Universidad de Murcia

Facultad de Informática

Propuesta de una infraestructura de clave pública y su extensión mediante un sistema de gestión distribuida de credenciales basado en delegación y roles

Tesis Doctoral

Presentada por:
Óscar Cánovas Reverte

Supervisada por:
Dr. Antonio Fernando Gómez Skarmeta
Departamento de Ingeniería de
la Información y las Comunicaciones

Murcia, Octubre de 2002

Universidad de Murcia

D. Antonio Fernando Gómez Skarmeta, Profesor Titular de Universidad del Área de Ingeniería Telemática en el Departamento de Ingeniería de la Información y las Comunicaciones de la Universidad de Murcia, AUTORIZA:

La presentación de la Tesis Doctoral titulada "Propuesta de una infraestructura de clave pública y su extensión mediante un sistema de gestión distribuida de credenciales basado en delegación y roles", realizada por D. Óscar Cánovas Reverte, bajo su inmediata dirección y supervisión, y presentada para la obtención del grado de Doctor por la Universidad de Murcia.

En Murcia, a 28 de Octubre de 2002
D. Antonio Fernando Gómez Skarmeta

Resumen

La criptografía de clave pública ha sido ampliamente reconocida como una tecnología fundamental sobre la cual pueden construirse varios servicios básicos de seguridad. Los principales esfuerzos de los últimos quince años se han concentrado en el problema de asignar de forma segura nombres a claves. De hecho, la comunidad científica ha ido progresivamente adoptando el uso de sistemas basados en el estándar X.509 con el fin de proporcionar servicios de seguridad a los sistemas distribuidos, los cuales dependen de la existencia de un método seguro y fiable de gestión de las claves públicas denominado infraestructura de clave pública (PKI, Public Key Infraestructure).

Las PKIs deben proporcionar mecanismos para gestionar todas las operaciones relacionadas con el ciclo de vida de los certificados digitales, es decir, su generación, distribución, validación y revocación. Sin embargo, la mayoría de los diseños e implementaciones actuales no proporcionan un tratamiento completo a todas estas cuestiones, especialmente a las relacionadas con la revocación, la validación o el cumplimiento de las políticas de certificación. La PKI que se presenta como una de las contribuciones de esta tesis subsana esta falta de soporte mediante la provisión de varios mecanismos diseñados para poder realizar una gestión completa de los certificados X.509.

Por otro lado, durante los últimos años, la criptografía de clave pública ha sido propuesta también como una herramienta para solucionar los problemas relacionados con la autorización y el control de acceso, es decir, para determinar qué están autorizadas a realizar las entidades de un sistema distribuido. De hecho, son varias las especificaciones existentes que proponen mecanismos capaces de ligar información de seguridad a claves públicas. Más concretamente, dichas especificaciones han desarrollado notaciones para asignar autorizaciones y para propagar dichas autorizaciones entre varias claves.

Sin embargo, tales propuestas no especifican cómo debe realizarse la gestión del ciclo de vida de los certificados de autorización. Si bien ciertos enfoques dependientes de la aplicación pueden dar resultado en entornos reducidos, su uso en escenarios complejos puede sacar a relucir varios problemas relacionados con su escalabilidad e interoperabilidad, lo cual hace necesario plantear un sistema que sea capaz de llevar a cabo dicha gestión de forma estructurada y distribuida. Por tanto, una de las principales contribuciones de esta tesis es la definición de una infraestructura de autorización. Dicha infraestructura está compuesta de un sistema distribuido de gestión de credenciales basado en delegación y roles, un marco para el intercambio seguro de información de autorización y una metodología de definición de políticas de autorización.

En consecuencia, tanto la PKI como la infraestructura de autorización constituyen un sistema global para la gestión distribuida de certificados digitales. Con el fin de demostrar su aplicabilidad como middleware de seguridad, se presenta también como parte de este trabajo la integración de dicho sistema en dos escenarios de aplicación distintos.

Abstract

Public key cryptography is widely recognized as being a fundamental technology on which several essential security services can be built. Much security discussion over the past 15 years has concentrated on the problem of assigning names to keys. Related to this, the Internet community is agreeing on the use of systems based on the X.509 standard in order to provide basic security services to distributed systems. These services need a practical and reliable method of publishing the public keys, called a Public Key Infrastructure (PKI).

PKIs must provide the mechanisms to manage all the operations related to the lifecycle of digital certificates, i.e., generation, distribution, validation, and revocation. However, most of the current designs and implementations of PKIs do not provide a wide coverage to all these issues, especially to those topics related to revocation, validation, or policy enforcement. The PKI presented as a contribution of this thesis overcomes this lack of support providing several mechanisms that have been designed to perform a complete management of X.509 certificates.

On the other hand, in recent years, public key cryptography has been also proposed as a tool for solving the problems related to authorization and access control, that is, for determining what the identities should be allowed to do in a distributed environment. Several specifications propose mechanisms for capturing security-relevant information and binding that information to public keys. They have also developed languages for assigning authorizations and for delegating those authorizations from one key to another.

Nevertheless, these proposals do not explain how the lifecycle of authorization certificates is performed. Although simple and not distributed approaches can constitute a good alternative for small scenarios, some problems derived from scalability or interoperability might arise in more complex environments. A structured and distributed system must be provided to manage the generation or revocation of those certificates. Therefore, one of the main contributions of this thesis is the definition of an authorization infrastructure. That infrastructure consists of a distributed credential management system based on delegation and role membership, a framework for secure exchange of authorization-related information, and a methodology for the definition of authorization policies.

The PKI and the authorization infrastructure constitute a global system for distributed management of identity and authorization certificates. In order to demonstrate their feasibility and their suitability as a security middleware, two different aplication environments making use of their mechanisms are presented.

Agradecimientos

Supongo que gran parte de lo que es mi vida actual está condicionada por el hecho de que mi padre quisiera comprarle un ordenador personal a un niño de 8 años que pasó gran parte de su infancia trasteando a 4 MHz (y eso que no me dejaba usarlo mucho porque decía que me ponía nervioso). Por tanto, el primer agradecimiento es para él y para mi madre, por haberme dado todo lo que estaba en sus manos para poder dedicarme a lo que me gusta y ganarme la vida con ello. También a mi hermana Fani, por haber prestado siempre tanta atención a lo que le cuenta su hermano (aunque no siempre me entienda) y por hacerme ver las cosas de otro color.

Por supuesto, nada habría sido lo mismo sin el constante apoyo y cariño de Noemí, quien ha visto como esta tesis nos robaba demasiadas horas y, no contenta con eso, tuvo incluso la paciencia de revisar el estilo de este documento (eliminando así innovadoras aportaciones al castellano que yo había propuesto reiteradamente).

Al grupo de investigación ANTS, especialmente a las personas con las que he tenido la oportunidad de colaborar en los proyectos de seguridad (Gabi, Félix, Rafa, Antonio y Gregorio). La posibilidad de haber analizado, debatido, defendido y cuestionado con ellos muchas de las ideas aquí presentes ha enriquecido sin duda el resultado final (y si no, siempre nos queda la posibilidad de aceptar aquella oferta que nos hicieron para formar parte de un circo).

A los compañeros de DITEC, por haber sabido lograr este ambiente tan cálido con el cual es mucho más agradable trabajar. En especial a Pepe, al cual le debo tantas cosas (por ejemplo, la sección 2.4.2 ;-) y del que he aprendido tanto que necesitaría un apéndice D para poder contarlo todo.

Por último, gracias a Antonio por haber confiado en mi y haberme dado la oportunidad de trabajar a su lado.

Índice General

Índice de Figuras

Capítulo 1

Introducción y objetivos

1.1 La seguridad como cuestión transversal en el diseño de sistemas informáticos

A lo largo de los últimos años, la naturaleza de los sistemas informáticos ha ido evolucionando de forma vertiginosa como consecuencia de los grandes avances tecnológicos de la segunda mitad del siglo XX. La *sociedad de la información* no es un término vacío de contenido, un vaticinio acerca de la influencia de la tecnología en nuestra vida cotidiana; todo lo contrario, se trata de una nueva revolución sociológica que ha modificado algunos de nuestros hábitos más cotidianos, que ha conseguido transmitir a la humanidad la sensación de proximidad, de la posibilidad de acceder a un número ilimitado de servicios y datos en continua expansión.

La red Internet es hoy en día uno de los medios de comunicación de mayor difusión e impacto, quizá el más directo y dinámico, lo que ha permitido poner en contacto a personas de todo el mundo a un coste asequible. El correo electrónico, especial precursor de esta red, ha asumido y ampliado gran parte de los servicios ofrecidos no hace mucho por el correo tradicional. Las aplicaciones de vídeoconferencia o voz vía Internet han dejado de ser herramientas utópicas para convertirse en componentes cotidianos, y términos como comercio electrónico, tele-trabajo, administración electrónica, redes privadas virtuales, identidad digital o tele-enseñanza son hoy comunes y hacen referencia al gran número de posibilidades y de nuevos servicios que las tecnologías de la información pueden proporcionar.

Sin embargo, parece ser un hecho que los términos *conectividad* y *seguridad* frecuentemente resultan antagónicos. En la evolución de las tecnologías de la información, la carrera entre ambos términos siempre ha tenido al primero de ellos como gran vencedor. Es un cliché que se repite, y es que los diseños de la mayor parte de los componentes relacionados con las redes de comunicaciones han estado caracterizados por desconsiderar de forma drástica la mayor parte de las cuestiones relacionadas con la seguridad. Incluso hoy en día, sigue siendo una práctica habitual posponer el diseño de los mecanismos de seguridad relacionados con una aplicación concreta, esperar a ver si la propuesta tiene éxito y después empezar a enmendar todo aquello que ha sido vulnerado. La explicación a esta

falta de coordinación puede deberse normalmente a la necesidad de resultados, entiéndase éstos como resultados económicos en el caso del sector privado o experimentales en el caso de la comunidad académica. Independientemente de las razones que hayan propiciado esta filosofía de diseño, lo cierto es que el principal trabajo de los profesionales de la seguridad de la información ha sido dotar de mecanismos de seguridad a aplicaciones y componentes ya existentes cuya vulnerabilidad ha quedado claramente expuesta, en lugar de considerar la seguridad como un aspecto transversal a tener en mente desde los primeros pasos de diseño.

Dada la creciente presencia de las tecnologías de información en la vida cotidiana, las limitaciones y vulnerabilidades detectadas toman una mayor importancia si tenemos en cuenta las implicaciones que éstas pueden llegar a tener sobre comunidades de usuarios extensas. Conocidos son los casos en los que un atacante ha podido tener acceso a números de cuentas bancarias o perfiles de consumo almacenados en un servidor indebidamente protegido, situaciones en las que la propagación incontrolada de virus ha tenido consecuencias catastróficas sobre un porcentaje considerable de los ordenadores de una organización, ataques que han dejado fuera de servicio durante horas a algunos de los servidores más visitados de Internet, suplantaciones de la identidad de altos cargos, interceptación de contraseñas transmitidas en una red de área local y un largo etcétera de actos que han aprovechado las numerosas vulnerabilidades de seguridad de los elementos que componen la red, y por extensión los sistemas distribuidos.

Quizá para entender la evolución de los mecanismos de seguridad en estos entornos habría que analizar primero las tres etapas distintas por las que, hasta ahora, ha atravesado el campo de los sistemas distribuidos. Originariamente, la principal preocupación era poder gestionar la presencia de múltiples usuarios en un mismo supercomputador. Por un lado había que establecer los criterios mediante los cuales se pudiera comprobar que sólo aquellos usuarios que habían sido autorizados podían tener acceso al ordenador. Por otro lado, se debía proteger la información asociada a cada usuario de la interceptación o modificación realizada de forma intencionada, o incluso casual, por parte de otros usuarios.

Posteriormente, debido a la revolución que supuso la fabricación masiva de ordenadores personales y a la extensión de la red Internet, el número de posibles tipos de amenazas de seguridad se multiplicó por varios órdenes de magnitud. En primer lugar, debe tenerse en cuenta que los protocolos sobre los cuales se sustenta la actual Internet (TCP/IP) fueron diseñados en la década de los setenta para poner en contacto a un conjunto de centros militares y de investigación de los Estados Unidos. La red era un recurso controlado que formaba parte del perímetro de seguridad y, en consecuencia, sólo se consideró prioritario que los protocolos proporcionaran las mejores prestaciones posibles en lo que a conectividad se refiere. El escenario que encontramos actualmente es muy distinto, ya que los mismos protocolos han seguido empleándose para crear una red pública de escala mundial, lo que ha sacado a relucir las vulnerabilidades inherentes al diseño inicial de Internet. En segundo lugar, los ordenadores personales fueron concebidos como una herramienta de trabajo, monousuario y sin previsión de ser conectados entre sí a través de una red de comunicación. Hoy en día, el panorama es completamente distinto, las posibilidades de este tipo de equipos parecen ser cada vez más ilimitadas y no se concibe el uso de un ordenador como un

elemento aislado del resto del mundo. Ha sido durante esta etapa cuando más esfuerzos ha realizado la comunidad científica a la hora de proporcionar algoritmos, protocolos, servicios y aplicaciones de seguridad capaces de afrontar las amenazas presentes, algunas heredadas de los primeros diseños y otras surgidas tras la creación de nuevos servicios.

La tercera etapa, la cual forma parte del mañana más cercano, plantea un escenario de millones de procesadores que formarán parte de los objetos más cotidianos y que harán uso de la tecnología de transmisión inalámbrica. Es un hecho que durante los próximos años habrá más teléfonos móviles conectados a Internet que ordenadores personales, y que la tecnología empieza a hacer realidad el concepto de computación ubicua, es decir, la interacción espontánea entre dispositivos digitales cuya principal misión es proporcionarnos servicio (electrodomésticos, alarmas, coches, etc.). No es una precipitación aventurar que la computación ubicua puede llegar a tener un impacto sobre la sociedad de dimensiones similares a las causadas por el nacimiento del Web. Sin embargo, hemos de tener en cuenta también los innumerables riesgos que comporta la adopción de este tipo de tecnologías, así como las repercusiones derivadas de la explotación de sus vulnerabilidades. Esto conlleva una gran responsabilidad para los miembros de la comunidad científica, los cuales deben estudiar en profundidad todos los aspectos relacionados con la seguridad antes de que las aplicaciones y servicios sean desarrollados y puestos en marcha en entornos reales.

A pesar de esta evolución de los sistemas informáticos y de su heterogeneidad, todos ellos comparten la necesidad de disponer del mismo conjunto de servicios de seguridad que permitan proteger tanto a los usuarios como a los datos y equipos implicados. Dicho conjunto lo forman los servicios de *confidencialidad*, *integridad*, *disponibilidad* y *no repudio*:

- El término confidencialidad hace referencia a la imposibilidad de acceder a información protegida por parte de todas aquellas entidades que no han sido autorizadas para tal efecto [21]. Dicha definición se aplica tanto a la información que pueda encontrarse almacenada en algún componente del sistema como a aquellos datos que son transmitidos a través de una red de comunicaciones.

- El servicio de integridad proporciona los mecanismos necesarios para detectar cualquier posible modificación o eliminación de información llevada a cabo por parte de alguna entidad no autorizada [159]. Para ello, se habilitan tanto mecanismos de prevención como de comprobación y recuperación de los datos involucrados.

- El término disponibilidad hace referencia al grado en el que un sistema o componente está operativo y accesible cuando es necesario hacer uso del mismo [159].

- El término no repudio ha sido adoptado de la literatura científica, donde originalmente hacía referencia a la imposibilidad de falsificar una firma digital por parte de una tercera entidad [71]. Hoy en día, el concepto del término se ha extendido hasta ser definido como la garantía de que tanto el emisor como el receptor de un mensaje poseen las evidencias necesarias como para que ninguno de ellos pueda negar su participación en la comunicación.

Como se puede apreciar, la protección suele estar basada en el hecho de poder diferenciar entre las entidades que han sido autorizadas y las que no. Discriminar entre ambos conjuntos conlleva normalmente la realización de un proceso que puede dividirse en tres etapas distintas: *identificación* (proceso mediante el cual un sistema de información suele reconocer a una entidad), *autenticación* (medida de seguridad diseñada para establecer la validez de una transmisión, mensaje o participante) y *autorización* (asignación de privilegios de acceso a usuarios, programas o procesos).

En otro orden de cosas, si analizamos los distintos niveles de abstracción de un sistema informático a los cuales se les ha dotado con alguno de los cuatro servicios básicos de seguridad comentados, apreciaremos que dicha protección abarca campos tan distintos como el diseño de coprocesadores o la computación distribuida basada en componentes software, lo cual indica que se trata también de una cuestión transversal no sólo en lo que a lo que a diseño de un componente específico se trata sino también respecto al conjunto de elementos que componen un sistema de información. En los últimos años hemos asistido a grandes avances en lo que respecta a cuestiones, pertenecientes a niveles de abstracción tan dispares, como el diseño de coprocesadores seguros que permiten detectar copias ilegales de código [63], técnicas de autenticación de memoria externa que permiten detectar las modificaciones que puedan haberse realizado sobre los datos durante su camino entre procesador y la memoria [83, 127], mecanismos que permiten administrar de forma remota y segura dispositivos autónomos [130, 183], protocolos seguros de comunicación asociados a los distintos niveles que forman parte de una arquitectura de red [9, 110, 111, 168], esquemas que permiten pagar de forma electrónica los bienes adquiridos a través de la red [142], y un larguísimo etcétera de propuestas que demuestran que la seguridad es un aspecto crucial independientemente del ámbito en el cual se encuentre ubicada.

Sin embargo, respecto a esa carrera anteriormente mencionada entre conectividad y seguridad, nos encontramos aún lejos de poder contemplar cómo ambos conceptos evolucionan a la par. La comunidad científica debe seguir realizando numerosas aportaciones que permitan que paulatinamente el grado de seguridad percibida en los sistemas informáticos alcance las cotas necesarias para trasladar a los usuarios la percepción de un concepto clave, la confianza.

1.2 Evolución de la certificación digital como herramienta de seguridad

Dos de las cuestiones a las que más respuestas se les ha intentado proporcionar durante la última década han sido las siguientes: por un lado, *¿quién es esta entidad?*, es decir, la necesidad de diseñar mecanismos de identificación y autenticación de entidades que permitan conocer quién se encuentra detrás de una determinada comunicación o información; por otro lado, una vez resuelta la primera cuestión, el siguiente paso consiste en responder a la pregunta *¿qué está autorizada a hacer esta entidad?*, esto es, determinar el conjunto de privilegios que tiene asociados una entidad con el fin de determinar si tiene derecho a

realizar la acción que está solicitando.

De entre el conjunto de técnicas que se han propuesto para resolver estas cuestiones, constituirá la piedra angular de este trabajo el uso de la certificación digital como herramienta de seguridad. Las aportaciones realizadas dentro del marco de esta tesis están íntimamente ligadas a las distintas tecnologías de certificación que han ido surgiendo durante los últimos años, las cuales se basan a su vez en la criptografía como herramienta fundamental. En consecuencia, a lo largo de esta sección se realizará una breve introducción de los conceptos fundamentales sobre criptografía y se comentará la evolución producida en el campo de la certificación digital, desde su concepción inicial como un mecanismo de definición de identidades hasta sus connotaciones más recientes relacionadas con la gestión de privilegios.

1.2.1 El papel de la criptografía

El término criptografía, procedente del griego *kriptos grafein* (escritura oculta), hace referencia al conjunto de técnicas y métodos que tienen como objetivo principal la transformación de información en una codificación que resulte ilegible para todas aquellas entidades que desconozcan alguno de los parámetros involucrados en dicha transformación. Se trata de una necesidad básica en el ámbito de la protección de las comunicaciones, al principio circunscrita al ámbito militar y diplomático, y que ha tomado gran importancia con el auge de las redes de comunicaciones.

Los elementos de un sistema criptográfico son los métodos utilizados para el cifrado y descifrado de información (los cuales pueden coincidir), la clave empleada como parámetro de dichos métodos, y el espacio en el cual se encuentran definidos tanto el mensaje a codificar (denominado también *texto en claro*) como su representación codificada (denominada también *criptograma*). En función de que la clave empleada para cifrar coincida o no con la clave utilizada para descifrar, los sistemas criptográficos se agrupan en dos grandes bloques: criptosistemas simétricos y criptosistemas asimétricos.

La *criptografía simétrica* se caracteriza por emplear la misma clave tanto para cifrar como para descifrar un mensaje. En la mayor parte de los casos, el parámetro que se supone secreto para que la información sea protegida de la interceptación de terceras personas es la clave empleada. Es un hecho contrastado que la seguridad de un criptosistema no puede recaer en el desconocimiento por parte de la comunidad del funcionamiento interno de los algoritmos de cifrado o descifrado. Un exponente de la robustez del sistema es que dichos algoritmos sean públicos, de forma que su fortaleza pueda ser sometida a escrutinio público y que la seguridad del sistema recaiga solamente en la no revelación de la clave empleada.

A este tipo de criptosistemas se les conoce también como sistemas de secreto compartido, y es que su correcto funcionamiento recae en la distribución confidencial de la clave a las entidades que formarán parte de la comunicación. Sin embargo, es la propia necesidad de compartir dicho secreto la que limita en muchas ocasiones el uso de este tipo de criptografía a la hora de poner en contacto entidades sin ningún tipo de relación previa. El uso aislado de la criptografía simétrica queda reducido a aquellos ámbitos en los cuales los participantes disponen de algún tipo de medio de comunicación externo mediante el cual

puedan realizar una transmisión confidencial previa del secreto a compartir. Otra posibilidad es emplear los denominados centros de distribución de claves (KDC, Key Distribution Center), los cuales actúan como terceras partes confiables a la hora de suministrar las claves que protegerán las futuras comunicaciones a realizar por parte de las entidades del sistema. No obstante, el uso de KDCs plantea también serios inconvenientes en lo que se respecta a privacidad, suplantación de identidad o disponibilidad [181].

El hecho de que coincida la clave empleada para cifrar y descifrar conlleva también otra serie de limitaciones en la aplicación de este tipo de criptografía. Por un lado, es necesario generar una clave distinta por cada par de entidades que deseen proteger su comunicación. En consecuencia, dado un sistema con n usuarios finales, el número total de claves necesarias para poner en conectar a todos los usuarios entre sí es del orden $O(n^2)$, lo cual puede llegar a limitar su escalabilidad. Por otra parte, el hecho de que dos usuarios compartan el mismo secreto hace imposible determinar con seguridad quién cifró o descifró una determinada información, lo cual impide que pueda ser utilizada como mecanismo de no repudio o de autenticación de la identidad.

La historia de la certificación digital comienza a germinarse a mediados de la década de los setenta, cuando Whitfield Diffie y Martin Hellman presentan un artículo titulado *"New Directions in Cryptography"* [60] que introduce el concepto de *criptografía asimétrica* o *criptografía de clave pública*. Su novedad fundamental con respecto a la criptografía simétrica es que las claves no son únicas, sino que forman pares compuestos por una *clave privada* y una *clave pública*.

Cada usuario del sistema dispone de un par de claves único y, a diferencia de lo que sucedía con la criptografía simétrica, la clave privada no es un secreto compartido sino que debe ser protegida por cada usuario. Sin embargo, la clave pública debe difundirse con el fin de que otras entidades puedan emplearla para proteger las comunicaciones realizadas con el usuario en cuestión. Este tipo de criptosistema se basa en el hecho de que resulta computacionalmente intratable intentar descubrir una clave a partir del conocimiento de la otra, lo cual anula la necesidad de establecer secretos compartidos entre entidades ya que basta con tener acceso a las claves públicas.

Lo que resulta interesante del uso de muchos de los sistemas basados en este tipo de criptografía es que las operaciones realizadas con una de las claves pueden revertirse empleando la otra. Por ejemplo, en el caso de que cierta información se cifre utilizando la clave pública de un usuario ésta podrá ser descifrada empleando la clave privada. Dado que sólo el usuario tiene acceso a dicha clave, obtenemos de esta forma un medio para proteger la *confidencialidad* de la información. Por otra parte, el cifrado realizado mediante la clave privada también puede deshacerse empleando la clave pública. Aunque esta operación carece de interés desde el punto de vista de la confidencialidad dado que la clave de descifrado es pública, y por tanto conocida por todos, representa un mecanismo muy robusto de *autenticación*. La razón es que sólo hay un usuario capaz de cifrar información que podrá ser descifrada posteriormente empleando la clave pública: el poseedor de la clave privada. Esta técnica, combinada con el uso de funciones de resumen digital, es lo que se ha dado a conocer como mecanismo de *firma digital*, ya que además de autenticación es capaz de proporcionar también los servicios básicos de *integridad* y *no repudio*.

Sin embargo, el hecho de que una clave sea pública, y que por tanto no sea necesario acceder a ella haciendo uso de un canal adicional, no implica que ésta sea auténtica. Es decir, resulta vital tener la certeza de que la clave pública pertenece a la entidad con la cual se desea establecer contacto. Un error en la asociación entre la identidad del usuario y el valor de su clave pública puede conllevar la transmisión de información sensible a terceras partes o la asociación de cierta información a la entidad equivocada. El artículo de Diffie y Hellman [60] proponía la utilización de un Archivo Público (Public File) que sería consultado por los usuarios para averiguar las claves públicas del resto de entidades del sistema. Para evitar que un atacante pudiera hacerse pasar por el archivo, todas las comunicaciones llevadas a cabo por éste debían estar firmadas digitalmente. Sin embargo, esta propuesta inicial presenta numerosos inconvenientes tanto desde el punto de vista del rendimiento como de la seguridad: el servicio puede convertirse en un cuello de botella que degradaría el rendimiento global del sistema; constituye un objetivo claro para cualquier tipo de ataque de denegación de servicio; cualquier modificación no detectada de los datos contenidos puede tener graves consecuencias para el resto de los usuarios.

Esta necesidad de asociar de forma confiable las claves públicas de los usuarios a su identidad nos lleva a los orígenes de la certificación digital. Como se verá en los siguientes apartados, la certificación digital resuelve satisfactoriamente éste y otros problemas relacionados con la gestión de privilegios, lo cual la define como una herramienta esencial de seguridad.

1.2.2 Certificación de la identidad

Consciente de las limitaciones del Archivo Público de claves, Loren Kohnfelder propuso en 1978 el concepto de *certificado digital de clave pública* o *certificado de identidad* [115]. Kohnfelder argumentaba que sólo hay dos formas posibles de realizar de forma segura la adquisición de las claves públicas, bien directamente mediante los usuarios implicados o bien a través de una tercera entidad confiable. Dados los numerosos inconvenientes asociados al Archivo Público, introdujo el concepto de certificado como un documento firmado digitalmente que contiene tanto la clave pública como la identidad del poseedor de la clave privada correspondiente. La diferencia principal respecto a la propuesta de Diffie y Hellman era la no dependencia de una conexión con la base de datos centralizada de claves, sino la posibilidad de que los certificados fueran emitidos por cualquier otra entidad que disfrutara de la confianza de los usuarios implicados, a la cual se le denominó *Autoridad de Certificación* (CA, Certification Authority). Realmente, este esquema de generación de documentos de identidad por parte de una tercera entidad confiable es muy común en otros aspectos de la vida real. Un ejemplo de ello es el documento nacional de identidad (DNI) emitido por el Ministerio del Interior.

El hecho de que un certificado se encuentre firmado digitalmente por una autoridad de certificación proporciona al documento un mecanismo de verificación de su integridad, lo cual hace que no sea necesario protegerlo. Esto conlleva que pueda ser almacenado en repositorios de información no confiables que podrán ser replicados y que deberán preocuparse simplemente de asegurar su disponibilidad.

Una década después de que Kohnfelder definiera el término, el estándar X.500 [38] incorporó el uso de certificados digitales X.509 [99] en su propuesta de un directorio global de entidades. De dicha propuesta nació también el concepto de *infraestructura de clave pública* (PKI, Public Key Infrastructure), el cual hace referencia al conjunto de elementos y procedimientos relacionados con la gestión del ciclo de vida de un certificado digital. Las infraestructuras de clave pública son elementos clave a la hora de dotar al sistema de la capacidad de gestionar todos aquellos aspectos implicados en la creación, publicación, renovación, validación y revocación de certificados.

A pesar de su importancia y del apogeo del que fueron protagonistas a finales de la década de los noventa, actualmente la implantación de las PKIs parece encontrarse en un estado de estancamiento [90], quizá debido a la multitud de mitos y falsedades [71] procedentes principalmente del sector privado. Durante un tiempo se dijo que las PKIs eran una necesidad imperiosa para que el comercio electrónico pudiera florecer. Posteriormente, se ha comprobado que el comercio electrónico ha ido evolucionando, a una velocidad más lenta de la que en un principio se esperó, sin la existencia de dichas PKIs. Posiblemente habría sido más correcto enunciar que eran las PKIs comerciales las que necesitaban el comercio electrónico para florecer.

Entre las causas del estancamiento de las PKIs podríamos encontrar dos grandes bloques, las relacionadas con la adopción de mecanismos que resultan obsoletos hoy en día y las asociadas a la falta de desarrollos que sean capaces de ofrecer algunos servicios básicos de seguridad no contemplados por las PKIs tradicionales, como por ejemplo la autorización y el control de acceso.

En relación con la herencia de mecanismos obsoletos, se puede afirmar que la tendencia general del mercado ha sido intentar adaptar el mundo real al diseño de las PKIs basadas en X.509 en lugar de realizar el esfuerzo de adaptar dicha tecnología a las necesidades del mercado. Por ejemplo, la idea del directorio global X.500 es uno de los ejes sobre los cuales se ha basado siempre el estándar X.509. Sin embargo, a día de hoy, dicho directorio no es una realidad, y su filosofía de identificación de entidades ha demostrado no ser la más apropiada [67]. Otro ejemplo lo constituye el uso de las listas de certificados revocados (CRL, Certificate Revocation Lists), las cuales son una adaptación de las listas negras de tarjetas de crédito que empleaban los comerciantes en los años setenta para detectar fraudes. Este tipo de sentencias negativas no responden a la pregunta de si un certificado sigue siendo válido, sólo son capaces de indicar si ha sido revocado, lo cual no es exactamente lo contrario. Como consecuencia general, los nuevos diseños de PKIs deben innovar ciertos aspectos de su funcionamiento y aportar nuevos servicios que permitan realizar una gestión más versátil y eficiente de los certificados digitales.

Respecto a las limitaciones en el campo de la autorización, hemos de tener en cuenta que el principal objetivo de las PKIs ha sido proporcionar mecanismos que permitieran establecer una relación entre el nombre y las claves públicas de las entidades. Sin embargo, el nombre no es más que un índice, un valor al cual habrá que asociar posteriormente una serie de atributos con el fin de determinar de qué privilegios dispone el usuario correspondiente (por ejemplo, si puede acceder a un determinado fichero, si tiene saldo en su cuenta corriente, si puede acceder a un laboratorio tras validarse frente a un dispositivo de

control, si tiene derecho a obtener un descuento en la compra de un producto, etc.). Tradicionalmente, la asignación de privilegios ha sido tarea de las aplicaciones finales, las cuales debían especificar y validar dichos permisos utilizando sus propios criterios y mecanismos. Como se verá a continuación, los nuevos enfoques de certificación de privilegios aportan un nuevo abanico de posibilidades a la hora de llevar a cabo los procesos relacionados con la autorización.

1.2.3 Certificación de los privilegios

Aunque los términos "certificado"y "certificado de identidad"se han venido utilizando como sinónimos, lo cierto es que un certificado digital puede interpretarse como un documento que recoge cierta información acerca de la entidad para la cual fue emitido. Dicha información no tiene que restringirse sólo al ámbito de la identificación, sino que puede tratarse de cualquier tipo de atributo o cualidad que desee ligarse a la entidad, como por ejemplo el conjunto de roles a los que pertenece dentro de una organización o los privilegios de los cuales disfruta dentro de un sistema. Este tipo de certificados que asocian competencias o capacidades a claves públicas se conocen con el nombre de certificados de atributo o autorización, y más genéricamente como *certificados de credencial* o simplemente *credenciales*.

Los certificados de credencial son documentos que pueden estar ligados a los certificados de identidad, pero que tienen una gestión totalmente independiente. La razón de que los privilegios aparezcan reflejados en documentos independientes al certificado de identidad está justificada por dos motivos. En primer lugar, hemos de tener en cuenta que la asignación y revocación de privilegios suele ser más dinámica que la gestión de la identidad, lo cual hace poco apropiado insertarlos en un certificado de este tipo ya que conllevaría la continua actualización del mismo. En segundo lugar, hemos de considerar que la delimitación de responsabilidades dentro de una organización puede propiciar que no resulte apropiado que una única entidad sea la encargada de determinar tanto la identidad como los privilegios de los usuarios. Normalmente, los certificados de credencial tienen una validez local, restringida a un subconjunto de elementos del sistema, lo cual hace que resulte conveniente que puedan ser gestionados de forma distribuida por autoridades de autorización locales (quizá esto último se entienda mejor extrapolándolo al mundo real, donde el conjunto de cualidades de una persona no se incluye como parte de su pasaporte, sino que están representadas por documentos independientes emitidos por otras entidades, como es el caso de un permiso de conducir, el carné de un club deportivo, una tarjeta universitaria o una tarjeta de crédito).

Los certificados de credencial constituyen una herramienta crucial en el desarrollo de sistemas de control de acceso descentralizados. Si las autoridades que los emiten se consideran confiables, las credenciales son pruebas suficientes para determinar el conjunto de privilegios asociados a una entidad, y por tanto para decidir si una determinada solicitud debe ser aprobada o denegada.

A lo largo de los últimos años, varias han sido las especificaciones en materia de certificados de credencial que han sido propuestas por parte de la comunidad científica [27, 69, 106]. Todas ellas se caracterizan por proponer mecanismos concretos de gestión de pertenencia

a grupos y de especificación de privilegios. Además, algunas proporcionan métodos genéricos de toma de decisiones de autorización e introducen el mecanismo de delegación de privilegios como herramienta fundamental de gestión de la autorización.

Sin embargo, es un campo abierto de investigación el diseño e implementación de mecanismos de gestión del ciclo de vida de este tipo de certificados. Actualmente, no existe un modelo claro que especifique cómo debe realizarse tanto el proceso de especificación de políticas de autorización como el control de su cumplimiento. Del mismo modo, son líneas de investigación activas las relativas a la transmisión, almacenamiento y revocación de credenciales. Algunas de estas cuestiones, junto con otras enumeradas en el apartado anterior, forman parte del ámbito en el cual se encuentran ubicadas las aportaciones de este trabajo de tesis.

1.3 Objetivos y aportaciones propias

El trabajo aquí presentado tiene tres objetivos fundamentales. Por un lado, la propuesta de una infraestructura de clave pública versátil que sea capaz de proporcionar mecanismos avanzados de gestión, los cuales permitirán integrar dicha infraestructura en escenarios de aplicación con requisitos muy diversos. En segundo lugar, extender la PKI mediante un mecanismo de gestión distribuida de credenciales capaz de dotar al sistema de los servicios básicos de autorización y control de acceso. Por último, ilustrar cómo es posible integrar las propuestas anteriores en entornos de aplicación reales, siguiendo además un enfoque metodológico estructurado.

Para la consecución del primer objetivo, este trabajo de tesis realiza las siguientes aportaciones:

- *Diseño de un sistema de certificación avanzado.* El diseño de la infraestructura propuesta en la sección 3.2 presenta algunas diferencias respecto a otros esquemas tradicionales, sobre todo en lo que respecta a la versatilidad ofrecida a la hora de ofrecer sus servicios y en la visión unificada de su gestión.

- *Definición de un mecanismo de políticas de gestión de PKIs.* Se trata de una de las aportaciones más innovadoras en lo que respecta al bloque de certificación de identidad. Las políticas permiten reflejar las condiciones expresadas en las prácticas de certificación y asegurar su cumplimiento. Como se verá en la sección 3.4, la aportación concreta consiste en la definición de dichas políticas y la provisión de los mecanismos necesarios para su edición, distribución y aplicación.

- *Propuestas avanzadas de revocación y validación de certificados.* Este trabajo, en su sección 3.5, presta especial atención a dos de las operaciones que más han sido descuidadas tradicionalmente. Por un lado, ofrece diversas soluciones al problema de la revocación, especialmente desde el punto de vista de la disponibilidad y versatilidad del servicio. En segundo lugar, ofrece mecanismos alternativos de validación de certificados basados en sentencias positivas, es decir, en la construcción de documentos que demuestren por sí mismos la validez de un certificado.

Una vez definida la infraestructura que proporcionará los servicios de identificación digital, el siguiente paso será llevar a cabo su extensión para dotar al sistema de servicios de autorización. La consecución de este objetivo abarca la mayor parte de las aportaciones propias de este trabajo, las cuales pueden agruparse de la forma siguiente:

- *Análisis del control de acceso basado en delegación.* La sección 4.4 presenta un análisis cuya meta principal es identificar tanto los aspectos a incluir en una infraestructura de autorización como las principales carencias y oportunidades de este tipo de entornos de cara a proponer soluciones reales.

- *Propuesta de un marco de intercambio de información de autorización.* Se trata de definir una propuesta que permita realizar la transmisión segura de información relativa a autorización (credenciales, políticas, solicitudes, etc.). El marco presentado en la sección 5.2 puede adaptarse a entornos caracterizados por distintos parámetros, como por ejemplo el método de distribución de credenciales, la estrategia de revelación de políticas o la optimización del proceso de solicitud.

- *Propuesta de un sistema distribuido de gestión de credenciales.* Esta propuesta constituye una de las aportaciones cruciales de este trabajo de tesis. La sección 5.3 presenta una infraestructura de autorización que permite extender los servicios ofrecidos por la PKI, de tal forma que es posible gestionar el conjunto de credenciales de un sistema concreto haciendo uso de los conceptos de delegación y control de acceso basado en roles. La propuesta contempla tanto la arquitectura del sistema como la definición de todos los elementos de información involucrados en el proceso de generación de credenciales.

  El último objetivo global es la definición de un marco metodológico que permita integrar las soluciones aportadas en escenarios reales. En relación con ello, las aportaciones realizadas son las siguientes:

- *Metodología de definición de estructuras de gestión.* La metodología propuesta en la sección 5.4 permite abordar la puesta en marcha de un escenario de autorización siguiendo un enfoque estructurado basado en la identificación de los distintos elementos que compondrán el sistema y la relación entre los mismos.

- *Integración en escenarios de aplicación concretos.* La viabilidad de todos los mecanismos de autorización ha sido verificada mediante su integración en escenarios de aplicación concretos. El capítulo 6 aporta un punto de vista práctico de las propuestas realizadas, el cual permite comprobar cómo tanto la infraestructura de clave pública como las aportaciones realizadas en materia de autorización son capaces de proporcionar soluciones reales a escenarios concretos.

Como consecuencia de dichas aportaciones, este trabajo de tesis ha permitido definir una infraestructura completa de servicios de identificación y autorización basada en el uso de la certificación digital como herramienta esencial.

1.4 Desarrollo de la Tesis

Este primer capítulo ha presentado el contexto en el cual se encuadra la tesis, haciendo especial énfasis en la certificación digital como línea argumental. Además, se han definido los objetivos principales del trabajo y se han enumerado las principales aportaciones realizadas.

El capítulo 2 muestra todos aquellos aspectos relacionados con la certificación de identidad desde el punto de vista de los modelos de confianza propuestos, el formato de los certificados y la gestión del ciclo de vida. Se recogen las principales especificaciones realizadas por la comunidad científica y se presentan algunas de las iniciativas de certificación más importantes que se han desarrollado a nivel tanto internacional como nacional, así como las propuestas que han surgido dentro del seno del grupo de investigación.

En el capítulo 3 se detalla el diseño de la PKI que forma parte del marco de esta tesis. Para ello, se introduce el diseño general de la PKI, es decir, sus elementos constituyentes y la relación entre los mismos. A continuación, se describen las operaciones básicas de gestión realizadas por dicha infraestructura. Por último, se detallan 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.

El capítulo 4 introduce la certificación de privilegios. En primer lugar se identifican las principales carencias de los sistemas tradicionales de certificación de identidad en materia de control de acceso. Posteriormente, se analizan los diferentes modelos de control de acceso que han surgido a lo largo del tiempo. A continuación, se exponen las diferentes especificaciones existentes en materia de certificados de credencial y se realiza un estudio acerca del estado del arte de la delegación en sistemas distribuidos como mecanismo de gestión de autorizaciones. El capítulo concluye con la identificación de las propuestas en materia de autorización que forman parte de esta tesis.

El capítulo 5 presenta los detalles relativos a los componentes de la infraestructura de autorización. En primer lugar introduce el marco de intercambio de información relativa a autorizaciones, tanto su diseño general como el protocolo que implementa las recomendaciones. A continuación, se describen tanto las entidades como las especificaciones relativas al sistema de gestión distribuida de credenciales basado en delegación y roles. Por último, el capítulo concluye con la presentación de la metodología que permitirá afrontar la puesta en marcha de un sistema de control de acceso de forma estructurada y haciendo uso de la infraestructura de autorización.

En el capítulo 6 se analiza la viabilidad de las propuestas formuladas en la tesis. Para ello, en primer lugar, se comentan los detalles relativos a la implementación tanto del marco de intercambio como del sistema de gestión de credenciales. A continuación, se describe cómo se integran parte de las aportaciones realizadas en dos escenarios de aplicación, concretamente en un entorno de control de acceso físico y en un sistema de suscripción electrónica basada en un protocolo seguro de pagos. Por último, se realiza un análisis del rendimiento de las propuestas con objeto de extraer conclusiones acerca de la sobrecarga que pueden llegar a introducir dentro de cualquier escenario de autorización.

El capítulo 7 presenta las conclusiones derivadas de este trabajo y las posibles vías de

investigación que quedan abiertas a partir de lo realizado.

El documento incluye además tres apéndices. El apéndice A contiene la especificación completa de los elementos de política empleados para gestionar la PKI descrita en el capítulo 3. El apéndice B proporciona todos los detalles relativos a los mensajes del protocolo de intercambio de autorizaciones. Finalmente, el apéndice C contiene la especificación de los elementos de información definidos por el sistema de gestión distribuida de credenciales.

Capítulo 2

Ciclo de vida de la identidad digital: infraestructuras de clave pública

El objetivo principal de este capítulo es mostrar todos aquellos aspectos relacionados con la certificación de identidad, tanto desde el punto de vista del formato de sus certificados como de la gestión de su ciclo de vida. Para ello, inicialmente se expondrán los principales estándares relacionados con este tipo de certificación y se analizarán sus modelos de confianza correspondientes. A continuación, se detallará el formato de los certificados X.509, estándar de certificación que más reconocimiento ha alcanzado entre la comunidad científica y empresarial. Posteriormente, se analizarán cuáles son las operaciones implicadas en el ciclo de vida de un certificado de identidad y se recogerán las principales recomendaciones realizadas por la comunidad científica en dicha materia. Por último, se presentarán algunas de las iniciativas de certificación más importantes que se han desarrollado a nivel tanto internacional como nacional, así como las propuestas que han surgido dentro del seno del grupo de investigación.

