los elementos afectados cualquier cambio producido, sino que serán éstos los que comprueben periódicamente si hay una nueva versión de la política mediante la conexión con un elemento centralizado encargado de suministrar la instancia más actual de dicha política.

La operación de validación consiste en la verificación de la firma digital contenida en la propia política. Sin embargo, la determinación de la entidad encargada de la generación de dicha firma es un punto conflictivo. Si estuviera generada usando la clave privada del administrador que la crea o modifica, el proceso de verificación podría llegar a complicarse ya que se necesita un canal alternativo para indicar a cada entidad dependiente de la política cuáles son los administradores válidos en cada momento, problema que se agrava si el número de administradores es alto y la pertenencia a este rol es muy dinámica. La alternativa de que las políticas estén firmadas por la autoridad de certificación resulta mucho más escalable, ya que su certificado se encuentra distribuido entre todas las entidades del sistema, lo cual hace más fácil su verificación. Así pues, aunque son los administradores los que interactúan con el servidor de solicitudes para crear o modificar las políticas, éstas son firmadas finalmente por la autoridad de certificación y publicadas en los repositorios de datos tanto públicos como internos.

Aunque podría pensarse que este esquema de generación y uso de las políticas es demasiado rígido, está justificado por la naturaleza inherentemente centralizada de las infraestructuras de clave pública, y por el hecho de que el número de administradores y

actualizaciones de la política no será tan alto como para pensar en esquemas más descentralizados.

3.4.3 Estructura de una política de PKI

La política está codificada utilizando la notación ASN.1 [105]. Actualmente, muchas de las propuestas acerca de políticas utilizan XML (eXtensible Markup Language) [31] como lenguaje de especificación por su legibilidad. Sin embargo, las recomendaciones PKIX están basadas en el uso de ASN.1 para todas sus especificaciones, lo que lo convertía en el lenguaje más natural a la hora de implementar este tipo de políticas. A continuación, se muestra la especificación general de una política de PKI.

```
PKIPolicy ::= SEQUENCE {
  serialNumber   INTEGER,
  thisUpdate     Time,
  nextUpdate     Time OPTIONAL,
  elements       SEQUENCE OF PolicyElement
}
```

Como vemos, se compone de un número de serie, una fecha de emisión, una fecha de próxima emisión y un conjunto de elementos de política. Con el fin de dotarle de integridad, a partir de la codificación DER de este objeto ASN.1 se construye un documento firmado digitalmente y codificado siguiendo el estándar PKCS#7 [118], el cual representa realmente a la política.

Los elementos de política representan las reglas derivadas a partir de lo especificado en las prácticas de certificación. En la siguiente sección se describe la estructura general de estos elementos de política así como los distintos tipos.

Elementos de política

La estructura general de los elementos de política es la siguiente:

```
PolicyElement ::= SEQUENCE {
  entities     SEQUENCE OF GeneralName OPTIONAL,
  rule         Rule
}
```

Un elemento de política contiene una especificación acerca del conjunto de entidades afectadas por dicho elemento (la especificación se realiza utilizando la estructura *GeneralName*, la cual abarca los Distinguished Names X.500) y una regla relacionada con el parámetro que está siendo controlado (para una descripción completa de los elementos de política consultar el apéndice A). Dichos parámetros pueden ser agrupados en varias categorías:

- **Parámetros de solicitud de certificación**. Se trata de los parámetros empleados para controlar algunos de los campos incluidos en las solicitudes de certificación. Generalmente estos campos son validados por las autoridades de registro cuando tramitan las solicitudes realizadas por los usuarios. Más concretamente, los parámetros controlados son:

  - *KeyType*. Tipo de clave que contendrá el certificado.
  - *RSAKeyLength*. Longitud máxima y mínima que debe tener la clave RSA contenida en la solicitud.
  - *DSAKeyLength*. Longitud máxima y mínima que debe tener la clave DSA contenida en la solicitud.
  - *AlternativeSubject*. Nombres alternativos permitidos.
  - *UniqueIdentifier*. Indica si es obligatoria la utilización de un campo de identificador único de usuario.
  - *CertNetscape*. Extensiones de tipo Netscape que puede contener el certificado a generar.
  - *KeyExtUsage*. Uso que se le puede dar a la clave a certificar.

- **Parámetros de emisión**. Estos parámetros controlan las características relacionadas con el periodo de validez por defecto de los certificados. Son validados por la autoridad de certificación.

  - *ValidityDates*. Contiene información acerca del periodo de validez que tendrá el certificado a generar.

- **Parámetros de renovación**. Estos parámetros controlan si el certificado puede ser renovado y, en caso afirmativo, el nuevo periodo de validez que tendrá el certificado.

  - *RenewalValidity*. Contiene tres clases de información. La primera de ella es el periodo a partir del cual se puede solicitar la renovación del certificado, es decir, el número mínimo de días que deben faltar para que el certificado caduque. El segundo tipo de información es el relativo al nuevo periodo de validez del certificado, es decir, el número de días por el cual será renovado. El tercero hace referencia al periodo máximo en días durante el cual una clave puede ser renovada.

- **Parámetros de revocación**. Estos parámetros especifican cómo debe gestionarse la emisión de listas de certificados revocados.

  - *CRLIssuance*. Indica si la emisión de CRLs debe realizarse tras la notificación de una revocación, de forma periódica cada cierto número de días, o de ambas formas (es decir, tras la notificación de una revocación y tras un número determinado de días sin ninguna notificación).

Además de los elementos aquí presentados, el mecanismo de definición de políticas es lo suficientemente versátil como para permitir que la política puede ser extendida con el fin de incluir nuevos elementos que se pudieran considerar necesarios en un futuro.