2.1 Estándares de certificación digital de identidad

Tal y como se ha comentado en el capítulo anterior, a lo largo de los últimos años son muchos los esquemas de certificación de identidad que han ido surgiendo. De entre todos ellos, sólo dos han obtenido una gran aceptación por parte de la comunidad de Internet y de los organismos gubernamentales: el estándar X.509 [99] y el sistema PGP (Pretty Good Privacy) [37]. Otros esquemas que también ofrecen servicios de certificación de identidad, como SPKI [69] (Simple Public Key Infrastructure) o SDSI [170] (Simple Distributed Security Infrastructure), serán analizados en capítulos posteriores como parte de infraestructuras más complejas que abarcan además servicios de autorización, definición de grupos, asignación de permisos a roles, etc. En esta sección, no se entrará en los detalles acerca de la estructura de los certificados X.509 o PGP, sino que se analizará su propuesta desde el punto de vista de las relaciones de confianza existentes entre los participantes, su adecuación a los principales entornos de aplicación de Internet y su grado de implantación.

2.1.1 Modelos de confianza

Antes de analizar los modelos de confianza de ambos sistemas, quizá sería necesario empezar definiendo qué entendemos por dichos modelos. En torno al concepto de confianza se han realizado numerosas definiciones, clasificaciones, recomendaciones y formalizaciones por parte de la comunidad científica en los últimos años [3, 108, 135].

Según Gambetta [82], *"la confianza es un nivel subjetivo de probabilidad con la cual un agente realizará una acción concreta"*. Es decir, en cierto sentido, es la cantidad de riesgo que estamos dispuestos a asumir de que una determinada acción en un determinado instante pueda realizarse de forma incorrecta. Hay tres puntos importantes derivados de este tipo de definiciones: el primero es que la confianza es subjetiva; el segundo es que afecta a aquellas acciones que no podemos controlar; el último es que el nivel de confianza depende de cómo nuestros actos se vean afectados por el comportamiento del agente en el cual confiamos.

Entendemos por tanto que por modelos de confianza de los sistemas de certificación se hace referencia al tipo de relación que se establece entre las entidades emisoras de certificados, los poseedores de dichos certificados y las entidades encargadas de verificar los documentos relacionados con los mismos. X.509 y PGP tienen modelos de confianza muy distintos, tanto en lo que a relación entre entidades certificadoras y entidades certificadas se refiere, como entre las propias entidades certificadoras.

Modelo basado en autoridades de certificación específicas

El estándar X.509 considera que los certificados deben ser emitidos por entidades especiales, a las cuales denomina autoridades de certificación, que tienen la potestad especial de poder emitir certificados que serán considerados como válidos por parte de una comunidad de usuarios más o menos extensa. Se tiene así una relación asimétrica de confianza entre las entidades del sistema, donde sólo algunas de ellas tienen la capacidad de crear nuevas identidades digitales. Es un modelo donde la confianza le viene impuesta tanto a los poseedores de certificados como a las entidades encargadas de verificarlos. Cada certificado está firmado por una, y sólo una, autoridad de certificación. El modelo es muy similar al empleado en el ámbito gubernamental, donde una entidad, el Estado, es la encargada de emitir documentos que son considerados como válidos por el resto de las entidades. Por supuesto, X.509 no restringe el modelo a la existencia de una única autoridad de certificación mundial, sino que contempla la posibilidad de que muchas autoridades de certificación independientes puedan operar de forma simultánea. La existencia de relaciones de confianza entre las distintas autoridades de certificación es lo que da lugar a las distintas configuraciones posibles de confianza entre entidades emisoras: modelo jerárquico, certificación cruzada y modelo basado en autoridad de certificación puente [79].

En el modelo jerárquico mostrado en la figura 2.1 se puede observar la existencia de una autoridad de certificación raíz cuya clave pública está contenida en un certificado autofirmado, el cual debe ser distribuido de forma confiable a todas las entidades del sistema ya que no proporciona de por sí autenticación, sino sólo integridad sobre la información

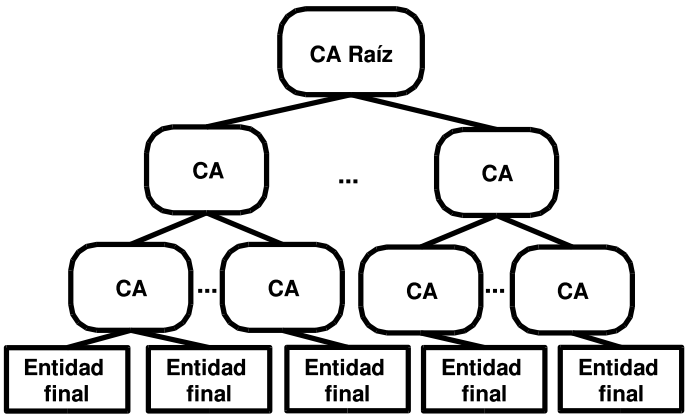

Figura 2.1: Modelo de confianza jerárquico

contenida. El resto de autoridades de certificación están subordinadas, lo que requiere que establezcan algún tipo de relación de dependencia con respecto a una autoridad de mayor nivel. Este esquema plantea algunas desventajas importantes: en primer lugar, la clave privada de la autoridad raíz representa un punto de ataque potencial que puede tener consecuencias sobre todos los certificados del sistema, puesto que en caso de compromiso de dicha clave, todos los certificados del sistema deberían ser revocados y refirmados; en segundo lugar, el enfoque jerárquico implica una cierta relación de dominancia o subordinación entre las organizaciones a las cuales están ligadas las autoridades de certificación, lo cual no es siempre cierto.

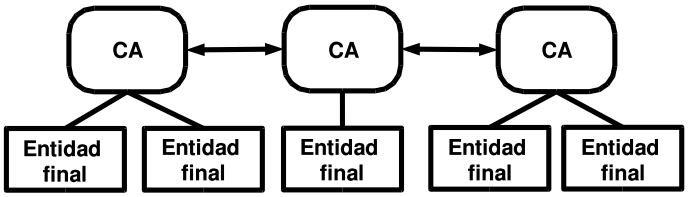

Figura 2.2: Modelo de certificación cruzada

Cuando las autoridades de certificación operan bajo la premisa de una relación de igualdad, el modelo jerárquico puede ser sustituido por la certificación cruzada o certificación punto a punto. Según este modelo, cada par de autoridades de certificación que desean establecer una relación de confianza, intercambian sus claves públicas y se certifican la una a la otra. La figura 2.2 muestra tres autoridades donde la primera y la segunda tienen una certificación cruzada, al igual que la segunda y la tercera. Con este modelo, cada autoridad de certificación es como una autoridad raíz.

Sin embargo, el modelo de la certificación cruzada puede llegar a generar del orden de $O(n^2)$ certificados en el caso de querer establecer relaciones de confianza entre un conjunto de N autoridades. La figura 2.3 muestra un esquema de certificación donde las autoridades no se certifican entre sí, sino respecto a una entidad central llamada autoridad de certificación puente [10], lo cual en el caso de disponer de N autoridades de certificación requiere

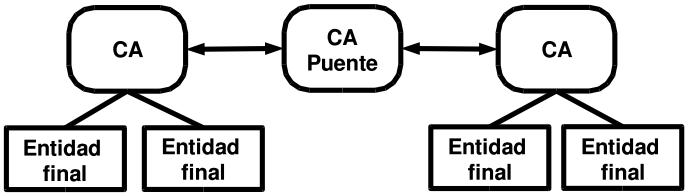

Figura 2.3: Modelo de autoridad de certificación puente

sólo N certificados punto a punto, un número sustancialmente menor que el necesario para el caso de la certificación cruzada.

Telaraña de confianza (web of trust)

En PGP no hay concepto de autoridad de certificación como tal; cualquier usuario puede certificar la clave pública de cualquier otro usuario PGP. Sin embargo, dicho certificado tendrá sólo validez para aquellas entidades que hayan decidido reconocer al signatario como certificador de confianza (*trusted introducer*, en terminología PGP). De esta forma, los usuarios PGP pueden construir caminos de certificación arbitrarios a través de toda la comunidad PGP, de ahí que este modelo de confianza se suela denominar *telaraña de confianza*.

El proceso de gestión de certificados es totalmente manual. Cada usuario dispone de una colección de claves públicas a las cuales se les asocia dos indicadores: uno que ilustra si la clave se considera válida, o no; el otro indica el nivel de confianza que se deposita en dicha clave con el propósito de que sirva como certificador de otras claves adquiridas en el futuro. El hecho de que una clave sea válida, o no, es independiente de que sea confiable como autoridad de certificación.

De hecho, mediante PGP sería posible simular el modelo de confianza de X.509 basado en autoridades de certificación específicas. Para ello, los usuarios tendrían que aceptar como certificadores válidos a aquellas entidades que asumirían el mismo rol que tiene una autoridad de certificación X.509, y considerar como no confiables al resto de claves.

2.1.2 Adecuación a entornos de aplicación e implantación

El modelo de confianza de PGP permite que cualquier entidad actúe como una autoridad de certificación capaz de emitir certificados para cualquier otra entidad. Este modelo funciona muy bien con comunidades relativamente pequeñas en las que no hay demasiada interacción entre los individuos. Debe tenerse en cuenta que PGP nació como el propósito personal de Phil Zimmerman [189] de ofrecer un servicio de confidencialidad, integridad y autenticación para el correo electrónico, entorno al cual está completamente ligado PGP. El hecho de que muchos individuos deban realizar decisiones importantes acerca de la gestión de la confianza conlleva un alto riesgo, puesto que determinaciones incorrectas pueden afectar a la comunidad en general. Se puede afirmar que, en comunidades reducidas de usuarios PGP, donde los usuarios de las claves públicas tienen una relación muy directa con

los certificadores de dichas claves, el sistema puede resultar muy apropiado. Sin embargo, esta característica no se extiende a entornos más conflictivos, automatizados, o distribuidos, como es el caso del comercio electrónico o las comunicaciones móviles. Además, el sistema PGP está muy ligado a las direcciones de correo electrónico como mecanismo de identificación de usuarios y certificadores, lo cual hace excesivamente complejo adaptarlo a otros escenarios que hagan uso de otros esquemas de nombramiento o que necesiten asociar otro tipo de atributos a las claves públicas de las entidades participantes.

Por otro lado, el esquema centralizado en el cual se basa el estándar X.509 ha sido adoptado con mayor facilidad por gran parte de organismos gubernamentales y empresas privadas. Hemos de considerar que el uso de autoridades de certificación centralizadas se asemeja bastante a la mayoría de los esquemas de funcionamiento interno de las grandes organizaciones, ya que en estos casos la emisión de documentos oficiales, la toma de decisiones, o la certificación en general suele realizarse por un número reducido de entidades, claramente identificado, y de gran confianza dentro del sistema. Además, los modelos de interrelación entre autoridades de certificación derivados a partir del estándar X.509 logran reflejar de forma bastante fiel la naturaleza de las relaciones que se establecen entre distintas organizaciones, ya sea ésta de tipo jerárquica, punto a punto, o a través de entidades mediadoras. Por último, tal y como se verá en la sección 2.2, el formato del certificado X.509 ha permitido implantar este sistema de certificación en múltiples entornos además del correo electrónico seguro, como pueden ser la navegación web segura, la protección de comunicaciones en el nivel de red o la certificación de código ejecutable. Debido a este alto grado de implantación y de aceptación, este capítulo detallará las características de este sistema de certificación y expondrá las distintas propuestas realizadas en lo que a gestión del ciclo de vida de este tipo de certificados se refiere.

2.2 El estándar X.509

X.509 [106] es el marco de trabajo de autenticación que fue inicialmente diseñado para dar soporte a los servicios de directorio X.500 [38]. Tanto X.509 como X.500 son parte de las series X de los estándares internacionales propuestos por la ISO (International Organization for Standardization) y la ITU (International Telecommunication Union). Los estándares X.500 se diseñaron para proporcionar servicios de directorio a las grandes redes de ordenadores, mientras que X.509 proporcionó el marco para autenticar dichos servicios.

El formato de los certificados X.509 ha evolucionado a través de tres versiones en diferentes ediciones del estándar. X.509v1 fue diseñado en 1988 para certificar las claves públicas de aquellas entidades que tenían asociado, de forma única, un nombre X.500 (para más información acerca de los nombres X.500 ver sección 2.2.1). La segunda versión del estándar, propuesta en 1993, proporcionaba un rango mucho más flexible de identificadores que asociar a las entidades. X.509v3, publicado en 1997, mejoró enormemente la flexibilidad de los certificados mediante la provisión de un mecanismo genérico para añadir extensiones. Esta última versión permite el uso de nombres locales en los certificados, puesto que se ha reconocido que un esquema de nombramiento único mundial es inabordable.

Son muy numerosos los estándares y los productos de certificación que se han desarrollado a partir del marco X.509. X9.55 [11], por ejemplo, es un estándar ANSI (American National Standard Institute) desarrollado por la Asociación Americana de Bancos, el cual es muy similar a X.509 pero más enfocado al sector de los servicios financieros. Algunas de las implementaciones de los certificados X.509 incluyen los sistemas de correo electrónico seguro PEM (Privacy Enhanced Mail) [109] y S/MIME (Secure/Multipurpose Internet Mail Extensions) [168], el sistema Fortezza (el estándar para correo electrónico seguro y cifrado de ficheros adoptado por el Departamento de Defensa de los Estados Unidos), los protocolos de seguridad SSL versión 3 (Secure Socket Layer) [9] y TLS (Transport Level Security) [59], y el protocolo de pago electrónico SET (Secure Electronic Transaction) [137]. Mención especial merece el grupo de trabajo PKIX [99] del IETF, el cual en los últimos años ha ido proponiendo una larga serie de borradores y especificaciones con el fin de generalizar el uso de este tipo de certificados en Internet.

En esta sección, se describirá la última versión del estándar X.509. Para una mejor comprensión del mismo, se introducirá primero el concepto de directorio y nombramiento basado en la propuesta X.500.

2.2.1 Directorio X.500

El estándar X.500 define tanto el protocolo utilizado para acceder a la información contenida en el directorio (DAP, Directory Access Protocol), como el modelo que define cómo se almacenan y gestionan los datos. El directorio X.500 es muy similar a un listín telefónico donde, dado el nombre de una persona, se puede encontrar información adicional acerca de la misma. De hecho, la idea original del proyecto X.500 era definir una única infraestructura pública formada por múltiples directorios gestionados de forma independiente, pero estructurados según un único espacio de nombres.

Una entrada en un directorio X.500 puede contener un conjunto de atributos, como el nombre de la organización para la cual trabaja la persona, su puesto de trabajo, su dirección de correo electrónico o sus certificados digitales, por nombrar sólo algunos de ellos. Estas entradas pueden representar a cualquier entidad del mundo real, no sólo personas sino también ordenadores, periféricos, compañías o naciones.

La indexación del directorio se realiza mediante la utilización de nombres globalmente únicos llamados nombres distinguidos *(DN, Distinguished Names)*. Con el fin de intentar asegurar su unicidad, los nombres se asignan de forma jerárquica siguiendo una estructura denominada *árbol de información del directorio (DIT, Directory Information Tree)* (ver figura 2.4).

Cada nodo, o vértice, del árbol tiene un nodo padre (excepto la raíz) y cualquier número de nodos hijo. Cada nodo, excepto la raíz, tiene asignado un nombre distinguido relativo *(RDN, Relative Distinguished Name)*, el cual es único entre todos los descendientes del nodo. Los RDNs de cada uno de los antecesores de un nodo se concatenan con el RDN del propio nodo para formar su nombre distinguido (DN). En la figura 2.4 se ilustra este proceso. Tras el nodo raíz, hay una entrada para cada uno de los países del mundo. Esas entradas tienen un RDN representado por un código único de dos letras asignado por la

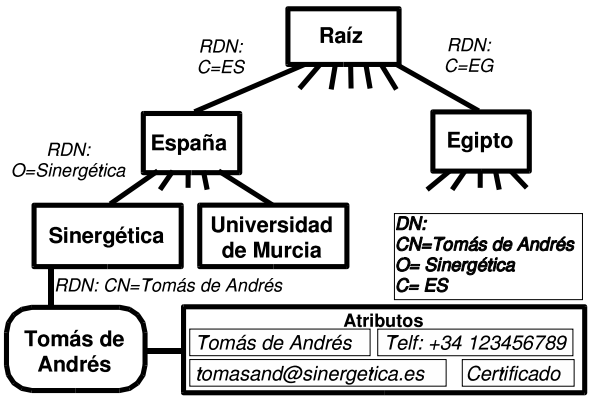

Figura 2.4: Árbol de directorio X.500

ISO. De los nodos de cada país, descienden cada una de las entradas correspondientes a las organizaciones de dicho país, como las empresas privadas, administraciones públicas, etc. Finalmente, cada organización crea entradas para cada una de sus unidades organizativas *(OU, Organizational Units)* y para sus empleados, máquinas, y cualquier otra entidad que quieran registrar. En el ejemplo de la figura, Tomás de Andrés trabaja para Sinergética, una compañía española. Sinergética ha asignado un RDN a Tomás que especifica su nombre *(CN, Common Name)*. Su entrada de directorio, accesible mundialmente haciendo uso de su DN, contiene algunos atributos, tales como su teléfono, correo electrónico, certificado, etc.

Por otro lado, con el fin de atender la necesidad de proporcionar un método de acceso y consulta a los directorios, se creó el Protocolo de Acceso Ligero a Directorios *(LDAP, Lightweight Directory Access Protocol)*. En 1997, el IETF propuso como estándar LDAPv3, publicado como RFC (Request For Comments) 2251 [188]. LDAP hace uso de una pila estándar TCP/IP y es mucho menos exigente, en lo que a recursos se refiere, que el protocolo de acceso propuesto por el estándar X.500. De hecho, LDAP se convirtió rápidamente en el estándar *de facto* a la hora de acceder a información contenida en directorios públicos.

2.2.2 Formato de los certificados X.509v3

En el periodo comprendido entre 1993 y 1994, cuando se intentó por primera vez implantar a gran escala los certificados X.509, se constató el hecho de que las versiones 1 y 2 de dichos certificados resultaban bastante deficientes en varios aspectos. Las principales razones que llevaron a considerar la posibilidad de que los certificados incluyeran otro tipo de información, que además pudiera ser extensible, fueron:

- Dado que una misma entidad puede disponer de varios certificados distintos, con claves públicas diferentes, empleados para propósitos muy diversos, es necesario poder identificar cada uno de ellos de forma independiente.

- Algunas aplicaciones necesitan identificar a los usuarios por nombres específicos, no

haciendo uso de los nombres X.500. Por ejemplo, en el ámbito del correo electrónico seguro, es más importante ligar una clave pública a una dirección de correo electrónico que a un nombre X.500.

- Los diferentes certificados pueden ser emitidos siguiendo distintas políticas y prácticas de certificación (para más información al respecto, ver sección 2.3.6), lo cual implica la necesidad de constatar las garantías aplicables a cada certificado.

Con el fin de satisfacer estos y otros requisitos, era necesario incorporar nuevos campos al formato de los certificados. De hecho, lo que se constató fue que progresivamente irían surgiendo nuevas necesidades que conllevarían la inclusión de nuevos campos, por lo que la tercera versión del estándar introdujo un mecanismo genérico de extensión de los certificados X.509. La figura 2.5 ilustra el contenido de un certificado X.509v3.

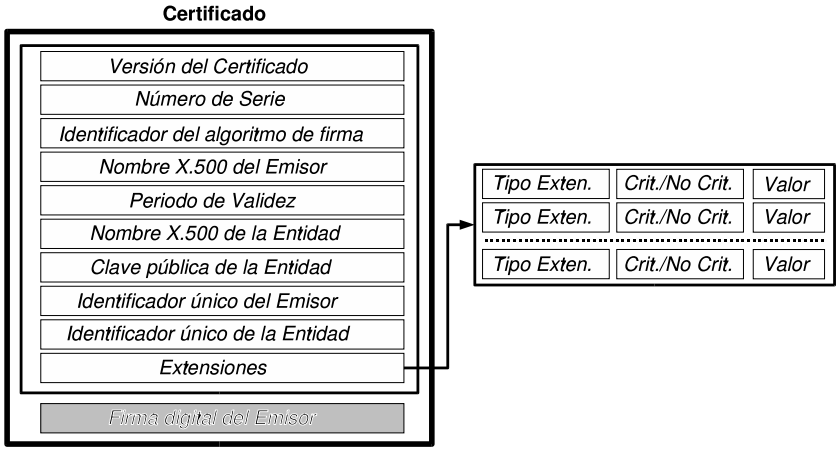

Figura 2.5: Certificado X.509v3

Los campos del certificado se interpretan de la siguiente forma:

- *Versión.* Indicador de la versión del certificado (en este caso, versión 3).
- *Número de serie.* Identificador único del certificado, asignado por la autoridad emisora del mismo.
- *Algoritmo de firma.* Identificador del algoritmo empleado por la entidad emisora para firmar el certificado.
- *Emisor.* Nombre X.500 de la entidad emisora
- *Validez.* Periodo durante el cual el certificado se considera válido, salvo revocación.
- *Entidad certificada.* Nombre X.500 de la entidad poseedora de la clave privada que está asociada con la clave pública contenida en el certificado.

- *Clave pública de la entidad.* El valor de la clave pública de la entidad, junto con un identificador del algoritmo con el cual debe usarse dicha clave.

- *Identificador único de emisor.* Secuencia de bits utilizada para identificar unívocamente a la entidad emisora, incluso en el supuesto de que el nombre de la entidad hubiera sido asignado a otras entidades a lo largo del tiempo.

- *Identificador único de entidad.* Secuencia de bits utilizada para identificar unívocamente a la entidad receptora del certificado, incluso en el supuesto de que el nombre de la entidad hubiera sido asignado a otras entidades a lo largo del tiempo.

- *Extensiones.* Las extensiones tienen un tipo asociado, que debe ser registrado mediante la asignación de un identificador único de objeto *(OID, Object Identifier)*, un indicador acerca de la criticidad y un valor. El indicador de criticidad refleja si la extensión puede ser ignorada por aquellos sistemas que no la reconozcan. Por último, el campo denominado *Valor* contiene los datos asociados a la extensión, siendo su estructura interna dependiente del tipo de extensión.

- *Firma digital.* La firma digital es un valor criptográfico obtenido a partir del resumen digital del certificado y la clave privada de la entidad emisora. Se trata del elemento que le confiere al certificado integridad y autenticidad.

Una de las diferencias más importantes entre la versión 3 y las anteriores hace referencia al mecanismo de nombramiento. La versión 3 no siguió imponiendo el sistema X.500 como único esquema de nombramiento para identificar a los emisores y los receptores de los certificados. Cualquier entidad puede ser identificada por uno o más nombres expresados mediante esquemas distintos. Es perfectamente razonable emitir un certificado que contenga varios nombres para una misma entidad, los cuales pueden ser usados por cualquier aplicación que reconozca su formato. Entre los formatos de nombramiento reconocidos explícitamente en el estándar X.509 encontramos: direcciones de correo electrónico, nombres de dominios de Internet, direcciones de correo electrónico X.400, nombres X.500, identificadores uniformes de recursos *(URI, Uniform Resource Identifier)* y direcciones del protocolo de Internet *(IP, Internet Protocol)*.

2.3 Ciclo de vida de un certificado digital

El ciclo de vida de un certificado digital de identidad, y más concretamente un certificado X.509, abarca todas aquellas operaciones de gestión de la información contenida en el mismo (par de claves, extensiones, identificadores, periodos de validez, datos sobre la entidad emisora, etc.) realizadas por las distintas entidades que componen el sistema de certificación. En esta sección, además de detallar las operaciones relacionadas con los certificados, se analizarán algunas cuestiones asociadas con la gestión de los pares de claves pública-privada, en especial las operaciones de generación y protección de claves.

2.3.1 Gestión de claves

El proceso de creación de claves criptográficas requiere aleatoriedad, de forma que el par de claves generado no sea fácilmente predecible. En el supuesto de que el valor de las claves se pudiera averiguar, o que el espacio de búsqueda se redujera de forma tan drástica que pudiera ser recorrido en un tiempo razonablemente corto, la seguridad de todo el sistema de gestión del ciclo de vida de los certificados digitales se vería seriamente afectada. Además, las circunstancias en las cuales se generen los pares de claves dependerán del uso que se les vaya a dar. Hay dos alternativas básicas a la hora de generar pares de claves:

- *Generación por parte del propietario.* El par de claves se genera en el mismo sistema (posiblemente en el mismo token hardware o módulo software) en el cual la clave privada va a almacenarse y a usarse posteriormente. Para el caso de las claves privadas de firma digital, esta alternativa resulta la más conveniente, ya que la información privada nunca abandona su lugar de generación, lo cual puede llegar a ser incluso un requisito en ciertos entornos como el descrito en el estándar X9.57 [12].

- *Generación en un elemento central.* El par de claves se genera en algún sistema central, y la clave privada se transporta de forma segura al equipo del usuario correspondiente. Este enfoque es necesario cuando los sistemas de almacenamiento de los usuarios, como ciertas tarjetas inteligentes, tienen unos recursos de memoria y de procesamiento muy limitados, o cuando la generación por parte de dichos dispositivos no es posible. La generación basada en un elemento centralizado es también aconsejable por otras razones, como por ejemplo la posibilidad de generar claves de mayor calidad o de realizar copias de seguridad de las mismas, si bien este último punto ha sido tradicionalmente foco de controversia entre la comunidad científica [4].

El uso de una u otra alternativa implicará una serie de variaciones en los procedimientos de gestión de los certificados digitales ofrecidos por un determinado sistema, aunque lo más común es que ambas alternativas sean una opción válida dentro del mismo entorno de gestión de certificados.

Una vez generado el par de claves, es necesario proteger convenientemente la clave privada, ya que la aplicabilidad de los certificados digitales recae en el hecho de que éstas sólo sean utilizadas por la persona o el dispositivo al cual pertenecen. Normalmente, las claves privadas son protegidas utilizando alguno de estos métodos:

- Almacenamiento en un módulo hardware, como una tarjeta inteligente o tarjeta PCMCIA.

- Almacenamiento en un fichero cifrado contenido en un disco duro u otro medio de almacenamiento de datos.

- Almacenamiento en un servidor de credenciales, el cual distribuye la clave privada al usuario después de haberlo autenticado.

En todos los casos, el acceso a la clave necesita estar protegido mediante el uso de uno o más mecanismos de autenticación personal. Los más comunes son los basados en PIN (Personal Identification Number) o password (palabra de paso). Otros métodos de autenticación están basados en análisis de datos biométricos [107].

2.3.2 Emisión de certificados

Como ya se comentó en la sección 2.1.1, la emisión de certificados digitales necesita de la actuación de una autoridad de certificación. Ahora bien, las interacciones entre dicha autoridad y los subscriptores o usuarios del servicio se realizan a través de entidades intermediarias conocidas como *autoridades de registro (RA, Registration Authorities)*. Estas entidades están encargadas de verificar la identidad del solicitante mediante la comprobación de los documentos acreditativos presentados de forma personal. La autoridad de registro no emite los certificados, sino que simplemente valida o rechaza las solicitudes que se le presentan. Es la autoridad de certificación la que posteriormente se encarga de emitir y publicar todas aquellas solicitudes que fueron previamente validadas. Entre las funciones ligadas a una autoridad de registro encontramos:

- Validar solicitudes de certificación.
- Aprobar o rechazar modificaciones sobre los atributos contenidos en los certificados de los usuarios.
- Generar, hacer copia de seguridad y recuperar pares de claves.
- Aceptar y autorizar solicitudes para la revocación o suspensión de certificados existentes.
- Distribuir físicamente tokens personales que contengan información criptográfica.

Una vez que las solicitudes han sido validadas y autorizadas por parte de alguna de las autoridades de registro presentes en el sistema, la siguiente etapa correspondiente al ciclo de vida del certificado es la generación del mismo. Este proceso implica los siguientes pasos:

1. La autoridad de certificación recibe la solicitud de certificación previamente validada.
2. La autoridad de certificación confirma que el certificado cumple la política de certificación y que está en consonancia con lo especificado en las prácticas de certificación (para más información al respecto, ver sección 2.3.6).
3. El certificado es firmado por un dispositivo de firma que contiene la clave privada de la autoridad de certificación. Este dispositivo puede ser un tarjeta criptográfica, un módulo software o incluso una tarjeta inteligente.

4. Se envía una copia del certificado al usuario y/o a un repositorio público de certificados.

5. Como servicio opcional, la autoridad de certificación puede archivar una copia del certificado con el fin de proporcionar una base de evidencias que podría ser utilizada en un futuro para servicios de no repudio.

6. La autoridad de certificación puede registrar ciertos detalles del proceso de generación de certificados, e incluso almacenar una copia de la clave privada asociada a la clave pública que acaba de certificar.

2.3.3 Distribución de certificados

Para poder cifrar datos o verificar firmas digitales, un usuario necesita el certificado asociado a la clave pública correspondiente, además de todos aquellos certificados asociados a las autoridades de certificación necesarias para completar el camino de certificación. Se trata de una cuestión de distribución de información, y no de un problema de seguridad, ya que los certificados no tienen que ser protegidos por tratarse de documentos firmados digitalmente. En esta sección se analizan las dos formas fundamentales de distribución de certificados.

En relación con las operaciones de firma digital, hay una forma muy apropiada para distribuir los certificados. Dado que el signatario dispone normalmente de una copia de su propio certificado, puede adjuntarlo al documento firmado digitalmente para que cualquiera que desee verificar la firma disponga de la información necesaria. De igual modo, el signatario puede añadir todos aquellos certificados que puedan ser necesarios para validar su propio certificado. Sin embargo, hay varias razones por las cuales esta técnica podría no resultar siempre apropiada. Por un lado, se puede realizar un gasto excesivo de los recursos de comunicación o de almacenamiento, ya que el usuario encargado de la verificación de la firma podría disponer previamente de una copia de los certificados. Por otro lado, no es siempre fácil averiguar qué certificados necesita el receptor para validar el mensaje, ya que las cadenas de certificación pueden ser muy complejas.

El otro método de distribución más ampliamente utilizado es el basado en servidores de directorio. Para las operaciones de cifrado de información, el acceso a certificados mediante consultas a servidores de directorio es uno de los métodos más habituales. Además, dichos servidores pueden proporcionar otro tipo de información acerca del destinatario, como la dirección de correo electrónico, información laboral, etc. Como ya se comentó en la sección 2.2.1, el estándar X.500 y el protocolo LDAP constituyen los ejes fundamentales sobre los cuales gira este servicio.

2.3.4 Renovación de certificados

Los certificados tienen un tiempo de vida limitado y, en general, deben ser renovados tras su expiración. Dicha renovación puede conllevar también un cambio de pares de claves,

aunque no es siempre obligatorio ya que es posible emitir nuevos certificados que incluyan las claves contenidas previamente en certificados ya caducados. Esta renovación puede realizarse de forma totalmente transparente de cara al usuario, con el fin de ocultarle los detalles relativos a periodos de validez, cambios de claves, y otros procesos de gestión, o por contra puede implicar la actuación del mismo, sobre todo en aquellos casos en los que se requiera la notificación del cambio de algunos datos contenidos en el certificado.

2.3.5 Revocación de certificados

Cuando se emite un certificado, se espera que el intervalo de uso del mismo coincida con el que se encuentra reflejado en el periodo de validez. Sin embargo, bajo ciertas circunstancias, los usuarios deben dejar de confiar en ciertas claves antes de que éstas caduquen. Tales circunstancias incluyen el conocimiento o la sospecha del compromiso de una clave privada, el cambio de nombre, o el cambio de relación entre la entidad certificada y la autoridad de certificación (por ejemplo, causado por una baja laboral). En estos casos, se debe revocar el certificado, logrando de esa forma que el periodo operativo del mismo sea inferior al proyectado inicialmente.

La decisión de revocar un certificado, generalmente, es responsabilidad de la autoridad de certificación, la cual actúa en respuesta a una solicitud formulada por parte de alguna entidad autorizada. El conjunto exacto de personas que está autorizado a revocar un certificado depende de las prácticas de certificación. Generalmente, el usuario afectado está autorizado a solicitar su propia revocación, además de los operarios de las autoridades de certificación o de las autoridades de registro.

Después de decidir que un certificado debe ser revocado, una autoridad de certificación debe propagar la noticia con el fin de evitar que el certificado en cuestión siga empleándose. El método más común para proporcionar información acerca de las últimas revocaciones es la emisión periódica de las llamadas listas de certificados revocados *(CRL, Certificate Revocation List)*. De hecho, el concepto de CRL forma parte del propio estándar X.509 [99]. Una CRL puede ser descrita como un documento firmado digitalmente por la autoridad de certificación que contiene una entrada para cada uno de aquellos certificados que han tenido que ser revocados antes de su expiración. Cada entrada puede contener información suplementaria, como el motivo de la revocación, o la fecha a partir de la cual el certificado debe considerarse como no válido. Estos documentos se emiten de forma periódica, con un intervalo que depende completamente de las prácticas de certificación de la autoridad, y con independencia de si han producido, o no, nuevas revocaciones. Sin embargo, como veremos en la sección 2.4, existen nuevos métodos de validación delegada y validación en línea que tratan de solventar algunas de las deficiencias asociadas a las listas de certificados revocados [113, 172].

2.3.6 Políticas y prácticas de certificación

El grado mediante el cual una entidad puede confiar en la información contenida en un certificado depende de muchos factores. Estos factores incluyen las prácticas seguidas

por la autoridad de certificación a la hora de autenticar a la entidad, las obligaciones del usuario (por ejemplo, a la hora de proteger su clave privada), y las obligaciones legales de la autoridad de certificación, es decir, garantías y limitaciones en la responsabilidad. De acuerdo con el estándar X.509, una política de certificación es un conjunto identificado de reglas que indica la aplicabilidad de un certificado a un entorno concreto de aplicación. La política de certificación, normalmente reflejada mediante una extensión contenida en cada certificado emitido, puede ser utilizada por parte de un usuario para decidir si el enlace entre identidad y clave, establecido por un determinado certificado, se puede considerar lo suficientemente confiable para el entorno de aplicación en cuestión. Se podría decir que se trata de un identificador digital que resume el conjunto de acciones llevadas a cabo por cierto sistema a la hora de gestionar el ciclo de vida de los certificados que emite.

Por otro lado, las prácticas de certificación son una declaración de los detalles del sistema y de los procedimientos seguidos a la hora de realizar las operaciones relacionadas con los certificados. Al tratarse de un documento que será consultado por los usuarios del servicio, debe ser bastante explícito, y debe proporcionar una descripción muy en profundidad de los servicios ofrecidos, responsabilidades y garantías. Ahora bien, desde el punto de vista de la interoperabilidad entre distintos sistemas de certificación, las políticas de certificación constituyen un vehículo más apropiado. Una autoridad de certificación con unas únicas prácticas de certificación puede tener asociadas distintas políticas de certificación, cada una quizá para un entorno de aplicación distinto. Para una información más en profundidad a cerca de aspectos legales relacionados con prácticas y políticas de certificación, consultar [18, 79].

2.4 Recomendaciones PKIX para el desarrollo de PKIs

Una infraestructura de clave pública *(PKI, Public Key Infrastructure)* puede ser definida como un conjunto de recursos software, hardware y humanos que posibilitan el uso de de la criptografía de clave pública para proporcionar los servicios básicos de seguridad de confidencialidad, autenticación, integridad y no repudio. En esta tesis, el concepto de infraestructura de clave pública está principalmente centrado en los componentes que forman parte de la gestión del ciclo de vida de los certificados de identidad. A lo largo de esta sección, se detallarán algunos aspectos fundamentales a considerar a la hora de construir PKIs que puedan dar soporte a grandes comunidades de usuarios. Para ello analizaremos las propuestas que ha ido desarrollando el grupo de trabajo PKIX [103] del IETF (Internet Engineering Task Force) en materia de servicios de certificación.

El grupo de trabajo PKIX se formó en Octubre de 1995 con el fin de desarrollar los estándares de Internet relacionados con el diseño de PKIs. El primer documento de trabajo fue la especificación del formato de certificación X.509 en base a la recomendación del ITU-T. A lo largo de su existencia, este grupo de trabajo ha publicado varias recomendaciones en materia de especificación de certificados de identidad y de atributo, listas de certificados revocados, mecanismos de validación de certificados, servicios de sellado de tiempo, prácticas de certificación, protocolos de gestión, o protocolos operacionales, por

citar sólo algunas de ellas.

2.4.1 Arquitectura de una PKI

Una PKI, según el modelo propuesto por el grupo PKIX, contiene cinco tipos de elementos:

- Una autoridad de certificación que emita y revoque los certificados de clave pública.
- Autoridades de registro que atestigüen la relación entre las claves públicas y la identidad de los usuarios.
- Poseedores de los certificados, los cuales pueden firmar documentos digitalmente y descifrarlos usando sus claves privadas.
- Usuarios de los certificados, los cuales pueden validar las firmas digitales y las cadenas de certificación originadas a partir de una autoridad de certificación raíz confiable, y cifrar documentos utilizando las claves contenidas en los certificados.
- Repositorios que almacenen y publiquen los certificados y las listas de certificados revocados.

La figura 2.6 ilustra la relación existente entre estos elementos. En ella se muestran las operaciones que pueden ser solicitadas por partes de los usuarios finales, las distintas funciones asociadas a cada uno de los elementos de gestión, y la relación que puede existir con otras autoridades de certificación pertenecientes a PKIs distintas. Cómo se produce la relación entre estos elementos, cuál es el formato de los mensajes intercambiados, y cuáles son los nuevos servicios de valor añadido que complementan la gestión básica del ciclo de vida de un certificado es la materia de los próximos apartados.

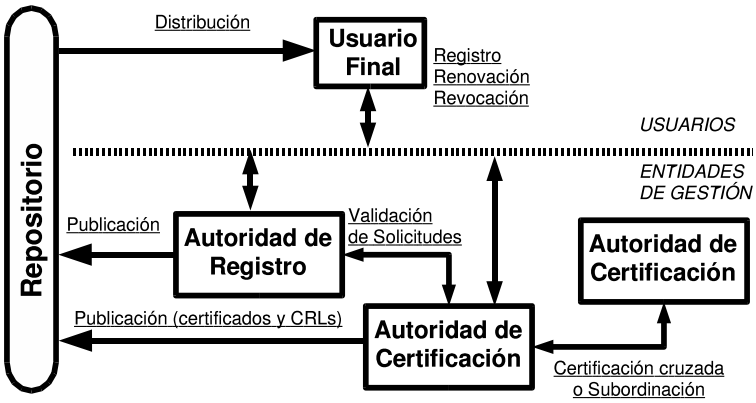

Figura 2.6: Entidades de una PKI

2.4.2 Protocolos de gestión

Los protocolos de gestión son el mecanismo mediante el cual se producen las interacciones entre los usuarios de la PKI y las entidades de gestión de la misma. Por ejemplo, un protocolo de gestión puede emplearse para transmitir la información de registro de un usuario, una solicitud de revocación, etc. Estos protocolos tienen dos componentes fundamentales: por un lado, el formato de los mensajes a enviar; por otro, el conjunto de reglas que gobierna la transmisión de esos mensajes.

Respecto al formato de los mensajes, algunas de las especificaciones PKIX se han basado en parte de la serie de estándares PKCS (Public Key Cryptography Standard) desarrollados por RSA Security, en concreto en dos ellos. El estándar PKCS#10 [120] define el formato de un mensaje de solicitud de certificación mediante el cual un solicitante puede codificar sus valores de clave pública, así como otros datos adicionales a incluir en el certificado. Por otro lado, el estándar PKCS#7 [118] especifica un formato de envoltura digital que, entre otros usos, resulta muy apropiado para encapsular los certificados emitidos como respuesta a las solicitudes en formato PKCS#10. La combinación de estos dos estándares ha sido muy popular entre la mayor parte de las implementaciones de PKI, sobre todo debido al hecho de que el software necesario para construir y decodificar estas estructuras de datos está ampliamente extendido en la mayoría de las librerías criptográficas.

Sin embargo, a mediados de la década de los 90, el grupo de trabajo PKIX inició un proyecto que tenía como fin desarrollar un protocolo de gestión capaz de tratar todas las operaciones relacionadas con el ciclo de vida de los certificados. Como consecuencia, se realizaron dos propuestas totalmente distintas. La primera de ellas se basaba en el diseño de un nuevo protocolo llamado Protocolo de Gestión de Certificados *(CMP, Certificate Management Protocol)* [7], mientras que la segunda era una propuesta que optaba por incrementar las especificaciones PKCS mediante la inclusión de nuevas características. El decepcionante resultado fue que ambas líneas de trabajo avanzaron de forma paralela, provocando que implementaciones de distintas compañías ofrecieran los mismos servicios mediante protocolos de gestión que, aunque conceptualmente iguales, eran incompatibles técnicamente. La propuesta basada en los estándares PKCS evolucionó hacia lo que hoy se conoce como CMC *(Certificate Management over CMS)* [151].

Certificate Management Protocol (CMP)

CMP especifica los mensajes a enviar en las comunicaciones entre elementos de la PKI, o entre elementos de la PKI y aplicaciones o servicios que hacen uso de la misma. Estos mensajes se definieron con el fin de dar soporte a las siguientes funciones:

- *Certificación.* Se especifica el formato de los mensajes de solicitud y de respuesta para la operación de solicitud de certificación. El formato generalmente utilizado es CRMF (Certificate Request Message Format) [149], aunque también se permite el uso de PKCS#10.

- *Prueba de posesión.* Cuando se emite un certificado, la autoridad de certificación

debe estar segura de que el solicitante está en posesión de la correspondiente clave privada. CMP especifica los intercambios de mensajes a realizar para este propósito.

- *Renovación de certificado.* Tanto para el caso de que las claves se mantengan como para el caso en el que sean reemplazadas.

- *Revocación de certificado.* Se proporciona soporte para que la solicitud pueda realizarla tanto el usuario afectado como cualquier otra entidad autorizada.

- *Notificaciones.* Se han definido mensajes para informar acerca de hechos puntuales, como la actualización del par de claves de la autoridad de autorización, emisión de CRLs, etc.

- *Envoltura digital.* CMP utiliza su propio formato de envoltura digital en caso de que sea necesario proteger un mensaje para propósitos de autenticación, integridad o confidencialidad.

Como puede comprobarse, los objetivos de CMP eran bastante ambiciosos. CMP define 25 tipos de mensajes, incluyendo solicitudes y confirmaciones. Su complejidad frente a los enfoque basados en PKCS#10 y PKCS#7 ha sido uno de sus principales puntos débiles a la hora de extenderse entre la comunidad científica y los productos comerciales.

Certificate Management over CMS (CMC)

La especificación CMC define un conjunto de mensajes destinados principalmente al proceso de registro y emisión de certificados. CMC también proporciona soporte para las solicitudes de revocación y renovación, no presentes en los sistemas PKCS. El sistema de envoltura digital está basado en la sintaxis CMS (Cryptographic Message Syntax) [98] de S/MIME [168], dado que la idea principal de CMC es emplear S/MIME como protocolo de de gestión de la PKI. El enfoque seguido por esta especificación se basa en la combinación de técnicas existentes, más que en el diseño de una propuesta completamente nueva (como CMP).

2.4.3 Protocolos operacionales

Los protocolos operacionales tienen la función de distribuir información acerca de los certificados, las listas de certificados revocados o la relación existente entre un documento y un determinado instante de tiempo. La distribución se puede realizar utilizando distintos medios, como DNS (Domain Name System) [64], LDAP [188], HTTP (Hypertext Transfer Protocol) [77] o X.500 [38]. Especial atención merecen aquellos protocolos relacionados con la distribución de información acerca del estado de los certificados, también conocidos como protocolos de verificación en línea, y los mecanismos de sellado digital de tiempo.