3.4.4 Cumplimiento de las políticas

Las autoridades de registro comprueban de forma periódica la existencia de una nueva política, de forma que siempre disponen de la última versión de la misma. Al recuperarla, verifican la integridad de la misma mediante la comprobación de la firma digital, y la almacenan para validar las solicitudes realizadas por los usuarios. En el caso concreto mostrado en la figura 3.3, la autoridad debe validar una solicitud creada previamente por el usuario, el cual desea que sea tramitada por la autoridad de certificación.

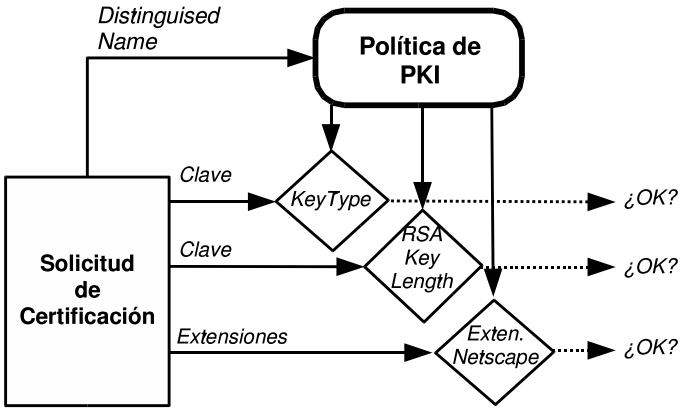

Figura 3.3: Ejemplo de cumplimiento de la política

La autoridad busca en la política todas aquellas reglas que sean aplicables al DN (Distinguished Name) que aparece en la solicitud. Una vez identificadas, en este caso tres de ellas, se comprueba individualmente que cada campo de la solicitud afectado por las reglas cumpla lo establecido en los elementos de política. Así pues, en este caso concreto, se verifica el tipo de clave contenida en la solicitud, la longitud de la misma, y las extensiones de tipo Netscape que se han solicitado. Si la solicitud cumple la política, la autoridad de registro la tramita y la envía al servidor de solicitudes para que sea procesada posteriormente por la autoridad de certificación.

El proceso llevado a cabo por el resto de las entidades de la PKI que se ven afectadas por la política es muy similar al presentado aquí para las autoridades de registro.

3.5 Nuevas propuestas de servicios de valor añadido para PKIs

Además de los servicios básicos de certificación que se han expuesto, la infraestructura incorporó algunos servicios más innovadores en materia principalmente de revocación y

validación. Se trata de propuestas propias que aportan versatilidad al sistema diseñado y que pueden ser vistas como mecanismos complementarios a los tradicionales. En primer lugar, analizaremos dos enfoques distintos para el servicio de autorrevocaciones, entendiendo como autorrevocación la capacidad de que un usuario sea capaz de revocar de forma inmediata su propio certificado, sin necesidad de intervención de otro operador humano del sistema. En segundo lugar se presentará un esquema de validación de certificados que, en ciertas condiciones, resulta más eficiente que los esquemas de validación en línea. Los dos servicios están íntimamente ligados ya que la provisión de mecanismos de autorrevocación facilitará disponer de información más actualizada en lo que respecta al estado de los certificados digitales.

3.5.1 Servicio de autorrevocaciones

En la sección 3.3.3 se ha descrito un mecanismo de revocación que se basa en las autoridades de registro para tramitar las solicitudes efectuadas por los usuarios. En ocasiones, dicho mecanismo puede no ser lo suficientemente versátil como para publicar o notificar incidentes acaecidos en cualquier instante, como la pérdida de una tarjeta inteligente fuera del horario laboral. Las autoridades de registro están controladas por uno o varios administradores con privilegios especiales a la hora de realizar ciertas tareas, pero en la mayoría de las organizaciones dichos operarios ejercen su labor durante un periodo laboral determinado. Por tanto, fuera de dicho horario de atención, el usuario no tiene opción de solicitar alguno de los servicios ofrecidos por las autoridades de registro. Si bien servicios como la certificación o renovación se han considerado generalmente como no críticos, es decir, los usuarios pueden esperar a realizar sus solicitudes durante el horario habilitado para tal efecto, el servicio de notificación de revocaciones debe merecer una atención especial (imagínese las consecuencias que podrían derivarse del compromiso de una clave privada durante un viernes noche que no puede notificarse hasta el lunes a primera hora).

Si analizamos la cuestión de quién debería estar autorizado a notificar la revocación de un certificado, nos damos cuenta inmediatamente que hay dos posibles entidades candidatas. Una de ellas podría ser una entidad con privilegios especiales, como en el caso de las autoridades de registro. La otra es el propio usuario afectado por la revocación. En contraste con el esquema tradicional de X.509, PGP [37] propone que sean los propietarios de las claves los responsables de la publicación de la revocación, lo que se conoce como autorrevocación. Si contrastamos este hecho con otros similares, como la pérdida de una tarjeta de crédito, nos daremos cuenta de que una actuación inmediata es necesaria. Resulta, hasta cierto punto, injustificable que la revocación como método para evitar situaciones comprometedoras, como el acceso a la información confidencial o la suplantación de identidad, quede retrasada durante un intervalo de tiempo por cuestiones administrativas o de otra índole.

En esta sección vamos a exponer las dos formas distintas de llevar a cabo una autorrevocación dentro la PKI. La primera de ellas está basada en el supuesto de que el usuario sigue disponiendo de su par de claves pública y privada que desea revocar, y que utilizará para notificar la revocación. La segunda alternativa es una solución para aquellas situa-

ciones en las que el usuario ha dejado de tener acceso a dicha información y por tanto no puede utilizarla para autenticarse ante el sistema y activar el proceso. Tanto la una como la otra consiguen el mismo objetivo, notificar de forma inmediata la revocación para que el mecanismo de validación en línea sea capaz de informar a partir de ese instante de la nueva situación.

Revocación mediante conexión segura autenticada

La primera forma de revocación se basa en el uso de una conexión segura SSL con autenticación de los dos extremos. El cliente, si aún se encuentra en posesión de su par de claves, puede emplear un servicio de autorrevocación que le permite tramitar de forma inmediata la revocación de su certificado. Tal y como se puede apreciar en la figura 3.4, la secuencia de acciones que desemboca en dicha tramitación es la siguiente:

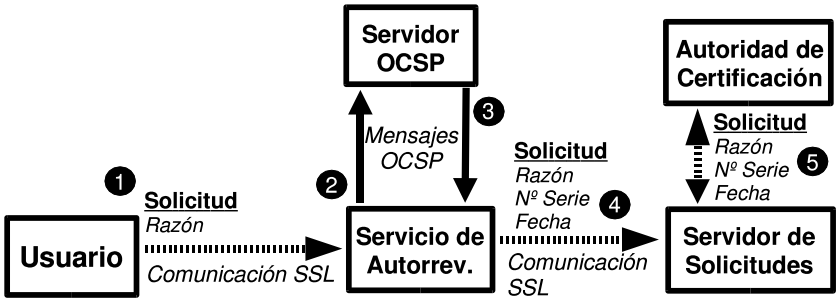

Figura 3.4: Autorrevocación mediante conexión autenticada

- El servicio de autorrevocaciones autentica al usuario como un usuario válido dentro del sistema, es decir, verifica su cadena de certificación y comprueba que el certificado no está revocado mediante una consulta al servidor OCSP.

- El usuario selecciona una razón por la cual desea revocar su certificado.

- El usuario no puede revocar otro certificado que no sea el que está usando en esos momentos para establecer la conexión. Se intenta evitar de esta forma que un usuario trate de revocar certificados a los cuales no tiene acceso.

- El servicio de autorrevocaciones inserta una nueva solicitud de revocación en el servidor de solicitudes, anotando el número de serie del certificado a revocar, la razón de revocación y el instante en el que se produjo la solicitud (la marca temporal se obtiene a través del servicio de sellado de tiempo).

- La autoridad de certificación recoge la solicitud, modifica la base de datos interna que es consultada por el servidor de OCSP y crea una entrada en la lista de certificados revocados que será publicada en su próxima emisión. La entrada contiene como

serialNumber el número de serie del certificado, como *reasonCode* la razón proporcionada por el usuario, y como *revocationDate* el instante notificado por el servicio de autorrevocaciones.

Es importante hacer mención a un detalle significativo. Como se ha visto, no se permite la introducción de una fecha de invalidación (*invalidityDate*). Dicha extensión de las entradas de una CRL tiene como propósito notificar el instante a partir del cual se tiene la sospecha de que el certificado dejó de ser válido, el cual puede ser anterior a la fecha de tramitación de la revocación. Sin embargo, si permitiésemos al usuario introducir dicho valor, podríamos incurrir en un error grave de seguridad. Hemos de tener en cuenta que este servicio sólo puede usarlo un usuario certificado, es decir, un usuario que tiene en su poder un par de claves que siguen considerándose válidas. Ahora bien, dicho usuario no tiene porque ser el poseedor original del par de claves, sino que puede ser un impostor que haya tenido acceso a las mismas. Aunque en un principio parece claro que el impostor no va a querer revocar el certificado robado, ya que entonces dejaría de poder hacer uso del mismo, no podemos decir lo mismo en el caso de que el servicio de autorrevocaciones permitiera introducir la fecha de invalidación.

En dicho caso, supongamos que el impostor decide autorrevocar el certificado. Tras introducir una razón de revocación, especifica una fecha de invalidación muy anterior al instante actual. El efecto futuro de ese dato podría asemejarse con el de un ataque de denegación de servicio, ya que aquellos documentos digitales firmados por el usuario original antes del compromiso de las claves, pero con posterioridad a la fecha de invalidación introducida por el impostor, no podrán ser considerados como válidos, aunque en realidad sí lo sean. El problema viene derivado del hecho de no poder determinar si el solicitante de la revocación es el usuario original, el cual se supone que actuará de buena fe, o un impostor. Por tanto, la única solución posible es deshabilitar la opción.

Revocación mediante autenticación en dos fases

El segundo método de autorrevocación está basado en el trabajo presentado en [52]. En la sección anterior partíamos del supuesto de que el usuario que desea revocar su certificado está aún en posesión del par de claves a anular. Sin embargo, circunstancias como la pérdida de una tarjeta inteligente o el robo de un disco duro imposibilitan hacer uso de dicho servicio. Contemplando esta posibilidad, se diseñó un sistema de dos fases mediante el cual cualquier usuario puede revocar su propio certificado incluso después de haber perdido su identidad digital. Un sistema similar al que aquí se presenta fue propuesto de forma paralela por parte del grupo de trabajo PKIX del IETF [151].

La primera fase transcurre durante el intervalo de tiempo en el que el usuario dispone de su certificado digital, con totales garantías de seguridad respecto al mismo. En ella el usuario crea una solicitud de revocación que permanecerá almacenada de forma segura hasta que tenga que ser tramitada. La protección de dicha solicitud recae en el hecho de que se encuentra cifrada mediante una clave simétrica derivada a partir de un password que el usuario introdujo. La segunda fase entra en acción una vez que se ha producido

el evento que obliga a la revocación del certificado. Durante la misma, el usuario deberá introducir el password con el cual recuperar y tramitar la solicitud creada anteriormente.