Validación de certificados

Por protocolos operacionales de validación de certificados entendemos aquellos mecanismos que proporcionan información acerca del estado actual del certificado (revocado, válido o estado desconocido), o relativa a la cadena de certificación necesaria para validar la autenticidad del certificado. Para ello, hay definida una serie de protocolos de entre los cuales analizaremos OCSP (Online Certificate Status Protocol), DPD (Delegated Path Discovery), DPV (Delegated Path Validation) y SCVP (Simple Certificate Validation Protocol).

Una de las principales deficiencias de las CRLs es que la periodicidad con la cual se publican las últimas revocaciones no está bajo el control de las aplicaciones que deben validar los certificados. De forma ideal, el conocimiento acerca de si un certificado se encuentra revocado, o no, debería ser adquirido en el mismo momento en el cual necesita utilizarse el certificado. Mediante un mecanismo de verificación en línea, una aplicación o usuario puede obtener respuestas instantáneas acerca del estado de los certificados implicados. Conscientes de este hecho, el grupo de trabajo PKIX desarrolló OCSP (Online Certificate Status Protocol) [150], un protocolo que define el formato estándar de las solicitudes y respuestas intercambiadas para averiguar el estado de un certificado. Este protocolo se basa en la existencia de un servidor OCSP, encargado de procesar las solicitudes de validación que recibe, verificar el estado de los certificados implicados utilizando algún mecanismo confiable y responder con una sentencia firmada digitalmente que incluya dicho estado.

Los protocolos DPV (Delegated Path Validation) y DPD (Delegated Path Discovery) [165] están relacionados con el procesamiento de cadenas de certificación. Ambos están basados en la posibilidad de delegar, en una tercera entidad confiable, la realización de ciertas comprobaciones que involucran a varios certificados relacionados por formar parte de la misma cadena de confianza. DPV establece los formatos de solicitud y respuesta necesarios para consultar si los certificados implicados siguen siendo válidos (es decir, si no están revocados y siguen unidos por una relación de confianza). Por otro lado, los servidores DPD tienen la función de descubrir, en nombre de sus usuarios, toda la información de estado para validar localmente un certificado (certificados de autoridades de certificación, listas de certificados revocados, respuestas OCSP, etc).

SCVP (Simple Certificate Validation Protocol) [134] es una línea actual de trabajo que intenta ocultar a los clientes los detalles concretos del método de validación de certificados que se está empleando en un sistema para averiguar el estado de los mismos. Independientemente de que la consulta sea a una CRL, mediante OCSP o cualquier otro mecanismo, SCVP proporciona una única visión de la operación de validación con el fin de evitar que las distintas aplicaciones deban conocer los detalles concretos de los métodos de validación empleados. De esta forma, el cliente delega en un servidor SCVP la responsabilidad de obtener (mediante consultas OCSP, CRLs, DPV, DPD, etc.) toda la información necesaria para determinar si un conjunto de certificados es confiable.

Sellado de tiempo

El sellado de tiempo (del inglés, *time-stamp*) se emplea para atestiguar que un determinado evento se produjo en un determinado instante de tiempo. Por ejemplo, en una transacción económica, el sellado de tiempo puede emplearse para establecer una relación entre la factura y el instante de compra, o entre la distribución del producto comprado y el momento en el cual se efectuó dicha distribución. Sin embargo, su uso no está sólo restringido a escenarios de pago, sino que puede ser empleado igualmente en entornos de validación de documentos, notaría electrónica, bases de datos, etc. En un sentido amplio del término, un sellado de tiempo es el establecimiento de una relación confiable entre un determinado elemento digital y un instante de tiempo en el cual se quiere dejar constancia de que el elemento digital existió. No es necesario que dicho sellado contenga completamente el elemento digital a sellar, sino que bastará en la mayoría de los casos con la presencia de su resumen digital.

Para que el sellado de tiempo pueda ser considerado como confiable, la estructura de datos que lo contiene debe estar protegida criptográficamente, y además la marca de tiempo debe haber sido obtenida de una fuente confiable. Esto último puede asegurarse mediante la utilización de proveedores de valores temporales basados en UTC (Universal Time Coordinated). Dichos proveedores son entidades nacionales o internacionales que funcionan de forma totalmente independiente a los servicios que hacen uso de ellos y que aseguran una alta precisión en los datos temporales que suministran.

Respecto a la protección criptográfica del documento que contiene la asociación entre el evento y el instante de tiempo en el cual fue sellado, debe ser suficiente como para hacer imposible que de forma retroactiva pueda modificarse la información contenida. Uno de los mejores métodos para lograr este objetivo es que el documento se encuentre firmado digitalmente por una entidad confiable.

En concreto, el grupo de trabajo PKIX define el sellado de tiempo como un servicio en el cual una tercera parte confiable (a la cual denominan autoridad de sellado de tiempo -*TSA, Time Stamp Authority*-) firma un mensaje con el fin de probar que existía antes de un determinado instante de tiempo. En el documento de definición del servicio [6], se especifica el protocolo basado en mensajes de solicitud y respuesta que debe emplearse para asociar marcas temporales confiables a documentos digitales.

Frente a este modo de operación básico, se han propuesto también esquemas encadenados que tratan de minimizar las consecuencias derivadas del compromiso de la clave de la TSA [32, 62]. Dichos esquemas se basan en el uso de secuencias lógicas de resumenes digitales distribuidas entre los distintos sellos de tiempo, las cuales establecen un orden cronológico que dificulta la falsificación posterior de alguno de los elementos de información contenidos en el sello.

2.5 Entornos de PKI

Durante los últimos años, muchos han sido los desarrollos en materia de PKI que han sido llevados a cabo en distintos países e instituciones con el fin de dotar de servicios de certificación a comunidades de usuarios extensas. En esta sección, vamos a describir las iniciativas internacionales y nacionales más importantes que tienen como base el estándar X.509 y las recomendaciones PKIX. Por último, se analizarán los desarrollos realizados previamente por el grupo de investigación ANTS en materia de certificación, con el fin de ubicar la línea de partida para el diseño y la implantación de la infraestructura de clave pública del Proyecto PISCIS, proyecto dentro del cual se encuadran los desarrollos de gestión de identidad digital que forman parte de esta tesis.

2.5.1 Desarrollos nacionales e internacionales

La gran mayoría de las PKI desarrolladas con éxito están bajo el control de una única empresa, y su ámbito de aplicación no va más allá del uso interno o de sus filiales. Sin embargo, en este apartado nos vamos a centrar en aquellas infraestructuras de clave pública que implican a múltiples organizaciones, ya sean nacionales o internacionales. No se trata de describir los detalles concretos de las implementaciones asociadas a estas infraestructuras, sino de conocer los propósitos de dichos sistemas, las comunidades de usuarios implicadas y los servicios ofrecidos.

PEM (Privacy Enhaced Mail)

En el año 1993, la comunidad científica de Internet completó el desarrollo de un conjunto de estándares para el sistema PEM [109], los cuales incluían protocolos de correo electrónico seguro y especificaciones relacionadas con la PKI de soporte. La infraestructura PEM, ilustrada en la figura 2.7, seguía el modelo de confianza jerárquico. A pesar de que el sistema PEM no tuvo gran aceptación por parte de las empresas privadas y de la comunidad Internet en general, el diseño de su PKI has sido lo suficientemente significativo como para que muchas PKIs posteriores hayan basado sus desarrollos en este modelo.

El modelo PEM está basado en tres tipos de autoridades de certificación:

- *Internet Policy Registration Authority (IPRA).* Se trata de la autoridad raíz de la infraestructura. Está controlada por la *Internet Society*, una organización internacional sin ánimo de lucro.

- *Policy Certification Authorities (PCAs).* Las PCAs son las únicas autoridades certificadas por la IPRA. Una PCA debe registrarse frente a la IPRA y publicar su política de certificación de usuarios y autoridades de certificación subordinadas. Dado que cada camino de certificación PEM contiene exactamente una única PCA, los usuarios pueden identificar fácilmente las políticas de certificación seguidas.

- *Certification Authorities (CAs).* Estas entidades representan, por ejemplo, organizaciones privadas, unidades organizativas o áreas geográficas concretas.

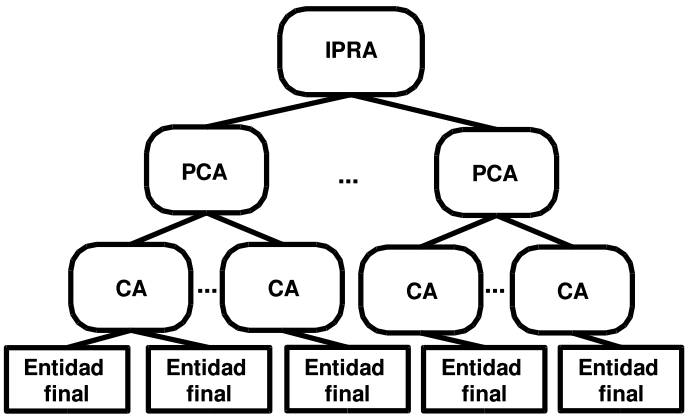

Figura 2.7: Infraestructura PEM

El desarrollo de PEM presentó algunas soluciones a problemas muy importantes. Por ejemplo, fue pionero en el concepto de política de certificación. Sin embargo, la implantación del sistema PEM a gran escala fracasó por varios motivos, entre ellos la aparición de nuevas tecnologías de correo electrónico seguro como S/MIME [168].

Secure Electronic Transaction (SET)

Las principales compañías de tarjetas de crédito, lideradas por Visa y MasterCard, desarrollaron el sistema SET [137], el cual está compuesto por un complejo protocolo de intercambio seguro de datos y una infraestructura de clave pública. El sistema SET está destinado a dar soporte a los pagos basados en tarjetas de crédito a través de Internet, y está compuesto por varios tipos distintos de entidades, como los emisores de tarjetas, los usuarios de las tarjetas, bancos de clientes, comerciantes, bancos de los comerciantes, autoridades de certificación y pasarelas de pago (las cuales conectan el sistema con la red interbancaria, medio en el cual se realizan realmente las transacciones).

La criptografía de clave pública se emplea para proporcionar servicios de autenticación y confidencialidad a todas las partes implicadas en una transacción electrónica. La estructura jerárquica de la PKI de SET, mostrada en la figura 2.8, incluye los siguientes tipos de autoridades de certificación:

- *Autoridad de Certificación Raíz.* Esta autoridad se mantiene desconectada (off-line) y se utiliza sólo para emitir certificados a las distintas autoridades de certificación de las compañías de tarjetas de crédito.

- *Autoridad de Certificación de las Compañías.* Cada compañía, como Visa o MasterCard, dispone de una autoridad de certificación a este nivel. Desde el punto de vista de las políticas de certificación, cada compañía puede establecer sus propios criterios.

- *Autoridad de Certificación Geopolítica.* Este nivel opcional de la jerarquía permite a las compañías distribuir la responsabilidad de la gestión de los certificados entre distintas regiones geográficas o políticas.

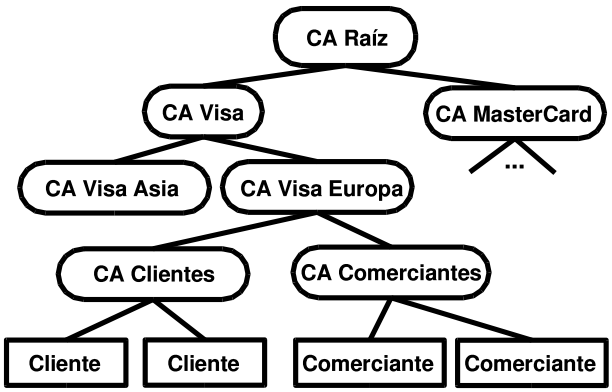

Figura 2.8: Infraestructura SET

- *Autoridad de Certificación de los Clientes.* Son las encargadas de emitir los certificados digitales a todos aquellos subscriptores de la compañía correspondiente. Estas autoridades suelen ser gestionadas por entidades financieras con las cuales el cliente tiene una relación contractual.

- *Autoridad de Certificación de los Comerciantes.* Emiten certificados digitales a los comerciantes que forman parte del sistema SET.

A pesar de todos los esfuerzos de estandarización, las detalladas especificaciones del sistema y el gran interés que despertó en su momento esta iniciativa entre las principales entidades financieras, el sistema SET ha ido decayendo en los últimos años, hasta ser hoy en día una propuesta que apenas ha llegado a implantarse a nivel mundial. Las claves de su anunciado fracaso pueden encontrarse en la complejidad intrínseca del sistema (protocolos con gran carga criptográfica, gran número de mensajes, complicados procedimientos de registro, necesidad de entidades mediadoras como las pasarelas de pagos), la necesidad de la colaboración de muchas entidades con intereses muy distintos, la reticencia de las entidades financieras a usar la red Internet como mecanismo de transporte y la influencia muy negativa que ha tenido la escasa implantación de la tecnología de PKI durante los últimos años.

Identrus

Identrus [102] se formó en 1999 como un consorcio de las principales entidades financieras del mundo, entre las cuales también se encuentran las principales entidades bancarias españolas. Su principal objetivo es potenciar el comercio electrónico B2B (*Business to Business* o comercio entre empresas). Las entidades financieras proporcionan los servicios de gestión de confianza, intentando crear un proceso comercial donde los participantes posean una única identidad digital que habilite los mecanismos de no repudio.

El núcleo de Identrus es una PKI de escala internacional donde las empresas son certificadas a través de sus instituciones financieras. Toda la infraestructura está dotada de un

sistema de validación en línea del estado de los certificados que opera mediante el protocolo OCSP. La PKI está estructurada jerárquicamente en los siguientes niveles:

- *Autoridad Raíz Identrus.* La raíz establece los procedimientos y las políticas de gestión de riesgos del resto del sistema. A este nivel se encuentra definido un repositorio público que proporciona información en tiempo real del estado de las entidades de nivel 1.

- *Autoridad de nivel 1.* Este nivel está formado por las principales instituciones financieras, las cuales certifican a las entidades de nivel 2 y a sus propios clientes.

- *Autoridad de nivel 2.* En este nivel se encuentran las instituciones financieras de menor entidad, las cuales deben ser certificadas por las instituciones de nivel 1.

- *Empresas.* Se trata de los clientes comerciales de las instituciones tanto de nivel 1 como de nivel 2, las cuales establecen transacciones entre sí haciendo uso de los servicios proporcionados por Identrus.

Este sistema se encuentra actualmente en marcha, y supone una simplificación sustancial frente a SET, ya que el sistema se ha reducido a un mecanismo de certificación digital para empresas, con las cuales es más fácil mantener relaciones comerciales confiables y de largo plazo.

EuroPKI

En lo que respecta a infraestructuras de certificación de ámbito europeo, varias han sido las iniciativas llevadas a cabo durante los últimos siete años destinadas a desarrollar un sistema de tales características. Tomando en consideración los resultados de proyectos como ICE-TEL [101] y ICE-CAR [100], varias organizaciones europeas definieron en Diciembre de 1999 la infraestructura denominada EuroPKI [72].

EuroPKI proporciona una autoridad de certificación raíz y un marco sobre el cual desarrollar pruebas de certificación jerárquica y cruzada entre las organizaciones participantes. La estructura actual de dicha infraestructura está formada por los siguientes elementos:

- Una autoridad raíz (denominada *Top Level CA*) getionada por el Instituto Politécnico de Turín.

- Varias autoridades de certificación subordinadas de nivel nacional, entre las cuales se encuentran la *CA Italiana*, *CA Eslovena*, *CA Polaca*, *CA Noruega (UNINETT)*, *CA Británica (University College of London)*, *CA Irlandesa (Trinity College of Dublin*, *CA Austriaca (IAIK)* y la *CA Española (IRIS-PCA).*

- Autoridades de certificación de tercer nivel asociadas a distintas instituciones y empresas de cada país.

Actualmente, este iniciativa se encuentra en una fase de indefinición, ya que tras haberse realizado las pruebas pertinentes en lo que respecta a interoperabilidad de los certificados emitidos por las distintas autoridades, la propuesta no ha establecido ningún objetivo concreto desde mediados del año 2001, y el número de países y organizaciones participantes sigue siendo muy bajo.

Proyecto CERES

La iniciativa española de certificación puesta en marcha por la Administración es el denominado Proyecto CERES (CERtificación ESpañola) [58], el cual se encuentra liderado por la Fábrica Nacional de Moneda y Timbre (FNMT). Su objetivo principal es el establecimiento de una autoridad de certificación pública que permita dotar de servicios básicos de autenticación y confidencialidad a las comunicaciones, realizadas vía Internet, entre las administraciones públicas y las empresas o ciudadanos.

El proyecto, además del uso de ficheros criptográficos, incluye la posibilidad del uso de dispositivos de almacenamiento seguro de información, como tarjetas inteligentes que contengan las claves privadas asignadas a las empresas y ciudadanos.

Su infraestructura de clave pública no es jerárquica, ya que está constituida por una única autoridad de certificación raíz gestionada por la FNMT. El proyecto especifica que las candidatas a constituirse como autoridades de registro son las oficinas de correos, presentes en la mayoría de localidades españolas.

En la actualidad, el principal entorno en el cual se ha hecho uso de este proyecto es en la presentación telemática del Impuesto sobre la Renta de las Personas Físicas (IRPF). Mediante dicho servicio, los ciudadanos tienen la posibilidad de presentar su declaración a la agencia tributaria a través de Internet previa obtención de un certificado digital X.509, el cual puede solicitarse actualmente sólo en las oficinas de la Agencia Tributaria, ya que las oficinas de correos no actúan en este momento como autoridades de registro.

2.5.2 Desarrollos previos realizados en la Universidad de Murcia

En el año 1997, la Universidad de Murcia inició un proyecto bajo el nombre de Proyecto SSL [35], cuyo objetivo principal era dotar a sus miembros de mecanismos que hicieran posible el establecimiento de comunicaciones seguras a través de la red corporativa. Se trataba de un problema bastante complejo debido a la gran cantidad de usuarios implicados (cerca de 40.000) y la heterogeneidad de los mismos (profesores, investigadores, personal de administración y servicios, alumnos). Uno de los objetivos fundamentales era que los desarrollos pudieran ser utilizados desde los puestos de trabajo, desde el hogar, o desde las ALAs (Aulas de Libre Acceso), lo que conllevaba tener que considerar varios sistemas operativos, dispositivos, navegadores, lectores de correo electrónico, etc.

El sistema diseñado finalmente tenía tres piedras angulares. En primer lugar, se basaba en el uso de las tarjetas inteligentes que la Universidad de Murcia proporciona a todos sus miembros, las cuales tienen capacidad para almacenar información confidencial. El segundo concepto clave era hacer uso de los estándares de seguridad existentes en dicho momento

para proteger las comunicaciones entre los miembros de la comunidad, más concretamente el protocolo SSL [9] para las comunicaciones HTTP seguras, y el protocolo S/MIME [168] para el envío de correo electrónico. La última piedra angular era hacer uso del estándar de certificación X.509 [99] para dotar de identidad digital a todos los usuarios y procesos.

Respecto a este último punto, es importante recalcar que en aquellos momentos los desarrollos en materia de certificación X.509 eran escasos y que, ya desde el principio, se tomó conciencia de que el ciclo de vida de los certificados y su integración con las tarjetas inteligentes sería la cuestión más compleja a resolver. La infraestructura de clave pública desarrollada finalmente [36] tomaba como base la herramienta Netscape Certificate Server (ahora conocida como Certificate Management System [155]), sobre la cual se construían el resto de los elementos del sistema, tales como las autoridades de registro o los servidores de directorio. El otro punto importante del proyecto respecto a la integración de los certificados X.509 y las tarjetas inteligentes fue el desarrollo de un módulo que seguía las especificaciones del estándar PKCS#11 [121]. Mediante dicho software era posible hacer uso de la información contenida en las tarjetas inteligentes (claves privadas y certificados de identidad) a la hora de establecer conexiones seguras mediante SSL o de intercambiar correos confidenciales y autenticados mediante S/MIME.

Centrándonos en los detalles de la infraestructura de clave pública, se desarrolló un sistema distribuido formado por varias autoridades de registro, una autoridad de certificación, y un servidor de directorio. Sólo las autoridades de registro se diseñaron e implementaron como parte del proyecto, siendo el resto de elementos productos comerciales. La PKI resultante proporcionaba servicios de certificación a través de las autoridades de registro, almacenamiento de información en las tarjetas inteligentes, publicación de datos en el servidor de directorio y gestión de solicitudes de revocación. Se trataba de un sistema piloto, muy limitado en ciertos aspectos, pero que era capaz de proporcionar la mayor parte de los servicios demandados en dicho momento.

La concesión posterior en el año 1999 del proyecto de investigación PISCIS (Piloto de definición de una Infraestructura de Seguridad para el Comercio Inteligente de Servicios), conllevaba la necesidad de diseñar y desarrollar una infraestructura mucho más versátil, capaz de proporcionar más servicios que los desarrollados en el marco del Proyecto SSL, así como de adaptarse a las nuevas tecnologías surgidas durante los últimos años tanto en materia de PKI como de tarjetas inteligentes e interfaces criptográficas. En el siguiente capítulo se abordarán los detalles de dicha PKI, diseñada e implementada en el marco de esta tesis.

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.

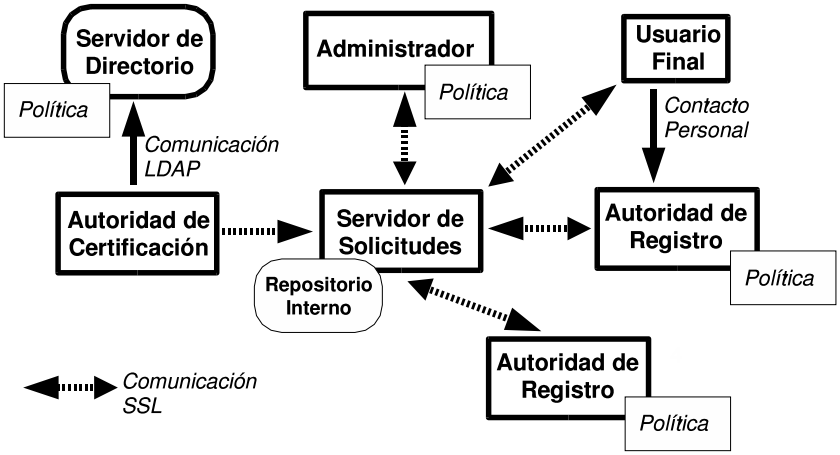

Figura 3.1: Colaboración entre las entidades de la PKI

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.

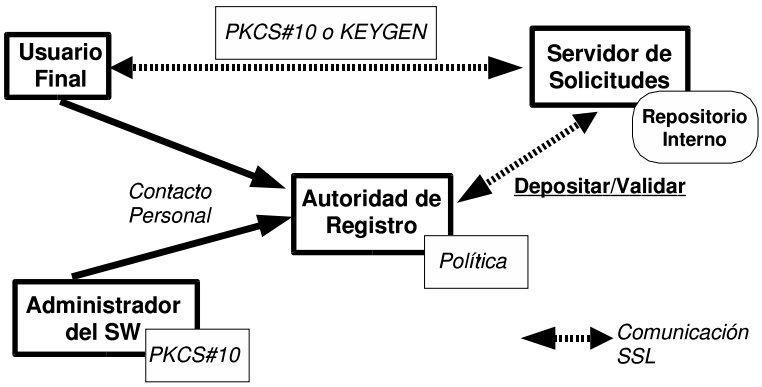

Figura 3.2: Alternativas del proceso de certificación

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.

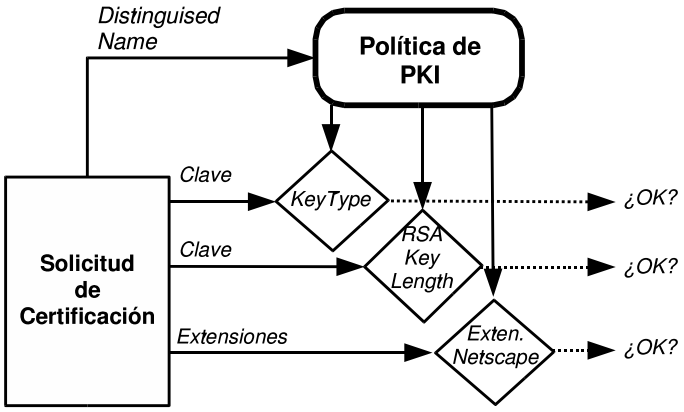

Figura 3.3: Ejemplo de cumplimiento de la política

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:

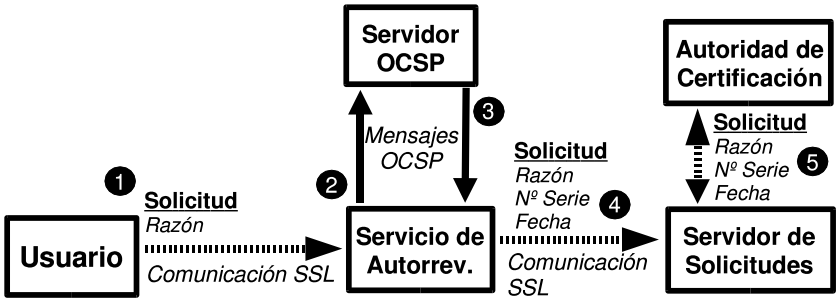

Figura 3.4: Autorrevocación mediante conexión autenticada

- 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.

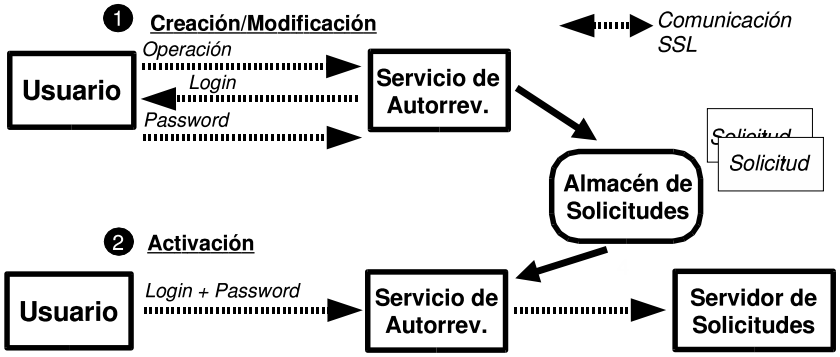

Figura 3.5: Autorrevocación en dos fases

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.

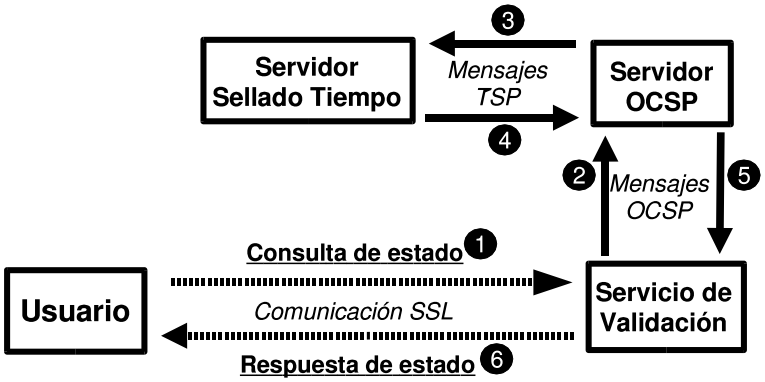

Figura 3.6: Validación del certificado a refirmar

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.

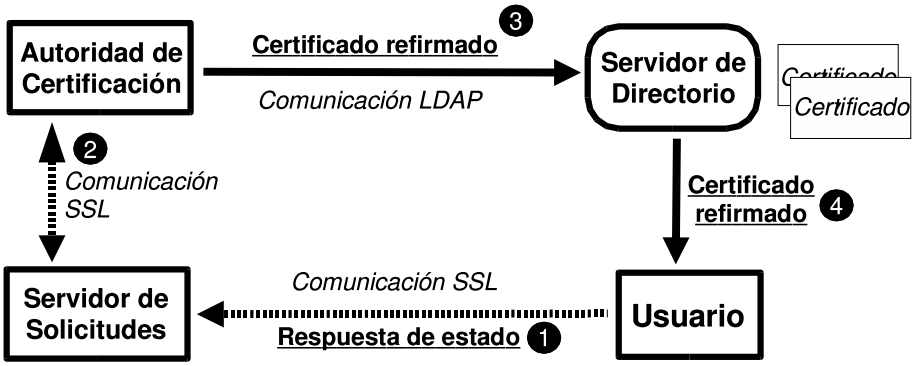

Figura 3.7: Obtención del certificado refirmado

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.

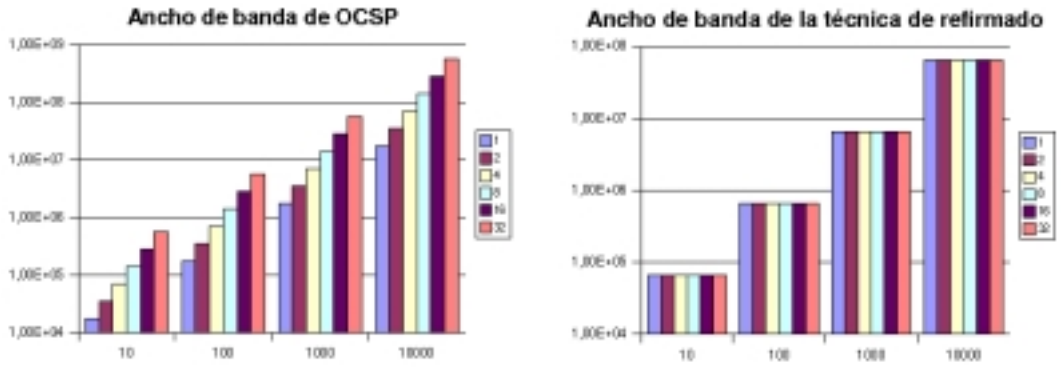

Figura 3.8: Comparativa entre OCSP y refirmado

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.

Capítulo 4

La certificación digital como mecanismo de autorización

En este capítulo se introducirán las principales carencias de los sistemas tradicionales de certificación de identidad a la hora de abordar un servicio básico de seguridad tan importante como el control de acceso. Se analizarán los diferentes modelos de control de acceso que han surgido a lo largo del tiempo, haciendo hincapié tanto en el control de acceso basado en roles como en el modelo basado en delegación. A continuación, se expondrán las diferentes especificaciones existentes en materia de certificados de credencial y se realizará un estudio acerca del estado del arte de la delegación en sistemas distribuidos como mecanismo de gestión de autorizaciones. El capítulo concluye con la identificación de las propuestas en materia de autorización que forman parte de esta tesis.

4.1 Carencias generales de los sistemas de certificación tradicionales

Aun contemplando la gran gama de servicios desarrollados en torno a los sistemas de certificación X.509, un análisis más en detalle de dichas propuestas revela una serie de carencias o incógnitas difíciles de resolver a la hora de aplicar dichos sistemas de forma más genérica al ámbito de los sistemas distribuidos. Si bien es cierto que el problema de la identidad digital queda bien resuelto haciendo uso de las infraestructuras de clave pública basadas en X.509, no podemos ignorar que el establecimiento de dicha identidad no es más que la asignación de un identificador a una clave pública, lo cual, como veremos en este apartado, no resuelve otras cuestiones relacionadas con el control de acceso, anonimato o la delegación de privilegios.

Antes de entrar en detalle con el análisis de dichas carencias, conviene definir cuáles son los servicios de seguridad contrastados a la hora de identificar las limitaciones de los sistemas de certificación de identidad:

- *Control de acceso.* El control de acceso comprende tanto los medios como los métodos

mediante los cuales se limita a los usuarios y otras entidades software, como los hilos de ejecución independiente o procesos en general, su capacidad de acceder y utilizar de alguna forma los recursos almacenados en un sistema de computación.

- *Anonimato.* El anonimato hace referencia a la posibilidad que tiene una entidad de acceder a los servicios ofrecidos por un sistema distribuido sin tener que revelar su identidad. Es conveniente recalcar que existe una clara diferencia entre identificador e identidad de la entidad solicitante de los servicios. Por identidad consideraremos todos aquellos identificadores que establecen una relación unívoca entre las acciones del solicitante y su identidad en el mundo real. Identificador es un término más amplio que abarca tanto los patrones empleados para identificar personas como patrones menos estructurados, como los resúmenes digitales o los valores de las claves públicas.

- *Delegación.* La delegación se entiende como la distribución de privilegios entre las entidades de un sistema distribuido. El privilegio obtenido por la entidad receptora es independiente del privilegio asociado a la entidad que lo emitió, en el sentido de que la revocación de este último no implica la revocación del primero.

En los siguientes apartados se analizarán las carencias más importantes del estándar X.509 respecto a los servicios anteriormente definidos.

4.1.1 Respecto al control de acceso y la autorización

En la mayoría de los sistemas distribuidos, la interacción de los usuarios con el sistema puede dividirse en dos fases bien diferenciadas. Por un lado, encontramos la fase de establecimiento de sesión, en la cual el usuario se identifica frente al sistema de cara a ser autenticado. En la segunda fase, una vez que se ha determinado que es una entidad válida dentro del entorno, se entra en un ciclo indefinido de tramitación de solicitudes, las cuales serán aceptadas, o no, dependiendo de la política de control de acceso de cada sistema, de los permisos asignados al usuario y de los parámetros del contexto.

En contraste con la identificación digital, para la cual se han desarrollado los estándares e infraestructuras analizados en los capítulos anteriores, el control de acceso ha sido tradicionalmente una función dependiente de la aplicación en cuestión y, por tanto, muy ligada al entorno en el cual se encontrara ubicada. En general, el proceso llevado a cabo está basado en la recepción de solicitudes firmadas digitalmente, la determinación del signatario de las solicitudes y la comprobación de si dicho signatario tiene concedido el acceso a los recursos necesarios para efectuar la acción solicitada. Sin embargo, hay varias razones por las cuales se considera necesaria la especificación de mecanismos eficientes de control de acceso en sistemas distribuidos:

- El número de entidades solicitantes puede llegar a ser muy elevado, al igual que el número de solicitudes.

- Ambos conjuntos cambian dinámicamente y no pueden ser conocidos completamente de antemano.

- La certificación de identidad proporciona simplemente un índice, al cual hay que asociar después los privilegios o permisos que puede ejercer.

- En algunos sistemas de certificación no resulta trivial determinar quién firma la solicitud, especialmente cuando se trabaja con cadenas de certificación largas.

- Resulta muy complejo calcular previamente todos los derechos de acceso con el fin de codificarlos como parte de la aplicación.

Tal y como se vio en la sección 2.2, los certificados de identidad constan de campos que contienen información relativa a la entidad certificadora, periodo de validez, información de la clave pública de la entidad certificada y uno o varios identificadores para dicha entidad. Por tanto, por sí mismos no constituyen ningún mecanismo de especificación de permisos, es decir, no explicitan qué recursos son accesibles por parte de los usuarios y bajo qué circunstancias. Por supuesto, no se puede hacer uso sólo de la identidad con el propósito de autorizar el acceso a recursos, ya que la identidad nos es más que un índice con el cual acceder a otro tipo de información y no un fin en sí mismo para estos escenarios. Por ejemplo, un mecanismo bastante extendido para controlar el acceso a páginas Web consiste en la comprobación de que el usuario está certificado por una autoridad de certificación reconocida como confiable por el servidor. Dicho mecanismo no permite realizar un tratamiento pormenorizado de los distintos privilegios que pueden disponer las comunidades de usuarios certificadas por la misma autoridad, incluso en el supuesto de que las decisiones se tomaran en función de los nombres X.500 asociados a dicha autoridad, lo cual implica que la mayoría de las decisiones deban realizarse como una función interna preprogramada de los servicios ofrecidos. Como veremos más adelante, el esquema de nombramiento X.500 no logra reflejar de forma apropiada los distintos roles que los usuarios juegan dentro del sistema y, por tanto, un nombre distinguido puede resultar inútil a efectos de autorización.

Parte de la comunidad científica consideró en un principio que el mecanismo de extensiones de los certificados X.509v3 podría asimilar las necesidades en lo que a especificación de privilegios se refería. La idea consistía en incluir los privilegios de cada usuario como parte del certificado de identidad, codificados como parte de una extensión (más concretamente de la extensión *subjectDirectoryAttributes*). Sin embargo, esta alternativa pronto encontró serios inconvenientes, los cuales pueden resumirse en los siguientes puntos:

- La entidad encargada de certificar quiénes son los usuarios puede no estar autorizada a determinar qué pueden hacer los mismos. Ha de tenerse en cuenta que la emisión de certificados de identidad puede considerarse un proceso centralizado, caracterizado por la confianza absoluta de los usuarios hacia la autoridad de certificación. Por el contrario, la determinación de los criterios por los cuales se autoriza a los usuarios suele tener un carácter más local. Dichos criterios suelen ser establecidos por una o más entidades especiales con un control más directo sobre los servicios proporcionados del que puede tener una autoridad de certificación central.

- La utilización de un mismo documento, el certificado de identidad, conlleva a que cualquier cambio en alguno de los permisos contenidos en el mismo (por ejemplo, una extensión o revocación de los mismos, o bien la inclusión de nuevos privilegios) produzca la revocación y emisión de un nuevo certificado que refleje la situación actual. Dado que los privilegios asociados a una entidad cambian de forma mucho más dinámica que su identidad, esto conllevaría a una reemisión bastante continua de los certificados, mucho más de lo especificado en el periodo de validez de los mismos. Este hecho tiene dos consecuencias muy negativas: la primera es que produce una sobrecarga en el sistema de gestión, el cual tendrá que tramitar de forma continua los cambios que se vayan produciendo; la segunda es que el mecanismo de revocaciones y validación de certificados se ve gravemente afectado, ya que la modificación de cualquiera de los certificados existentes conlleva la revocación de los anteriores, haciendo que las listas de control de certificados revocados crezcan de forma alarmante.

- El periodo de validez asociado a una identidad es generalmente mucho mayor que el asociado a un permiso. Sin embargo, dicha diferencia es difícil de constatar en los certificados de identidad debido a que poseen un único campo habilitado para reflejar dicho intervalo. Por ejemplo, el uso de intervalos temporales grandes no reflejarían de forma apropiada las políticas de autorización del sistema, al igual que los periodos cortos podrían ir contra lo especificado por las prácticas de certificación.

- En la mayoría de entornos de control de acceso, la gestión de los permisos se suele realizar teniendo en cuenta que los usuarios suelen formar grupos como consecuencia de desempeñar los mismos roles (el control de acceso basado en roles se analizará más en detalle en la sección 4.2.3). El uso de roles simplifica enormemente el control de acceso, ya que los permisos pueden ser asociados directamente a los roles. Sin embargo, el uso de extensiones nos permite sólo asociar información a las entidades que están certificadas, es decir, a aquellas entidades que poseen una clave pública. Los roles, al ser simplemente agrupaciones conceptuales de usuarios, no tienen asociada ninguna clave pública, lo cual entra conflicto con el criterio de certificación X.509.

Como conclusión respecto al control de acceso, se puede afirmar que el simple uso de certificados de identidad, con o sin extensiones, no introduce grandes ventajas a la hora de implementar este tipo de servicio. Veremos más adelante que las autorizaciones pueden tratarse como documentos digitales independientes, emitidos por distintas entidades denominadas autoridades de autorización o autoridades de atributo, que pueden contener una referencia a los certificados de identidad con los cuales están relacionados y que permiten que su gestión sea independiente de las aplicaciones que hagan uso de ellos.

4.1.2 Respecto al anonimato

En el ámbito de los sistemas distribuidos, muchas son las aplicaciones en las cuales no es necesario, o incluso aconsejable, revelar la identidad de los participantes a la hora de realizar ciertas operaciones o de utilizar ciertos servicios [46]. En algunos casos, la identidad

sólo se desvela en situaciones de conflicto, cuando se ha producido alguna amenaza o ataque al sistema. Por ejemplo, en entornos de comercio electrónico, los clientes podrían tener derecho a realizar sus compras de forma anónima, justo como lo hacen cotidianamente al pagar el periódico en un kiosco, o un libro en una librería. En otros entornos de control de acceso, como el control de acceso físico a instalaciones, puede no resultar necesario identificarse antes de abrir una puerta haciendo uso de un dispositivo instalado para tal efecto, ya que en la vida real no decimos en voz alta nuestro nombre al abrir la puerta de un laboratorio de investigación, sino que simplemente usamos la llave, es decir, el elemento que nos autoriza a entrar. Es obvio que en este último caso primero debemos haber obtenido una copia válida de la llave, proceso para el cual debemos haber demostrado que teníamos derecho a dicha copia, pero se trata de un proceso de autenticación inicial, no reiterado con cada acceso.

Sin embargo, el uso de certificados X.509, los cuales contienen uno o varios identificadores que pueden llegar a proporcionar gran cantidad de información acerca de los usuarios, complica la construcción de servicios basados en el anonimato.

Si nos centramos en el ámbito de la autorización, lo deseable en algunos entornos es establecer algún mecanismo que asegure que los permisos o la pertenencia a roles sigue siendo válida y confiable, pero sin que ello conlleve revelar la identidad. Como veremos en la sección 4.4.3, este mecanismo de anonimato puede llevarse a cabo haciendo uso de varias técnicas basadas en claves temporales y reducción de autorizaciones.

4.1.3 Respecto a la delegación de privilegios

Por delegación de privilegios entendemos el acto de propagar a otra entidad, ya sea un ser humano o un proceso software, parte de los permisos asociados al usuario que los delega. Mediante este mecanismo, la entidad delegada queda autorizada a realizar las operaciones implicadas como si de la entidad original se tratara. En algunos campos de los sistemas distribuidos, como por ejemplo el campo de los agentes inteligentes mediadores [157], dicha delegación puede llegar a ser necesaria para que la entidad que actúa como representante pueda llevar a cabo la tarea que le ha sido encomendada.

Además, en ciertos entornos autónomos, como el de los agentes de comercio electrónico [129], no es factible pretender que el usuario se encuentre disponible para verificar su identidad o autoridad siempre que se requiera. El sistema perdería parte de su autonomía si se requiriera la intervención humana en aquellas operaciones que representan una verificación de la autoridad del usuario.

Es necesario dejar claro que se está hablando en todo momento de delegación de la autoridad de realizar una tarea y no de delegación de identidad, puesto que la identidad es única e intransferible. El hecho de que la identidad sea única no debe confundirse con el hecho de que el esquema de identificación empleado sea globalmente único, sino que podría serlo simplemente a nivel local como se verá más adelante.

Los certificados de identidad X.509 ofrecen pocas alternativas en lo que a soporte para delegación se refiere. De nuevo, el único mecanismo a utilizar podría ser el sistema de extensiones, en las cuales podríamos especificar los privilegios delegados y las entidades

receptoras de los mismos. Sin embargo, esto implicaría saber de antemano cuáles van a ser dichas entidades y dicho conjunto de privilegios, lo cual en la mayoría de los casos es imposible debido a la dinamicidad de la mayoría de los sistemas distribuidos. Un cambio en las condiciones de delegación implicaría un cambio en el certificado, con todos los problemas que derivados de ello que fueron analizados en la sección 4.1.1.

La delegación como mecanismo de gestión de autorizaciones se analizará en detalle en las secciones 4.2.4 y 4.4.