La figura 3.5 muestra un esquema general de este servicio. En la primera fase, el cliente establece una conexión SSL con autenticación de los dos participantes mediante la cual puede elegir si crear o modificar una solicitud de autorrevocación. El servidor construye un login único a partir de los datos contenidos en el certificado del usuario (dos buenos campos para ello son el identificador único de usuario y el número de serie del certificado, puesto que un mismo usuario puede tener varios certificados) y se lo presenta al usuario para que éste le asocie un password. A partir del password introducido por el usuario, y usando una función de resumen digital, el servidor deriva una clave simétrica con la cual cifrar una solicitud de revocación que quedará almacenada en un almacén de solicitudes. La segunda fase tiene lugar cuando el usuario ya no tiene acceso a su par de claves y, por tanto, está basada en una conexión SSL donde sólo el servidor es autenticado. En dicha fase, el cliente proporciona el login y el password asociados a la solicitud que quiere que se haga efectiva. Usando el login, el servidor recupera la solicitud de revocación y si puede descifrarla a partir del password generado, procederá a su tramitación mediante el envío al servidor de solicitudes de la PKI.

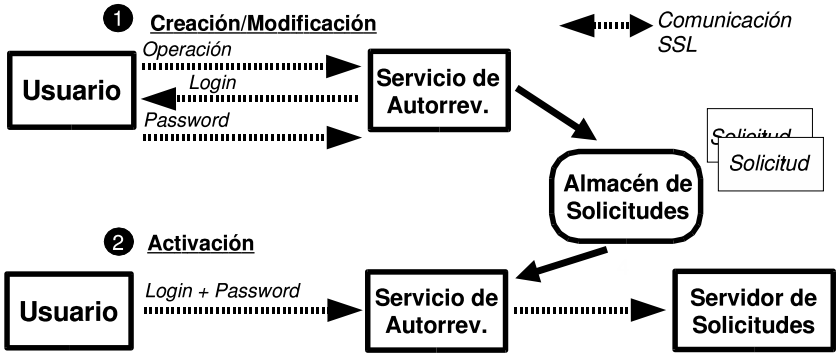

Figura 3.5: Autorrevocación en dos fases

3.5.2 Servicio de refirmado de certificados

En ciertas ocasiones, cuando la comunicación llevada a cabo entre dos entidades se considera de alto valor, o también denominado comúnmente como de alta seguridad, es necesario utilizar mecanismos muy precisos que permitan a los participantes discernir si los certificados en uso siguen siendo válidos en ese momento o por contra han sido revocados. Tal y como hemos visto, el mecanismo original propuesto por el estándar X.509 está basado en listas de certificados revocados, método que no es aconsejable en escenarios de alta seguridad debido a que el periodo de emisión de las mismas puede ser mayor que el deseado por ciertas aplicaciones que necesitan información de validación relativamente reciente. Entre los escenarios que requieren esta precisión en la validación podemos citar los entornos de

banca a distancia, transacciones financieras, intercambio de información jurídica o el acceso a historiales médicos.

Las listas de certificados revocados han sido ampliamente cuestionadas por varios autores que han propuesto varios mecanismos alternativos [148, 172]. Según la filosofía de las CRLs, las listas son emitidas periódicamente por las autoridades de certificación, lo que conlleva que el intervalo de actualización de la información esté impuesto por dichas autoridades y no por las entidades que deben validar el certificado. Por ejemplo, una CRL emitida semanalmente no satisface los requisitos de validación diaria necesarios para una aplicación de las mencionadas anteriormente. Rivest opina que para este tipo de escenarios *"el signatario puede (y debería) proporcionar todas las pruebas que la entidad receptora pudiera necesitar, y más concretamente información reciente de validación [...] la representación más simple de este tipo de evidencia es un certificado emitido recientemente"* [172].

Como ya se vio en la sección 2.4.3, hoy en día existen propuestas como OCSP (Online Certificate Status Protocol) [150] que pueden llegar a proporcionar información casi instantánea acerca de la validez de los certificados. Sin embargo, no en todos los entornos y situaciones un mecanismo de consulta en tiempo real es la mejor solución posible. Este tipo de mecanismos en línea necesitan de un cierto ancho de banda y pueden llegar a degradar el rendimiento global del sistema debido a la gran cantidad de mensajes introducidos en la red. Además, sólo son de utilidad para aquellas aplicaciones o dispositivos que disponen de una conectividad permanente. Escenarios como el control de acceso físico a laboratorios llevado a cabo por dispositivos especiales ubicados a las entradas, pero que carecen de conectividad o ésta no es permanente, son también entornos de alta seguridad donde se necesita información bastante reciente y que pueden no encontrar solución en propuestas como OCSP. En general, es necesaria una solución intermedia entre la validación fuera de línea clásica (CRL) y la validación en línea (OCSP), de forma que la fiabilidad del sistema se vea aumentada sin afectar al rendimiento global o sin depender de una conectividad permanente.

Otros autores proponen modelos intermedios entre los enfoques clásicos basados en CRLs y el esquema de OCSP. La mayoría de estos modelos están basados en el uso de resúmenes digitales en lugar de firmas digitales a la hora de validar los certificados. Silvio Micali propuso el Sistema de Revocación de Certificados (CRS) [140] que finalmente ha acabado derivando en el sistema NOVOMODO [141]. Otras propuestas han sido las realizadas por Kocher acerca de los Árboles de Certificados revocados [113] y otras modificaciones posteriores a este esquema [153]. En general, todos ellos proponen el uso de cadenas de información que son reveladas sistemáticamente para mantener el estado de los certificados. En esta tesis, la intención era diseñar un sistema que no supusiera un cambio en las aplicaciones actuales y la adopción de nuevas propuestas, sino habilitar un mecanismo basado en los estándares que ya están soportados. Al mismo tiempo, dicho mecanismo debería hacer uso de sentencias positivas, es decir, sentencias que indiquen si un certificado sigue siendo válido en lugar de expresar que no ha sido revocado.

Tal y como se verá a continuación, las dos diferencias principales entre el sistema propuesto de refirmado de certificados [51] y OCSP es que los signatarios o solicitantes, y

no los servidores, son los encargados de obtener y presentar la información de validación, y que la respuesta a esta consulta no será un nuevo tipo de mensaje firmado digitalmente por un validador delegado, sino que se tratará de un nuevo certificado X.509 emitido por la CA. Los certificados son el elemento ideal para realizar afirmaciones acerca de otras claves o certificados, lo cual puede llegar a eliminar la necesidad de introducir mensajes con nueva sintaxis. Como veremos, el mecanismo proporcionado puede ser empleado por todas las aplicaciones basadas en X.509, puesto que no introduce más requisito que la interpretación de un certificado.

El servicio tiene ciertas similitudes con el sistema Kerberos [114], puesto que los certificados refirmados se generan siguiendo un esquema basado en servidores de validación y agentes de refirmado (concepto afín al de servidores de tickets de Kerberos). Aunque más adelante se realizarán varias analogías con el sistema Kerberos, es importante dejar claro que no se está proponiendo un sistema de control de acceso. El control de acceso es una fase posterior, la cual sería realizada una vez que el usuario ha sido validado, cuestión que es la que nos incumbe en estos momentos.

Diseño del sistema

El sistema de refirmado de certificados (o certificados que rejuvenecen) está diseñado para aquellas aplicaciones que necesitan saber si un certificado presentado por un usuario era válido no hace más de X horas. El valor concreto de X depende de la política de cada aplicación y podría fluctuar entre unos pocos segundos hasta varias horas. Necesitamos entonces un elemento de información (*ticket* en la terminología Kerberos) capaz de establecer la validez del certificado del usuario durante las últimas X horas, y es importante mencionar que esta información no debe depender del protocolo o aplicación en uso. Como se ha mencionado reiteradamente en este documento, los certificados pueden realizar afirmaciones acerca de claves, identidades, autorizaciones o cualquier otra cosa que sea digitalizable. Además, una sentencia acerca de la validez de una clave pública es semánticamente equivalente a un certificado emitido para dicha clave [80]. Es más, al usar certificados X.509 para nuestros propósitos, eliminamos la necesidad de introducir una nueva sintaxis para las sentencias de validación. En conclusión, se definirá el ticket de validación como un certificado refirmado que complementará al certificado original de cada usuario y que en la mayoría de los escenarios lo suplantará. Este nuevo certificado incluirá nueva información acerca del estado del certificado original, pero preservará los valores principales del mismo (clave pública, nombre, extensiones, etc.).

En contraste con OCSP, podemos decir que éste emplea un nuevo tipo de sentencias firmadas digitalmente por el validador cuyo soporte no está ampliamente extendido a pesar de estar estandarizado desde hace tiempo. La otra diferencia principal es que el certificado refirmado es una información presentada por el poseedor del mismo, liberando de esta forma a los servidores de la obligación de consultar a un validador OCSP acerca de la validez de cada certificado recibido. Supongamos un escenario con N usuarios y M servidores que necesitan información de validación reciente acerca de los certificados de los usuarios. Con OCSP, en el periodo de X horas, habrá $N*M$ consultas acerca del estado de los certificados,

en el supuesto de que todos los usuarios intentaran acceder una vez a todos los servidores, lo cual representa $N * M$ sentencias firmadas digitalmente. En contraste, con el servicio aquí presentado, los N usuarios deben obtener un certificado refirmado X.509 con información reciente acerca de su estado. A continuación, los certificados pueden emplearse para probar a los servidores que eran usuarios válidos hace T horas, donde T es el periodo de tiempo transcurrido entre la re-emisión de los certificados y el instante actual. Si el periodo T es inferior al periodo máximo X establecido por la política de control de acceso de los servidores, el mismo certificado puede ser empleado varias ocasiones con varios servidores sin necesidad de realizar nuevas consultas o emitir nuevas sentencias.

La cuestión ahora es qué diferencia hay entre el certificado original y el refirmado que hace que este último incorpore información más reciente acerca del estado del mismo. La primera idea intuitiva quizá sea incorporar una nueva extensión al certificado que incluya una fecha indicando la última vez que se comprobó que el certificado no estaba revocado. Sin embargo, desde un punto de vista flexible, podemos considerar que ya hay un campo en los certificados que puede desempeñar perfectamente ese papel, el campo *Not Before*. Realmente, este valor podría ser considerado no sólo como la fecha de creación del certificado, sino como el instante último de chequeo del estado del mismo. Si tenemos en cuenta la falta de soporte de CRLs en la mayoría de las aplicaciones actuales, darle esa semántica al campo *Not Before* no está tan lejos de la verdad, puesto que en ausencia de mecanismos de revocación, la validez del certificado se comprueba sólo en el momento de su creación. Así pues, en este sistema, los certificados refirmados sólo se diferencian de los originales en el valor almacenado en el campo *Not Before*, que en este caso contiene una fecha de creación más reciente (por eso se dice que el certificado rejuvenece).

Los clientes pueden usar este certificado para demostrar que siguen siendo usuarios válidos dentro del sistema, ya que se demuestra que los certificados no estaban revocados en el momento de su rejuvenecimiento. Ahora las aplicaciones pueden establecer su criterio para considerar si un certificado sigue siendo válido después de haberlo refirmado: tener como máximo X horas de antigüedad. La siguiente cuestión es cómo refirmar estos certificados de forma segura, considerando como seguro que sólo entidades autorizadas sean capaces de crear certificados válidos.