4.1.4 Conclusiones

Las infraestructuras de clave pública clásicas y, por tanto, el sistema de certificación X.509, han alcanzado un estado de madurez considerable en lo que a identidad digital se refiere. Por otro lado, como se ha visto en esta sección, no ofrecen por sí mismas mecanismos sólidos sobre los cuales ofrecer de forma distribuida servicios de control de acceso o de autorización en general. Por tanto, se puede afirmar que estas infraestructuras de clave pública son sólo el primer paso hacia la creación de sistemas distribuidos seguros, una herramienta inicial que contesta a la pregunta de *¿quién es esta entidad?* y sobre la cual se puede seguir construyendo una serie de servicios adicionales más enfocados a contestar la otra gran pregunta, *¿qué puede hacer esta entidad?*. A lo largo de la siguiente sección veremos los distintos enfoques que se han seguido en los últimos años en el ámbito de los sistemas distribuidos para proporcionar soluciones al problema del control de acceso. Las cuestiones relacionadas con el anonimato y la delegación de privilegios se analizarán en la sección 4.4.

4.2 Modelos de control de acceso

El propósito de un sistema de control de acceso es mantener la confidencialidad, integridad y disponibilidad de los recursos mediante la provisión de mecanismos que hagan muy difícil a entidades no autorizadas acceder a los mismos.

Butler Lampson [125] fue el primero en definir formalmente los conceptos básicos relacionados con la protección de sistemas distribuidos: la matriz de control de acceso y sus dos posibles representaciones, las listas de control de acceso *(ACL, Access Control List)* y las listas de capacidades o competencias *(capability lists)*. En la primera de ellas, se almacena una lista de usuarios autorizados por cada recurso, mientras que en la segunda se almacena una lista de derechos de acceso por cada usuario. Aunque estas definiciones son del año 1974, los conceptos siguen siendo lo suficientemente generales como para seguir empleándose hoy en día.

Se desprende de la definición, tanto de las ACL como de las listas de capacidades, que ambos elementos deben ser protegidos de modificaciones no autorizadas, ya que dichas modificaciones afectan directamente al modo de controlar el acceso a los recursos. De algún modo, ambos objetos forman parte también del propio sistema de control de acceso y, por tanto, es necesario también controlar qué entidades son capaces de modificarlas.

Como se verá en apartados posteriores, esta tesis está centrada especialmente en las listas de capacidades y, más concretamente, en los derechos de acceso firmados digitalmente, a los cuales también se les suele denominar credenciales.

4.2.1 Mandatory Access Control (MAC)

En 1973, Lampson identificó también lo que llamó el problema de la reclusión [124] *(confinement problem)*, es decir, cómo prevenir la filtración de información confidencial a través de canales de comunicación protegidos. Esta definición implicaba una desconfianza explícita hacia los usuarios y los componentes del sistema, lo cual propició un gran esfuerzo de investigación en materia de seguridad informática por parte del ejército americano durante la década de los ochenta. Se trataba de diseñar sistemas MAC *(MAC, Mandatory Access Control)* que mediante medios técnicos hicieran imposible el acceso a información y el establecimiento de comunicaciones no autorizadas por la política de seguridad del sistema. Este tipo de mecanismos que proporcionan un control total sobre las acciones de los usuarios recibe el nombre genérico de base de computación confiable *(TCB, Trusted Computing Base)*.

El modelo más extendido de política de seguridad MAC está basado en acreditaciones y niveles de seguridad. Cada recurso está etiquetado con un nivel de seguridad que puede tomar los valores no catalogado, desclasificado, restringido, confidencial, secreto y muy secreto. A cada usuario se le asigna una acreditación, la cual especifica un cierto nivel de seguridad que le concede el acceso a todos los recursos etiquetados con el mismo o inferior nivel de seguridad, pero nunca superior. Por ejemplo, el modelo de Bell-LaPadula [22] protege la confidencialidad de los datos impidiendo el flujo de recursos etiquetados con un alto nivel de seguridad hacia usuarios con una acreditación baja.

El estricto control impuesto por los sistemas MAC hacía imposible que los usuarios pudieran establecer sus propias políticas de control de acceso, ni siquiera para los datos que ellos mismos creaban. En entornos civiles, esta rigidez suponía un gran problema, ya que los usuarios están acostumbrados a disponer de mayor autonomía e independencia en lo que a decisiones de seguridad se refiere. Como consecuencia, se propusieron algunas modificaciones al modelo MAC, como el sistema ORGCON [5] o el sistema ORAC [139]. Sin embargo, el modelo que más se impuso en los entornos no militares fue el control de acceso discrecional *(DAC, Discretionary Access Control)*, el cual se analizará en el apartado siguiente.

4.2.2 Discretionary Access Control (DAC)

El control de acceso discrecional [61] difiere del MAC en el hecho de que los usuarios están autorizados a asignarse permisos entre sí, es decir, los usuarios pueden permitir o denegar el acceso a los recursos que ellos mismos controlan.

El control de acceso discrecional suele articularse mediante el empleo de listas de control de acceso. En dichas listas, los distintos controladores de recursos especifican qué usuarios o grupos de usuarios pueden acceder a sus recursos y qué operaciones pueden realizar sobre

cada uno de ellos. Por ejemplo, en el caso de los sistemas operativos convencionales, los usuarios tienen derecho a especificar la política de acceso a sus ficheros, política que será impuesta por el propio sistema operativo, el cual actúa como controlador o monitor. En este caso, los permisos pueden especificarse en forma de grupos de usuarios y patrones de bits que representan los permisos asociados a los ficheros, lo cual es una variante de la matriz de control de acceso introducida por Lampson.

Sin embargo, en el caso de los sistemas distribuidos, en los cuales la asignación de permisos y el cumplimiento de políticas debe realizarse de forma descentralizada, veremos que el enfoque seguido para implementar los sistemas DAC hace uso de otros mecanismos. En estos escenarios, las ACL no son del todo apropiadas por varias cuestiones:

- Muchos sistemas distribuidos operan en entornos muy abiertos, en los cuales la identidad de los posibles usuarios del sistema no puede ser conocida de antemano. Esto hace imposible usar las ACL clásicas como mecanismo de control de acceso. En tal caso, lo que se necesita es un método y una infraestructura capaz de añadir y eliminar usuarios de forma dinámica.

- El modelo clásico de ACL es muy estático y su administración se realiza normalmente de forma centralizada, es decir, mediante un conjunto de administradores que realizan los cambios en la ACL cuando es necesario.

- Las ACL no reflejan el modelo de gestión de responsabilidades derivado a partir de la estructura de una organización, es decir, no se articula según el organigrama de la misma. Esto hace difícil descentralizar la gestión de los permisos entre varias entidades de mayor peso, las cuales pueden administrar de una forma más eficiente un conjunto de permisos y/o recursos.

- Se concede demasiado poder a los administradores de las listas de control de acceso, ya que tienen plenos poderes para modificar la totalidad de la lista. Este hecho tiene dos inconvenientes principales: el primero es que hace difícil detectar un posible abuso de poder por parte de los administradores; el segundo es que en sistemas distribuidos que abarcan a varias organizaciones, la gestión del control de acceso basada en omnipotentes administradores choca con la mayoría de las políticas de seguridad de dichas organizaciones, las cuales requieren un control más descentralizado del sistema.

4.2.3 Role Based Access Control (RBAC)

El concepto de control de acceso basado en roles (RBAC) surgió con la aparición de los sistemas multiusuario y multiaplicación en la década de los 70. El concepto principal del RBAC es que los permisos se asocian a los roles y los usuarios son agrupados en distintos roles, lo cual simplifica enormemente la gestión de los privilegios [75]. RBAC puede verse como un sistema de control de acceso independiente, el cual puede coexistir con MAC y DAC, pero que proporciona otras ventajas a la hora de modelar de forma más intuitiva las políticas de control de acceso.

Las relaciones entre usuarios y roles, y entre roles y permisos se tratan de forma totalmente independiente, en el sentido de que los usuarios pueden ser incorporados o eliminados de ciertos roles con independencia de los permisos que son asignados a dichos roles. Se puede decir que los roles constituyen un concepto semántico sobre el cual se articulan las políticas de control de acceso. Se trata de un elemento estable dentro del sistema, puesto que si bien el conjunto de usuarios que pertenece a un rol, o la serie de permisos asignados al mismo, puede ser muy dinámico, el rol permanece inalterado al ser un concepto ligado a la estructura organizacional del sistema que se está modelando, lo cual es inherentemente más estático. Las relaciones entre roles, usuarios y permisos puede contemplarse en la figura 4.1.

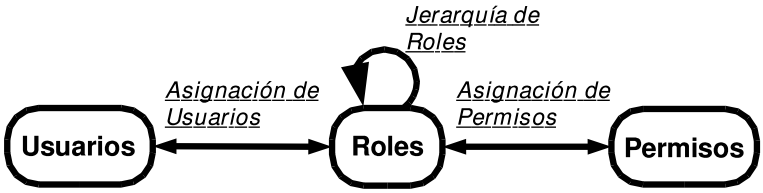

Figura 4.1: Relación entre elementos RBAC

En 1996, Ravi Sandhu [178] especificó una familia de modelos de referencia RBAC formada por cuatro modelos distintos:

- El modelo básico, denominado $RBAC_0$, especifica los elementos mínimos que debe contener un sistema RBAC. Entre dichos elementos encontramos el conjunto de usuarios, el concepto de rol, el conjunto de permisos y el concepto de sesión. Un usuario establece una sesión para activar uno o varios de los roles de los cuales es miembro. En dicho caso, el conjunto de permisos de los que puede disfrutar el usuario es fruto de la unión de los permisos asignados a todos los roles que han sido activados. En cierto modo, las sesiones representan la dinámica del sistema.

- El modelo $RBAC_1$ introduce las *jerarquías de roles* (ver figura 4.1). Dichas jerarquías son un medio natural de estructurar los roles con el fin de reflejar la estructura de autorización y responsabilidad de una organización. En un sistema $RBAC_1$, los permisos asignados a los roles fluyen a través del árbol que forma la jerarquía, pudiendo ser limitados en cualquier nodo del árbol de forma que no sean heredados por sus descendientes.

- El modelo $RBAC_2$ introduce el concepto de *restricción*. Realmente no se trata de una evolución del modelo $RBAC_1$, sino una ampliación independiente del modelo $RBAC_0$. Las restricciones son un aspecto importante del RBAC ya que sirven para especificar, por ejemplo, que dos roles son disjuntos (en el sentido de que un mismo usuario no puede pertenecer a ambos roles).

- Por último, el modelo $RBAC_3$ es la integración del modelo $RBAC_1$ y del modelo $RBAC_2$, es decir, un sistema RBAC con jerarquía de roles y restricciones al cual se

le denomina *modelo consolidado*. En este modelo, las restricciones se pueden aplicar a la propia jerarquía de roles, permitiendo así controlar el número de ascendientes o descendientes que puede llegar a tener un nodo de la misma.

Una última cuestión es la determinación de quién está autorizado a modificar los propios conjuntos de usuarios, roles, permisos y las relaciones entre ellos. Para ello, se definen los llamados permisos administrativos, los cuales deben ser explícitamente definidos como parte del sistema y pueden ser ejercidos por varias entidades descentralizadas.

Gran parte de las aportaciones y los desarrollos que forman parte de esta tesis están basados en el modelo RBAC, más concretamente en el modelo $RBAC_1$, puesto que como se verá, el sistema distribuido de gestión de credenciales está basado en la agrupación de usuarios en roles y la estructuración de dichos roles en forma de jerarquías. Los permisos administrativos son asignados de forma descentralizada a distintas entidades haciendo uso de los mecanismos de delegación presentados en el próximo apartado.

4.2.4 Control de acceso distribuido basado en delegación

El control de acceso discrecional ha ido evolucionando hasta convertirse casi en un nuevo modelo totalmente descentralizado donde las operaciones de gestión de permisos pueden ser realizadas por cualquier usuario. Las listas de control de acceso se han transformado en certificados firmados digitalmente que expresan los permisos que los usuarios pueden ejercer dentro de un determinado escenario. Estos certificados, al estar protegidos criptográficamente, pueden ser ampliamente difundidos y utilizados más allá de los límites del propio sistema en el cual fueron creados.

Si además consideramos a todos los usuarios al mismo nivel, es decir, con capacidad para emitir certificados de autorización a cualquier otra entidad del sistema, encontramos un nuevo modelo de gestión de los derechos de acceso basado en lo que se ha venido a denominar certificados de delegación [15, 16]. Como veremos en el apartado 4.4, dichos certificados pueden formar una red compleja que, a diferencia de la estructura arbitraria del modelo de confianza de PGP (ver sección 2.1.1), refleja las relaciones organizacionales existentes entre las claves contenidas en dichos certificados, y en consecuencia entre sus poseedores.

Un certificado de delegación es un documento firmado digitalmente mediante criptografía asimétrica en el cual una entidad concede ciertos privilegios a otra entidad. La estructura general de este tipo de certificados es la mostrada por la figura 4.2.

La semántica de este certificado puede interpretarse como que la entidad emisora concede los privilegios especificados a la entidad receptora, la cual podrá hacer uso de ellos durante el periodo de validez especificado y podrá a su vez propagarlos siempre que así se especifique en las restricciones de propagación. La sintaxis del privilegio es dependiente de cada entorno de aplicación, de forma que en función del entorno se definirán las reglas utilizadas para combinar y comparar los privilegios.

La delegación sólo tiene efecto siempre que el emisor del certificado tenga la autoridad que está intentando delegar. No obstante, es perfectamente posible delegar un privilegio

Figura 4.2: Certificado de delegación

antes de tenerlo, ya que el orden seguido para formar la cadena de delegación no tiene por qué coincidir con el orden de los certificados dentro de la misma.

Dichas cadenas de delegación se forman cuando una entidad delega en otra y ésta a su vez transfiere parte de los permisos en una tercera, y así sucesivamente. Si todos los certificados que forman la cadena delegaran los mismos permisos durante el mismo periodo de validez, la cadena podría verse como una propagación de los permisos asociados a la primera entidad de la misma. Sin embargo, lo más común es que tanto el conjunto de permisos delegados como los periodos de validez no coincidan. En consecuencia, el último elemento de la cadena obtiene los permisos resultantes de la intersección de los derechos asociados al primer emisor de la cadena y de las autorizaciones especificadas en cada uno de los certificados de la misma. De igual forma, el periodo de validez de la cadena es igual a la intersección de los periodos de validez contenidos en ella. Por ejemplo, en la cadena de la figura 4.3, la clave $K1$ autoriza a $K2$ a acceder al servidor FTP *ftp.delegation.org* durante el mes de octubre del año 2002. A su vez, ésta autoriza a la clave $K3$ a acceder al directorio X contenido en dicho servidor FTP, y lo hace sin límite de tiempo. Por último, la clave $K4$ es autorizada por $K3$ a acceder a cualquier servidor FTP en el periodo de tiempo comprendido entre septiembre y noviembre del año 2002. Supongamos que $K4$ quisiera acceder al servidor de FTP *ftp.delegation.org* y que este servidor hubiera sido configurado de forma que sólo concede el acceso a la clave $K1$ y sus posibles delegados. Esto implicaría que $K4$ debería presentar toda la cadena de delegación para poder demostrar que existe un camino de autorización desde $K1$ hasta $K4$, el cual autoriza a $K4$ a acceder al directorio X de dicho servidor sólo durante el mes de octubre. Esto es así porque el periodo de validez y los permisos obtenidos por $K4$ a partir de la cadena son el resultado de la intersección de todos los certificados que la forman.

Una vez vistos los conceptos básicos, en el apartado 4.4 se realizará un análisis más detallado de las ventajas y retos de los sistemas basados en delegación.

Figura 4.3: Cadena de delegación

4.3 Estudio de las especificaciones sobre certificados de credencial

Tal y como se definió anteriormente, se denomina certificado de credencial (en ciertos foros se usan los términos certificado de autorización o de atributo para hacer referencia al mismo concepto) a una sentencia firmada digitalmente, la cual especifica un conjunto de privilegios asignados a una entidad por parte de un emisor.

En los últimos años, varias han sido las propuestas realizadas en materia de certificados de credencial. Como se verá, cada una de ellas adopta enfoques distintos a la hora de intentar aportar una representación de las listas de capacidades introducidas en el apartado 4.2, teniendo siempre como característica común la creación de certificados firmados digitalmente mediante criptografía asimétrica. Todas estas propuestas son conscientes de que las infraestructuras de clave pública basadas en X.509 constituyen el pilar fundamental en lo que a distribución de claves y asignación de identidad digital se refiere, y por tanto todas ellas toman como base este tipo de infraestructuras a la hora de incorporar mecanismos de autorización. Como se verá, el punto de partida para la mayoría de ellas son las claves públicas contenidas en los certificados de identidad, a partir de las cuales son capaces de establecer las distintas políticas de seguridad, documentos acreditativos de autorización y relaciones de confianza entre las distintas entidades del sistema.

Sin embargo, el sistema por el cual las credenciales son distribuidas entre los clientes o los servidores, su método de publicación o creación, la implementación concreta de los repositorios públicos de autorizaciones o de los sistemas de revocación, son aspectos que no suelen ser tratados ni definidos en la especificación de estos sistemas. Con esto se quiere decir que el objetivo de dichas especificaciones no es la definición de un marco completo de implantación y uso de las mismas, sino que suelen centrarse en la propuesta de un lenguaje que satisfaga las necesidades concretas de cada entorno de control de acceso, pero sin entrar en detalle de cómo realizar la mayoría de los pasos relacionados con la dinámica del sistema. La forma concreta de llevar a cabo esta dinámica es parte del trabajo de esta tesis, como se verá en detalle en las secciones 5.3 y 5.4.

El estudio aquí realizado permitirá determinar cuál de estas especificaciones será empleada para desarrollar los servicios de autorización introducidos en el siguiente capítulo.

4.3.1 PolicyMaker

PolicyMaker es un modelo de gestión de confianza basado en un lenguaje de especificación de acciones confiables y relaciones de confianza. Su artículo introductorio [27] tiene como eje central la distinción entre política, credencial y relación de confianza, así como el lenguaje de especificación de éstas. De hecho, este artículo acuña el término de *gestión de confianza descentralizada* (Decentralized Trust Management), en contraposición con los esquemas tradicionales de certificación claramente centralizados.

PolicyMaker está basado en los siguientes principios fundamentales:

- *Mecanismo unificado.* Las políticas, credenciales y relaciones de confianza se expresan como programas (o parte de programas), empleando un lenguaje de programación seguro (entendiendo como seguro el hecho de que la ejecución de sus programas está confinada dentro de un entorno controlado).

- Debe tratarse de un sistema lo suficientemente *rico expresivamente* como para soportar relaciones de confianza complejas. Al mismo tiempo, políticas y relaciones más sencillas, como por ejemplo las derivadas de X.509 y PGP, pueden emplearse en PolicyMaker introduciendo simplemente ligeras modificaciones.

- *Localidad del control.* Cada entidad debe ser capaz de decidir cuándo aceptar las credenciales presentadas o en quién delegar las tareas de comprobación. Este control local de las relaciones de confianza evita realizar suposiciones globalmente conocidas y aceptadas, como sucede con las jerarquías de certificación tradicionales.

- *Diferencia entre política y mecánica.* El mecanismo de verificación de credenciales no depende del tipo concreto de credencial o de la semántica de la aplicación que las emplea.

En PolicyMaker se pretende que una aplicación, para permitir cierta acción, siga los siguientes pasos:

1. Obtener los certificados, verificar las firmas y determinar la clave pública de los solicitantes.

2. Verificar que los certificados no han sido revocados.

3. Enviar la solicitud, los certificados y la descripción de la política de la aplicación a un motor de gestión de la confianza (herramienta que precisa si una determinada solicitud está en consonancia con la política de seguridad del sistema).

4. Permitir el acceso si la respuesta ha sido satisfactoria.

Este método de control de acceso, es decir, la posibilidad de construir credenciales y políticas sin hacer referencia a identificadores, y por tanto sin emplear nombres que estén asociados a las autorizaciones, resulta muy apropiado para sistemas que requieren el anonimato.

Arquitectura del sistema

PolicyMaker es un servicio ofrecido a las aplicaciones, bien en forma de librería de enlace dinámico, o bien como servicio independiente accedido mediante una interfaz bien definida. Es similar a un motor de consulta a bases de datos. Acepta como entrada un conjunto de políticas locales, credenciales y acciones que se pretenden realizar, y devuelve una respuesta positiva o negativa acompañada, opcionalmente, por una serie de anotaciones que justifican la decisión tomada. Tanto las credenciales como las políticas están definidas en términos de predicados, llamados filtros, asociados a claves públicas de cualquier criptosistema asimétrico.

Una acción se considera aceptable (o que conforma con la política), si puede construirse una cadena de confianza desde la política hasta la clave solicitante, a través de la cual los filtros son satisfechos. Sin embargo, PolicyMaker no determina el formato concreto de las acciones, dejando a cada aplicación que las exprese de la forma más adecuada.

Lenguaje de autorización

Una consulta al sistema PolicyMaker es una solicitud para determinar si una determinada clave pública está autorizada a realizar cierta acción de acuerdo con la política local. El formato de la consulta es el presentado por la figura 4.4.

key1, key2, .., keyN **REQUESTS** *ActionString*

Figura 4.4: Consulta PolicyMaker

Las consultas son procesadas basándose en la información contenida en las credenciales (*asserts* en terminología de PolicyMaker), las cuales son sentencias que confieren autorizaciones a las claves. Los elementos de dichas credenciales (ver figura 4.5) son los siguientes:

Source **ASSERTS** *AuthorityStruct* **WHERE** *Filter*

Figura 4.5: Credenciales PolicyMaker

- *Source*: Hace referencia a la entidad que crea la credencial. Puede contener el valor *Policy*, cuando se trata de una credencial que forma parte de la política local, o la clave pública de la entidad que firma la credencial.
- *AuthorityStruct*: Conjunto de claves públicas a las que se aplica.
- *Filter*: Predicado que debe cumplir el *ActionString*.

En resumen, cada credencial establece la confianza en las claves públicas de la *AuthorityStruct* para realizar la acción que satisface el filtro, donde por credencial entendemos tanto los certificados firmados digitalmente como las políticas (iguales que los certificados

pero sin firmar, ya que son locales y válidas incondicionalmente). La política local puede autorizar directamente a ciertas claves para realizar determinadas acciones, pero normalmente delegará en emisores de credenciales en los cuales confía, puesto que dichos emisores, en general, tendrán un mayor conocimiento del entorno de aplicación y una relación más estrecha con los solicitantes.

Semántica de las consultas

PolicyMaker exige que al menos una de las credenciales sea local (es decir, parte de la política), aunque el resto sean proporcionadas en la propia consulta. La razón es que la prueba de la conformidad o no de una acción se construye tomando siempre como raíz de la cadena de confianza una credencial local. La construcción de estas pruebas está basada en la definición de grafos dirigidos donde cada vértice es una clave o una política y los arcos son filtros. El núcleo del sistema de comprobación de conformidad se encuentra descrito en [29].

Firmas digitales y lenguaje de programación de los filtros

Una cuestión que hay que aclarar es que PolicyMaker no verifica por sí mismo las firmas digitales. En este sistema, las claves públicas siempre identifican el programa con el cual deben procesarse (p.e.: PGP:0x01234567...) de forma que es un programa o librería externa quien se encarga de realizar este tipo de comprobaciones. La justificación de este hecho es la adaptabilidad a distintos sistemas criptográficos que puedan ir surgiendo, favoreciendo que PolicyMaker no se encuentre restringido a un conjunto de sistemas predeterminado.

Por otro lado, los filtros son programas interpretados dentro de un entorno de ejecución confiable. Los datos de entrada para dichos filtros son las acciones, contexto (fecha, hora, datos del sistema) y cadenas de credenciales. Aunque podría emplearse cualquier lenguaje interpretado, PolicyMaker se decanta por AWK, sin descartar la posibilidad de emplear Java o TCL.

Escenarios de uso

Los escenarios de uso que han sido propuestos y/o implementados mediante PolicyMaker son:

- *Sistemas de correo electrónico.* En [27] se muestra cómo PolicyMaker puede emplearse para dotar de autenticidad a los mensajes de correo electrónico (controlando la identidad de las claves, así como la vinculación organizacional del poseedor de la clave). También se propone emplear PolicyMaker (mediante credenciales con anotaciones) para obtener todos aquellos datos con los cuales asegurar un envío confidencial de los mensajes (tipo de cifrado, clave de cifrado, etc).

- *Servidores de validaciones.* Ya que PolicyMaker está basado en sentencias positivas (no puede depender de negaciones), en [27] se propone diseñar un servicio de emisión

de credenciales que especifiquen la validez de los certificados (bien bajo demanda o mediante multidifusión).

- *Sistemas sencillos de Workflow.* En [27], la intención es ilustrar la capacidad del sistema para tratar con solicitudes que deben ser validadas por varias entidades o que deben atravesar varias etapas hasta ser autorizadas.

- *Control de acceso a contenidos WWW mediante la utilización de un sistema de etiquetado de la información* [28]. Se propone el uso de PolicyMaker junto con sistemas como PICS [169], de forma que el usuario especifique claramente cuál es la política seguida para permitir o no la visualización de ciertos contenidos, atendiendo a criterios como la cantidad de violencia, sexo u otros factores presentes en la información. Para ello, empleando PolicyMaker, se propone tomar las decisiones en función de la valoración realizada sobre los contenidos por parte de entidades confiables, las cuales puedan ser autorizadas a emitir su juicio mediante certificados de credencial o mediante la propia política del usuario. Se trata de un sistema con organizaciones etiquetadoras, organizaciones que han obtenido su capacidad de etiquetar de forma delegada, contenidos etiquetados y usuarios finales. Posteriormente han surgido sistemas más avanzados, como REFEREE [44].

4.3.2 KeyNote

En 1998, los autores de PolicyMaker analizan en [26] varios de los modelos existentes de lo que ellos llaman gestión de confianza (*Trust Management*) en sistemas distribuidos. Entre ellos se encuentra KeyNote, la evolución de su PolicyMaker.

KeyNote [25] fue diseñado con dos objetivos muy claros en mente: su estandarización y la facilidad de integración en las aplicaciones. Para conseguir este objetivo, KeyNote asigna una mayor responsabilidad al motor de conformidad y menos a la aplicación (por ejemplo, ahora la verificación de las firmas digitales las realiza el propio motor y no una aplicación externa). Además, las credenciales y las políticas deben estar escritas en un lenguaje más cercano al motor. Las razones de este cambio en el lenguaje están relacionadas con la eficiencia, interoperabilidad y la tendencia a propiciar que credenciales y políticas sean reutilizadas fácilmente.

Al igual que PolicyMaker, requiere que las aserciones posean la propiedad de la monoticidad, evitando de esa forma que fallos de trasmisión en la red que impidan el envío de credenciales provoquen autorizaciones erróneas.

El motor de evaluación de KeyNote recibe como entrada una lista de credenciales y políticas, las claves públicas del solicitante y un entorno de acción *(AE, Action Environment)* creado por la aplicación, el cual contiene a su vez toda la información considerada relevante y necesaria para tomar la decisión de conformidad. La lista de pares (atributo, valor) que forman el AE debe reflejar de forma precisa los requisitos de seguridad de la aplicación, y quizá sea la tarea más importante a la hora de integrar KeyNote en las aplicaciones. La salida del motor de evaluación es una cadena de caracteres definida por la aplicación que

suele ser tan simple como "autorizado" o "no autorizado", en contraste con el mecanismo de anotaciones que aportaba PolicyMaker.

Las diferencias principales entre PolicyMaker y KeyNote son:

- Los predicados de KeyNote están escritos mediante una notación sencilla similar a las expresiones en lenguaje C y a las expresiones regulares.
- Los filtros KeyNote siempre devuelven un valor booleano como respuesta.
- La verificación de las firmas digitales asociadas a las credenciales forma parte del propio sistema KeyNote.
- Las acciones se describen de forma sencilla como pares atributo/valor.

Como veremos a continuación, las políticas y las credenciales siguen compartiendo la misma sintaxis. Ambos tipos de aserciones se escriben de forma independiente y son programas autónomos sin dependencias entre ellos. Al contrario de lo que sucedía con PolicyMaker, en el lenguaje de aserciones de KeyNote no hay bucles ni llamadas a funciones. La idea es diseñar un motor sencillo que pueda estar embebido en las aplicaciones o en el propio sistema operativo.

Sintaxis de las aserciones

La estructura de las aserciones en KeyNote (ver figura 4.6) es similar a la de las cabeceras del correo electrónico.

```
<Assertion>:: <VersionField>? <Authorizer> <LicenseesField>?
              <LocalConstantsField>? <ConditionsField>?
              <CommentField>? <SignatureField>?
```

Figura 4.6: Aserciones KeyNote

Los campos más importantes de dichas aserciones son:

- *Authorizer*. Identificador del emisor de la aserción (*Policy* en el caso de las políticas).
- *Licensees*. Los receptores de los permisos que concede la aserción. Puede estar formado por el Y lógico de varias claves, el O lógico o por la expresión *k-of-n*.
- *Local-Constants* permite definir valores locales dentro de una aserción y suele emplearse por claridad.
- *Conditions*. Condiciones bajo las cuales el emisor confía en los receptores para que realicen el acceso. Son predicados que operan con un conjunto de atributos escritos en un lenguaje de expresiones regulares, asignaciones y comparaciones similar a C.

- *Signature.* Contiene la firma de la aserción en el caso de que sea un certificado. Las aserciones no firmadas pueden emplearse sólo para especificar políticas.

Como puede apreciarse, no se incluye ningún campo relacionado con la validez de la aserción (periodo de validez, localización de un servidor de confirmación, etc.). En su lugar, en KeyNote se propone la utilización del campo Conditions para realizar comprobaciones con la fecha actual o mecanismos similares. Como se indicará más adelante, la caducidad o revocación de credenciales es un asunto no abordado en KeyNote.

Semántica de evaluación de las consultas

Los parámetros de una consulta KeyNote son los siguientes:

- Identificador de la(s) entidad(es) que solicitan la acción.
- El conjunto de atributos que describen la acción.
- El conjunto de valores de conformidad de interés para la aplicación, ordenados de menor a mayor.
- Las aserciones que se emplearán en la evaluación.

Para que se pueda realizar el cálculo de conformidad, los identificadores de las entidades deben estar normalizados, es decir, que las comparaciones entre identificadores se realizan siempre tras convertir la representación de las claves a una forma canónica.

En cuanto a lo que se refiere al cálculo del valor de conformidad, KeyNote no emplea el modelo de pizarra compartida de PolicyMaker. En su lugar, utiliza una búsqueda en profundidad que intenta satisfacer recursivamente un política. Los resultados intermedios son utilizados por el motor y, a diferencia de PolicyMaker, nunca hay comunicación entre las aserciones. El funcionamiento del motor [25] está caracterizado por su cálculo totalmente monotónico. El suministro de credenciales no apropiadas no significa la aprobación de acciones ilegales, así como la inserción de aserciones a una consulta nunca resulta en una respuesta de menor conformidad (de igual forma, la falta de credenciales no provoca considerar válidas acciones que no lo son).

Escenarios de uso

Los escenarios de uso que han sido propuestos y/o implementados mediante KeyNote son:

- *Seguridad en el nivel de red.* El encapsulado de mensajes mediante protocolos como IPSec es un terreno sencillo y bastante bien explorado. Sin embargo, la dificultad se encuentra a la hora de gestionar la política que gobierna el envío o la recepción de paquetes, siendo este problema especialmente complejo en los firewalls. En [30] se sugiere un marco sencillo de gestión de la confianza a este nivel que hace uso de KeyNote.

- *Redes activas* [26]. KeyNote se ha aplicado en el campo de las redes activas en el proyecto SANE.

- *Código móvil* [26]. Se emplean credenciales para expresar las condiciones bajo las cuales el código fue certificado, así como para describir el conjunto mínimo de características que el equipo receptor debe proporcionarle al código móvil para ejecutarse.

- *Firewalls*. En [104] se propone el uso de aserciones KeyNote para regular el tráfico que circula a través de los firewalls. Se trata de un esquema que intenta descentralizar el cumplimiento de las políticas de seguridad mediante una distribución de aserciones KeyNote basada en IKE [94] como protocolo de intercambio de las mismas.

Conclusiones

Tanto KeyNote como PolicyMaker adoptan una visión clara y concisa del problema de la gestión de la confianza en sistemas distribuidos. La distinción clara entre los conceptos de política, credenciales y relaciones de confianza permite adaptar el sistema a casi cualquier entorno de aplicación. Además, presentan no sólo una sintaxis de descripción de la información, sino también un motor de conformidad que independiza a las aplicaciones de la necesidad de realizar ellas mismas los cálculos de conformidad, proporcionando una herramienta común a todas las entidades implicadas en un sistema distribuido. Otra cuestión que resulta interesante es que la descripción del entorno de acción (AE) está definida por las aplicaciones implicadas, y no impuesta por el propio sistema (que ni siquiera conoce su estructura), lo cual le confiere una gran versatilidad y adaptabilidad a multitud de entornos.

Sin embargo, presentan algunos inconvenientes que impiden su uso más extendido y su popularidad. Sin duda, el hecho de que sus aserciones sean programables es su mayor ventaja y desventaja, puesto que implica el esfuerzo de plasmar en un lenguaje de programación decisiones de política que en ocasiones no resulta fácil traducir. Además, obligan a construir complejas herramientas de creación automática de aserciones a partir de los requisitos del usuario, intentando de esta forma ocultar al usuario final la complejidad del lenguaje de aserciones.

También en lo que a aspectos de infraestructura se refiere, carecen de un entorno definido de creación y distribución de credenciales, que al fin y al cabo es necesario para poder hacer uso del motor en aplicaciones concretas. Los propios autores proponen como vías futuras la resolución del descubrimiento de credenciales por parte del motor KeyNote, así como la comprobación de la información relacionada con revocaciones de las credenciales, o la generación y distribución de las mismas.

4.3.3 PMI (Privilege Management Infrastructure)

La cuarta edición del estándar X.509 [106], publicada por la ITU-T en el año 2001, es la primera edición que propugna la estandarización de los certificados asociados a una Infraestructura de Gestión de Privilegios *(PMI, Privilege Management Infrastructure)*, término

que se empleará en este documento única y exclusivamente para hacer referencia a las infraestructuras de gestión de autorizaciones basadas en el estándar X.509. Hasta la fecha, las versiones anteriores de X.509 se habían concentrado exclusivamente en el problema de la identidad digital y su gestión. Hoy en día, varios autores hablan de lo que sería el siguiente paso en la certificación digital X.509: la integración de las PKIs y las PMI en lo que se vendría a denominar Infraestructuras de Autenticación y Autorización *(AAI, Authentication and Authorization Infrastructures)* [57, 132].

Se podría decir que la PMI es a autorización lo que la PKI es respecto a la autenticación, y por tanto muchos de los conceptos de ambas infraestructuras son muy similares. El elemento clave a partir del cual giran las PMI es el certificado de atributo *(AC, Attribute Certificate)*, el cual establece una vinculación entre una entidad y un conjunto de atributos o privilegios (los términos atributo y privilegio se utilizarán indistintamente en esta sección). Este tipo de certificados son emitidos por las autoridades de atributo *(AA, Attribute Authorities)*, las cuales también se disponen de forma jerárquica, al igual que se vio en la sección 2.1.1 para las PKI, siendo la entidad raíz la denominada Fuente de Autoridad *(SOA, Source of Authority)*. Las SOAs pueden delegar parte de la autoridad que poseen en AAs subordinadas, distribuyendo de esta forma la responsabilidad en lo que a gestión de privilegios se refiere.

El grupo de trabajo PKIX ha publicado recientemente una especificación acerca de los certificados de atributo X.509 [74], la cual contempla sólo un subconjunto de las recomendaciones reflejadas por el documento de la ITU-T. A lo largo de esta sección, se remarcará claramente si lo presentado se corresponde con las propuestas PKIX o con los contenidos del estándar.

Certificados de atributo X.509

La estructura general de un certificado de atributo X.509 es la presentada en la figura 4.7. Como se puede apreciar, es muy similar a la de un certificado de identidad X.509, siendo dos las principales diferencias entre ambas especificaciones.

La primera de ellas hace referencia al campo denominado *Poseedor*. Este campo, utilizado para denotar a la entidad que recibe los privilegios, puede contener tres tipos distintos de identificadores.

- *Número de serie.* Este tipo de identificador se emplea para hacer referencia (mediante el número de serie) al certificado de identidad asociado al usuario que recibe el privilegio. Esto implica que se basa tanto en la existencia previa de dicho certificado como en una política de asignación de identificadores únicos a autoridades de certificación.

- *Nombre de Entidad.* El uso de nombres es especialmente útil en dos situaciones concretas. La primera de ellas hace referencia a la posibilidad de asignar privilegios a usuarios que no han recibido todavía su certificado de identidad, puesto que al basarnos en nombres y no en números de serie es posible emitir el certificado de atributo de forma independiente. La segunda razón está relacionada con escenarios

Figura 4.7: Certificado de atributo X.509

donde no se emplean certificados de identidad por estar basada la autenticación en otros métodos (por ejemplo en un login y password). En estos casos, los certificados de atributo pueden emplearse para aportar información acerca de privilegios una vez que la sesión ha sido iniciada y el usuario ha sido autenticado.

- *Resumen digital.* Sin duda alguna, se trata del identificador más versátil a la hora de hacer referencia a la entidad poseedora del privilegio. Mediante este mecanismo, es posible vincular los atributos a cualquier objeto cuyo resumen digital pueda ser calculado, como por ejemplo un código ejecutable, un certificado de identidad o incluso una clave pública, lo cual nos proporcionaría un buen mecanismo para asignar directamente privilegios a claves, sin necesidad de usar nombres.

La segunda gran diferencia respecto a los certificados de identidad es la inclusión del campo *Atributos*. Se trata del elemento que contiene los datos relativos al privilegio que se está asignando y, por tanto, puede contener cualquier tipo de información. No obstante, en la propuesta PKIX [74], se definen algunos atributos estándar que pueden ser empleados de forma genérica:

- *Información de autenticación.* Se trata de un atributo diseñado para proporcionar compatibilidad a sistemas ya existentes basados en login y password. Este atributo puede almacenar de forma cifrada ambos elementos de información con el fin de que el usuario pueda autenticarse frente al sistema mediante la presentación del certificado.

- *Identidad de acceso.* Sirve para identificar al poseedor del certificado frente a un determinado sistema. Por tanto, el identificador contenido en este atributo dependerá completamente del entorno de aplicación.

- *Grupo y rol.* Se emplean para contener información acerca de la pertenencia del poseedor a ciertos grupos o roles (este atributo se analizará en detalle más adelante).

- *Acreditación.* Este tipo de atributo está muy relacionado con los sistemas MAC ya que contiene el nivel de acreditación asociado con el poseedor del certificado.

Al igual que sucedía con los certificados de identidad, los certificados de atributo están dotados también de un mecanismo de extensiones que permite incluir información adicional acerca del certificado que se está emitiendo. En concreto, se han especificado algunas extensiones básicas que permiten matizar la información contenida en el campo relativo a atributos. Por ejemplo, la extensión *Time-specific* permite especificar el periodo de tiempo durante el cual tendrá vigor el atributo que se está declarando, y la extensión *TargetingInformation* puede emplearse para identificar el conjunto de aplicaciones a las cuales va destinado el atributo.

Delegación

La delegación es quizá el mecanismo en el cual difieren más los enfoques adoptados por el grupo de trabajo PKIX y por el propio estándar de la ITU-T. Mientras que la recomendación estándar incluso sugiere un modelo de PMI basado en delegación, la propuesta del grupo PKIX no considera aconsejable el uso de cadenas de delegación a la hora de gestionar los privilegios asignados por las distintas AAs. Desde este foro, se sugiere que cada AA gestione conjuntos disjuntos de privilegios, los cuales deben ser asignados directamente a los usuarios sin el uso de entidades intermedias, es decir, sin emplear AAs subordinadas. De esta forma, la SOA actuaría en un primer nivel, concediendo conjuntos de privilegios independientes a cada AA mediante certificados de atributo.

Sin embargo, la recomendación de la ITU-T sí contempla la delegación como un mecanismo aconsejable a la hora de gestionar ciertos escenarios de autorización. En estos casos, tanto las distintas SOAs como las AAs son capaces de asignar privilegios a otras AAs, así como de restringir la capacidad de éstas a la hora de seguir propagando los privilegios. De esta forma, se puede llegar a obtener una cadena de certificados de atributo arbitrariamente larga, cuya fuente es una de las SOA del sistema y cuyo último elemento será un usuario final. Para poder verificar que el usuario puede ejercer su privilegio, es necesario disponer de toda la cadena. Es importante remarcar el hecho de que una entidad podrá actuar como autoridad de atributo sólo en el caso de que así haya sido reconocida por otra autoridad o por la SOA, es decir, no es posible que cualquier entidad pueda constituirse en una emisora de privilegios si no llega a ser reconocida en ningún momento como tal.

La restricción en la propagación de la delegación se lleva a cabo empleando dos extensiones ya definidas. *BasicAttributesConstraints* sirve para especificar si el poseedor del certificado puede actuar a su vez como autoridad de atributo. Por otro lado, con el fin de controlar qué usuarios pueden recibir los privilegios, se ha habilitado otra extensión denominada *DelegatedNameConstraints* que especifica qué conjunto de nombres puede formar parte de la cadena de delegación.

Modelos de PMI

Se contemplan cuatro modelos distintos de PMI con posibilidades de ser usados en escenarios con distintas características.

El *modelo general* ofrece un marco abstracto en el cual pueden encuadrarse el resto de modelos. Considera sólo tres tipos de entidades, objeto, poseedor del privilegio y verificador del privilegio, que interaccionan en un escenario genérico de autorización. El objeto es el recurso protegido, sobre el cual se pueden realizar varias operaciones distintas que son controladas por el verificador del privilegio tras la solicitud realizada por el poseedor del mismo. Las decisiones se toman considerando las políticas de autorización del sistema, cuya definición no está contemplada en el estándar al asumir que es tarea del sistema final.

El modelo más sencillo que puede derivarse a partir del general es el *modelo de control*, indicado para escenarios de control de acceso. Dicho modelo no presupone una estructura de agrupamiento de usuarios ni la existencia de cadenas complejas de certificación.

El *modelo de roles* está completamente basado en RBAC y se estructura según los mecanismos vistos en la sección 4.2.3. Cabe destacar que, si bien la pertenencia a roles por parte de los usuarios se materializa mediante certificados de atributo [162], la asignación de los privilegios a dichos roles no forma parte de los mecanismos proporcionados por este modelo. El sistema final debe decidir cuál es el método más apropiado para reflejar esta relación.

Por último, el *modelo de delegación* contempla aquellos entornos de aplicación en los cuales puede ser aconsejable la existencia de varias autoridades a través de las cuales van fluyendo los privilegios de una forma más descentralizada. Sus características fueron ya expuestas en el apartado anterior.

Escenarios de uso

La especificación de este tipo de certificados es relativamente reciente, razón por la cual el número de escenarios en los cuales se ha aplicado con éxito es todavía reducido. Una de las iniciativa más antiguas es el denominado proyecto Akenti [184], centrado principalmente en la provisión de mecanismos de control de acceso a un escenario caracterizado por tres tipos de entidades: proveedores de contenido, usuarios y emisores de atributos. Los proveedores de contenido especifican las condiciones en las cuales conceden el acceso a los distintos usuarios, las cuales están codificadas como un conjunto de requisitos sobre unos atributos concretos. Dichos atributos son asignados a los usuarios mediante la utilización de certificados de atributo emitidos por las autoridades reconocidas del sistema. Sin embargo, hay que dejar constancia de que dichos certificados, aunque similares conceptualmente, no siguen el esquema X.509 sino que se trata de un formato de certificación desarrollado dentro del mismo proyecto.