Kerberos basa su sistema en tres entidades principales: el servidor de autenticación (que valida a los usuarios), el servidor de tickets (que proporciona los tickets para acceder a los recursos del sistema) y el servidor final que proporciona el servicio demandado. En este sistema encontramos elementos muy similares: un proceso que verifica que los certificados siguen siendo válidos en el momento de la consulta, un sistema automático de actualización, refirmado y publicación de los certificados, y varios servidores en cuya política se establece que sólo serán aceptados aquellos certificados con menos de X horas de antigüedad, donde X puede ser distinta para cada servidor. El proceso que valida los certificados debe ser una entidad confiable dentro del sistema con el fin de asegurarnos que no miente acerca del estado de los mismos. Como veremos, se hace uso de un servidor OCSP que emite sentencias firmadas que indican qué certificado no está revocado y en qué momento no lo está. Estas sentencias serán la entrada del sistema automático que producirá los certificados rejuvenecidos. Por último, los usuarios podrán acceder a su certificado más reciente, que

en muchos casos sustituirá al anterior, para acceder a los servicios denominados de alta seguridad. En las siguientes secciones se explica en detalle cómo está implementado el sistema.

Dinámica del sistema

Es este apartado se presenta cómo se llevan a cabo tanto el proceso de validación de los certificados como el de refirmado. Se mostrarán los distintos elementos integrantes de este servicio así como la relación entre los mismos.

El sistema puede tomar como punto de partida la necesidad por parte del usuario de acceder a un servidor que exige certificados no más antiguos de un día. Supongamos una comunicación SSL a través de la cual el usuario presenta su certificado para identificarse ante el servidor. La comprobación del campo *NotBefore* determinará si el usuario será considerado como válido (lo cual no quiere decir que esté autorizado a usar el servicio).

Realmente, el servicio se puede dividir en tres fases distintas: validación del certificado, refirmado y recuperación del nuevo certificado, y uso del mismo.

Primera fase: Validación del certificado

El sistema está diseñado de forma que en todo momento es el usuario el que tiene el control de lo que está sucediendo. La figura 3.6 muestra un esquema general de la fase de validación. En primer lugar, el usuario solicita al servidor de validación que le genere una sentencia firmada que demuestre que su certificado sigue siendo válido en ese instante. Esto genera una solicitud OCSP por parte del servidor que tiene como fin comprobar el estado actual del certificado. El servidor OCSP verifica el mismo y, en caso positivo, realiza una consulta al servidor de sellado de tiempo [6] con el fin ligar el estado del certificado con el instante actual. La fecha devuelta por el servicio de sellado se almacena en la respuesta OCSP y esta respuesta es reenviada al usuario como prueba de validez de su certificado.

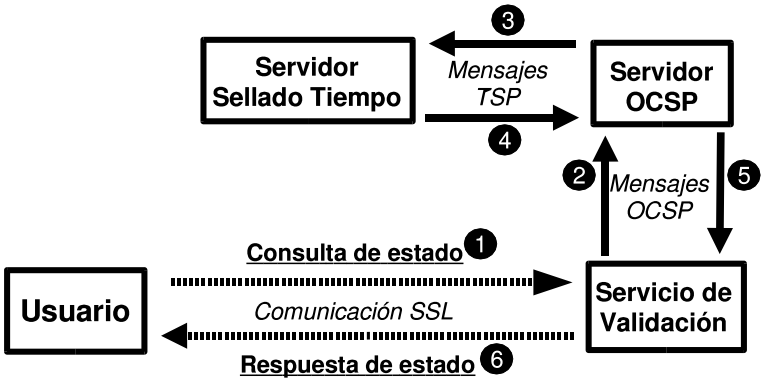

Figura 3.6: Validación del certificado a refirmar

A continuación se detallan los valores más importante contenidos en los mensajes intercambiados durante esta fase:

- **Solicitud OCSP (OCSPRequest)**
  - *requestList*. Una lista de un solo elemento.
  - *certID*. Resumen digital del nombre, la clave pública del emisor del certificado y número de serie del mismo.
- **Solicitud TSP (TimeStampReq)**
  - *messageImprint*. Hash del certificado verificado.
  - *nonce*. Carga aleatoria para evitar ataques de reenvío.
- **Respuesta TSP (TimeStampResp)**
  - *genTime*. Fecha de sellado del documento.
  - *messageImprint*. Mismo valor que el contenido en la solicitud.
- **Respuesta OCSP (OCSPResponse)**
  - *tbsResponseData*. Datos acerca de las respuesta a la solicitud.
  - *responses*. Lista de respuestas (en este caso sólo una).
  - *producedAt*. Instante de tiempo en el que se produjo la verificación.
  - *certID*. Identificador del certificado analizado.
  - *certStatus*. Contendrá el valor *good* en el caso de que no se encuentre revocado. En caso contrario el valor será *revoked*.

En esta fase hay varias decisiones a justificar. El servicio OCSP se utiliza porque está disponible en la PKI, y es quizá la mejor forma de delegar la comprobación de la validez de un certificado digital. La diferencia principal frente a su uso convencional es que la consulta acerca del estado la inicia el poseedor del mismo, en lugar de la entidad encargada de verificarlo. El uso del servicio de sellado de tiempo tiene como finalidad obtener una fecha confiable del instante en el cual se está haciendo la validación. Esta fecha, devuelta en la respuesta del protocolo TSP (campo *genTime*), se incluirá como parte del mensaje OCSP de respuesta (concretamente en el campo *producedAt*). Por tanto, el uso de estos servicios nos proporciona, por un lado una validación confiable del certificado (al estar basada en un elemento confiable como es el servidor OCSP), y por otro una marca temporal producida también por otro elemento confiable, ya que la alteración de dicha marca podría tener consecuencias graves como se verá posteriormente.

Una vez recibida la respuesta de estado, se podría pensar que no necesitamos refirmar el certificado ya que dicha respuesta ya es una demostración de por sí de que el certificado era válido en un determinado instante. Podría contemplarse la posibilidad de que fuera esa sentencia lo que el usuario empleara para convencer a un servidor acerca de su validez.