Uno de los proyectos más recientes que hace uso de este tipo de certificados es el Proyecto PERMIS [40, 41]. El proyecto tiene como meta construir una PMI X.509 basada en el modelo de roles que pueda ser utilizada para varias aplicaciones distintas en tres ciudades de Europa, concretamente Barcelona, Bolonia y Salford. Los entornos de aplicación van

desde el acceso a bases de datos de multas de tráfico por parte de compañías de alquiler de coches, hasta acceso a mapas urbanos por parte de arquitectos. En general, se trata de aplicaciones de control de acceso a recursos centralizados.

Por último, dejar constancia de los desarrollos que nuestro grupo de investigación ha realizado en materia de certificados de atributo X.509. En concreto, se ha propuesto una extensión de la PKI del proyecto PISCIS con el fin de dar soporte a la creación y publicación de este tipo de certificados [84]. Dicha extensión se emplea para generar los certificados que serán posteriormente utilizados en distintos entornos de aplicación, entre los cuales se encuentra ya desarrollado un escenario de control de inicio de sesión en sistemas operativos Windows NT/2000 que hace uso de este tipo de certificados para codificar datos relativos a la sesión del usuario (login, password, dominio de sesión).

Conclusiones

Sin duda alguna, la nueva recomendación de la ITU-T supone un paso importante hacia la creación de sus denominadas PMIs. La propuesta, en la mayoría de los aspectos, es lo suficientemente genérica y amplia como para dar cabida a varios escenarios, así como a diferentes modelos de gestión de los privilegios.

Una de las principales ventajas que aporta es que no presupone la existencia previa de una PKI, permitiendo que los certificados puedan ser ligados a entidades identificadas mediante mecanismos distintos de un certificado de identidad. Asimismo, aunque sí se ha proporcionado un conjunto estándar de atributos, el sistema es lo suficientemente flexible como para incorporar nuevos tipos de atributos que puedan resultar necesarios en cada entorno.

Sin embargo, hay varias cuestiones que quedan sin resolver una vez examinado el estándar. En primer lugar, el modelo basado en roles puede parecer incompleto, puesto que, si bien se proporcionan los mecanismos necesarios para reflejar la pertenencia de los usuarios a los roles, no se hace tanto hincapié en cómo reflejar la asignación de permisos a roles mediante los propios certificados de atributo (a diferencia de otros sistemas como el que veremos en la próxima sección) o, incluso, en cómo especificar una jerarquía de roles que sigue un modelo $RBAC_1$ (ver sección 4.2.3). Dejar al sistema final la decisión de cómo especificar este tipo de relaciones puede derivar en problemas de interoperabilidad. Una segunda desventaja es la falta de propuestas en lo referente a la especificación de privilegios, es decir, el planteamiento de unas directrices que puedan ser utilizadas para codificar los permisos de los distintos entornos de aplicación. Asociado a esto, falta un mecanismo genérico de cálculo de autorizaciones como el que presentan el resto de propuestas que forman parte de este análisis. Por último, el sistema no acaba de ser tan descentralizado como sería deseable, ya que las autoridades de atributo deben ser reconocidas como tales por otra entidad de nivel superior, lo cual rompe con la idea de descentralización en la que una entidad se constituye en autoridad en el momento en el que un controlador de recursos la considera como tal, sin necesidad de que otras autoridades tengan constancia de ello.

4.3.4 SPKI/SDSI

En 1996, Ronald Rivest y Butler Lampson proponen la primera versión del sistema SDSI (Simple Distributed Security Infrastructure) [170]. Según la percepción de los autores, los sistemas de clave pública existentes en ese momento eran demasiado complejos e incompletos. SDSI se trataba de una nueva propuesta que intentaba combinar la funcionalidad de una infraestructura de clave pública sencilla con mecanismos de definición de grupos, listas de control de acceso y definición de espacios de nombres locales. Sin duda alguna, la propuesta de definición de nombres locales, tanto para la identificación de claves públicas como para la formación de grupos, constituyó la primera ruptura drástica con las propuestas de la comunidad X.509. En contraste con la utilización de nombres X.500 globales, SDSI nació con la filosofía de proporcionar mecanismos para la definición de espacios locales de nombres, muy ligados al entorno organizacional en el cual fueran utilizados, y que pudieran ser fácilmente enlazados entre sí [1, 93, 126].

Al mismo tiempo, desde el grupo de trabajo SPKI (Simple Public Key Infrastructure) del IETF, y especialmente debido al trabajo personal de Carl Ellison, se proponía un sistema similar más enfocado a la definición de condiciones de control de acceso y más sencillo que SDSI desde el punto de vista de la cantidad de estructuras de datos distintas a emplear. Si bien ambas líneas de trabajo empezaron a avanzar en sus progresos de forma paralela, surgiendo multitud de trabajos [65, 81] relacionados con las primeras versiones de estos sistemas, dos años más tarde decidieron unificar sus infraestructuras [69], ya que ambas proponían sistemas y marcos de trabajo similares que podrían verse mejorados por la selección de lo mejor de cada propuesta.

Así pues, en esta sección describiremos el sistema resultante, denominado SPKI/SDSI aunque comúnmente denominado simplemente como SPKI, que es la alternativa más seria hasta el momento para la construcción de infraestructuras de autorizaciones. Como se verá a continuación, SPKI/SDSI aporta su propio mecanismo de definición de nombres locales, separa las distintas clases de certificación en tres categorías independientes y proporciona un método genérico de representación intermedia de autorizaciones y de reducción de las mismas.

Terminología

La propuesta SPKI/SDSI (de ahora en adelante SPKI) está caracterizada por una gran cantidad de notación propia y una forma radicalmente distinta de concebir el diseño de una infraestructura de certificación en relación con las tradicionales. A continuación se definen algunos conceptos de la terminología SPKI:

- *Certificado.* Se trata de un documento, firmado digitalmente mediante criptografía asimétrica, que asigna un privilegio o un identificador a una entidad. Contiene al menos un emisor y una entidad receptora (subject), y puede contener periodos de validez, información de autorización e información de delegación. En realidad hay tres categorías de certificados: ID (relación <nombre, clave>), Atributo (relación

<autorización, nombre>) y Autorización (relación <autorización, clave>). Un certificado de autorización o de atributo puede autorizar la propagación de todo o parte del poder que se recibe del emisor del certificado (delegación).

- *Keyholder.* La persona o entidad que posee y controla una determinada clave privada.

- *Principal.* Clave criptográfica capaz de verificar una firma digital. En general, este concepto hace referencia al componente público del par de claves asociados a una entidad, por lo que en la mayoría de los casos se tratará de un sinónimo de clave pública.

- *Entidad (subject).* Se trata del elemento al que se le asigna cierto identificador o autorización, bien a través de un certificado o mediante una entrada de una lista de control de acceso. Puede tomar la forma de una clave, un nombre, el resumen digital de un objeto o un conjunto de claves de una función umbral *k-of-n*.

- *S-expresión.* Es el formato de datos elegido por SPKI, similar a las expresiones empleadas en LISP pero con la limitación de que no se permiten las listas vacías y que el primer elemento de cualquier S-expresión debe ser una cadena de caracteres, llamada el tipo de la expresión.

Nombres SDSI

Tal y como se comenta en el estándar SPKI, los nombres son un mecanismo definido simplemente por conveniencia humana, ya que las claves criptográficas satisfacen totalmente cualquier necesidad de nombramiento que pudieran tener las entidades software. Como veremos más adelante, el sistema de cálculo de autorizaciones de SPKI acaba reduciendo todos los nombres en claves criptográficas.

En SPKI no hay reglas de nombramiento, puesto que se supone que cada emisor puede definir su propia política de asignación de nombres dentro de su entorno de aplicación. Estos nombres tienen un significado local (nombres similares a los empleados en las agendas personales, o a los seudónimos introducidos en los agentes de correo electrónico). Son nombres que no necesitan ser globalmente únicos, sino que deben ser únicos simplemente en el espacio local donde han sido definidos (aunque ello no quiere decir que no puedan emplearse para definir identificadores globalmente únicos). Su simplicidad y su escalabilidad hicieron que el sistema de nombramiento definido en SDSI fuera adoptado en el sistema SPKI/SDSI.

Un nombre básico SDSI tiene la forma *(name k n)*, que simplemente representa al nombre n definido en el espacio de nombres de la clave criptográfica k. A partir de los nombres básicos pueden emplearse nombres compuestos, como por ejemplo el nombre *(name (name k n) m)* que hace referencia al nombre m definido por la clave nombrada como n por k. Los nombres compuestos (al igual que el uso de nombres de grupos) tienen la ventaja de que al ser indirecciones, cualquier cambio en la definición del nombre se ve inmediatamente difundido entre todas las referencias.

Certificados SPKI de identidad

Los certificados de identidad SPKI pueden ser empleados principalmente para tres propósitos distintos. En primer lugar, pueden emplearse de forma similar a los certificados X.509, es decir, para asociar un identificador a una clave pública. La principal diferencia respecto a X.509 es que dicho identificador será considerado único dentro del espacio de nombres del emisor del certificado, y no globalmente. En segundo lugar, los certificados de identidad pueden emplearse como mecanismo de definición de grupos de principales. La creación de un grupo se consigue mediante la emisión de varios certificados que asocian el mismo nombre a distintos principales. Por último, este tipo de certificados puede emplearse para crear relaciones de inclusión entre grupos, ya que la entidad a la que se le asocia el nombre puede tratarse a su vez de un identificador de grupo.

La estructura [68] de los certificados de identidad (ver figura 4.8) está formada por tres campos principales: *issuer* (emisor), *subject* (receptor), *valid* (validez). El elemento denominado *principal* es el espacio de nombres en el cual se está definiendo el nombre *name*, *subject* es la entidad a la que hará referencia el nombre (que puede ser a su vez otro nombre, un principal, o un resumen digital de un objeto) y *valid* es un elemento opcional que hace referencia al método de validación del certificado (los métodos de validación se verán más adelante).

```
(cert
  (issuer (name <principal> <name>))
  (subject <subject>)
  (valid <valid>)?
)
```

Figura 4.8: Certificado SPKI de identidad

La concatenación de la clave pública de la entidad emisora (o incluso su resumen digital) junto con el nombre que se está definiendo da lugar a los identificadores globalmente únicos. Es más, esta forma de definición de nombres globales nos permite adaptar los nombres de los certificados X.509 de forma bastante inmediata, ya que estos pueden considerarse como *(name <clave CA> <DN de la entidad certificada>)*.

Certificados SPKI de autorización y de atributo

Ambos tipos de certificados poseen estructura similar [68], ya que la principal diferencia se encuentra en el campo *subject*, el cual puede hacer referencia a un principal (certificado de autorización) o a un nombre (certificado de atributo). Los certificados de autorización se emplean para asignar privilegios directamente a claves, mientras que los certificados de atributo son útiles para asignar privilegios a grupos de entidades. En el caso de que una aplicación posea simplemente un certificado de atributo, es necesario obtener uno de identidad para tener la relación completa <autorización,nombre,clave>. Los principales campos de este tipo de certificados son los mostrados por la figura 4.9.

```
(cert
  (issuer <principal>)
  (subject <principal> | <name>)
  (propagate)?
  (tag <tag>)
  (valid <not-before>? <not-after>? <online-test>?)?
)
```

Figura 4.9: Certificados SPKI de autorización y atributo

Algunos de los campos ya han sido comentados. Sin embargo, otros son nuevos en este tipo de certificados y serán explicados en éste y en apartados posteriores.

- *(propagate).* Si está presente, sirve para indicar que la autorización puede delegarse a su vez.

- *(tag).* Se trata del campo donde se establecen de forma concreta los privilegios asignados al *subject*. La estructura interna de este campo no está determinada, y se deja que cada aplicación haga el uso de ella que más le convenga para sus intereses. De todas formas, aunque no se fuerce ningún patrón, sí que se habilitan algunas restricciones y operaciones útiles. Por ejemplo, las s-expresiones que empiezan con el operador * sirven para hacer referencia a especificaciones más complejas: *(* set)* se emplea para enumerar un conjunto de elementos; *(* prefix)* se utiliza para hacer referencia a cadenas de caracteres que empiezan con un determinado prefijo; *(* range)* sirve para hacer referencia a un rango de valores; por último, *(tag *)* es equivalente a *todas las autorizaciones*. Los tags se asumen como posicionales, por tanto, los parámetros de un tag tienen un significado dependiente de su posición. En las secciones 5.3 y 6.4.4 se verán en detalle ejemplos de utilización de las s-expresiones para codificar tags de autorización.

Validación en SPKI

Las condiciones de validez de los certificados SPKI pueden expresarse de forma directa en cada uno de los mismos, aunque también es posible omitirlas, lo cual le confiere al certificado una validez indefinida. Entre los distintos mecanismos disponibles encontramos tanto los tradicionales basados en comprobación de fechas (mecanismos *off-line*), como aquellos basados en consultas instantáneas (*on-line*).

El mecanismo tradicional basado en fechas hace uso de dos límites (opcionales cada uno de ellos) para definir tanto la fecha máxima como mínima de validez de la sentencia.

Los test de validación en línea permiten obtener un nivel de información más ajustado acerca de la validez de cierto certificado. Hay un total de 4 formas de test en línea:

- *(crl).* Obtención de una lista de certificados revocados.

- *(reval)*. Sirve para obtener las fechas de validez de un certificado no revocado.
- *(one-time)*. Es una prueba de validez que no emplea fechas (similar a OCSP).
- *(new-cert)*. Sirve para obtener la copia más reciente de un certificado. Se emplea cuando se hace uso de certificados con ciclo de vida muy corto.

Para todos estos mecanismos en línea es necesario disponer de un punto de consulta, es decir, de la localización del elemento que realiza esta función (el cual no tiene por qué ser la misma entidad que emite los certificados a verificar). Dicha localización forma parte también del campo *(valid)* del certificado.

Listas de control de acceso (ACL) y secuencias

Las ACLs son listas de sentencias, partes de certificados que no necesitan campos de emisor o firmas (puesto que se suponen que están controladas localmente por el poseedor del recurso al cual se le está controlando el acceso). Si todos los campos opcionales se dejan en blanco, la entidad obtiene indefinidamente los permisos especificados en el campo *tag*, pero sin capacidad para delegarlos. Su sintaxis es la mostrada en la figura 4.10.

```
(acl
  (entry
    (subject <principal> | <name>)
    (propagate)?
    (tag <tag>)
    (valid <not-before>? <not-after>? <online-test>?)?
  )
  ...
)
```

Figura 4.10: Lista de control de acceso SPKI

Por otro lado, las secuencias son listas ordenadas de objetos que se suelen suministrar al verificador para que éste considere si concede o no el acceso a un recurso. Suelen incluir las firmas de los certificados (una o varias firmas, dependiendo de cuántos principales realicen la solicitud) y otra información útil. En [70] se muestran varios ejemplos de secuencias y certificados SPKI.

Cálculo de autorizaciones

Por cálculo de autorizaciones se hace referencia al método mediante el cual se determina si una solicitud satisface una política concreta. Hay que tener en cuenta que dicha determinación no es evidente, y no se limita a constatar simplemente que el principal que presenta la solicitud de acceso al recurso está reflejado directamente en la ACL. El método debe ser capaz de resolver los casos en los que la clave del solicitante no aparezca listada

explícitamente en la ACL, como cuando el acceso está basado en la pertenencia a un grupo determinado o en cadenas de delegación (o incluso en ambas cosas a la vez).

Para resolver estas situaciones hay que considerar previamente cuáles son las posibles entradas al proceso de cálculo de autorizaciones. Encontramos que este proceso recibe tanto certificados de nombres (ID), como certificados de atributos, certificados de autorización, listas de control de acceso y las claves públicas de los solicitantes. En primer lugar, se procede a la validación de los certificados presentados (verificación de las firmas digitales y chequeo del estado de los certificados). A continuación, se procede a la conversión de los certificados y las listas de control de acceso en tuplas. Posteriormente, las tuplas que representan nombres se reducen hasta obtener sólo las claves asociadas. Por último, las tuplas ligadas a los certificados de atributo y de autorización se reducen para calcular el resultado final de autorización.

Los certificados de autorización y de atributo dan lugar a las denominadas 5-tuplas, las cuales son un formato de representación adecuado para realizar los cálculos de autorización. De hecho, su representación es lo suficientemente genérica como para permitir que otro tipo de certificados (KeyNote, X.509 AC, etc.) puedan ser transformados en 5-tuplas. Los elementos que las componen son:

- *Emisor*: clave pública, resumen digital o la palabra *self* (en el caso de las listas de control de acceso). Es quien firma la autorización.
- *Subject* (entidad): clave (o resumen digital) del receptor.
- *Delegación*: valor booleano que se utiliza para especificar si la autorización se puede propagar.
- *Autorización*: una S-expresión.
- *Fechas de validez*: inicio y fin de validez (pueden estar deducidas a partir de test en línea).

Estas tuplas suelen representarse mediante la notación <*I,S,D,A,V*>, donde cada uno de los componentes hace referencia a los elementos que acaban de ser descritos. Se dice que dos 5-tuplas se reducen si se cumple lo especificado en la figura 4.11.

$$< I1, S1, D1, A1, V1 > + < I2, S2, D2, A2, V2 > \ \ se\ reducen\ en$$
$$< I1, S2, D2, AIntersect(A1, A2), VIntersect(V1, V2) > \ \ si$$
$$AIntersect(A1, A2) \neq \emptyset \ \ \wedge \ \ VIntersect(V1, V2) \neq \emptyset \ \wedge$$
$$S1 = I2 \ \ \wedge \ \ D1 = verdadero$$

Figura 4.11: Reducción de autorizaciones SPKI

Las funciones $AIntersect$ y $VIntersect$, definidas en [69], son operaciones de intersección de conjuntos encargadas de hallar las autorizaciones comunes a ambos certificados (AIntersect) y el periodo de validez resultante (VIntersect).

Por otro lado, los certificados SPKI de identidad se convierten en 4-tuplas para ser reducidos finalmente a una clave criptográfica concreta. Contienen la siguiente información:

- *Emisor*: clave pública o resumen digital.
- *Nombre*: una cadena de caracteres.
- *Subject*: clave pública, resumen digital o nombre.
- *Fechas de validez* inicio y fin de validez (pueden estar deducidas a partir de test en línea).

Cálculo de la cadena de certificación

El proceso de descubrimiento de cadenas de certificación, es decir, del cálculo de conformidad, es un proceso complejo. Los certificados de nombres pueden componerse para derivar nuevos nombres, y los certificados de autorización pueden combinarse a su vez para derivar nuevas autorizaciones, y ambos pueden emplearse para deducir nuevas autorizaciones a nombres. El procedimiento seguido, ampliamente expuesto en [66], puede resumirse como la búsqueda en un grafo dirigido de un camino de certificación que tenga como nodo inicial la política de seguridad del sistema, y como nodo final la clave pública asociada al usuario que está realizando la solicitud. La construcción de dicho grafo está basada en el mecanismo de reducción de 5-tuplas visto en el apartado anterior.

Escenarios de uso

Desde que en 1996 se propusieran las primeras versiones tanto de SDSI como de SPKI, se han desarrollado varios trabajos relacionados con estas propuestas, y que en general tienen en común el intento de puesta en marcha de dichos sistemas en entornos convencionales. Pero es sobre todo con posterioridad a la integración de ambos sistemas cuando aflora el número de propuestas que hace uso de los mismos.

En 1998, Elien [66] realiza un estudio acerca del cálculo de autorizaciones y propone un algoritmo para el cálculo de cadenas de certificación, o lo que es lo mismo, hallar un método para calcular valores de conformidad a partir de un conjunto de ACLs y certificados SPKI.

Maywah [138] proporciona un mecanismo para limitar el acceso a recursos Web mediante el uso de certificados SPKI. Básicamente, se trata de una extensión del browser Netscape Communicator que permite realizar el intercambio de listas de control de acceso y certificados siguiendo un protocolo concreto.

Mención especial merecen los trabajos realizados en la HUT (Helsinki University of Technology) en relación con la propuesta SPKI. Entre ellos podemos citar las aportaciones de Nikander en materia de arquitecturas de autorización para sistemas orientados a objetos distribuidos [156, 163] o control de acceso WLAN [116], la propuesta de Lampinen sobre el uso de certificados SPKI para propósitos de autorización en CORBA [123], y muy especialmente los trabajos en materia de delegación realizados por Aura [15, 16, 17].

Conclusiones

Sin duda alguna, SPKI/SDSI es la propuesta más ampliamente analizada y utilizada en lo que a gestión de autorizaciones se refiere. Prueba de ello son las numerosas aportaciones realizadas por parte de la comunidad científica a lo largo de estos últimos años.

La clave de su versatilidad se encuentra en la distinción clara que se realiza entre los conceptos de clave, nombre y autorización. Como consecuencia, los tres tipos de certificados resultantes son capaces de aportar todos los mecanismos necesarios para modelar los sistemas clásicos de control de acceso, en especial DAC y RBAC. A diferencia de lo que sucedía con los certificados de atributo X.509, SPKI sí aporta mecanismos para reflejar los permisos asignados a los roles (mediante los certificados de atributo SPKI) e incluso para modelar las jerarquías de roles (mediante los certificados de identidad). Además, el uso de representaciones intermedias basadas en 5-tuplas y 4-tuplas posibilita, por un lado, la capacidad de convertir documentos expresados siguiendo otros sistemas de certificación en alguna de estas representaciones intermedias y, por otro lado, un mecanismo genérico de reducción de certificados independiente del entorno de aplicación. Dicha independencia se consigue mediante la provisión de unas directrices a la hora de especificar los permisos asignados mediante los certificados y ACLs.

El mecanismo de definición de tags basado en s-expresiones es lo suficientemente claro y estructurado como para poder expresar la mayorías de las condiciones de autorización derivadas de cualquier sistema. La traducción de solicitudes o políticas de seguridad en s-expresiones no es una tarea tan compleja como la especificación de dichos elementos mediante lenguajes como los utilizados por PolicyMaker o Keynote. Se podría incluso decir que la notación basada en s-expresiones es lo suficientemente clara como para poder ser interpretada sin problemas por un usuario medio.

Sin embargo, esta propuesta también presenta algunas carencias en lo que al formato de sus certificados se refiere. Quizá la más importante de ellas es la incapacidad para restringir a qué usuarios se puede propagar un privilegio, ya que el control de la delegación está basado simplemente en un valor booleano, sin que sea posible especificar ningún tipo de restricción adicional. Otras limitaciones serán analizadas en la sección 4.4.6.

4.3.5 Otros esquemas basados en XML

En los últimos años hemos visto aparecer gran cantidad de propuestas en materia de seguridad que se caracterizan por emplear XML (Extensible Markup Language) [31] como lenguaje de especificación. En materia de autorización, también se ha realizado un esfuerzo importante a la hora de definir esquemas que permitieran la codificación y el intercambio de este tipo de información.

AuthXML [158] permitía a distintas organizaciones intercambiar información relativa a autenticación, autorización, perfiles de usuario y sesiones. Este sistema se diseñó para simplificar las transacciones entre colaboradores que hicieran uso de aplicaciones de seguridad no interoperables, con el fin de crear un sistema de codificación común.

Tanto AuthXML como S2ML (otra propuesta similar propugnada por Sun Microsys-

tems y Verisign, entre otros) [158], se fundieron en un único estándar de muy reciente creación denominado SAML (Security Assertion Markup Language) [158], el cual incluye además nuevas características. Este estándar está impulsado por OASIS (Organization for the Advancement of Structured Information Standards), el cual es un consorcio internacional sin ánimo de lucro encargado de crear especificaciones basadas en XML.

4.3.6 Conclusiones

En relación con lo analizado en la sección 4.1, podemos comprobar que las especificaciones analizadas dentro de este apartado aportan mecanismos suficientes como para subsanar las carencias propias de los sistemas de certificación de identidad.

En lo que respecta al control de acceso, todas las especificaciones permiten la creación de certificados de credencial independientes que contienen los privilegios recibidos por las entidades del sistema. Del mismo modo, la mayor parte de las propuestas aportan elementos de definición de políticas de autorización.

Desde el punto de vista del anonimato, ninguna de las especificaciones requiere el uso de identificadores a la hora de asignar permisos a las entidades, ya que esta asignación puede realizarse empleando únicamente claves públicas, e incluso el resumen digital de las mismas.

Por último, todas ellas proporcionan mecanismos de propagación o delegación de permisos, así como medios para controlar la expansión de los mismos. De hecho, la delegación es uno de los conceptos fundamentales sobre el cual se apoyan todas estas propuestas a la hora de ser empleadas para la construcción de sistemas de autorización distribuidos. En consecuencia, el siguiente apartado tratará más en profundidad cuáles son las ventajas y limitaciones existentes en materia de delegación basada en certificados de credencial.

4.4 Análisis de las oportunidades y retos del control de acceso basado en delegación

Gran parte de las propuestas y los desarrollos que se enmarcan dentro de esta tesis hacen uso de la delegación como mecanismo básico de gestión de autorizaciones. En consecuencia, se considera importante realizar un estudio más en profundidad acerca de todos los aspectos relacionados con este enfoque.

Tal y como se vio en la sección 4.2.4, la idea principal que subyace en el modelo de control de acceso distribuido es que los controladores de recursos delegan en autoridades específicas la gestión de los accesos. De esta forma, dichas autoridades pueden emitir certificados que propaguen dichos permisos a otras autoridades subordinadas o a usuarios finales, los cuales transfieren un subconjunto de dichos certificados junto con sus solicitudes de acceso para probar que están autorizados a acceder a los recursos. El proceso finaliza de nuevo en el controlador, puesto que es el encargado de validar los certificados y contrastar si las evidencias presentadas cumplen la política de seguridad del sistema.

En esta sección, se presenta un análisis original a partir del cual se extraen las diferentes oportunidades y retos que implica el mecanismo de control de acceso basado en delegación, especialmente desde un punto de vista de la gestión de autorización. El análisis está estructurado atendiendo a los tópicos de gestión, cadenas de delegación, diferencias entre autoridad y posesión de permisos, anonimato, distribución de certificados y revocación. No se trata de describir una especificación concreta de certificados (a los cuales llamaremos bajo el nombre genérico de certificados de credencial), sino de abordar cada uno de los tópicos desde un punto de vista más abstracto con el fin de no limitarnos a lo especificado en alguna de las propuestas analizadas en el apartado 4.3 (la sección 4.4.6 contrastará lo aquí expuesto con dichas propuestas).

4.4.1 Estructuras de gestión

Gestión de permisos

Los certificados de credencial proporcionan un mecanismo para establecer estructuras organizacionales que pueden ser cambiadas de forma dinámica. La estructura de certificados refleja la composición de una organización concreta, y en contraste con las listas de control de acceso, el control de los permisos contenidos en los certificados está ampliamente distribuido [15]. Los cambios sobre la política de autorización no tienen que ser propagados a todas las ACLs que controlan el acceso a los recursos, y la gestión de los certificados es una tarea relativamente sencilla al estar distribuida entre varias entidades que controlan un subconjunto pequeño de los permisos. A continuación, se muestra un ejemplo acerca de cómo la delegación puede simplificar las listas de control de acceso y, por tanto, la lógica de los controladores de recursos. En §4.1 se muestran dos ACLs no basadas en delegación para dos controladores distintos. La ACL del *controlador1* concede dos permisos P^1 y P^2 a las claves públicas K_A y K_B. La ACL del *controlador2* asigna otros permisos a K_D y K_E.

$$ACL(controlador1) = (K_A, P^1), (K_B, P^2) \quad ACL(controlador2) = (K_D, P^3), (K_E, P^2) \quad (4.1)$$

Supongamos ahora que una nueva clave pública K_C debe ser autorizada por ambos controladores a realizar la operación P^2. Siguiendo este enfoque, las dos ACLs deberían ser modificadas para incluir (K_C, P^2). Aunque en un principio esto podría considerarse una tarea sencilla, no sucedería lo mismo si dicha modificación tuviera que aplicarse a varias listas de control de acceso distribuidas a lo largo de todo el sistema. Algunos aspectos como la consistencia, el ancho de banda consumido y la disponibilidad son críticos en las soluciones basadas en ACLs. Supóngase que se redefine la política de control de acceso mostrada en §4.1 haciendo uso de la delegación. La forma de expresar esta delegación será mediante el empleo de etiquetas contenidas en cada entrada de la ACL, las cuales especificarán la entidad autorizada a emitir certificados de credencial para determinados permisos.

En §4.2 se expresan los mismos criterios que en §4.1, pero haciendo uso de las etiquetas de delegación y de tres autoridades de autorización K_{auth1}, K_{auth2} y K_{auth3}.

$$\begin{aligned} ACL(controlador1) &= (K_{auth1}, P^1, propagar), (K_{auth2}, P^2, propagar) \\ ACL(controlador2) &= (K_{auth3}, P^3, propagar), (K_{auth2}, P^2, propagar) \end{aligned} \tag{4.2}$$

Finalmente, para poder autorizar a las entidades finales a acceder a los recursos, las autoridades deben emitir certificados de credencial asignando parte de los permisos obtenidos a través de las ACLs. En §4.3 se muestran los certificados necesarios para emular la política de autorización de §4.1 (las fechas de validez de los certificados se han omitido por simplicidad).

$$\begin{aligned} &autoriza(K_{auth1}, K_A, P^1) \quad autoriza(K_{auth2}, K_B, P^2) \\ &autoriza(K_{auth2}, K_C, P^2) \quad autoriza(K_{auth2}, K_E, P^2) \\ &autoriza(K_{auth3}, K_D, P^3) \end{aligned} \tag{4.3}$$

De este modo, la asignación a K_C del permiso para realizar P^2 sólo implica la generación de un nuevo certificado de credencial (K_{auth2}, K_C, P^2), sin que exista la necesidad de modificar alguna de las ACLs existentes. Este esquema puede incluso extenderse para crear jerarquías de gestión que reflejen la estructura organizacional. Por ejemplo, K_{auth2} podría también delegar un conjunto de permisos $P^{2\prime}$ a K_B mediante el certificado $(K_{auth2}, K_B, P^{2\prime}, propagate)$, lo cual puede tener sentido si pensamos en K_{auth2} como la clave pública asociada a un jefe de departamento y en K_B como la correspondiente al jefe de sección dentro del departamento.

Cadenas de delegación

Como se acaba de mencionar, los permisos pueden ser redelegados en otras claves, las cuales pueden a su vez redelegarlos y así indefinidamente. Por tanto, tal y como se vio en la sección 4.2.4, los certificados de delegación constituyen cadenas donde los permisos fluyen desde las autoridades hacia los usuarios finales (de hecho, como se comenta en [15], la delegación no genera cadenas sino grafos).

Sin embargo, la gestión de estas cadenas puede ser una tarea compleja. Las decisiones de autorización que deben ser tomadas en base a cadenas largas no son sencillas ya que la distribución y recuperación de varios certificados puede ser una tarea computacionalmente costosa. Además, desde el punto de vista de un atacante, dichas cadenas pueden revelar información muy valiosa acerca de la estructura de autorización del sistema (datos acerca de las autoridades, permisos concedidos, posibilidad de propagación, etc.). En consecuencia, en algunos entornos la información contenida en los certificados se considera confidencial, lo que implica la adopción de medidas destinadas a evitar que sea desvelada.

La reducción de certificados, ya comentada en la sección 4.3.4, es una de las técnicas que puede proporcionar mecanismos para eliminar el exceso de información contenido en las cadenas de certificación. Al analizar la cadena de certificados expresada en §4.4, es posible inferir la reducción presentada en §4.5.

$$autoriza(K_{auth1}, K_i, P^1, propagar) \quad autoriza(K_i, K_j, P^2) \tag{4.4}$$

$$autoriza(K_{auth1}, K_j, (P^1 \cap P^2)) \tag{4.5}$$

El certificado resultante no asigna ningún permiso nuevo, puesto que se trata simplemente de una versión simplificada de la cadena original. Sin embargo, además de ocultar algunos detalles presentes en la cadena, dicho certificado puede ser procesado de forma más rápida que la cadena completa. Su periodo de validez será el resultante de la intersección de los periodos de validez de los certificados de §4.4 y los permisos otorgados serán también el resultado de intersectar los permisos propagados por la cadena original.

Es importante dejar constancia de que en ocasiones no es posible realizar la reducción de una cadena sin perder algunas de las características contenidas en ella. Por ejemplo, cuando la validación de los certificados intermedios de la cadena debe realizarse utilizando algún sistema de chequeo en línea (como OCSP), ya que el certificado final no refleja la necesidad de dicha validación.

Control de la delegación

Hasta ahora, los ejemplos que se han mostrado realizan un control de la delegación mediante el uso de una etiqueta booleana que permite propagar o no el permiso. En contraste, varias son las alternativas que pueden emplearse a la hora de controlar dicha propagación. En [69], los autores de SPKI defienden el uso de este enfoque basado en un valor booleano frente a otras propuestas centradas en la limitación de la profundidad de delegación a un número determinado de niveles. Su justificación es que resulta imposible, en la mayoría de los casos, poder predecir de antemano la profundidad apropiada y que, en el caso de que esto fuera posible, no serviría de nada de cara a controlar la proliferación de permisos a lo ancho del árbol organizacional. No obstante, SPKI ofrece otra forma más elaborada de controlar la propagación haciendo uso de los certificados umbral (aquellos que exigen la participación de un conjunto de k entidades sobre una población de n). Supongamos que K_A quiere propagar ciertos privilegios P a K_B, y que a su vez quiere asegurarse de que estos no se propagan a otras entidades no contempladas en su política. Mediante el control booleano no puede asegurar dicha situación, pero podría realizar la propagación a K_B mediante un certificado en el que el campo del receptor tuviera la forma $\{(2-of-2)(K_B)(K_A)\}$, lo cual impediría a K_B propagar el privilegio P sin su consentimiento. Además de crear un problema de centralización en K_A, la cuestión queda sin resolver en el caso de que K_A decida propagar el mismo privilegio a K_C haciendo uso de la misma construcción. La razón es que tanto K_B como K_C pueden confabularse para propagar el privilegio a K_D, ya que ambos tienen la mitad de la autoridad necesaria para ello.

Por esta y otras razones, varios son los autores que consideran insuficiente el enfoque booleano. En [19] se presenta un mecanismo de control de la delegación que permite especificar de forma más concreta las entidades que en un futuro serán capaces de recibir los permisos que se están propagando. La limitación está basada en el uso de expresiones

regulares que establecen el subárbol de la organización que está autorizado a formar parte del camino de delegación. Esta propuesta es sin duda un paso importante hacia un mejor control de la delegación, aunque puede ser poco eficiente en aquellos escenarios donde la estructura del árbol sea demasiado dinámica. El principal problema es que la autorización de una nueva rama del árbol puede llegar a implicar la modificación de todos los certificados que forman parte del camino desde la raíz hasta la nueva rama, con el fin de poder reflejar el cambio en los criterios de propagación de los permisos.

4.4.2 Autoridad y posesión de permisos

Una de las cuestiones que más controversia ha producido entre la comunidad científica es: *¿puede una entidad ejercer los permisos que ella misma asigna a otras entidades?*

No hay un acuerdo general al respecto y algunos autores piensan que la autoridad siempre es capaz de emitir un certificado para una clave pública temporal generada por ella misma, asignándose de esta forma los permisos que de otra forma no podría ejercer. Desde un punto de vista general, parece apropiado que a un administrador se le pueda limitar en ciertos entornos el disfrute de los privilegios que gestiona, y que por tanto sería necesario habilitar los mecanismos necesarios para tal efecto.

Por ejemplo, algunos autores distinguen claramente entre gestionar un permiso y ser capaz de ejercerlo [176]. En general, el término *autoridad* hace referencia a la posibilidad de crear y delegar permisos, mientras que el término *privilegio* suele emplearse para referirse tanto a autoridad como a permiso. Sin embargo, la especificación de políticas de seguridad que permitan separar claramente los conceptos de autoridad y permiso es una línea de investigación en la cual debe realizarse todavía un esfuerzo importante.

Otro aspecto interesante relacionado con la posesión de privilegios es el concepto de *transferencia*. Es importante recalcar que el hecho de emitir un nuevo certificado no invalida ninguno de los existentes previamente, es decir, el emisor no pierde ninguno de los privilegios que posee. La *transferencia* es mucho más difícil de implementar que la delegación, puesto que implica la revocación de los privilegios tras la asignación de los mismos, es decir, es una operación que debe realizarse de forma atómica. Además, dado que los certificados de credencial sólo dan soporte a políticas de seguridad donde los privilegios crecen de forma monotónica, es imposible verificar que una entidad no tiene ciertos privilegios (no hay sentencias negativas).

4.4.3 Anonimato

La sección 4.4.1 introdujo los problemas derivados de la revelación de información sensible contenida en los certificados que forman parte de una cadena. No en vano, dicha estructura muestra las relaciones existentes entre claves, y es relativamente sencillo asociar dichas claves a usuarios reales cuando se utilizan certificados de identidad.

En [17] se presentan dos técnicas destinadas a evitar el rastreo de las claves: el uso de claves temporales y la reducción de certificados.

Claves temporales

El rastreo de claves puede dificultarse mediante el uso de claves temporales en las cuales redelegar parte de los permisos pertenecientes al usuario. Por ejemplo, un usuario podría crear claves temporales distintas para cada una de las tareas que realiza y utilizar éstas cada vez que realiza solicitudes de servicio con el fin de ocultar su clave pública original, la cual muy probablemente esté asociada a algún tipo de identificador mediante un certificado de identidad. En §4.6 se muestra una cadena de certificación en la cual el usuario K_U delega un conjunto de permisos $P^{1\prime}$ a una clave temporal K_T que ha sido generada por él mismo.

$$\begin{gathered} autoriza(K_{auth1}, K_U, P^1, propagar) \quad autoriza(K_U, K_T, P^{1\prime}) \\ donde \;\; P^{1\prime} \subseteq P^1 \end{gathered} \tag{4.6}$$

Es importante recalcar que la cadena de certificación es sólo válida si K_U tiene asignado el privilegio de poder propagar parte de los permisos que ha recibido. En algunos entornos de aplicación, como los sistemas de comercio electrónico, la redelegación no suele estar permitida dado que la adquisición de ciertos permisos puede implicar algún tipo de coste económico.

Sin embargo, el uso de claves temporales puede resultar complejo en aquellos casos en los que se utilicen mecanismos de control de la delegación. Tal y como se ha visto, este tipo de restricciones suelen estar basadas en la especificación de subárboles de entidades autorizadas a recibir los privilegios, subárboles que deben ser conocidos con anterioridad a la generación de los certificados. En contraste, las claves temporales se generan de forma dinámica, y su valor no puede ser conocido previamente, lo cual dificulta redelegar en ellas parte de los privilegios.

Se propone aquí una solución (ver §4.7) a este problema. Como puede apreciarse, la delegación está autorizada para los miembros del grupo G, el cual está dentro del espacio de nombres de la entidad K_M. Si en algún momento es necesario redelegar en una clave temporal, la solución pasa por conseguir que K_M considere a dicha clave como miembro de G.

$$\begin{gathered} autoriza(K_{auth1}, K_U, P^1, propagar(K_M\$G)) \quad autoriza(K_U, K_T, P^{1\prime}) \\ donde \;\; K_T \in K_M\$G \end{gathered} \tag{4.7}$$

Esta solución posibilita el uso de claves temporales sin que sea necesario modificar las condiciones de control de la delegación ni los certificados implicados dentro de la cadena, aunque requiere que dichas claves sean registradas por una entidad como parte de cierto grupo. No obstante, al tratarse de un enfoque basado en claves y no en nombres, es necesario asegurar que las claves pertenecen realmente a usuarios autorizados a recibir los permisos, y no a otros usuarios. Como se comenta en [17], el control de delegación y el uso de claves temporales depende inevitablemente de un mecanismo de identificadores únicos asociados a todas las claves propiedad de una entidad. Paradójicamente, obtenemos así que el uso de claves temporales para evitar el rastreo de la actividad de los usuarios pasa por la necesidad de autenticar de forma robusta dichas claves.

Reducción y reductores confiables

El apartado anterior introdujo el uso de claves temporales como mecanismo para ocultar la actividad de las claves privadas de los usuarios. Sin embargo, el simple uso de dicho tipo de claves no oculta la clave original del usuario en una cadena de certificación, ya que es necesario presentar toda la cadena para obtener la autorización (por ejemplo, K_U está incluida en la cadena tanto en §4.6 como en §4.7). Sin embargo, tal y como se vio en §4.5, un certificado reducido contiene sólo la primera de las claves de la cadena (la que verifica el certificado) y la última. Esto exige que la autoridad raíz de la cadena sea la encargada de emitir el certificado reducido para que éste pueda ser considerado como válido.

En contraposición, se presenta aquí un esquema que no requiere la intervención de dicha raíz durante el proceso de reducción. Este enfoque está basado en el concepto de *reductores confiables* como entidades específicas que han sido autorizadas a realizar reducciones en nombre de la autoridad raíz. Los reductores pueden ser habilitados para gestionar sólo un pequeño conjunto de los permisos que emanan de la autoridad raíz, aquellos que permiten ser reducidos. De esta forma, se libera a las autoridades raíz de la obligación de tener que reducir, posiblemente de forma relativamente continua, cadenas largas de certificados.

Los reductores confiables pueden ser habilitados como autoridades válidas utilizando dos técnicas distintas. En §4.8 se muestra la técnica basada en listas de control de acceso y en §4.9 se presenta la alternativa basada en autorizaciones.

$$ACL(controlador1) = (K_{root}, P^1, propagar), (K_{reducer}, P^{1\prime}, propagar) \tag{4.8}$$

$$ACL(controlador1) = (K_{root}, P^1, propagar) \tag{4.9}$$

$$autoriza(K_{root}, K_{reducer}, P^{1\prime}, propagar)$$

$$donde \;\; P^{1\prime} \subseteq P^1$$

La técnica basada en listas de control de acceso requiere la inclusión de las claves públicas de los reductores en dichas listas. Si el número de controladores y el número de reductores es elevado, o si estos conjuntos cambian de forma muy dinámica, esta alternativa puede ser desaconsejable. Por otro lado, la técnica presentada en §4.9 hace uso de certificados de autorización para dar de alta los nuevos reductores. El certificado emitido por K_{auth} a $K_{reducer}$ concede al reductor el derecho a generar reducciones relacionadas con los permisos contenidos en $P^{1\prime}$. En contraste con la alternativa basada en listas de control de acceso, el certificado reducido generado por el reductor no es suficiente para obtener el acceso a los recursos al no constituir una autorización directa realizada por alguna de las entidades contenidas en la ACL. Esto hace que sea necesario transferir de alguna forma al controlador el certificado que autoriza al reductor a comportarse como tal.

4.4.4 Distribución y recuperación de certificados

Una vez que los certificados son generados, parte de ellos se difundirán de forma pública al resto de componentes del sistema y otro subconjunto será protegido por contener

información considerada como confidencial. Por tanto, obtener los certificados necesarios para chequear si una solicitud debe ser autorizada no es una tarea sencilla. En primer lugar, dado que los certificados pueden estar ampliamente distribuidos entre varios emisores, repositorios y usuarios, es necesario descubrir la localización exacta de estas entidades (a las cuales agruparemos con el término genérico de suministradores). A continuación, dado que algunos certificados y políticas de seguridad contendrán información confidencial, será necesario proporcionar mecanismos de control de acceso a dichos elementos de información [179]. Como veremos, existen varias alternativas a la hora de consultar a los suministradores, las cuales se encuentran agrupadas en: dirigidas por el usuario, dirigidas por el controlador y distribuidas entre los suministradores.

Las distintas propuestas formuladas para solucionar el problema de la recuperación de certificados [14, 89, 165] intentan hacer frente a los problemas que se exponen a continuación.

El problema de la *pertenencia oculta*

En §4.10 se presenta lo que denominaremos como el problema de la *pertenencia oculta*, es decir, la determinación de si una clave pública concreta es miembro de un determinado grupo o rol. La ACL del *controlador1* especifica que sólo los miembros del grupo *personal* definido por K_{root} pueden realizar la operación P, la cual está siendo solicitada por K_U.

$$\begin{aligned} ACL(controlador1) &= (K_{root}\$"personal", P) \\ K_{root}\$"personal" &= \{K_{nivel1}\$"secA", K_{nivel1}\$"secB"\} \\ K_{nivel1}\$"secA" &= \{K_{nivel2}\$"depart1", K_{nivel2}\$"depart2"\} \\ K_{nivel2}\$"depart2" &= \{K_T, K_U, K_V\} \end{aligned} \tag{4.10}$$

Siguiendo el ejemplo, podemos comprobar que K_U es efectivamente un miembro del grupo *personal*, ya que es miembro del grupo *depart2*, que a su vez es un subconjunto del grupo *secA*, el cual está incluido en la definición de *personal*. Sin embargo, determinar dicha pertenencia puede implicar un análisis exhaustivo de todo el árbol que representa las relaciones entre los grupos existentes. Es más, el problema se complica si consideramos que las relaciones entre grupos no tienen porque formar un árbol, sino que podrían estar representadas por un grafo con ciclos.

El problema del *permiso oculto*

El problema del *permiso oculto* es muy similar al comentado en el apartado anterior. En §4.11 se muestra una ACL donde el *controlador1* delega la autoridad sobre el conjunto de permisos P a la entidad K_{root}. En este ejemplo, K_U solicita la operación P^4, la cual forma parte del conjunto de permisos P. La estructura de grupos es la misma que la mostrada en el ejemplo anterior.

$$ACL(controlador1) = (K_{root}, P, propagar)$$

$$\begin{gathered} autoriza(K_{root}, K_{nivel1}, P^1, propagar) \ \ donde \ \ P^1 \subseteq P \\ autoriza(K_{root}, K_{nivel1}\$"secA", P^2) \ \ donde \ \ P^2 \subseteq P \\ autoriza(K_{nivel1}, K_{nivel2}\$"depart1", P^3) \ \ donde \ \ P^4 \subseteq P^3 \subseteq P^1 \\ autoriza(K_{nivel1}, K_{nivel2}\$"depart2", P^4) \ \ donde \ \ P^4 \subseteq P^1 \end{gathered} \tag{4.11}$$

Siguiendo el ejemplo, es posible comprobar que la solicitud formulada por K_U debería ser autorizada por tratarse dicha clave de un miembro del grupo *depart2* y haber sido tal grupo autorizado a ejercer el permiso P^4. Como se comentó anteriormente, descubrir este camino de autorización puede implicar el recorrido de varias cadenas de delegación. De hecho, en el ejemplo podría hallarse un camino alternativo siempre que P^4 estuviera contenido en P^2. En conclusión, un buen método de descubrimiento de cadenas de certificación debe gestionar tanto la pertenencia a grupos como el cálculo de privilegios.

Propuestas para el descubrimiento de certificados

El descubrimiento de las cadenas de delegación puede realizarse empleando enfoques muy distintos: mediante la obtención de certificados a partir del solicitante, mediante la recuperación por parte del controlador o mediante la cooperación de distintos repositorios.

Tradicionalmente, el solicitante era el responsable de obtener los certificados necesarios a partir de repositorios públicos o tarjetas inteligentes. Sin embargo, resulta sorprendente la falta de mecanismos o protocolos de intercambio estándar capaces de transmitir certificados de credencial. Los protocolos de seguridad más comunes, como TLS (Transport Level Security) [59], IKE (Internet Key Exchange) [94], o S/MIME (Secure/Multipurpose Internet Mail Extensions) [168], están preparados para transmitir única y exclusivamente certificados de identidad. De hecho, las distintas propuestas que han ido apareciendo para incorporar a dichos protocolos la capacidad de intercambiar certificados de credencial son muy incompletas [73, 133]. La creación de un marco para el intercambio de información relativa a autorización es uno de los campos de trabajo a los que más esfuerzo se le ha dedicado en esta tesis, tal y como se verá en la sección 5.2.

Hoy en día, hay varias propuestas, como DPD (Delegated Path Discovery) [165], destinadas a ofrecer a los usuarios un servidor mediante el cual obtener dichos certificados en su nombre. En este último contexto, es el servidor el encargado de adquirir los datos que de otra forma tendría que recuperar el cliente utilizando distintos protocolos de acceso a repositorios.

El otro enfoque empleado para realizar el descubrimiento está basado en la cooperación distribuida de varios suministradores, una propuesta que emplea por ejemplo la arquitectura AAA (Authentication, Authorization and Accounting) [186]. En relación con §4.11, podríamos considerar que los certificados emitidos por K_{root} están almacenados en un suministrador distinto al de los certificados emitidos por K_{nivel1} o K_{nivel2}. De esta forma, una solicitud de descubrimiento de certificados enviada por el *controlador1* al suministrador de K_{root} podría ser parcialmente reenviada a otros suministradores con el fin de obtener los elementos de la cadena.

Este tipo de descubrimiento distribuido constituye una de las líneas de investigación a la que más esfuerzos debe prestarse con el fin de obtener métodos eficientes de búsqueda. Por un lado, uno de los principales retos lo constituye el control de la redundancia de consultas. Dado que el grafo de delegación puede contener reiteradas referencias a certificados de privilegios o de grupos almacenados por un determinado suministrador, es importante controlar que dicho elemento no sea consultado más veces de las estrictamente necesarias. Por otro lado, la eficiencia en las búsquedas debe ser compaginada con el control de la revelación de información confidencial y la gestión de la información relativa a revocaciones.

4.4.5 Revocación

Los certificados de credencial pueden ser revocados en el supuesto de que el privilegio especificado por el certificado haya dejado de ser válido. Normalmente encontramos dos tipos de situaciones en las cuales es necesario revocar un certificado. Una de ellas es cuando se produce un relevo de la persona hasta entonces encargada de gestionar un conjunto de permisos. En dicho caso, la medida más natural es revocar el certificado del antiguo administrador de forma que se imposibilite la asignación futura de privilegios por parte del mismo, pero respetando al mismo tiempo las asignaciones realizadas hasta el momento. El otro caso se da cuando se tiene conciencia de que un usuario ha estado asignando privilegios de una forma arbitraria, no conforme con la política de autorización de la organización. En dicho caso, lo aconsejable es revocar el certificado con efecto retroactivo, es decir, invalidando todos los certificados y sentencias emitidas en cualquier instante por el usuario.

La revocación suele tratarse siempre considerando la situación más sencilla, la que hace que un certificado no sea válido a partir del instante en el cual se realiza la revocación ([92] contiene una clasificación de los esquemas de revocación). Sin embargo, si se desea que una revocación pueda tener efectos retroactivos, es necesario distinguir entre el instante en el cual un certificado es revocado y el periodo durante el cual el privilegio tiene vigor.

En [175], los autores proponen algunos mecanismos para resolver los aspectos relacionados con la propagación de revocaciones. Dichos mecanismos hacen uso de certificados definidos como se expresa en §4.12.

$$autoriza(K_{auth}, K_U, P[I], sello-tiempo, id) \tag{4.12}$$

El sello de tiempo, generado por una entidad confiable, hace referencia al instante en el cual se crea el privilegio, e I es el intervalo durante el cual puede ejercerse el privilegio P. Los sellos de tiempo se utilizan para evitar que los certificados creados después de que el emisor haya perdido su autoridad puedan ser considerados como válidos, lo cual puede lograrse fácilmente mediante la falsificación del intervalo de tiempo I.

Por otro lado, las revocaciones se representan como se muestra en §4.13.

$$revoca(K_{auth}, id, [I], sello-tiempo) \tag{4.13}$$

Contienen el identificador *id* del certificado sujeto a revocación y un periodo de tiempo I denominado el periodo de deshabilitación. Dicho intervalo posibilita revocar certificados que fueron emitidos en el pasado. Por ejemplo, un periodo de deshabilitación con una fecha *not-before* anterior al sello de tiempo sirve para anular certificados anteriores, mientras que una fecha igual a dicho sello se utiliza para revocar sólo al certificado *id*.

Aunque la propuesta aporta soluciones al problema de la propagación de revocaciones, no es apropiada para todos los entornos de aplicación. En primer lugar requiere el uso de un sistema de sellado de tiempo confiable, el cual es un servicio inherentemente centralizado que choca con el enfoque claramente descentralizado de la delegación mediante certificados. De hecho, algunos sistemas suponen que los certificados de credencial pueden ser generados de forma *off-line*, lo cual imposibilita el uso de este tipo de servicios centralizados. Por otro lado, la revocación afecta a los certificados identificados por *id*. Si el mismo privilegio ha sido asignado mediante varios certificados, la revocación de uno de ellos no deshabilita el privilegio en sí, lo cual podría solventarse si la revocación hiciera referencia a los permisos y no a un número de serie.

4.4.6 Soporte para la delegación en las especificaciones analizadas sobre certificados de credencial

Una vez estudiados los aspectos más importantes relacionados con la delegación en sistemas distribuidos, se realizará una comparativa de las distintas especificaciones sobre certificados de credencial analizadas en la sección 4.3. El objetivo es mostrar qué características de las enumeradas a lo largo del análisis que se acaba de realizar están presentes en dichas propuestas.

La lista de propuestas contrastadas está formada por el sistema KeyNote, la PMI X.509 y la especificación SPKI/SDSI. Al ser KeyNote una evolución del sistema PolicyMaker, se ha decidido analizar exclusivamente la especificación más reciente.

Los aspectos de la delegación presentes en esta comparativa son:

- *ACLs o políticas basadas en delegación.* Soporte para la especificación de políticas o listas de control de acceso basadas en delegación (Sección 4.4.1).
- *Cadenas de delegación.* Posibilidad de construir cadenas de delegación (Sección 4.4.1).
- *Control de la propagación.* Provisión de mecanismos para controlar a qué entidades se puede extender la propagación de los privilegios asignados a una entidad (Sección 4.4.1).
- *Autoridad y posesión.* Posibilidad de separar los conceptos de autoridad y posesión de privilegios (Sección 4.4.2).
- *Transferencia.* Provisión de mecanismos para implementar la transferencia de privilegios (Sección 4.4.2).

- *Reducción de certificados.* Posibilidad de reducir cadenas de delegación de forma automática (Sección 4.4.3).

- *Descubrimiento de certificados.* Soporte para realizar el descubrimiento de certificados almacenados de forma distribuida (Sección 4.4.4).

- *Revocación.* Provisión de mecanismos para especificar revocaciones (Sección 4.4.5).

La tabla 4.1 contrasta dichos criterios respecto a las especificaciones ya estudiadas.

Criterio	**KeyNote**	**PMI X.509**	**SPKI/SDSI**
ACL/Política	Aserciones de tipo POLICY	No especificadas	Listas de control de acceso SPKI
Cadenas delegación	Soportadas	Soportadas (no recomendadas por PKIX)	Soportadas
Control propagación	Basado en funciones umbral k-of-n	Control booleano (mediante la extensión *Basic Attributes Constraints*) y control del subárbol (mediante *Delegated Name Constraints*)	Control booleano y basado en funciones umbral k-of-n
Autoridad y posesión	Sin distinción	Posible control mediante extensiones *Basic Attributes Constraints* y *Delegated Name Constraints*	Sin distinción
Transferencia	No soportada	No soportada	No soportada
Reducción	Mediante el motor de conformidad	No especificada	Mediante reducción de tuplas
Descubrimiento	No especificado	No especificado	No especificado
Revocación	No especificada	Mediante las listas de certificados de atributo revocados (ACRL), con efectos retroactivos mediante fechas de invalidación	Mediante CRLs y métodos en línea, sin efectos retroactivos

Tabla 4.1: Soporte para la delegación de las especificaciones estudiadas

4.5 Planteamiento de las soluciones proporcionadas

Al amparo de todo lo expuesto en este capítulo, parece claro que se ha llegado a un cierto nivel de madurez en lo que a especificaciones de certificados de credencial se refiere. En conclusión, podemos observar que si bien los lenguajes de codificación de dichas propuestas son capaces de soportar la mayoría de las exigencias derivadas del control de acceso distribuido, tanto basado en roles como en delegación, falta dotarle a estos planteamientos de un marco mediante el cual puedan adaptarse a entornos reales.

En cierto sentido, se podría afirmar que la autorización basada en certificados ha alcanzado un cierto reconocimiento en lo que a planteamiento se refiere, es decir, en lo que respecta a la parte más estática del enfoque: formatos de los certificados, formato de las listas de control de acceso, entidades que participan, etc. Sin embargo, es quizá la parte dinámica de este enfoque la que presenta mayores carencias y la que necesita un mayor esfuerzo por parte de la comunidad científica.

En consecuencia, parte del trabajo de esta tesis fue la definición de una infraestructura de autorización basada en certificados de credencial, la cual está destinada a proporcionar los mecanismos necesarios para la construcción de sistemas distribuidos basados en los conceptos de roles y delegación. Como veremos en los siguientes capítulos, dicha definición abarca los siguientes elementos de trabajo:

- *Marco de intercambio de información relativa a autorización.* Tal y como se comentó en la sección 4.4.4, sorprende la falta de propuestas relacionadas con el intercambio de información relativa a autorización. Este vacío motivó la definición de un marco que tiene por objetivo proporcionar los mecanismos necesarios para controlar el acceso a recursos protegidos en escenarios basados en el modelo cliente-servidor. Como se verá en la sección 5.2, este marco es capaz de negociar las características de seguridad de las sesiones establecidas entre los usuarios y los controladores de recursos, intercambiar información relativa a solicitudes de acceso, certificados de credencial y políticas de seguridad, proteger la transferencia de los recursos protegidos y optimizar las solicitudes realizadas dentro de una misma sesión. Se detallará además una implementación de dicho marco realizada mediante un protocolo de comunicaciones que puede actuar como una capa de transporte transparente para las aplicaciones.

- *Sistema distribuido de gestión de credenciales.* Si analizamos la evolución de los sistemas basados en X.509, podemos apreciar que a partir de la definición de los certificados se desarrollaron gran cantidad de soluciones destinadas a gestionar el ciclo de vida de los mismos. En este sentido, los sistemas X.509 cuentan con propuestas que hacen referencia a la arquitectura del sistema, protocolos de comunicación entre las entidades participantes, formatos de solicitud de certificados, servicios de validación, etc. Sin embargo, en materia de certificados de credencial, la gestión del ciclo de vida de los certificados ha sido un campo en el que apenas se ha realizado aportaciones. El sistema de gestión de credenciales presentado en la sección 5.3 ofrece los mecanismos necesarios para gestionar sistemas distribuidos basados en roles y delegación. Entre

las especificaciones de dicho sistema encontramos la definición de la arquitectura del mismo, identificación de las entidades participantes, mecanismos de comunicación entre las mismas, definición de los formatos de solicitud de certificados de credencial, definición de las políticas de concesión de privilegios, mecanismos de definición de roles y métodos de reducción automática de cadenas de delegación.

- *Metodología para la definición de estructuras de gestión de credenciales.* El sistema presentado en la sección 5.3 está compuesto por un gran número de componentes y elementos a gestionar. En concreto, encontramos autoridades de autorización, autoridades de nombramiento, puntos de acceso al servicio, entidades solicitantes, entidades receptoras, roles, relaciones entre los roles y privilegios. La metodología presentada en la sección 5.4 tiene como objetivo establecer un enfoque estructurado que permita modelar entornos de control de acceso complejos en los cuales el número de entidades participantes resulta demasiado elevado como para abordar la especificación de las estructuras de gestión de una forma arbitraria. Dicha metodología identifica los distintos niveles de establecimiento de dichas estructuras, los procedimientos a seguir en cada uno de dichos niveles y su materialización en el sistema presentado en 5.3.

- *Implementación e integración en entornos de aplicación reales.* Por último, el capítulo 6 mostrará los detalles de la implementación de los elementos de trabajo anteriormente descritos, así como la aplicación de dichos elementos en entornos de aplicación reales. De esta forma se podrá comprobar tanto su viabilidad como su integración con ciertas arquitecturas de seguridad (o también denominadas *middleware* de seguridad).

La definición de estos componentes y su integración en escenarios reales permitirá mostrar las posibilidades que pueden ofrecer las infraestructuras de autorización en el campo de los sistemas distribuidos.

Capítulo 5

Una infraestructura de autorización basada en certificados

Una vez analizadas las alternativas y las posibilidades de los certificados de credencial, se trata ahora de presentar el conjunto de componentes que dan lugar a la infraestructura de autorización diseñada. En primer lugar, se verá cuál es la estructura general del sistema y cómo está relacionada con la infraestructura de clave pública vista en el capítulo 3. A continuación se introduce el marco de intercambio de información relativa a autorizaciones, tanto su diseño general como el protocolo que implementa las recomendaciones. Posteriormente, se describen tanto las entidades como las especificaciones relativas al sistema de gestión distribuida de credenciales basado en delegación y roles. Por último, el capítulo concluye con la presentación de la metodología que permitirá afrontar la puesta en marcha de un sistema de control de acceso de forma estructurada y haciendo uso de la infraestructura de autorización.

5.1 Visión general del sistema

A la hora de extender la infraestructura de clave pública presentada en el capítulo 3 con el fin de incorporar mecanismos de autorización, era necesario identificar qué elementos compondrían dicha extensión y cuáles serían los nexos con el sistema de partida.

Desde el punto de vista de su funcionalidad, la PKI resulta el mecanismo ideal para la generación y distribución de claves criptográficas entre los usuarios del sistema. La emisión controlada de certificados de identidad y la difusión de los mismos a través de tarjetas inteligentes nos sitúa en el punto de partida a la hora de iniciar la tarea de asignar privilegios a las claves contenidas en dichos certificados. De esta forma, se puede decir que el proceso de gestión de las claves (y de la identidad) se mantiene independiente del proceso de gestión de las autorizaciones, puesto que todas las cuestiones relacionadas con la validez o revocación de claves forman parte de la responsabilidad de la PKI, lo cual permite definir sistemas de gestión de autorizaciones totalmente enfocados a las cuestiones de manejo de privilegios.

El hecho de partir de una infraestructura de gestión de identidades tiene como contrapartida la presencia de identificadores únicos, al menos en el ámbito organizativo en el cual está definido del sistema. Dichos identificadores contenidos en los certificados de identidad X.509 pueden ser un problema en algunos escenarios de control de acceso en los cuales el anonimato resulta un requisito forzoso. En consecuencia, el sistema de gestión de autorizaciones debe proporcionar los mecanismos necesarios para eliminar el enlace que existe entre las claves públicas de los usuarios y su correspondiente identificador único. Es decir, para llegar a un sistema de control de acceso anónimo a partir de una infraestructura de certificación de identidad será necesario desarrollar propuestas que oculten dicha transición.

La figura 5.1 muestra cuál es la conexión entre la infraestructura de clave pública y la infraestructura de autorización basada en certificados que se presenta en este capítulo.

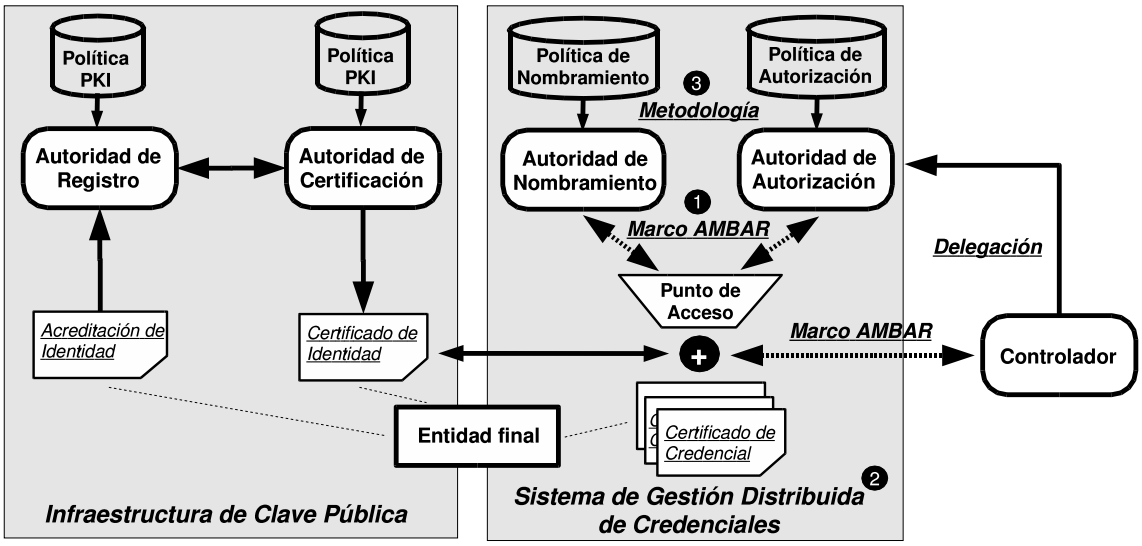

Figura 5.1: Visión general del sistema

Si se comparan las características de una PKI y un sistema de este tipo, es posible encontrar varias similitudes en lo que a estructura y funcionalidad se refiere.

En primer lugar, ambas necesitan una infraestructura formada por entidades emisoras, entidades intermedias (o mediadoras) y usuarios finales. En el caso de la PKI vimos como las autoridades de registro y las autoridades de certificación cooperan para tramitar las solicitudes de certificación presentadas por las entidades finales. En el caso de la infraestructura de autorización, es necesaria la presencia de elementos encargados de emitir los distintos tipos de credencial (autoridades de nombramiento y de autorización), así como la intervención de elementos intermedios capaces de poner en contacto a dichas autoridades con las entidades finales (puntos de acceso). Además, parte de ambas infraestructuras deben ser las especificaciones relacionadas con el formato de las solicitudes de certificación y formato de las políticas de seguridad, así como los medios utilizados para poner en contacto a las distintas entidades que la componen.

Por otro lado, en ambos casos las entidades emisoras deben seguir políticas concretas de certificación a la hora de atender las solicitudes presentadas por los usuarios finales. Vimos que las políticas de PKI permiten determinar si una solicitud o certificado cumplen con lo especificado por las prácticas de certificación. En el caso de la infraestructura de autorización, el uso de políticas está encuadrado en dos entornos distintos. En primer lugar, las políticas de nombramiento y de autorización permitirán determinar si una entidad concreta puede ser asociada a un conjunto de roles o de privilegios. En segundo lugar, el uso de políticas permitirá a los puntos de acceso tener un conocimiento de la estructura del sistema y de los requisitos de seguridad impuestos en el proceso de solicitud de credenciales. Dada la complejidad de ambos tipos de políticas, éstas deberán ser diseñadas siguiendo una metodología concreta que permita manejar de forma estructurada el gran número de usuarios y condiciones del sistema.

Finalmente, la última similitud está relacionada con la necesidad de mecanismos genéricos para el intercambio de los certificados generados, es decir, propuestas que permitan a las entidades participantes de una comunicación transmitir los certificados necesarios para el servicio que se está desarrollando. En el caso concreto de una PKI, dichos mecanismos los constituyen los distintos protocolos de seguridad con soporte para certificados X.509. Respecto a la autorización, es necesario un marco que permita intercambiar información relativa a autorización entre los controladores y las entidades finales.

Todas estas características propias de la infraestructura de autorización se agrupan en los tres bloques básicos presentados en este capítulo, los cuales aparecen numerados en la figura 5.1.

- *Marco AMBAR.* El marco AMBAR (Access Management Based on Authorization Reduction) es el mecanismo mediante el cual se transmite toda la información relacionada con autorización. Por un lado, el marco se emplea en las comunicaciones realizadas entre las entidades finales y los controladores de recursos con el fin de proporcionar un medio mediante el cual intercambiar certificados de identidad, certificados de credencial, políticas de autorización y recursos protegidos. Por otra parte, el marco forma también parte del sistema de gestión distribuida de credenciales ya que se emplea también para transmitir las solicitudes de certificación realizadas a las autoridades por parte de los usuarios finales. Los detalles de AMBAR se expondrán en la sección 5.2.

- *Sistema de gestión distribuida de credenciales.* Este sistema abarca tanto la definición de las entidades necesarias para la gestión de credenciales como la especificación de los elementos de información necesarios para dicho propósito. Está basado completamente en el mecanismo de delegación y en el concepto de rol, lo cual determina la mayor parte de sus características en lo que a estructura y notación se refiere. Tal y como se verá en la sección 5.3, el sistema realiza una distinción clara entre la gestión de la pertenencia a roles y la asignación de privilegios a dichos roles. Esta separación de conceptos puede apreciarse en la figura 5.1, donde el mecanismo de nombramiento (o pertenencia) dispone de sus propias autoridades y políticas independientes de la

autorización. Sin embargo, el acceso a la funcionalidad ofrecida por ambos subsistemas está agrupado en ciertos elementos mediadores denominados puntos de acceso, a través de los cuales es posible solicitar y obtener los certificados de credencial.

- *Metodología de definición de estructuras de gestión.* La puesta en marcha de un sistema de control de acceso basado en roles y delegación requiere una identificación muy concisa de los elementos participantes y de la relación entre ellos. Se trata de identificar todos los recursos que se desea proteger, determinar qué acciones realizadas sobre ellos deben controlarse, descubrir cuáles son los roles fundamentales del sistema, la política de pertenencia a dichos roles, el conjunto de privilegios asociados a los mismos, identificar a las entidades encargadas de emitir los certificados correspondientes y acotar los periodos de validez de los mismos, entre otras tareas. Debido al gran número de elementos involucrados y a la complejidad de las tareas asociadas, es necesario establecer una metodología genérica de construcción de políticas de autorización y de nombramiento, a partir de las cuales pueda abordarse el desarrollo del sistema de una forma estructurada. Dicha metodología de diseño se analizará en la sección 5.4.

Como se verá a lo largo de este capítulo, los distintos componentes se encuentran totalmente relacionados entre sí y, a su vez, con la infraestructura de clave pública ya descrita.

5.2 AMBAR: marco de intercambio de información relativa a autorización

En la sección 4.4.4 se analizaron las distintas alternativas posibles a la hora de obtener o descubrir las credenciales necesarias para tomar las decisiones de autorización. Como ya se comentó, los sistemas de control de acceso pueden emplear enfoques muy distintos en lo que a distribución de credenciales se refiere. Algunos de ellos determinan que la responsabilidad de obtener la información es del controlador de recursos, mientras que otros argumentan que deben ser los solicitantes los encargados de proporcionar la información relativa a sus privilegios. En general, no hay acuerdo acerca de cuál es la mejor alternativa ya que en la mayoría de los casos depende del entorno de aplicación concreto.

Por otro lado, también se ha comentado la falta de mecanismos genéricos relacionados con el intercambio de información de autorización. Los protocolos de seguridad más comunes, como TLS (Transport Level Security) [59], IKE (Internet Key Exchange) [94], o S/MIME (Secure/Multipurpose Internet Mail Extensions) [168], están preparados para transmitir única y exclusivamente certificados de identidad.

Como consecuencia, uno de los campos de trabajo a los que más esfuerzo se le ha dedicado en esta tesis es la definición de un marco que proporcione los mecanismos necesarios para controlar el acceso a recursos protegidos en escenarios basados en el modelo

cliente-servidor. Este marco, denominado *AMBAR (Access Management Based on Authorization Reduction)*, se ha diseñado siguiendo un enfoque estructurado mediante el cual se han identificado los distintos parámetros relacionados con el intercambio de información relativa a autorización, todo ello con el fin de adaptar el marco a los distintos enfoques ya comentados acerca de distribución de certificados de credencial.

En primer lugar se analizarán las limitaciones de los sistemas existentes y se contrastará la propuesta con otras iniciativas relacionadas. A continuación se enumerarán los requisitos del marco y se describirá su arquitectura. Posteriormente se detallará una implementación concreta del marco basada en un protocolo cliente-servidor denominado protocolo AMBAR. Por último se realizará un análisis de seguridad de dicha implementación del marco.

5.2.1 Análisis de las propuestas actuales

En esta sección se analizará cómo se lleva a cabo normalmente el control de acceso basado en certificados. Este análisis mostrará por qué las propuestas actuales pueden verse mejoradas mediante la utilización del marco que aquí se presenta. Se ha seleccionado un escenario basado en Web, donde el controlador de recursos toma decisiones en función de la información de autorización presentada por los clientes y de su propia política de control de acceso. Se supondrá que dicho controlador delega en autoridades externas el privilegio de determinar qué entidades están autorizadas a acceder a los recursos, determinación que realizarán éstas mediante la emisión de certificados de credencial. Así pues, el acceso será concedido siempre que el controlador disponga de toda la información necesaria para verificar que la solicitud cumple con su política de seguridad.

Uno de los enfoques más tradicionales que se pueden seguir a la hora de implementar este sistema es el mostrado en la figura 5.2. En ella podemos observar como tanto el controlador como el cliente disponen de módulos adicionales encargados de las funciones de control de acceso. Esta funcionalidad puede ser añadida al software del cliente mediante la utilización de *applets* o *ActiveX*. El servidor Web en el cual se encuentra ubicado el controlador puede realizar dicha función mediante el uso de *servlets* o extensiones de servidor específicas. Cuando un usuario solicita el acceso a un recurso, se establece una conexión SSL [9] con el fin de autenticar a los participantes de dicha comunicación y de proteger los datos que se enviarán a continuación. Acto seguido, se genera un mensaje HTTP [77] que especifica el recurso solicitado. Los certificados de credencial deben incluirse en el documento HTML o en alguna cabecera HTTP ya que SSL no proporciona mecanismos para intercambiar este tipo de información. Por tanto, solicitud y credenciales son encapsuladas en el mismo paquete SSL y enviadas al controlador. El paquete es procesado por el módulo SSL y parte de su contenido es entregado al software de control de acceso con el fin de determinar si la solicitud debe ser aprobada. Finalmente, el recurso se entrega al usuario y, opcionalmente, se adjunta un conjunto de certificados destinados a proporcionar una autorización directa que simplifique solicitudes posteriores.

Esta solución puede tener asociados varios inconvenientes. En primer lugar, las credenciales deben incluirse como parte de los datos de la aplicación (HTML). Con el fin de independizar el mecanismo de control de acceso del entorno de aplicación concreto, sería

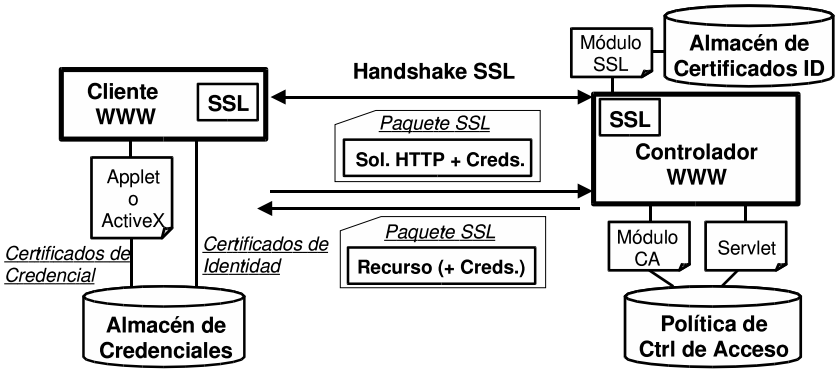

Figura 5.2: Enfoque común de control de acceso basado en certificados

conveniente no mezclar los datos de aplicación con la información relativa a autorización. De hecho, los certificados de identidad intercambiados durante la fase inicial de autenticación SSL se transmiten siguiendo este enfoque. Una forma de no combinar ambos tipos de datos es transmitir en primer lugar la solicitud, después obtener del controlador la política de control de acceso asociada y finalmente enviar los certificados de credencial. El problema está en que el controlador podría considerar que la política de control de acceso contiene información confidencial, la cual no debería ser difundida a usuarios desconocidos. Las credenciales transmitidas junto con la solicitud siguiendo el enfoque original pueden ayudar al servidor a determinar si puede desvelar su política.

Por otro lado, esta propuesta carece de una fase de negociación de los parámetros de autorización. Como se vio en la sección 4.3, son varias las especificaciones realizadas en materia de certificados de credencial, y varios los métodos de distribución de los mismos. Los participantes deben poder seleccionar si las credenciales serán proporcionadas por parte del cliente (método denominado *push*) o si bien serán recuperadas por parte del controlador de algún suministrador (método *pull*). Además, el método *push* puede subdividirse a su vez en función de si se realiza una difusión controlada de la política de control de acceso. Mediante la propuesta presentada en la figura 5.2 es difícil realizar una negociación de dichos parámetros que controle todas las solicitudes realizadas por parte del cliente dentro de la misma sesión [91].

Finalmente, dejar constancia de la dificultad existente con este enfoque a la hora de optimizar solicitudes subsecuentes. Es común que colecciones de recursos organizadas por directorios hereden los derechos de acceso conforme nos adentramos en el árbol de documentos. Por tanto, una vez que un usuario ha sido autorizado a acceder a un recurso en la misma sesión no resulta necesario retransmitir las credenciales implicadas en dicha decisión. Sin embargo, la implementación de sesiones, y por tanto la posibilidad de realizar optimizaciones dentro de la misma sesión, es un proceso que debería implementar la propia aplicación del controlador, lo cual complica su diseño.

Podemos encontrar en la literatura algunos sistemas de control de acceso que siguen este enfoque. En [138] se presenta un mecanismo de control de acceso a recursos Web basado en certificados SPKI y en el uso del protocolo HTTP como mecanismo de trans-

porte de información de autorización. En consonancia con lo que se acaba de comentar, dicho sistema carece de fase de negociación, mantenimiento de sesiones u optimización de solicitudes.

Enfoques alternativos

Conscientes de las limitaciones del esquema presentado en la figura 5.2, son varios los autores que han propuesto sistemas alternativos que intentan suplir algunas de las carencias anteriormente comentadas.

En [184] se presenta un mecanismo que emplea certificados digitales para definir y aplicar políticas de control de acceso sobre recursos ampliamente distribuidos. La arquitectura está basada en el modelo *pull*, donde el controlador se encarga de recuperar los certificados asociados a los usuarios con el fin de determinar si se cumplen las condiciones especificadas por los proveedores de los recursos a controlar. El sistema proporciona algunos mecanismos para optimizar solicitudes subsecuentes, como por ejemplo el uso de caches de certificados. Como se verá más adelante, el marco aquí presentado complementa esta propuesta ya que también es capaz de dar soporte a sistemas basados en delegación, los cuales suelen estar basados en el método de distribución *push*.

Otros trabajos están relacionados con el control de la difusión de políticas [179]. Dicha propuesta muestra cómo es posible diseñar sistemas de establecimiento automático de confianza (*automated trust establishment*) que controlen la revelación de información confidencial contenida en las políticas de control de acceso. Como se verá en la sección 5.2.3, este mecanismo puede incorporarse a uno de los módulos del marco AMBAR con el fin de controlar dicha difusión de las políticas.

Un enfoque que hace uso del protocolo SSL es el presentado en [97]. Esta propuesta hace uso de los certificados X.509 intercambiados durante la fase de negociación SSL para asignar a los usuarios solicitantes un rol dentro del sistema. Dicha asignación se realiza siguiendo una política de pertenencia a roles definida por el controlador de los recursos. El sistema proporciona sólo una solución parcial al problema ya que no determina los mecanismos mediante los cuales se asignan los privilegios a los roles, sino que simplemente proporciona una solución destinada al agrupamiento de usuarios en roles. A pesar de lo que se vio en la sección 4.3, los propios autores del sistema no consideran útiles los certificados de credencial a la hora de especificar la pertenencia a grupos ni de especificar los permisos asociados a los mismos.

El grupo de trabajo AAA (Authentication, Authorization and Accounting) del IETF ha propuesto también un marco de autorización [146, 186] destinado a la protección de recursos y servicios dentro del ámbito de Internet. El marco está basado principalmente en el control de acceso a la red, movilidad y calidad de servicio en IPv6. La principal diferencia entre esta propuesta y el marco AMBAR es que este último está más enfocado a aplicaciones y escenarios de alto nivel.

5.2.2 Objetivos generales del marco

El marco que aquí se presenta soluciona gran parte de los inconvenientes que se han mencionado en el apartado anterior. AMBAR da soporte a distintos tipos de certificados de credencial y de identidad, e incorpora un mecanismo de negociación diseñado para adaptar el marco a escenarios de control de acceso con distintas características. Tal y como se verá en apartados posteriores, AMBAR es un marco basado en sesiones que está constituido por varios módulos independientes. La figura 5.3 muestra cómo puede adaptarse dicho marco a escenarios basados en el Web.

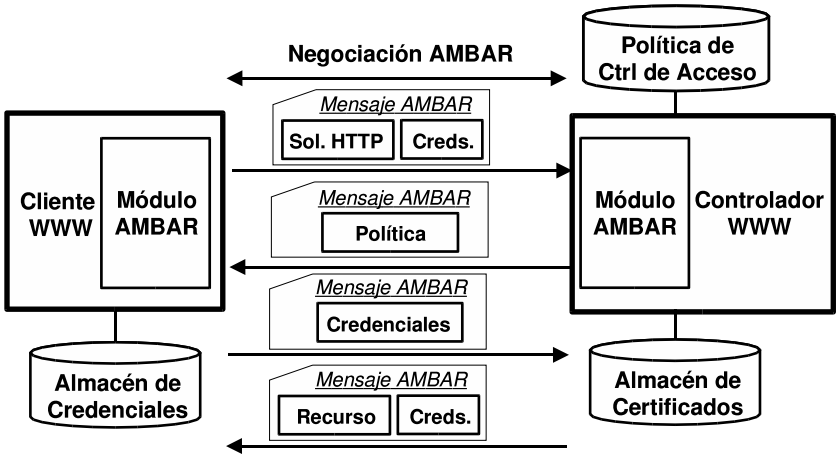

Figura 5.3: Control de acceso basado en AMBAR

Tanto el controlador como el cliente disponen de módulos adicionales AMBAR responsables de las funciones de control de acceso. Cuando un usuario solicita un recurso protegido, se inicia una fase de negociación de sesión con el fin de autenticar a las entidades participantes, negociar qué tipo de credenciales será utilizado, acordar cuál será el método de distribución de las mismas y determinar si la información intercambiada debe ser protegida de posibles ataques tanto pasivos como activos. Una vez que se establece la sesión, las solicitudes HTTP y las credenciales se encapsulan de forma separada dentro de paquetes AMBAR y se envían al controlador. El contenido de dichos paquetes puede ser procesado por un módulo del marco AMBAR o bien ser entregado a otra aplicación. En este ejemplo, el contenido es procesado por el módulo AMBAR, el cual determina la política de autorización relacionada con la solicitud (la selección de políticas puede estar basada en diversos métodos y suele ser dependiente de las credenciales contenidas en el primer mensaje). A continuación, el controlador transmite un paquete que incluye la política de autorización que especifica las credenciales necesarias para obtener el acceso. Finalmente, el cliente transmite dichas credenciales y el controlador suministra el recurso solicitado.

Hay varias ventajas en el hecho de utilizar un enfoque de este tipo. La primera es que las entidades pueden negociar los parámetros relacionados con el control de acceso. Además, las credenciales y los datos de aplicación se envían de forma separada, y es bastante sencillo intercambiar varios mensajes (solicitud, credenciales, políticas) para resolver una solicitud

de acceso. Al estar basado en un enfoque orientado a la sesión, es posible relacionar solicitudes entre sí con el fin de optimizar los cálculos o los envíos necesarios para tomar las decisiones.

En vista de lo analizado, hay tres objetivos principales que deben ser satisfechos por el marco AMBAR. En primer lugar, debe ser independiente del entorno de aplicación, es decir, debe dar soporte a cualquier tipo de política, privilegio o solicitud. En segundo lugar, debe ser capaz de operar con las principales especificaciones en materia de certificados de identidad y de credencial, así como de negociar los parámetros de autorización de cada sesión. Por último, su diseño debe ser extensible, estructurado y estar dividido en módulos con funciones bien definidas.

Con el fin de optimizar aquellos escenarios en los que varios mensajes de solicitud y respuesta se intercambian continuamente entre un cliente y un controlador, debe tratarse de un marco orientado a la sesión. En estos casos, la mayor parte de la información necesaria ya fue enviada con solicitudes anteriores, y algunas de las decisiones de autorización ya obtenidas pueden ser útiles para determinar si una nueva solicitud debe ser aprobada, sin la necesidad de calcular nada de nuevo. Como se verá en posteriores apartados, las caches de certificados y las reducciones de autorización son mecanismos muy indicados para estos propósitos.

5.2.3 Arquitectura del marco

El marco AMBAR está compuesto por diferentes módulos organizados, tal y como muestra la figura 5.4, en dos capas. La capa superior está formada por cinco módulos funcionales distintos: Gestión de Sesiones *(SM, Session Management)*, Gestión de Solicitudes *(RM, Request Management)*, Gestión de Resultados de Autorización *(ARM, Authorization Results Management)*, Gestión de Flujos de Datos *(DSM, Data Stream Management)*, y Gestión de Errores *(EM, Error Management)*. En el nivel inferior se sitúa la capa de Convergencia de Transporte *(TC, Transport Convergence)*. La capa TC encapsula toda la información generada por los módulos superiores de acuerdo con el mecanismo de transporte correspondiente. Opcionalmente, esta capa protege la confidencialidad y la integridad de la información mediante la aplicación de alguno de los mecanismos negociados durante una fase previa. Los siguientes apartados detallan la funcionalidad de cada módulo.

Figura 5.4: Arquitectura AMBAR

Session Management

Este módulo proporciona los mecanismos para negociar las diferentes opciones soportadas por el marco y para establecer los parámetros de la sesión. El módulo genera además todo el material criptográfico que pudiera ser necesario para la capa TC a la hora de proteger la información intercambiada.

Los solicitantes y los controladores de recursos pueden negociar los siguientes parámetros:

- *Cifrador simétrico.* Los participantes pueden seleccionar qué algoritmo de cifrado simétrico (y su longitud de clave) protegerá los datos intercambiados.
- *Modo de operación.* AMBAR proporciona dos modos de operación: modo anónimo (la identidad del solicitante no se revela) y modo identificado (donde tanto el solicitante como el controlador son identificados).
- *Certificados de identidad.* Los participantes pueden seleccionar qué tipo de certificados de identidad serán empleados para propósitos de autenticación (X.509, PGP, SPKI/SDSI, etc).
- *Certificados de credencial.* Los participantes pueden seleccionar qué tipo de certificados serán utilizados para propósitos de autorización (SPKI/SDSI, X.509 AC, KeyNote, etc).
- *Método de distribución.* Es posible negociar si las credenciales serán proporcionadas por parte del solicitante (*push*) o si bien serán obtenidas por parte del controlador desde algún suministrador. El método *push* puede a su vez subdividirse en varias posibilidades dependiendo del criterio seguido para la revelación de la política de autorización y de la entidad responsable de hallar la prueba de autorización.