Sin embargo, hay varios inconvenientes que desaconsejan esta opción. El primero estriba en que la mayoría de los protocolos de seguridad actuales no proporcionan más mecanismos que el simple intercambio de certificados, y en este caso necesitaríamos intercambiar también una especie de credencial de validez. Así pues, empleando protocolos como SSL, el envío de este tipo de sentencias sería complejo. La segunda razón que desaconseja esta opción es que obligamos al servidor a comprender la sintaxis de los mensajes OCSP, hecho que en gran parte de los servicios actuales no se cumple. Sin embargo, el certificado refirmado puede transmitirse sin problemas y ser verificado de forma sencilla.

Segunda fase: Refirmado del certificado

La siguiente fase consiste en el refirmado del certificado. Tal y como se aprecia en la figura 3.7, la respuesta de estado es el elemento de entrada para el sistema automático de re-emisión. Dicha respuesta es procesada por el servidor de solicitudes, el cual verifica la firma digital del documento, extrae la información relativa al certificado a procesar (el número de serie) y la fecha de comprobación del estado. Estas dos informaciones son las que utiliza posteriormente la autoridad de certificación para rejuvenecer el certificado. Por un lado, recupera el certificado original usando el número de serie, y por otro lado genera un nuevo certificado donde la fecha contenida en el campo *NotBefore* es la incluida en la respuesta de estado, y la fecha del campo *NotAfter* no se modifica. El nuevo certificado se emite y se publica en el servidor de directorio en la entrada correspondiente al usuario. Hay que recalcar que este certificado no sustituye al anterior, sino que se le adjunta, de forma que el usuario tiene en cada momento un mínimo de dos certificados válidos. Por último, el usuario obtiene el nuevo certificado que ya está listo para usarse.

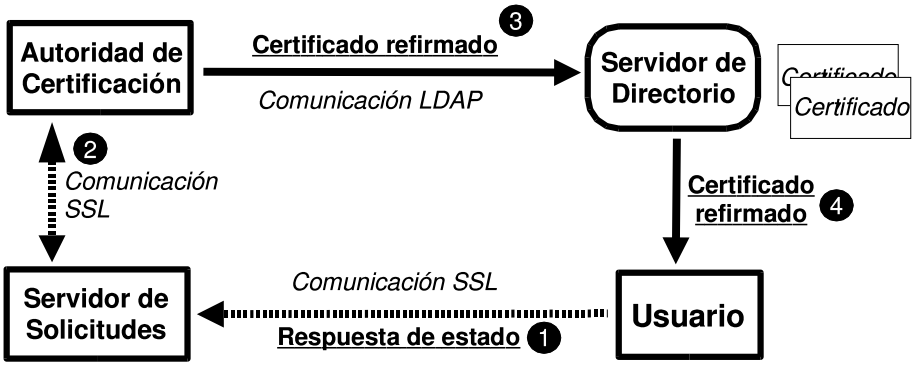

Figura 3.7: Obtención del certificado refirmado

La emisión del certificado rejuvenecido se lleva a cabo de igual forma que el resto de las operaciones realizadas por la PKI, es decir, mediante la recuperación periódica desde el servidor de las solicitudes pendientes y la tramitación de las mismas. El certificado nuevo no sustituye al anterior en el directorio con el fin de asegurar que documentos firmados digitalmente antes del refirmado puedan seguir verificándose usando el certificado original. Sin embargo, una solicitud posterior de rejuvenecimiento sí conllevaría la sus-

titución del certificado más joven, ya que no hay ninguna ventaja adicional en conservar dicho certificado en el servidor de directorio (el original permite validar cualquier documento y el último contiene la información de validación más reciente, el resto son redundantes).

Tercera fase: Uso del certificado refirmado

Cualquier servidor que por política exija certificados no más antiguos de X horas aceptará el nuevo certificado siempre que éste sea presentado antes de que transcurran X horas desde su emisión. Es importante volver a recalcar que el parámetro X es dependiente de cada servidor, por lo que un certificado podría ser válido para unos servicios pero no para otros. Por su parte, el servidor no necesita tener una lógica muy compleja para validar a usuarios, es decir, no debe realizar consultas a elementos externos o construir mensajes con determinada sintaxis. En su lugar, sólo debe analizar el contenido del campo *NotBefore* y determinar si dicho valor está en consonancia con su política.

Comparativa entre OCSP y la técnica de refirmado

Con el fin de ilustrar el rendimiento que puede llegar a obtenerse mediante la técnica de refirmado, en este apartado se va a realizar una comparativa entre la cantidad de información generada usando ambas técnicas para validar un mismo certificado. Nos interesa analizar el factor del ancho de banda consumido, es decir, el número de bytes enviados a través de la red para realizar la validación. El hecho de que no se realice un análisis respecto al tiempo se ve justificado por dos motivos: el primero es que el servicio de refirmado introduce una serie de retardos adicionales (como el periodo de obtención de solicitudes por parte de la autoridad de certificación) que hacen muy difícil establecer un criterio temporal común; el segundo motivo se debe a que, desde el punto de vista del servidor final que está validando el certificado del usuario, todo se reduce a la verificación de una firma digital (la del certificado o la de la respuesta OCSP), proceso en el cual no podemos encontrar diferencias significativas. Sin embargo, como ahora veremos, el análisis del ancho de banda nos proporcionará datos a partir de los cuales se pueden extraer conclusiones muy interesantes.

Durante una verificación basada en OCSP, dos son los mensajes involucrados en el proceso: la solicitud y la respuesta OCSP. En contraste, tal y como se vio en las figuras 3.6 y 3.7, el proceso de refirmado lleva asociado un mensaje de solicitud de validación (formado principalmente por el certificado a validar), una solicitud y respuesta OCSP, una solicitud y respuesta TSP, una respuesta de estado (formada básicamente por la respuesta OCSP), y la recuperación del certificado refirmado. La tabla 3.1 muestra la longitud media de todos los mensajes involucrados durante el proceso de validación (la determinación de dicha longitud se ha realizado a partir de la monitorización y extracción de datos reales de los servicios involucrados).