Todos estos parámetros se negocian en función de las políticas de seguridad específicas definidas por los sistemas finales. El objetivo principal del módulo SM es adaptar el marco a los distintos escenarios de control de acceso y crear sesiones AMBAR.

Request Management

Las decisiones de autorización, procesos de optimización o los algoritmos de control de difusión de políticas son ejecutados como parte del módulo RM. En general, este módulo está encargado de la gestión de todo lo relacionado con solicitudes, credenciales y políticas.

Las solicitudes pueden ser generadas por la aplicación del solicitante o pueden ser derivadas a partir de datos específicos de la aplicación dentro de este módulo. Por ejemplo, una solicitud HTTP puede ser convertida de forma automática a una s-expresión con el fin de simplificar el proceso de autorización, aunque tanto la solicitud HTTP como la s-expresión serían transmitidas al controlador. Las s-expresiones son especialmente útiles porque reflejan sólo aquella información relacionada con el proceso de decisión.

Las credenciales pueden ser recuperadas a partir de almacenes de certificados, de entidades emisoras o de los propios solicitantes. Dichas credenciales deben ser verificadas y validadas, aunque los mecanismos para dichos propósitos son completamente dependientes de la infraestructura disponible ya que, como se ha visto en capítulos anteriores, dichas comprobaciones pueden estar basadas en listas de certificados revocados, mecanismos de verificación en línea, certificados de corta duración, etc. Una consecuencia lógica de todo esto es que las carencias de la infraestructura en la cual se esté aplicando el marco pueden afectar a éste en lo que a seguridad se refiere.

Las políticas son emitidas con el fin de especificar qué credenciales son necesarias para obtener el acceso a los recursos que se están solicitando. El criterio de revelación de dichas políticas depende del controlador en cuestión. En general, hay tres alternativas posibles a la hora de efectuar dicha revelación: los controladores pueden difundir gradualmente las políticas [179]; pueden difundirlas sin ningún tipo de control; o pueden decidir no desvelarlas (más característico del modo *pull*).

Como ya se ha mencionado, un protocolo orientado a la sesión permite realizar algunas optimizaciones. El módulo RM es responsable del cálculo y las optimizaciones de las decisiones de control de acceso. Las optimizaciones están condicionadas por varios parámetros, como el tipo de certificado de credencial que se esté empleando y el método de distribución que se haya negociado.

Una de las formas más sencillas de minimizar el número de envíos es guardar una copia local de los certificados que han sido intercambiados. No obstante, hay que tener en cuenta que la validez de los certificados almacenados de forma local debe ser comprobada de forma periódica, sobre todo teniendo en cuenta que posibles revocaciones pueden alterar su estado.

Otro mecanismo que se emplea para reducir el ancho de banda utilizado es la transmisión de los resúmenes digitales de los elementos de información que ya han sido transmitidos previamente, lo cual es especialmente útil a la hora de retransmitir políticas de autorización extensas. Para ello, el módulo RM puede hacer uso de una tabla de dispersión que relacione los resúmenes digitales recibidos con los elementos previamente transmitidos.

Por otro lado, con el fin de simplificar el cálculo de autorización, es posible hacer uso de decisiones previas para determinar si una nueva solicitud de acceso debe ser aprobada o denegada. La figura 5.5 muestra un ejemplo de esta situación. Una vez que la solicitud de acceso al fichero C ha sido procesada, R no necesita presentar ninguna credencial nueva para acceder a los ficheros A o E ya que la reducción de autorización llevada a cabo por el controlador indica que tiene concedido el acceso a A, C y E. Esta reducción podría incluso ser utilizada durante otras sesiones para evitar la transmisión de credenciales incluidas en dicha reducción. El mecanismo de reducción empleado para realizar estas optimizaciones dentro del módulo será explicado detalladamente en la sección 5.3.5.

Authorization Results Management

El módulo ARM proporcionar los mecanismos necesarios para generar las notificaciones acerca de las decisiones de control de autorización, y puede usarse además para gestionar los

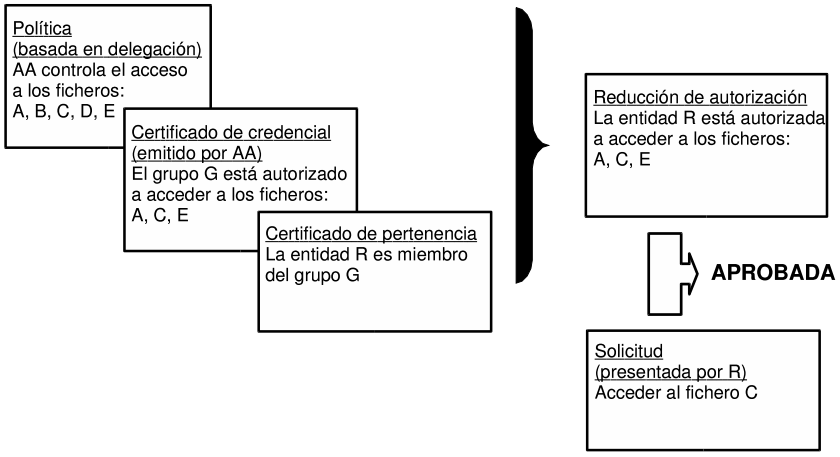

Figura 5.5: Ejemplo de optimización de solicitud de acceso

recursos solicitados. Se generarán notificaciones negativas cuando el acceso sea denegado. En el caso de que sea aprobado, dos son las posibles respuestas: una notificación positiva si el solicitante no desea obtener el recurso, sino sólo llevar a cabo alguna acción sobre él; o bien el recurso en sí. El módulo además habilita o deshabilita el módulo DSM siempre que se apruebe una solicitud relacionada con el inicio o la finalización de un flujo de datos.

Error Management

Las aplicaciones hacen uso del módulo EM para notificar situaciones de error o de alerta, como por ejemplo la verificación incorrecta de un mensaje, la recepción de un mensaje fuera de orden, la imposibilidad de negociación debida a preferencias incompatibles, la validación negativa de un certificado, etc. La información difundida acerca de estas situaciones contienen un nivel de gravedad y una descripción del error que se ha producido.

Data Stream Management

El modelo descrito basado en intercambios de solicitudes y respuestas no es apropiado si se desea utilizar el marco AMBAR para controlar comunicaciones bidireccionales basadas en flujos de datos. El módulo DSM, inicialmente deshabilitado, controla la transmisión de dichos flujos tras la aprobación de una solicitud de activación del mismo. Una vez que éste es activado, las aplicaciones son capaces de intercambiar datos libremente, que además puede ser protegidos por la capa TC si así se negoció durante la sesión (la sección 5.2.4 se detalla este proceso).

Transport Convergence

La capa TC codifica toda la información generada por los módulos superiores. Dicha codificación puede estar basada en XDR [143], XML o cualquier otro formato que resulte

apropiado. Además, la implementación del módulo TC depende del mecanismo de transporte concreto que se esté empleando para transmitir la información, el cual puede ser una conexión SSL, un socket TCP, etc. En el siguiente apartado se presenta una implementación concreta del marco AMBAR que hace uso de sockets TCP. En dicha implementación la capa TC ofrece servicios de confidencialidad y autenticación de los mensajes intercambiados.

5.2.4 El protocolo AMBAR como implementación del marco

Una vez que se ha analizado el diseño del marco AMBAR, en esta sección se expondrá cómo puede intercambiarse información relativa a autorización mediante el protocolo AMBAR [47]. El marco puede considerarse como las directrices a seguir a la hora de implementar un sistema de control de acceso concreto.

Parte del diseño del protocolo está basado en SSL. SSL es sin duda una aportación muy valiosa al campo de las comunicaciones confidenciales y su seguridad ha sido ampliamente estudiada durante los últimos años [145, 187]. AMBAR toma como punto de partida una modificación de la funcionalidad ofrecida por SSL y por tanto algunos mensajes se han visto simplificados o han sido eliminados. Dicha modificación se han realizado siguiendo algunas de las prácticas de diseño de protocolos criptográficos expuestas en [2, 13], las cuales hacen referencia al uso correcto de cargas aleatorias, la inclusión de información que pueda ser asociada a los participantes o la codificación de un mensaje dentro de una secuencia, entre otros factores. Es importante recalcar que el objetivo principal de esta implementación del marco no es la definición de un nuevo protocolo criptográfico, sino la especificación de un protocolo que cumpla con los requisitos expuestos en la sección 5.2.2.

La descripción del protocolo está estructurada atendiendo a los distintos módulos del marco, poniendo un énfasis especial en lo que a negociación de sesiones y gestión de solicitudes se refiere. Una especificación completa de los mensajes que componen el protocolo se encuentra en el apéndice B.

Notación empleada

La especificación de los mensajes está formada por 4 campos distintos. El primero de ellos hace referencia al orden del mensaje dentro de la fase en la cual se encuentra encuadrado. El segundo indica el nombre del mensaje. El tercer campo muestra si el mensaje es enviado desde el cliente al servidor o viceversa. Por último, el cuarto campo especifica los contenidos del mensaje.

Dichos contenidos se encuentran también expresados siguiendo una notación concreta. A continuación se detalla como debe interpretarse dicha notación:

- *item1+item2+item3*: Concatenación de varios elementos de información.
- k_X: Clave pública de X.

- $\{M\}_{k_X^{-1}}$: Mensaje M cifrado con la clave privada de X (en ocasiones equivalente a la firma digital de M).
- *SHA1(M) o MD5(M)*: Resumen digital de M calculado mediante las funciones SHA-1 [42] o MD5 [171].
- $\{M\}^{k_{MAC}}$: Código de autenticación de M calculado mediante la clave k_{MAC}.
- $\{M\}_{k_S}$: Mensaje M cifrado con la clave k_S.

Módulo SM

Los mensajes del módulo SM se usan para negociar las diferentes opciones soportadas por el marco y para establecer los parámetros de la sesión. Parte de los datos intercambiados se emplean para generar el material criptográfico que utiliza la capa TC para proteger la información (siempre que la opción de confidencialidad se haya negociado).

Cada mensaje SM contiene un campo que define el tipo concreto de mensaje. Los siguientes apartados detallan los mensajes que se intercambian durante la negociación tanto de una sesión identificada como de una sesión anónima.

Sesión identificada

1 **ClientInit** $C \Rightarrow S$ $Ver_c, N_c, Assert_c, Category_c, Suite_c, Identity_c, Distribution_c$

El mensaje *ClientInit* inicia la negociación AMBAR y contiene las preferencias del cliente. Ver_c identifica la versión de AMBAR del cliente, N_c es una carga aleatoria de 64 bytes que será empleada posteriormente para calcular el material criptográfico, $Assert_c$ indica las preferencias del cliente en lo que a certificados de credencial se refiere, $Category_c$ expresa el modo de operación propuesto (anónimo o identificado), $Identity_c$ indica las preferencias en lo que respecta a certificados de identidad y $Distribution_c$ contiene el modo de distribución de credenciales propuesto. $Suite_c$ es un campo que contiene una lista de los algoritmos de cifrado simétrico que soporta el cliente a la hora de proteger los datos de la capa TC. Cuando dicha lista contiene sólo el elemento *null* la capa TC no ofrece ningún tipo de servicio de protección de la información transmitida, y por tanto la fase SM se reduce al intercambio de los mensajes *ClientInit* y *ServerInit*.

2 **ServerInit** $S \Rightarrow C$ $Ver_s, N_s, Assert_s, SessionID, Category_s, Suite_s, Identity_s, Distribution_s$

El mensaje *ServerInit* es la respuesta del servidor a *ClientInit*. Las diferencias más significativas entre ambos mensajes son la presencia del identificador de sesión *SessionID* y la selección por parte del servidor de uno de los algoritmos simétricos propuestos. La elección de los parámetros por parte del servidor se realiza siempre teniendo en cuenta lo especificado por el cliente. Si alguna de las preferencias del cliente son incompatibles con las del servidor se produce el intercambio de mensajes EM para notificarlo.

3 **PKValue** $S \Rightarrow C$ $\{S, k_s\}_{k_{CA_1}^{-1}}$

El mensaje *PKValue* enviado por el servidor contiene información relativa a su identidad, y está compuesto normalmente por un certificado digital que incluye datos acerca de la clave pública del servidor k_s, su identificador asociado S y la entidad emisora CA_1.

4 **PKValue** $C \Rightarrow S$ $\{C, k_c\}_{k_{CA_2}^{-1}}$

Cuando el mensaje *PKValue* lo envía el cliente éste contiene un certificado digital con la identidad del cliente C, su clave pública k_c, y su entidad emisora CA_2 (CA_1 y CA_2 podrían hacer referencia a la misma autoridad, pero no es obligatorio).

5 **ActivateCrypto** $C \Rightarrow S$ $\{PreMasterSecret\}_{k_s}, \{SHA1(N_c + MasterSecret + N_s)\}_{k_c^{-1}}$

El mensaje *ActivateCrypto* se emplea para establecer el material criptográfico que protegerá los siguientes mensajes del protocolo y para verificar la identidad del cliente. En primer lugar, está compuesto de un valor de 64 bytes, denominado $PreMasterSecret$, que se transmite cifrado mediante la clave pública del servidor obtenida con el mensaje *PKValue*. Este $PreMasterSecret$ y las cargas aleatorias intercambiadas con los primeros dos mensajes dan lugar al $MasterSecret$. En segundo lugar, el mensaje contiene una cadena de bytes que representan la firma digital de la concatenación del $MasterSecret$ a dichas cargas aleatorias. Una vez que el servidor recibe el mensaje, éste descifra el $PreMasterSecret$, calcula a partir de él el valor del $MasterSecret$ y verifica la firma del cliente. De esta forma, el servidor puede averiguar si el cliente controla la clave privada asociada a la clave pública que fue transmitida en el cuarto mensaje y si ambos participantes han llegado al mismo $MasterSecret$. El cálculo del $MasterSecret$ se muestra en el apéndice B.

6 **InitSession** $C \Rightarrow S$ $\{SHA(MasterSecret + Issuer + SM\_Messages)\}_{k_{SYMM_s}}^{k_{MAC}}$

7 **InitSession** $S \Rightarrow C$ $\{SHA(MasterSecret + Issuer + SM\_Messages)\}_{k_{SYMM_c}}^{k_{MAC}}$

El último mensaje de la fase SM es *InitSession*. Sirve para indicar que la fase de negociación ha concluido y que se ha activado la protección criptográfica de los mensajes. Se trata del primer mensaje AMBAR protegido por la capa TC mediante los datos derivados a partir del $MasterSecret$. Su contenido está formado principalmente por el resumen digital de todos los mensajes intercambiados durante la fase de negociación (el contenido de este mensaje se analizará más en detalle en la sección 5.2.5).

Sesión anónima

Sólo hay una diferencia entre el modo identificado y el anónimo. Con el fin de preservar la identidad del cliente, el mensaje *PKValue* contiene en este caso sólo la clave pública del mismo.

4 **PKValue** $C \Rightarrow S$ k_c

Módulo TC

El módulo TC transforma los mensajes de la capa superior de acuerdo con lo negociado en la fase anterior. Los mensajes transmitidos con anterioridad a *InitSession* no están protegidos ya que las claves criptográficas se calculan a partir del $MasterSecret$.

El módulo proporciona además un formato común de encapsulamiento de los mensajes SM, RM, ARM, DSM y EM. En esta implementación se ha empleado la siguiente estructura para codificarlos:

AMBARMessage $C \Leftrightarrow S$ $tipo, longitud, datos$

El campo de *datos* contiene los mensajes, posiblemente protegidos, de la capa superior, cuyo tipo es *tipo* y de tamaño igual a *longitud*. Los mensajes protegidos están compuestos por dos campos: *contenido* y *MAC*. El *contenido*, el *tipo* y la *longitud* se autentican primero utilizando un algoritmo HMAC [42], y el código resultante se almacena en *MAC*. A continuación, tanto *contenido* como *MAC* se cifran utilizando el modo de cifrado CBC (Cipher Block Chaining) [122] y el sistema de relleno PCKS#5 [119]. Las claves empleadas para calcular los códigos de autenticación se denominan K_{MAC} y las claves de cifrado utilizadas son K_{SYMM_S} y K_{SYMM_C}. La forma de derivar estas claves a partir del $MasterSecret$ se muestra en el apéndice B.

Módulos RM, ARM y DSM

Con el fin de explicar el funcionamiento de estos módulos, se analizarán los distintos métodos de distribución a través de algunas secuencias típicas de mensajes. A lo largo de este apartado, se denominará *transacción* a los diferentes mensajes relacionados con una solicitud de acceso concreta, mientras que por *sesión* se entiende la secuencia de distintas transacciones.

Método de distribución push-calculation

En una sesión basada en el método *push-calculation* los clientes calculan la prueba de autorización tras la recepción de la política que protege los recursos gestionados por el controlador.

1	**Request**	$C \Rightarrow S$	$\{T_{ID}, T_{Step}, SFlag, Solicitud, [Asserts]^{0..N}\}^{k_{MAC}}_{k_{SYMM_s}}$
2	**Policy**	$S \Rightarrow C$	$\{T_{ID}, T_{Step}, ACL\}^{k_{MAC}}_{k_{SYMM_c}}$
3	**Calculation**	$C \Rightarrow S$	$\{T_{ID}, T_{Step}, Prueba\}^{k_{MAC}}_{k_{SYMM_s}}$
4	**Neg_Notification**	$S \Rightarrow C$	$\{T_{ID}, T_{Step}, Detalles\}^{k_{MAC}}_{k_{SYMM_c}}$
4	**Aff_Notification**	$S \Rightarrow C$	$\{T_{ID}, Detalles\}^{k_{MAC}}_{k_{SYMM_c}}$
4	**Resource**	$S \Rightarrow C$	$\{T_{ID}, Recurso\}^{k_{MAC}}_{k_{SYMM_c}}$

El mensaje *Request*, generado por el módulo RM, representa la solicitud de autorización formulada por el cliente. Contiene un identificador de transacción T_{ID}, un identificador

de secuencia dentro de la transacción T_{Step}, un valor $SFlag$ que indica si la solicitud está relacionada con la gestión de flujos de datos, un conjunto de credenciales relacionadas con la solicitud (las cuales pueden servir al controlador para decidir revelar su política) y la solicitud de acceso. Los datos están cifrados (si así se negoció) mediante la clave K_{SYMM_S} y autenticados con K_{MAC}. Todos los mensajes de esta sección están protegidos de la misma forma, por lo que no se volverá a hacer referencia a estas claves.

La respuesta del controlador, generada por el módulo RM, es el mensaje *Policy*. Incluye la lista de control de acceso que protege el recurso, el mismo identificados T_{ID} que aparecía en la solicitud y un valor T_{Step} incrementado en una unidad.

Una vez que el cliente recibe la política, se crea una prueba de autorización que contiene todos los certificados necesarios para formar una cadena de delegación desde la política hasta la clave del solicitante. Dicha prueba se envía al controlador como parte del mensaje *Calculation*. El uso de T_{ID} y T_{Step} es el ya comentado.

El último paso es la respuesta del servidor a la prueba. Si ésta fuera incompleta, el servidor mandaría un mensaje *Neg_ Notification*. Dicho mensaje podría contener los detalles (*Detalles*) de la negativa. Por otro lado, si la prueba fuera correcta, el controlador podría enviar un mensaje *Resource* (si se estaba solicitando acceder a un recurso concreto) o un mensaje *Aff_ Notification* (si la solicitud no lleva implícita la transmisión del recurso sino, por ejemplo, la ejecución remota de una operación sobre el recurso).

Método de distribución push-asserts

Cuando se selecciona el método de distribución *push-asserts* los controladores son responsables de todo el proceso de construcción de pruebas de autorización. Los solicitantes envían la solicitud y todas las credenciales necesarias para el acceso al recurso. Dicho envío de credenciales puede realizarse en demanda, es decir, en función de la información que éstos reciben mediante los mensajes *Policy* del controlador, o bien durante el envío de la solicitud en aquellos casos en los que el controlador no esté dispuesto a revelar ningún dato acerca de su política.

1	**Request**	$C \Rightarrow S$	$\{T_{ID}, T_{Step}, SFlag, Solicitud, [Asserts]^{0..N}\}^{k_{MAC}}_{k_{SYMM_s}}$
2	**Policy**	$S \Rightarrow C$	$\{T_{ID}, T_{Step}, ACL\}^{k_{MAC}}_{k_{SYMM_c}}$
3	**Asserts**	$C \Rightarrow S$	$\{T_{ID}, T_{Step}, [Asserts]^{1..N}\}^{k_{MAC}}_{k_{SYMM_s}}$
4	**Neg_Notification**	$S \Rightarrow C$	$\{T_{ID}, T_{Step}, Detalles\}^{k_{MAC}}_{k_{SYMM_c}}$
4	**Aff_Notification**	$S \Rightarrow C$	$\{T_{ID}, Detalles\}^{k_{MAC}}_{k_{SYMM_c}}$
4	**Resource**	$S \Rightarrow C$	$\{T_{ID}, Recurso\}^{k_{MAC}}_{k_{SYMM_c}}$

La principal diferencia con respecto al método anterior es el envío del mensaje *Asserts* por parte del solicitante tras la solicitud de la política del controlador. Los mensajes 2 y 3 pueden intercambiarse varias veces en función de la táctica de revelación de la política de control de acceso que siga el controlador. Por otro lado, también es posible que dichos mensajes no lleguen a intercambiarse en aquellos casos en los que el controlador no esté dispuesto a aportar ningún tipo de información. Esto implicaría que el solicitante debería

enviar todas las credenciales en el primer mensaje.

Método de distribución pull

En algunos casos, el controlador puede optar por recuperar las credenciales del solicitante a partir de un suministrador de información. Con este método de distribución *pull* el cliente sólo envía un único mensaje *Request* que contiene la solicitud de acceso.

1 **Request** $C \Rightarrow S \quad \{T_{ID}, T_{Step}, SFlag, Solicitud\}_{k_{SYMM_s}}^{k_{MAC}}$

Cuando el controlador recibe el mensaje intenta recuperar las credenciales necesarias para conceder el acceso. Los mensajes de respuesta ARM han sido omitidos por simplicidad.

Gestión de flujos de datos

Como se comentó anteriormente, el mensaje *Request* contiene un valor que indica si un flujo de datos debe ser establecido o cancelado. El valor *start_stream* representa la solicitud de un nuevo flujo de datos (cualquier posible flujo anterior sería cancelado), el valor *stop_stream* se usa para solicitar la finalización del flujo actual y el valor *no_stream* indica que la solicitud no está relacionada con los flujos de datos. El establecimiento de dichos flujos implica la colaboración de los módulos RM, ARM y DSM tal y como muestra la figura 5.6.

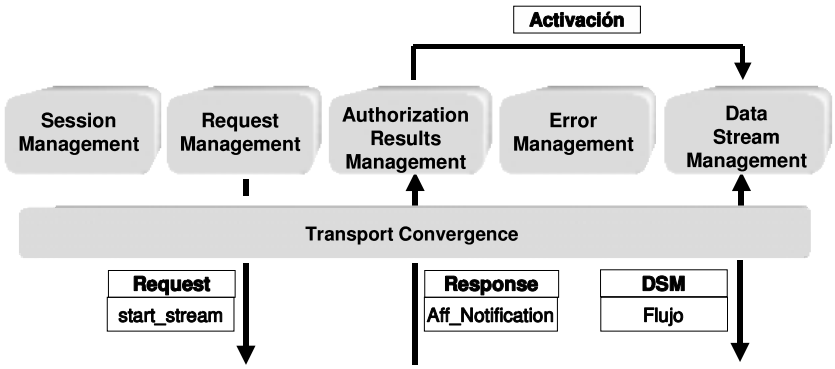

Figura 5.6: Gestión de flujos

Una vez que se establece un flujo, las aplicaciones pueden intercambiar datos libremente, los cuales estarán protegidos por la capa TC si así se negoció.

5.2.5 Análisis de seguridad del protocolo

Tal y como se ha visto en los apartados anteriores, el protocolo presentado ofrece mecanismos para la protección de los datos intercambiados. Esta sección presenta un breve

análisis técnico de la seguridad criptográfica de dicho protocolo que tiene como fin identificar las medidas tomadas contra ataques pasivos y activos bien conocidos. Análisis de seguridad más exhaustivos, por ejemplo empleando lógicas de autenticación [34], están fuera del ámbito de este trabajo.

Análisis del módulo TC

Las claves de sesión que protegen la información (claves de cifrado y de autenticación) son generadas a partir de las cargas aleatorias intercambiadas al principio de la sesión y del secreto compartido $PremasterSecret$. Además, se emplean claves de sesión independientes para cada sentido de la conexión.

Centrándonos en la autenticación, recalcar que se hace uso de las funciones HMAC, las cuales están basadas en resúmenes digitales. Las claves criptográficas utilizadas como parámetro para las funciones HMAC tienen una longitud mínima de 128 bits, lo cual proporciona un nivel de seguridad más que aceptable.

Este método de autenticación constituye también una buena defensa frente a algunos tipos de ataques activos. Uno de los ataques activos más comunes relacionados con el método de cifrado simétrico de bloque CBC es el denominado ataque de *Cut-and-paste* [181]. El ataque está basado en la sustitución de bloques del criptograma actual por bloques de criptogramas anteriores, formando así un nuevo criptograma que al descifrarse contendrá tres tipos de información: información correcta que no ha sido alterada, información falsa del criptograma anterior y datos aleatorios. Sin un método de control de la integridad de los mensajes, el receptor de los mismos podría interpretar como correcta tanto la información aleatoria como la proveniente de criptogramas anteriores. El protocolo AMBAR previene este tipo de ataques al calcular y transmitir siempre un código de autenticación (MAC) del mensaje en claro que va a ser cifrado, código que no puede ser falsificado por un tercero al estar derivado a partir de una clave simétrica de autenticación compartida por cliente y controlador.

Sin embargo, el simple uso de un código de autenticación no es suficiente para detener posibles ataques de reenvío. Este tipo de ataques está basado en el envío por parte de un atacante de mensajes anteriores pertenecientes a la misma sesión que se desea atacar. El mecanismo utilizado por el protocolo AMBAR para evitar este tipo de ataques es la asignación de identificadores únicos a cada uno de los mensajes que se transmite, lo cual permite detectar la recepción de un mensaje reenviado. Dichos identificadores están presentes en cada uno de los mensajes RM, ARM, EM y DSM.

Análisis de los módulos RM y ARM

El módulo RM asigna un identificador único T_{ID} a cada transacción y un identificador de secuencia T_{Step} a cada mensaje intercambiado dentro de cada transacción. De esta forma, mensajes RM anteriores que pudieran ser insertados por un atacante (como por ejemplo la solicitud de un fichero de gran tamaño para provocar situaciones de denegación de servicio) son detectados e ignorados.

El módulo ARM también evita este tipo de ataques mediante la inclusión de números de secuencia en sus mensajes. Por ejemplo, un ataque basado en la transmisión de mensajes *Neg_ Notification* anteriores puede ser detectado examinando el valor de T_{ID}.

Análisis del módulo SM

El diseño de un protocolo de seguro de intercambio de información confidencial es un proceso complejo ya que no es fácil determinar si el protocolo no es susceptible de ningún tipo de ataque conocido. Este análisis se centra en tres tipos de ataques: falsificación de la identidad, alteración de los parámetros negociados y ataques de reenvío.

En relación con la falsificación de la identidad, el protocolo asume la existencia de autoridades de certificación confiables. Se da por supuesto que las entidades comunicantes disponen de los medios necesarios para comprobar la validez de los certificados intercambiados.

Respecto a los parámetros negociados, es relativamente sencillo para un atacante modificar la lista de valores contenidos en los dos primeros mensajes del protocolo, especialmente en lo que respecta a los algoritmos criptográficos, con el fin de forzar la utilización de las opciones más débiles. La sencillez de este tipo de ataque estriba en el hecho de que todos los mensajes transmitidos durante la fase SM antes de *InitSession* no están protegidos. Esta falta de protección hace que un atacante pueda interceptar y modificar los mensajes de negociación. Sin embargo, AMBAR detecta este tipo de ataque mediante la inclusión en el mensaje *InitSession* de un código de autenticación de todos los mensajes SM intercambiados. En el caso de que no se produzca ningún tipo de ataque, tanto cliente como servidor deben obtener el mismo código de autenticación. Por el contrario, si alguno de los mensajes fue alterado durante su trayecto los códigos de autenticación diferirán, lo cual invalidará completamente la negociación.

El último tipo de ataque está relacionado con el reenvío de información anterior. Un atacante podría guardar toda la información enviada por un cliente a un servidor, e instantes después iniciar una nueva comunicación con dicho servidor. Realmente, el atacante se limitaría a reenviar todos los mensajes que ha registrado anteriormente en respuesta a los mensajes del servidor. Incluso teniendo en cuenta que dicho atacante no es capaz de descifrar parte de la información que está enviando, inicialmente logra hacerse pasar por el cliente original con éxito. Sin embargo, el reenvío se detecta tras la recepción del mensaje *ActivateCrypto*, el cual incluye la firma digital del valor $MasterSecret$. Dicho valor está derivado a partir de las cargas aleatorias N_C y N_S, las cuales son distintas en cada ejecución del protocolo, imposibilitando así la reutilización de mensajes anteriores.

5.2.6 Ventajas de AMBAR

El marco AMBAR es capaz de proporcionar los mecanismos básicos de seguridad necesarios para llevar a cabo el intercambio de información de autorización en escenarios de control de acceso. Mediante este marco, es posible negociar los parámetros de autorización más apropiados para cada entorno, optimizar solicitudes de acceso haciendo uso de deci-

siones de autorización anteriores y de información previamente intercambiada, y proteger la integridad de los datos que se están intercambiando. Su uso libera a las aplicaciones de alto nivel de la necesidad de tener que codificar la información relativa a autorización como parte de los datos de alto nivel, lo cual permite aislar claramente el mecanismo de control de acceso del propósito específico de la aplicación.

5.3 DCMS: Sistema de gestión distribuida de credenciales

El segundo componente principal de la infraestructura de autorización que se presenta en este capítulo es el sistema DCMS (Distributed Credential Management System) [48, 49]. Hasta el momento se han detallado tanto las especificaciones referentes a certificados de credencial (ver sección 4.3) como el marco diseñado para poder intercambiar dicha información (ver sección 5.2). Sin embargo, es necesario un paso intermedio que conecte ambos mecanismos, es decir, un sistema capaz de crear y distribuir los certificados de credencial para que éstos puedan ser utilizados por las entidades finales a la hora de acceder a los recursos protegidos. Realizando una analogía con los sistemas de certificación de identidad, nos encontraríamos en una situación en la que, tras definir el formato de certificación X.509 y los protocolos TLS o S/MIME, sería necesario especificar todos los pasos relacionados con la gestión del ciclo de vida de dichos certificados, es decir, con la especificación de una PKI. En el caso concreto que aquí se describe, se ha definido un sistema de gestión del ciclo de vida de certificados SPKI/SDSI. Como se comentó en la sección 4.3.4, SPKI/SDSI supone la alternativa más seria hasta el momento en lo que a autorización basada en certificados se refiere, de ahí que se haya elegido como especificación a utilizar a la hora de construir el sistema.

Si bien hay gran multitud de propuestas que hacen uso de este tipo de certificados a la hora de implementar escenarios de aplicación concretos, como el acceso a objetos distribuidos CORBA [123], el control de acceso a redes WLAN [116], la protección de recursos en entornos de agentes móviles [154] o el WWW [45], la mayoría de estas iniciativas carecen de un sistema genérico de gestión de los certificados. En consecuencia, la forma en la que los usuarios solicitan los certificados de autorización, el medio por el cual se distribuyen, o la política de autorización seguida para tal efecto suele ser dependiente del sistema y está implementada, a menudo, de forma demasiado sencilla y no distribuida. Si bien este enfoque puede funcionar correctamente en determinados escenarios, entornos más complejos pueden sacar a relucir ciertas carencias en materia de escalabilidad o interoperabilidad. La generación y revocación de este tipo de certificados debería realizarse de forma estructurada y completamente distribuida.

La propuesta aquí presentada define cómo deben expresarse las solicitudes de certificación, proporciona mecanismos para satisfacer las distintas políticas de seguridad, identifica las entidades involucradas en un escenario de certificación y qué tipo de colaboración se establece entre ellas. DCMS constituye una aportación muy valiosa a la definición de siste-

mas capaces de proporcionar servicios de autorización a la mayoría de escenarios basados en delegación y roles, independientemente del entorno de aplicación en el cual se encuentren éstos ubicados. Cómo se verá en los siguientes apartados, DCMS se ha centrado principalmente en las operaciones de creación y distribución de certificados y políticas de autorización, si bien puede integrarse fácilmente con otras propuestas existentes en materia de revocación y validación de certificados SPKI [117] o de publicación en repositorios públicos [8, 95].

Esta sección está estructurada de la siguiente manera. En primer lugar se presentará un escenario de autorización genérico con el cual justificar las decisiones de diseño que han dado lugar a la definición de DCMS. A continuación, se presentará la estructura general del sistema, es decir, los principales componentes funcionales que lo componen y su relación. Posteriormente, se proporcionarán los detalles relativos a cada subsistema, en especial los relacionados con el formato de las solicitudes de certificación, políticas de autorización y entidades participantes. Finalmente, se expondrá cómo se integra el marco AMBAR con DCMS a la hora de actuar como mecanismo de intercambio de información de autorización.

5.3.1 Motivación

Con el fin de ilustrar cuáles han sido los criterios de diseño a la hora de construir DCMS, se mostrará a continuación un entorno de control de acceso basado en delegación, roles y certificados de credencial SPKI. El objetivo del estudio de dicho entorno es la extracción de las características comunes a cualquier escenario de control de acceso basado en estos elementos, lo cual nos permitirá determinar cómo debe estructurarse DCMS y qué mecanismos debe ofrecer de cara a proporcionar la máxima escalabilidad, interoperabilidad y adaptabilidad.

Los escenarios de control de acceso basados en el concepto de delegación y en el agrupamiento de usuarios mediante roles presentan una estructura similar a la mostrada en la figura 5.7.

Uso de autoridades de autorización

En estos entornos, los controladores delegan gran parte de su gestión del control de acceso en terceras partes confiables denominadas de forma genérica autoridades de autorización. De esta manera, la determinación de qué usuarios, o grupos de usuarios, están autorizados a acceder a los recursos se realiza de forma distribuida por parte de cada una de dichas autoridades, las cuales actuarán según lo especificado en su política de autorización. Es decir, se considera que una autoridad de autorización puede ser cualquier entidad final del sistema a la cual se le hayan conferido los privilegios de gestión de un conjunto de recursos por parte del controlador de los mismos. El número, la localización o la responsabilidad de cada una de ellas es un factor totalmente dependiente del entorno de aplicación específico, y posiblemente muy dinámico, lo cual es un aspecto a tener en cuenta a la hora de diseñar el sistema DCMS con vistas a ofrecer una solución que abarque todas las posibles configuraciones.

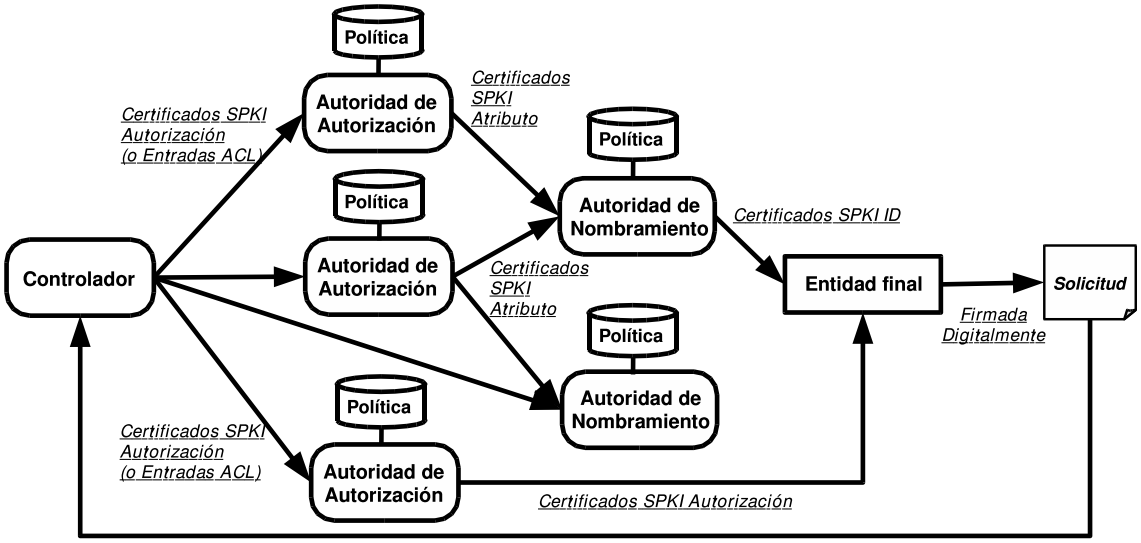

Figura 5.7: Elementos de un entorno de control de acceso basado en delegación y roles

Por otro lado, la forma mediante la cual los controladores pueden especificar esta delegación de la gestión en las autoridades puede variar mucho de un entorno a otro, sobre todo en función de la especificación de certificados de credencial que se esté empleando. En este caso concreto, debido al uso de la especificación SPKI, la delegación es posible plasmarla de dos formas distintas:

- *Mediante certificados de autorización.* Cuando el controlador correspondiente dispone de un par de claves asimétricas, es posible generar un certificado de autorización que tenga al controlador como entidad emisora y a la autoridad de autorización como entidad receptora. El conjunto de recursos que podrán ser administrados de forma descentralizada por la autoridad está contenido en el campo *tag*, siendo dicha gestión efectiva durante el periodo de validez contenido en el documento (salvo revocación). Para que la autoridad pueda actuar como tal, el certificado debe permitir la propagación de los privilegios a otras entidades del sistema.

- *Mediante entradas de una ACL.* En el caso de no disponer de dicho par de claves, el controlador puede especificar la delegación mediante el uso de listas de control de acceso SPKI. Las entradas de una ACL contienen el mismo tipo de información que un certificado de autorización, excepto en lo que respecta al campo del emisor puesto que éste está implícito. La diferencia principal entre ambos mecanismos es que la constatación de la delegación no puede hacerse pública en este último caso, al ser la ACL un documento de uso local que carece de mecanismos de protección de integridad.

Una vez que las autoridades de autorización han obtenido la responsabilidad de gestionar un conjunto de los recursos del sistema, deberán proceder con la asignación de tales

privilegios al conjunto de entidades correspondientes. Dicho conjunto, dependiente totalmente de la autoridad en cuestión, forma parte de lo que se conoce como la política de autorización de dicha autoridad. La política contiene tanto el conjunto de entidades que pueden recibir los privilegios como qué parte de los mismos y durante qué intervalo de tiempo serán asignados. Es decir, la política de autorización puede verse como una sentencia que especifica cuáles son los certificados que la autoridad estará dispuesta a emitir cuando le sean solicitados. Es importante recalcar que aunque la autoridad pueda conocer de antemano los certificados que generará en un futuro, no los emite hasta que las entidades involucradas así lo soliciten. Esto evita que, sobre todo en entornos con gran cantidad de usuarios o recursos que proteger, se produzca una generación desmesurada de certificados de credencial que conlleve a la emisión y distribución de un porcentaje de autorizaciones muy superior al que se va a hacer efectivo frente a los controladores. Como consecuencia, la especificación y el cumplimiento de las políticas de autorización deben ser otros de los mecanismos incluidos en el sistema DCMS.

Por otro lado, se pueden identificar dos tipos de entidades receptoras de los privilegios administrados por una autoridad de autorización. En primer lugar, los privilegios pueden ser asignados a un nombre previamente definido. Este nombre puede hacer referencia a un grupo de usuarios (rol) o bien a un único usuario al cual se le ha asignado un identificador dentro del sistema. La asignación a un nombre de grupo es un mecanismo implícito de redelegación característico de los sistemas basados en roles, ya que cualquier miembro del rol obtiene inmediatamente el privilegio concedido. No obstante, los privilegios también pueden ser asignados directamente a entidades finales, es decir, a claves públicas asociadas a usuarios del sistema. Este enfoque puede emplearse en los casos en los que no se haga uso del concepto de rol, o más genéricamente, cuando no se emplee ningún tipo de identificador de usuarios además de las propias claves criptográficas.

Uso de autoridades de nombramiento

La autoridades encargadas de gestionar la pertenencia a roles se denominan bajo el nombre común de autoridades de nombramiento. Al igual que sucedía con las autoridades de autorización, una autoridad de nombramiento puede estar formada por cualquier entidad final del sistema a la cual se le hayan reconocido los privilegios de gestión de un conjunto de nombres del sistema. Es importante recalcar que dicho conjunto de nombres no tiene porque hacer siempre referencia a nombres de grupo, sino que puede tratarse también de un conjunto de identificadores únicos de usuario, de ahí que se les denomine con el nombre genérico de autoridades de nombramiento. En el caso concreto que aquí nos ocupa, las autoridades reflejan la pertenencia a roles o la asignación de identificadores mediante el uso de certificados de identidad SPKI.

Como puede apreciarse en la figura 5.7, la relación entre los controladores y las autoridades de nombramiento puede ser de dos formas. Por un lado, los roles o identificadores definidos por una autoridad de nombramiento pueden estar referenciados mediante los certificados de atributo, los cuales asocian privilegios a nombres y son emitidos por las autoridades de autorización. En este sentido, quedan autorizados a acceder a los recursos

gestionados por el controlador todos aquellos usuarios que ejercen el rol especificado en dicho certificado. Por otro lado, un rol puede ser autorizado directamente por parte de un controlador mediante una entrada de su ACL o mediante un certificado de atributo emitido por dicho controlador.

Al igual que sucedía con las autoridades de autorización, cada autoridad de nombramiento está regulada por una política, en este caso denominada de nombramiento, que especifica qué elementos del sistema pertenecen a un determinado rol y durante qué periodo. Por elementos del sistema se hace referencia no sólo a entidades finales o claves públicas sino también a otros roles contenidos en uno de mayor nivel, lo cual nos lleva a la definición de sistemas $RBAC_1$ (ver sección 4.2.3).

El papel de las entidades finales

Como se ha comentado, tanto las autoridades de autorización como las de nombramiento definen en sus políticas cuáles serán los criterios a seguir a la hora de emitir nuevos certificados de credencial. Es decir, los certificados son emitidos bajo demanda, sólo cuando las entidades receptoras de los privilegios así lo solicitan a algunas de estas entidades. En consecuencia, además de todos los mecanismos identificados hasta el momento, es necesario dotar al sistema DCMS de las herramientas necesarias para que el proceso de solicitud y distribución pueda llevarse a cabo con éxito.

Dicho proceso abarca tanto la definición de un formato de solicitud de certificación (en este caso, para los tres tipos de certificados SPKI) y un sistema de comunicación entre las entidades solicitantes y las autoridades. Además, y en relación con lo comentado en la sección 4.4.3 acerca de la reducción de cadenas de delegación y anonimato, será necesario proporcionar a los usuarios finales un sistema que permita reducir parte de sus certificados de credencial en aquellos entornos en los que dicha reducción se considere un requisito desde el punto de vista del anonimato o de la eficiencia. Tanto las reducciones como las solicitudes de certificación pueden ser realizadas a través de entidades intermedias denominadas *puntos de acceso*, las cuales introducen ventajas adicionales, tal y como se verá en la sección 5.3.3.