Con el fin de establecer una comparativa entre las dos técnicas, se han determinado dos variables a analizar. Considerando un periodo de tiempo T durante el cual no es necesario refirmar el certificado, se analiza una comunidad de usuarios que oscila entre 10 y 10.000 miembros, y un número de validaciones de los certificados de dichos usuarios que oscila

Mensaje	Longitud
Solicitud de validación	800 bytes
Respuesta de estado	1052 bytes
Solicitud OCSP	714 bytes
Respuesta OCSP	1052 bytes
Solicitud TSP	53 bytes
Respuesta TSP	1046 bytes
Certificado refirmado	800 bytes

Tabla 3.1: Valores para la comparativa OCSP vs Refirmado

entre 1 y 32 validaciones dentro del periodo T. Como ya se dijo en la sección 3.5.2, el número de mensajes generados por la técnica de OCSP responde a la fórmula de $N * M$, donde N es el número de usuarios y M es el número de validaciones. En el caso de la técnica de refirmado, el ancho de banda consumido es $N * R$, donde R hace referencia a todos los bytes necesarios para generar un nuevo certificado refirmado, el cual se calcula como se ha comentado en el párrafo anterior. La figura 3.8 muestra el ancho de banda consumido por ambas técnicas conforme se incrementa el número de usuarios y el número de validaciones distintas.

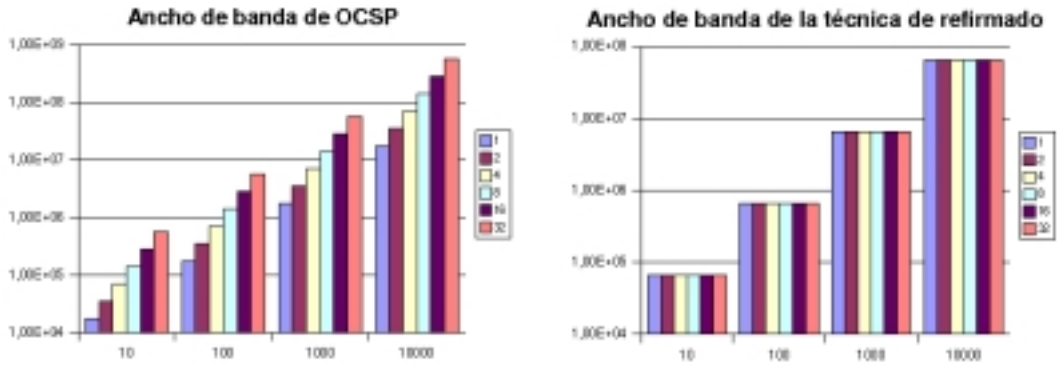

Figura 3.8: Comparativa entre OCSP y refirmado

Como podemos observar, el mecanismo de OCSP requiere menos ancho de banda cuando el número de validaciones es inferior a 4 dentro de un intervalo T. A partir de ese número, la técnica de refirmado ofrece mejores prestaciones en relación con dicho parámetro, ya que la cantidad de información transmitida se divide por 1,08 para 4 consultas, un factor de 2,1 para 8 consultas, 4,3 para 16 consultas y un 8,6 para 32 validaciones. Esto se deriva del hecho de que un incremento en el número de validaciones no conlleva un incremento del número de mensajes a generar para el caso de refirmado (como puede apreciarse en la gráfica al ver que el ancho de banda consumido es independiente del número de validaciones).

Comentarios finales

La técnica de refirmado de certificados es un mecanismo adicional proporcionado dentro del marco de la PKI. Su gran ventaja es que no sustituye a otros mecanismos de validación, sino que los complementa de cara a ofrecer unas mejores prestaciones en determinados entornos, tal y como hemos visto en el apartado anterior. De hecho, puede proporcionar un sistema fiable de validación a servicios y aplicaciones que no tengan soporte para OCSP o incluso CRLs. Además, sigue la filosofía expresada en esta tesis de basar toda decisión en sentencias positivas, y un certificado refirmado lo es.

3.6 Conclusiones

La infraestructura de clave pública aquí presentada está caracterizada por su gran versatilidad a la hora de gestionar las distintas operaciones involucradas en el ciclo de vida de los certificados digitales de identidad. Como se puede apreciar, su diseño general está basado en las principales recomendaciones realizadas por parte la comunidad científica y hace uso de los estándares más ampliamente reconocidos en lo que a gestión se refiere (por ejemplo, las series PKCS y los protocolos OCSP, TSP, LDAP y SSL)

Por otro lado, incorpora ideas innovadoras en materia de servicios de valor añadido. En primer lugar, el uso de políticas de seguridad permite descentralizar la administración de la infraestructura y adaptarla a su vez a escenarios con distintas prácticas de certificación. El mecanismo es lo suficientemente genérico y estructurado como para manejar los posibles requisitos de cada sistema. En segundo lugar, el conjunto de servicios basados en autorrevocaciones y refirmado de certificados ofrece unas características adicionales destinadas a dotar de una mayor fiabilidad a los procesos de validación de certificados, tradicionalmente descuidados por parte de la comunidad científica.

Por tanto, esta infraestructura constituye un punto de partida muy sólido a la hora de afrontar el segundo objetivo parcial de esta tesis, es decir, la especificación de mecanismos de autorización basados en certificados digitales. De alguna forma, a la hora de ofrecer sevicios de autorización es necesario partir de un sistema fiable de autenticación, con el cual poder etiquetar las entidades a las que posteriormente hay que asignar privilegios.