Las entidades finales, una vez que obtienen los certificados correspondientes a partir de las autoridades del sistema, generan solicitudes de acceso a los recursos protegidos por los controladores. Dichas solicitudes deben estar firmadas digitalmente mediante la clave privada asociada a la clave pública contenida en los certificados de credencial. Una vez que esto sucede, tanto las credenciales como la solicitud se envían al controlador para que contraste la veracidad de las mismas y compruebe que existe un camino de delegación desde su propia clave pública (o lista de control de acceso) hasta la clave pública del solicitante. Consecuentemente, la cadena de delegación se valida en el mismo punto en el cual se origina, lo cual es conocido como *bucle de autorización* [17, 33].

La redelegación en claves temporales generadas por las propias entidades finales será posible siempre que los mecanismos de control de la delegación así lo permitan. En el caso concreto de SPKI, esta redelegación será factible siempre que los certificados tengan activado el campo de la propagación. Es importante recalcar que dicha redelegación no

implica a ninguna autoridad, y que puede realizarse de forma totalmente descentralizada por parte de las entidades finales, las cuales determinarán qué parte de sus privilegios propagan a sus claves temporales.

5.3.2 Estructura general de DCMS

DCMS, tal y como muestra la figura 5.8, está divido en dos grandes bloques: el subsistema NMS (Naming Management System) gestiona todos los aspectos relacionados con los certificados de identidad SPKI, es decir, con la pertenencia a roles y la identificación de entidades finales; por otro lado, encontramos el subsistema AMS (Authorization Management System), responsable de la gestión de los certificados de atributo y de autorización SPKI, es decir, de la asignación de privilegios.

Además de estos dos bloques básicos, DCMS dispone de un servicio automático de reducción de autorizaciones (RMS, Reduction Management System), el cual podría encuadrarse dentro del subsistema AMS, aunque será estudiado de forma independiente.

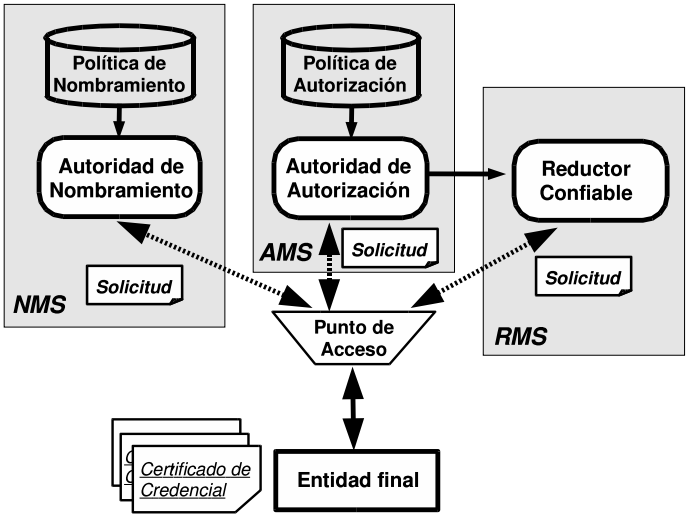

Figura 5.8: Estructura general de DCMS

Cada uno de estos bloques serán analizados siguiendo el mismo enfoque. En primer lugar, se identificarán los elementos que forman parte de su arquitectura. A continuación, se detallará tanto el formato empleado para representar las solicitudes de certificación como para reflejar las políticas de seguridad de las autoridades. Por último, se presentarán varios casos de uso que ilustran el funcionamiento de cada subsistema.

5.3.3 NMS (Naming Management System)

El subsistema NMS es responsable de las operaciones de certificación relacionadas con los certificados de identidad SPKI. Este tipo de certificados se utiliza normalmente para ligar un nombre a una determinada clave pública, así como para definir la pertenencia a grupos.

En relación con lo visto en la sección 5.3.1, un controlador podría tomar la determinación de autorizar el acceso a todos aquellos usuarios de un determinado grupo. En este caso, el sistema NMS será empleado por las entidades finales para obtener un certificado de pertenencia a dicho grupo, el cual es emitido por una autoridad de nombramiento concreta.

Entidades participantes

La figura 5.9 muestra los tres tipos de entidades que forman parte de NMS: solicitantes, puntos de acceso al servicio y autoridades de nombramiento.

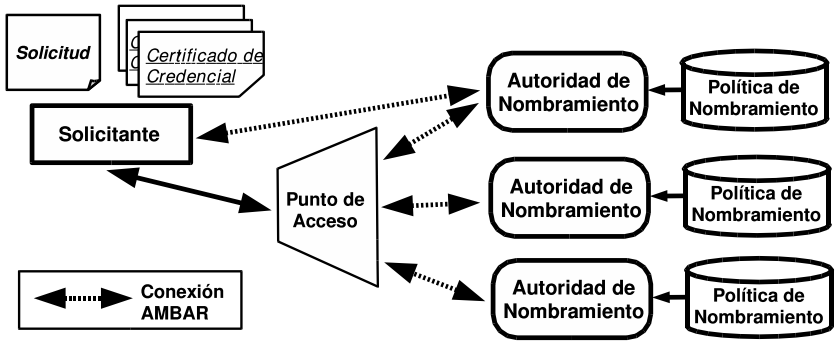

Figura 5.9: Entidades de NMS

- *Solicitante.* Son los usuarios que desean obtener un nuevo certificado de identidad SPKI. Para ello generan una solicitud de certificación y la envían a una autoridad de nombramiento en particular con el fin de obtener el certificado solicitado. Este envío puede realizarse a través de un punto de acceso o bien mediante una conexión directa AMBAR entre solicitante y autoridad. La solicitud podrá estar acompañada de otros certificados con el fin de satisfacer la política de seguridad de la autoridad.

- *Punto de acceso al servicio.* Los solicitantes pueden hacer uso de los puntos de acceso a la hora de enviar sus solicitudes de certificación a las autoridades de nombramiento apropiadas. Si bien los puntos de acceso son elementos opcionales, pueden ser considerados elementos muy útiles para los solicitantes. En primer lugar, pueden encargarse de ocultar la localización concreta de las distintas autoridades, lo cual puede ser conveniente en escenarios con varias autoridades donde resulta complicado averiguar qué autoridad es la indicada para emitir ciertos certificados, especialmente los de pertenencia a grupos. Los puntos de acceso pueden adquirir dicha información a partir de sentencias firmadas digitalmente, similares a las políticas, que contienen información acerca de la estructura del sistema y sus propiedades. De hecho, resulta más sencillo distribuir ese tipo de información al conjunto de puntos de acceso que a todas las entidades del sistema. Por otro lado, su uso puede liberar a los solicitantes de la necesidad de tener que disponer de software AMBAR para comunicar la información de la solicitud. La comunicación entre los solicitantes y los puntos de acceso

es totalmente dependiente del sistema, y podría comprender el uso de conexiones seguras o de terminales públicos.

- *Autoridad de Nombramiento (NA, Naming Authority).* Estas autoridades emiten certificados SPKI de identidad en función de las solicitudes recibidas a través de los puntos de acceso o bien directamente de los solicitantes. Están controladas por políticas de nombramiento concretas que determinan los requisitos mínimos para obtener los certificados, las cuales pueden estar implementadas mediante listas de control de acceso SPKI o mediante otros mecanismos. Cuando una NA recibe una solicitud y los posibles certificados adicionales, ejecuta un algoritmo de descubrimiento de caminos de certificación [66] con el fin de determinar si la solicitud debe aprobarse o no. La comunicación con las autoridades de nombramiento se realiza mediante el marco AMBAR. Como se comentó en la sección 5.2, AMBAR proporciona la funcionalidad necesaria para intercambiar información relativa a autorización. De esta forma, las entidades pueden autenticarse, se procede a la protección de los mensajes y es posible realizar ciertas optimizaciones destinadas a evitar el envío o el cálculo de algunas autorizaciones. La integración entre DCMS y el marco AMBAR se analizará en la sección 5.3.6.

- *Política de Nombramiento.* Se trata de documentos digitales que condicionan la toma de decisiones de las autoridades. Una política de nombramiento establece el conjunto válido de solicitantes de certificados de identidad SPKI. Asimismo, indica tanto las entidades que pueden ser asociadas a un conjunto determinado de roles como la jerarquía que forman estos últimos. Conceptualmente son muy similares a las políticas de PKI que se analizaron en la sección 3.4, en el sentido de que determinan si una solicitud cumple los requisitos necesarios para ser procesada. Sin embargo, veremos en siguientes apartados que ambas difieren tanto en el formato empleado para codificarlas (aquí se emplearán s-expresiones) como en la entidad responsable de su especificación y la metodología empleada para ello.

Formato de las solicitudes

Las solicitudes contienen información acerca del emisor del nombre, el propio nombre, el solicitante y el periodo de validez. El sistema de codificación empleado está basado en s-expresiones [170] ya que no se considera necesario hacer uso de una nueva sintaxis distinta de la utilizada en SPKI (los elementos de información empleados dentro del sistema DCMS se encuentran especificados completamente en el apéndice C). Es importante constatar el hecho de que los elementos de datos contenidos en una solicitud son los mismos que forman parte de un certificado de identidad SPKI, y que por tanto podemos hacer uso de dicha estructura para expresar las solicitudes. Las s-expresiones definidas tienen el formato presentado en la figura 5.10.

- *cert-request.* Identifica la s-expresión como una solicitud de certificación.

```
(cert-request
  (issuer (name NA_i N_i^j))
  (subject P)
  (valid ..)
)
```

Figura 5.10: Solicitudes NMS

- NA_i es la clave pública de la autoridad de nombramiento encargada de generar el certificado solicitado. En este caso, emite certificados para el nombre N_i^j.
- N_i^j. N^j es uno de los nombres definidos en el espacio de nombres de la autoridad NA_i.
- P. Es la entidad que solicita el certificado de identidad. P podría ser:
  - Una clave pública.
  - Un conjunto de entidades referenciado mediante un nombre de grupo, por ejemplo `(name NA N)`.
- *valid*. Hace referencia al periodo de validez durante el cual se solicita la asociación de P al nombre N_i^j . La codificación empleada para dicho periodo sigue el estándar SPKI [68].

En el caso de que la solicitud sea aprobada, se generará un nuevo certificado cuyo emisor será NA_i, P será el *subject*, y N_i^j será el nombre ligado a P, el cual será válido, como máximo, durante el periodo de tiempo especificado.

Las solicitudes de certificación se codifican como secuencias de dos elementos. El primer elemento es la s-expresión que codifica la solicitud y el segundo es la firma digital de la solicitud. Las firmas se codifican empleando la estructura *signature* definida en [68], y se realizan siempre usando la clave privada del solicitante.

Formato de las políticas de nombramiento

Las políticas de nombramiento se codifican utilizando una estructura muy similar a la definida por la especificación SPKI para las listas de control de acceso. Cada una de las entradas de dicha lista especifica las condiciones que deben cumplirse para generar los certificados de identidad correspondientes. El formato de dichas entradas es el mostrado en la figura 5.11.

- R hace referencia al solicitante o conjunto de solicitantes válidos de los certificados especificados en el campo *tag*. R puede ser una clave pública, un nombre `(name NA N)` o incluso un conjunto de claves públicas expresado con la construcción `(* set)`. El campo *subject* de las entradas de la políticas es opcional, lo cual implica que en el caso de que no esté presente se asumirá que el conjunto de solicitantes válidos es

```
(entry
  (subject R)?
  (tag
    (cert-request
      (issuer (name NA_i N_i^j))
      (subject P)
      (valid V_1)
    )
  )
  (valid V_2)?
)
```

Figura 5.11: Políticas NMS

igual al especificado en el campo *subject* de la s-expresión *cert-request* contenida en el campo *tag*.

- El campo *tag* especifica qué entidades pueden recibir los certificados de identidad. Tiene una estructura muy similar a la de las solicitudes NMS, aunque presenta algunas diferencias respecto a ella:

  - N_i^j puede hacer referencia a un conjunto de nombres mediante el uso de las construcciones `(* set)` y `(* prefix)`.
  - P puede hacer referencia a un conjunto de entidades. Hay dos posibilidades a la hora de expresar un conjunto de entidades. Por un lado, es posible utilizar un nombre de grupo, por ejemplo `(name NA N)`. Por otro lado, es posible usar la construcción `(* set)`.
  - V_1 hace referencia al periodo máximo durante el cual el certificado de identidad tendrá vigor.

- El valor V_2 incluido en el campo *valid* indica el periodo de tiempo durante el cual podrá solicitarse la creación de los certificados especificados en el campo *tag*. Dicho campo es opcional, lo que implica que en el caso de que no esté presente se asumirá que el periodo de solicitud es el mismo que el de validez de los certificados, es decir, igual al especificado en el campo *valid* de la s-expresión *cert-request*.

Esta política de nombramiento puede interpretarse como que la entidad (o entidades) R puede solicitar que la entidad (o entidades) P quede ligada a alguno de los nombres contenidos en N_i^j mediante la emisión de un certificado de identidad por parte de la autoridad NA_i. Dicha solicitud podrá realizarse durante el periodo V_2 y el certificado resultante tendrá una vigencia mámima V_1.

Casos de uso

Con el fin de aclarar cómo cooperan las entidades NMS para generar certificados de identidad, en este apartado se analizarán dos solicitudes de certificación. En primer lugar, se expondrá cómo crear certificados de pertenencia a grupos. A continuación, se mostrará cómo puede utilizarse NMS para definir subgrupos o jerarquías de roles. Todos los ejemplos omiten el campo relacionado con los periodos de validez por simplicidad, así como el contenido de las firmas digitales.

Pertenencia a grupos

En este ejemplo, P es una entidad que solicita un certificado de pertenencia al grupo N^j, el cual está gestionado por la autoridad NA_i. Para ello, P formula la siguiente solicitud:

```
(sequence
   (cert-request
     (issuer (name NA_i N_i^j))
     (subject P))
   (signature ..)
)
```

Esta solicitud se envía a NA_i para obtener el certificado correspondiente. Ésta será autorizada sólo en el caso de que NA_i pueda encontrar una cadena de certificación desde su ACL hasta la clave pública del solicitante. En nuestro caso, la política de autorización de la autoridad está expresada mediante la siguiente ACL:

```
(acl
  (entry
    (subject (name NA_l N_l^k))
    (tag (cert-request
      (issuer (name NA_i N_i^j))
      (subject (* set P Q R))
    ))
  )
)
```

Esta ACL especifica que sólo aquellos miembros del grupo N_l^k pueden solicitar un certificado de pertenencia para N_i^j. En el caso de que P, Q o R sean miembros de N_l^k, éstos podrán solicitar su propio certificado. De lo contrario, N_l^k puede ser considerada como una tercera parte confiable autorizada a realizar la solicitud. En este caso se asumirá que P es miembro de N_l^k, lo que conlleva que deba enviar el siguiente certificado para ser autorizado:

```
(cert
```

```
  (issuer (name NA_l N_l^k))
  (subject P)
)
```

Por último, una vez que la autoridad ha comprobado que el usuario está autorizado, se emite el certificado solicitado.

```
(cert
  (issuer (name NA_i N_i^j))
  (subject P)
)
```

Definición de subgrupos

Los subgrupos se crean mediante certificados de identidad en los que el campo *subject* es también un nombre. Hay una diferencia significativa en lo que respecta a este tipo de certificados frente a los de pertenencia a grupos. Un certificado de pertenencia suele ser solicitado por parte de la entidad que desea pertenecer al grupo, pero el certificado de subgrupo no puede ser solicitado por el subgrupo en sí. Los solicitantes válidos son dependientes de la política de nombramiento concreta, aunque algunos candidatos válidos pueden ser la propia autoridad de nombramiento que define el grupo o incluso un miembro del mismo. En el ejemplo que aquí se muestra, el solicitante autorizado es la autoridad de nombramiento, si bien ésta ha delegado dicha autorización en una tercera entidad R con el fin de evitar usar su clave privada para firmar solicitudes de certificación.

Esta es la solicitud enviada por R a NA_i con el fin de definir a N_l^k como subgrupo de N_i^j (está firmada con la clave pública de R):

```
(sequence
   (cert-request
     (issuer (name NA_i N_i^j))
     (subject (name NA_l N_l^k)))
   (signature ..)
)
```

La política especifica que NA_l puede solicitar certificados de identidad para N_i^j, y que además puede delegar dicho privilegio.

```
(acl
  (entry
    (subject NA_l)
    (propagate)
    (tag (cert-request
      (issuer (name NA_i N_i^j))
```

```
      (subject (name NA_l N_l^k))
    ))
  )
)
```

R envía además el siguiente certificado de autorización con el fin de demostrar que NA_l delegó en ella el privilegio de realizar cualquier tipo de solicitud de certificación:

```
(cert
  (issuer NA_l)
  (subject R)
  (tag (cert-request *))
)
```

Finalmente, NA_i utiliza los datos obtenidos a partir de la decisión de autorización para crear el certificado.

```
(cert
  (issuer (name NA_i N_i^j))
  (subject (name NA_l N_l^k))
)
```

5.3.4 AMS (Authorization Management System)

El subsistema AMS es responsable de las operaciones de certificación relacionadas con los certificados de atributo y de autorización SPKI. Como se verá en los siguientes apartados, tiene gran número de similitudes con el subsistema NMS.

Entidades participantes

NMS y AMS están basados prácticamente en los mismos elementos. Tanto los solicitantes como los puntos de acceso forman parte también de AMS, mientras que las autoridades de nombramiento se ven sustituidas por las autoridades de autorización.

Un solicitante AMS es una entidad que pide la generación de un nuevo certificado de atributo o de autorización. Para ello, debe construir una solicitud de certificación con información acerca de los privilegios que desea obtener (los privilegios son totalmente dependientes del entorno de aplicación). En AMS hay dos tipos de solicitantes distintos: por un lado tenemos aquellos usuarios que solicitan un certificado de autorización directa para una determinada clave pública; por otro, están aquellos que solicitan un certificado de atributo para un nombre determinado. Como veremos a continuación, los dos casos se tratan de forma distinta.

Formato de las solicitudes y las políticas

Las s-expresiones utilizadas en AMS para especificar solicitudes de certificación y las políticas de autorización están también basadas en la estructura definida por SPKI para los certificados de autorización y de atributo. Las principales diferencias con respecto a las expresiones de NMS son la presencia del campo *tag* para especificar el permiso concreto que se está solicitando o concediendo y la incorporación de un valor de control de la propagación. En los siguientes casos de uso veremos ejemplos de este tipo de s-expresiones.

Casos de uso de los certificados de autorización

En este primer ejemplo, P es una entidad que solicita un certificado de autorización con el tag tag^A a la autoridad AA_i.

```
(sequence
   (cert-request
     (issuer AA_i)
     (subject P)
     (tag tag^A))
   (signature ..)
)
```

La solicitud se envía a AA_i, y ésta examina su política de autorización para determinar si debe ser aceptada. Dicha política es la siguiente:

```
(acl
  (entry
    (tag (cert-request
      (issuer AA_i)
      (subject (* set P Q))
      (tag tag^B)
    ))
  )
)
```

Esta ACL, al omitir su campo *subject*, especifica que P y Q pueden solicitar un certificado de autorización que conceda los permisos expresados por tag^B (o un subconjunto de ellos). Si suponemos que tag^A es un subconjunto de dichos permisos, el usuario obtendría finalmente el certificado requerido.

```
(cert
  (issuer AA_i)
  (subject P)
```

```
  (tag tag^A)
)
```

Una de las principales ventajas de este enfoque es que es posible especificar un conjunto de privilegios, posiblemente infinito, sin la necesidad de tener que emitir todos los certificados asociados, ya que el conjunto necesario de éstos será emitido bajo demanda. La definición de conjuntos infinitos de certificados viene derivada de la utilización de expresiones basadas en el operador *.

Casos de uso de los certificados de atributo

Los certificados de atributo son especialmente útiles cuando se trabaja con roles, ya que pueden emplearse para especificar los permisos asignados a un determinado rol (recordemos que un certificado de atributo es aquél que contiene como campo *subject* un nombre y no una clave pública). Ahora bien, en relación con la gestión de este tipo de certificados, surge la siguiente duda: "*¿Quién debería ser el solicitante de un certificado de atributo?*". ¿Un usuario del rol al que hace referencia? ¿La autoridad que define el rol? ¿Otra entidad?

Para contestar a esta pregunta debemos considerar que los certificados son emitidos por las autoridades de autorización, y que por tanto los solicitantes válidos serán aquellos que estén reflejados en la política de seguridad de las mismas. DCMS mantiene las políticas inherentes del sistema tan al mínimo como es posible, con el fin de que sean los diseñadores del sistema los que establezcan su propia política. Por tanto, los solicitantes válidos pueden variar desde miembros del propio rol hasta gestores de rol (role managers). Esta última alternativa es muy interesante ya que divide de forma estructurada las tareas de administración entre varias entidades independientes. La forma en la que las autoridades expresan qué *role managers* pueden gestionar cada rol puede verse de la forma siguiente:

$$AA_i \Rightarrow RM_i^1(N_l^k, N_f^g), RM_i^n(N_j^h) \tag{5.1}$$

Esta expresión denota que la autoridad AA_i autoriza al *role manager* RM^1 a solicitar certificados de atributo para el grupo N^k definido por NA_l, y para el grupo N^g definido por NA_f. AA_i además autoriza a RM^n para solicitar certificados de este tipo para el grupo N^h definido por NA_j.

A continuación, vamos a ver cómo puede implementarse esta relación mediante AMS. En este ejemplo, RM_i^1 solicita un certificado de atributo para N_f^g con el privilegio tag^A. La solicitud enviada por RM_i^1 a AA_i es la siguiente.

```
(sequence
   (cert-request
     (issuer AA_i)
     (subject (name NA_f  N_f^g))
     (tag tag^A))
   (signature ..)
```

```
)
```

La política de autorización de la autoridad es la implementación de la expresión §5.1.

```
(acl
  (entry
    (subject RM_i^1)
    (tag (cert-request
      (issuer AA_i)
      (subject (* set
        (name NA_l N_l^k)
        (name NA_f N_f^g)))
      (tag tag^B)))
  )
  (entry
    (subject RM_i^n)
    (tag (cert-request
      (issuer AA_i)
      (subject (name NA_j N_j^h))
      (tag tag^C)))
  )
)
```

Finalmente, la autoridad utiliza los datos obtenidos de la decisión de autorización para crear el certificado solicitado.

```
(cert
  (issuer AA_i)
  (subject (name NA_f N_f^g))
  (tag tag^A)
)
```

5.3.5 Reduction Management System (RMS)

En la sección 4.4.3 se comentó la posibilidad de utilizar la reducción como mecanismo no sólo destinado a gestionar de forma eficiente un conjunto de credenciales sino también a proporcionar un servicio de anonimato a los usuarios del sistema. Se introdujo el concepto de *reductores confiables* como elementos capaces de simplificar un conjunto de certificados de credencial en un único documento que contuviera los privilegios derivados a partir del conjunto de partida.

Dentro de DCMS, se ha definido un sistema automático de reducción de credenciales denominado RMS (Reduction Management System), el cual ofrece la posibilidad de poner en contacto a los usuarios finales del sistema con el conjunto de reductores confiables dispo-

nibles. De esta forma, es posible enviar solicitudes de reducción y obtener los certificados resultantes como parte de los servicios disponibles en DCMS.

Entidades participantes

La principal diferencia entre RMS y los anteriores subsistemas en lo que se refiere a entidades participantes se encuentra en la presencia de los ya citados reductores confiables. Dichos elementos se relacionan con las entidades finales y las autoridades de autorización tal y como se muestra en la figura 5.12.

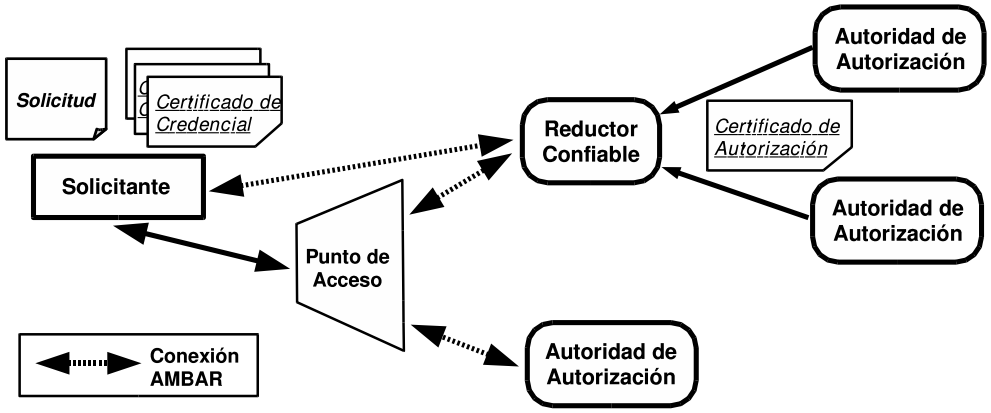

Figura 5.12: Entidades de RMS

Se considera reductor confiable a todo aquel elemento que ha recibido por parte de alguna autoridad de autorización el privilegio de emitir parte de los permisos que ésta gestiona. Dicha concesión se materializa mediante la creación de un certificado de autorización que contiene la siguiente información:

- *Emisor*: clave pública de la autoridad de autorización
- *Receptor*: clave pública del reductor confiable
- *Propagación*: activada
- *Tag*: especificación de los privilegios concretos que pueden reducirse a partir de los certificados originales. Mediante este campo es posible controlar que sólo sean reducidos aquellos privilegios que puedan ser ejercidos de forma anónima.
- *Validez*: periodo durante el cual se pueden realizar las reducciones correspondientes.

Como se muestra en la figura 5.12, los reductores pueden ser entidades confiables para más de una autoridad de autorización, lo cual implica que éstos sean capaces de realizar reducciones relacionadas con conjuntos distintos de recursos. Del mismo modo, las autoridades pueden prescindir de estos elementos a la hora de ofrecer el servicio de reducción,

asumiendo ellas mismas la responsabilidad de ofrecer el servicio de forma directa a los usuarios. El uso, o no, de reductores es totalmente dependiente del entorno de aplicación y de la política de seguridad seguida por cada autoridad, y se debe a cuestiones relacionadas con la seguridad, disponibilidad y eficiencia de dichas autoridades.

Formato de las solicitudes

Una solicitud RMS, tal y como muestra la figura 5.13, debe contener información acerca de la autoridad raíz de la cadena de delegación, el nodo final de la cadena y el conjunto de privilegios que se pretende que estén incluidos en el certificado reducido.

```
(sequence
  (chain-reduction
    (issuer AA_i)
    (subject P)
    (tag tag_R)
  )
  (certificate ...)
  ...
  (certificate ...)
)
```

Figura 5.13: Solicitudes RMS

Mediante esta solicitud, el principal P pide a AA_i la reducción del conjunto de certificados contenidos en la secuencia. El certificado resultante contendrá todos los privilegios contenidos en tag_R que puedan ser derivados a partir de dicho conjunto.

Los certificados a reducir pueden enviarse como parte de la solicitud o bien de forma separada en el campo *Asserts* del marco AMBAR. La elección de una u otra opción es un aspecto totalmente dependiente de la implementación.

Formato de las políticas de reducción

En el caso de los reductores confiables, la política de reducción se obtiene directamente a partir de los certificados recibidos por parte de las autoridades de autorización. La figura 5.14 muestra la política derivada a partir de un certificado de autorización emitido por la entidad AA_i para el conjunto de permisos tag_R con validez dentro del intervalo V.

A diferencia de las políticas de los subsistemas NMS y AMS, el campo *subject* de cada entrada de la política no especifica el conjunto de solicitantes válidos de la reducción sino que hace referencia a la entidad raíz de la cadena a reducir. El sistema RMS exige que los solicitantes de un certificado reducido sean siempre las entidades situadas al final de la cadena de delegación, es decir, los propios receptores de los privilegios.

```
(entry
  (subject AA_i)
  (propagate)
  (tag tag_R)
  (valid V)
)
```

Figura 5.14: Política de reducción

5.3.6 Integración del marco AMBAR y DCMS

El uso del marco AMBAR a la hora de realizar el envío de solicitudes y certificados aporta varias ventajas al sistema DCMS. Frente a protocolos como SSL [9], AMBAR libera a las entidades DCMS de la necesidad de encapsular la información relativa a autorización. Además, la posibilidad de mantener sesiones abiertas entre los puntos de acceso y las autoridades permite realizar optimizaciones en el envío de información entre ambos participantes, así como evitar cálculos de autorización frecuentes.

Es importante recalcar que no resulta conveniente que las autoridades utilicen sus claves privadas de firma para establecer conexiones AMBAR. Constituye una alternativa más correcta que éstas generen pares de claves temporales destinadas a proteger las comunicaciones. Para que dichas claves sean consideradas como válidas por el resto de entidades del sistema, las autoridades deben emitir certificados de autorización que les confieran el privilegio de actuar como su *interfaz de red*. Todos aquellos certificados que contengan el tag `(tag dcms-comm)` serán interpretados, tanto por los puntos de acceso como por los solicitantes, como *cartas de presentación* de las claves destinadas a establecer conexiones AMBAR.

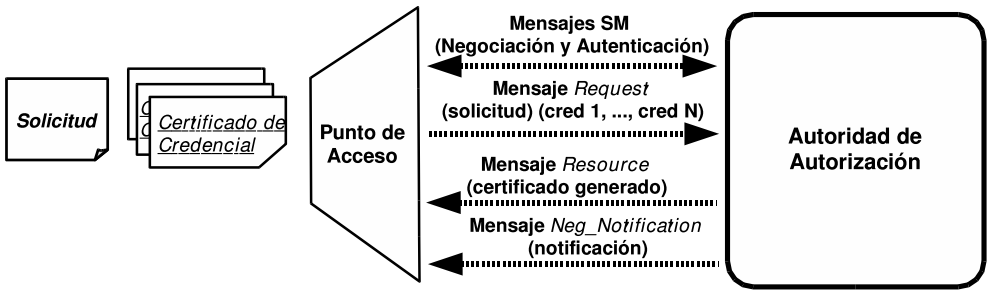

Figura 5.15: Comunicación AMBAR entre punto de acceso y autoridad

La figura 5.15 muestra los detalles de una comunicación AMBAR (en modo *pull*) entre un punto de acceso y una autoridad de nombramiento. Durante la fase SM, ambas entidades son autenticadas y se negocian las opciones de configuración de la comunicación. Las solicitudes y las credenciales se envían como parte del mensaje *Request* (perteneciente al módulo RM). La respuesta puede transmitirse empleando dos tipos de mensajes ARM. En caso de aceptación, el certificado generado se transmite dentro de un mensaje *Resource*,

mientras que en caso de rechazo éste se notifica por medio de un mensaje *Neg_ Notification*. Es importante dejar constancia de que la negociación se realiza sólo una vez por sesión y que conforme aumenta el número de solicitudes es más probable que éstas puedan ser optimizadas haciendo uso de datos anteriores.

Por otro lado, también puede negociarse el método de distribución de las credenciales necesarias para obtener el certificado. No es obligatorio que el solicitante proporcione toda la información necesaria durante el inicio de la transacción (tal y como se muestra en la figura 5.15). Otros modos posibles de distribución son, por ejemplo, el envío de credenciales tras la recepción de parte de la política de autorización de la autoridad (es decir, el envío de mensajes *Asserts* tras la recepción de mensajes *Policy*).

5.4 Metodología para la definición de estructuras de gestión de credenciales

La puesta en marcha de un sistema de control de acceso basado en roles y delegación requiere una identificación muy concisa de los elementos participantes y de la relación entre ellos. Se trata de identificar todos los recursos que se desea proteger, determinar qué acciones realizadas sobre ellos deben controlarse, descubrir cuáles son los roles fundamentales del sistema, la política de pertenencia a dichos roles, el conjunto de privilegios asociados a los mismos e identificar a las entidades encargadas de emitir los certificados correspondientes, entre otras tareas.

Cuando el desarrollo del sistema está condicionado por la utilización de enfoques de gestión estructurados como el seguido en DCMS, resulta apropiado determinar una metodología que permita coordinar tanto a las autoridades como a las entidades solicitantes o receptoras de certificados.

La especificación de metodologías para la construcción de sistemas de control de acceso basados en roles y delegación representa un campo abierto de investigación. Así pues, encontramos en la literatura algunos trabajos muy genéricos que proporcionan enfoques metodológicos de alto nivel, como las alternativas *top-down* y *bottom-up* presentadas en [20], o la identificación de niveles de control realizada en [177].

La metodología introducida en esta sección está también estructurada en niveles de gestión, tal y como muestra la figura 5.16. Como puede apreciarse, la mayor parte de los procedimientos llevados a cabo en esta metodología están organizados en dos bloques funcionalmente distintos, si bien ambos toman como punto de partida la delegación de la gestión en autoridades (nivel 0). Tal y como se verá en las próximas secciones, la determinación de los niveles pertenecientes a los bloques AMS y NMS sigue un enfoque *bottom-up* donde el diseño del sistema se realiza partiendo de los detalles más concretos hasta llegar a las características de alto nivel.

El bloque AMS está compuesto por 4 niveles de gestión distintos: identificación de relaciones entre operaciones, asignación de permisos a entidades receptoras, determinación de solicitantes y periodos de solicitud, y modos de acceso a la autoridad. El objetivo conjunto

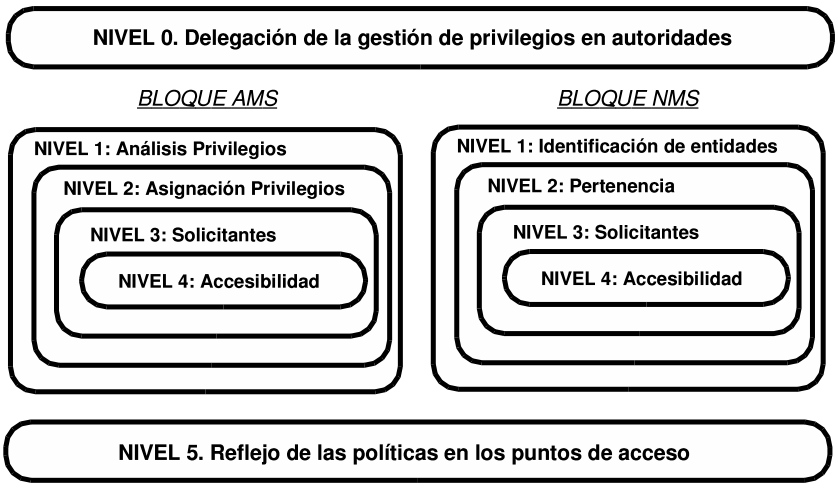

Figura 5.16: Metodología de definición de estructuras de gestión

de los procedimientos de este bloque es proporcionar a las autoridades de autorización un mecanismo para especificar sus políticas de autorización, es decir, para crear las listas de control de acceso que codifiquen dichas condiciones. El bloque será ejecutado por una autoridad siempre que ésta reciba mediante delegación la responsabilidad de asignar un conjunto de permisos relacionados con uno o más recursos del sistema.

Por otro lado, el bloque NMS está compuesto también por 4 niveles de gestión distintos: identificación del conjunto de entidades, determinación de la pertenencia, determinación de solicitantes y periodos de solicitud, y modos de acceso a la autoridad. El objetivo conjunto de los procedimientos de este bloque es proporcionar a las autoridades de nombramiento un mecanismo para especificar sus políticas de nombramiento, es decir, para crear las listas de control de acceso que codifiquen dichas condiciones. El bloque será ejecutado por una autoridad siempre que ésta reciba la responsabilidad de gestionar la pertenencia a un determinado rol del sistema.

Aunque inicialmente los procedimientos asociados tanto a los niveles AMS como NMS serán ejecutados siguiendo el orden impuesto, las continuas revisiones del sistema de control de acceso así como la necesidad de reflejar nuevas situaciones de autorización pueden hacer que la metodología deba emplearse de nuevo partiendo de alguno de los niveles intermedios. En la mayoría de los casos, la ejecución de los procedimientos de un nivel inferior (por ejemplo una nueva asignación de privilegios o la identificación de nuevos recursos a proteger) conllevará la realización de los de nivel superior (por ejemplo cambiar la forma de acceso a las auutoridades si los nuevos privilegios son considerados como críticos dentro del sistema), de ahí que en la figura 5.16 los niveles superiores aparezcan contenidos dentro de los de nivel inferior.

Por último, a partir de las políticas de autorización obtenidas, pueden derivarse los documentos de configuración del sistema que ayudan a los puntos de acceso DCMS a conocer la estructura del mismo. Como ya se comentó en la sección 5.3.3, estos documentos digitales permiten proporcionar a los usuarios finales un servicio de autorización más trans-

parente, a la vez que posibilita la ocultación de autoridades que sólo podrán ser accedidas a través de los correspondientes puntos de acceso.

5.4.1 Delegación de la gestión de privilegios en autoridades

El nivel 0 de la metodología está encargado de la especificación de los procedimientos a seguir para delegar la gestión de privilegios en distintas autoridades. Sus cuatro etapas principales tienen como objetivo determinar qué recursos protegidos por controladores serán gestionados de forma descentralizada por alguna de las autoridades del sistema.

Especificación de los recursos a proteger y sus operaciones

La primera acción a realizar es la especificación de los recursos que se desea proteger. Dicha especificación comprende la definición de un esquema de identificación de recursos, la identificación de las operaciones realizadas sobre dichos recursos, y la publicación de dicha especificación. Frente al punto de vista de algunos autores como [164], los cuales consideran que los recursos deberían ser identificados como objetivos lógicos con el fin de ocultar los detalles físicos de bajo nivel, la especificación que aquí se presenta está enfocada a la identificación de recursos de bajo nivel. Por supuesto, aplicando este mismo enfoque desde un punto de vista de más alto nivel sería posible especificar objetivos lógicos más complejos.

1. *Esquema de identificación de recursos.* Los recursos a proteger en el sistema deben ser identificados de forma que sean únicos dentro del mismo, o al menos dentro del controlador por el cual se encuentran protegidos. Esto posibilita tanto que las credenciales emitidas por las autoridades no sean ambiguas como que los usuarios finales puedan hacer referencia de forma unívoca al elemento al cual quieren acceder. A la hora de definir la notación debe considerarse la posibilidad de representar de forma apropiada los agrupamientos, colecciones o estructuras presentes entre recursos. Por ejemplo, las colecciones de ficheros se encuentran siempre organizadas de forma jerárquica, por lo que el uso de notaciones basadas en prefijos agiliza su gestión.

2. *Especificación de las operaciones sobre los recursos.* Cada tipo de recurso se caracteriza por tener asociadas un conjunto de operaciones que lo involucran. La ejecución de parte de ellas, las que tienen interés desde el punto de vista de esta metodología, debe ser protegida frente a usuarios no autorizados. La especificación de dichas operaciones abarca tanto la determinación de identificadores asociados a las mismas como la identificación de los parámetros relacionados y las posibles limitaciones temporales. Por ejemplo, en el caso de un sistema de ficheros, sería necesario concretar las operaciones a proteger (lectura, escritura, modificación, creación de directorios, etc), los parámetros de dichas operaciones (tamaño máximo de los ficheros, cuota de disco, etc) y las limitaciones temporales que pudieran imponerse (utilización exclusiva durante el horario laboral).

3. *Publicación de la especificación.* Una vez concretada la notación de los recursos y sus operaciones, es decir, la especificación de los permisos a gestionar por el sistema, ésta debe hacerse pública con el fin de que las autoridades de autorización sean capaces de emitir credenciales que conformen con la misma. Es importante recalcar que dicha especificación deber ser la misma tanto para los usuarios solicitantes como para autoridades y controladores, que o bien la emplean directamente o bien deben conocerla para realizar las traducciones pertinentes a su formato de representación interno. La publicación puede realizarse utilizando esquemas XML [31], estructuras ASN.1 [105] o bien mediante módulos software capaces de generar dichos permisos mediante mecanismos de más alto nivel.

Como resultado principal de esta etapa se extrae el formato del campo tag contenido en la s-expresión *cert-request* empleada para codificar tanto las solicitudes como las políticas de autorización.

Determinación de los controladores y los recursos implicados

Una vez que se conoce cómo hacer referencia a los recursos del sistema, el siguiente paso es determinar qué parte de ellos formarán parte del entorno de autorización. Los recursos identificados serán normalmente agrupados por conjuntos, y cada uno de estos conjuntos será asignado a un controlador distinto para que actúe como punto de cumplimiento de la política. La agrupación por controladores presenta varias ventajas: por un lado, recursos que requieren el mismo nivel de seguridad pueden ser protegidos utilizando los mecanismos ofrecidos por un mismo controlador; por otro lado, la visión de conjunto de dichos recursos puede simplificar la gestión de sus privilegios y la notación empleada para codificarlos.

No es un requisito que los controladores se encuentren en línea, ni tampoco que dispongan de su propio par de claves criptográficas. La metodología es igualmente aplicable a escenarios donde, por ejemplo, los controladores de los recursos no dispongan de conectividad, bien por tratarse de dispositivos muy sencillos (control de iluminación, apertura de puertas, etc.) o bien porque el entorno no lo requiera.

Determinación de las autoridades

A continuación, cada controlador (más concretamente la persona encargada de administrarlo) debe determinar cuáles serán las entidades que ejercerán como autoridades gestoras de los permisos involucrados. Dicha determinación es completamente dependiente del sistema, si bien suele hacer referencia a entidades del sistema con cierto peso administrativo, como responsables de administración, jefes de sección o figuras similares.

Junto con la determinación de las autoridades, los controladores deben identificar el conjunto de permisos que delegan en ellas. Dicha identificación implica la delimitación de un subconjunto de los recursos protegidos por el controlador y la selección de una serie de operaciones aplicables sobre los mismos. Una vez planificada esta asignación, sólo queda especificarla mediante alguno de los mecanismos que ya han sido analizados, es decir, mediante certificados de credencial o listas de control de acceso en el caso de SPKI. Es

importante recalcar que la delegación puede realizarse bien directamente sobre una única entidad o sobre un conjunto de elementos en forma de grupo. En el primer caso, nos encontramos con un sistema de gestión basado en autoridades de autorización, las cuales propagan los privilegios mediante certificados SPKI de autorización o de atributo. En el segundo caso, el disfrute de los privilegios está ligado a la pertenencia a cierto grupo, lo que implica que la gestión quede en manos de la autoridad de nombramiento correspondiente. Estas dos vertientes dan lugar a los bloques AMS y NMS, respectivamente, de la metodología.

En el momento en el que un controlador toma la determinación de delegar en cierta entidad la gestión del acceso a sus recursos, dicha entidad se convierte automáticamente (si no lo era ya) en una autoridad dentro de la estructura de gestión. Esto implica que dicha autoridad debe seguir los procedimientos asociados a los distintos niveles que forman parte bien del bloque AMS o del bloque NMS, según corresponda, con el fin de gestionar los privilegios que acaba de recibir.

5.4.2 Procedimientos asociados a las autoridades de autorización

El bloque de procedimientos asociados a las autoridades de autorización tiene como objetivo construir las políticas de autorización de las mismas. Lo especificado en este bloque debe ser seguido por dichas autoridades cada vez que se delega en ellas la gestión de nuevos privilegios o bien se modifican las condiciones de los actuales. Esto puede suceder tras la ejecución de los procedimientos de nivel 0, es decir, tras la identificación de nuevos recursos o controladores, o bien tras la propagación de la gestión desde una autoridad a otra. A continuación se exponen los cometidos de cada uno de los niveles pertenecientes a este bloque.

Nivel 1: Identificación de relaciones entre las operaciones

La especificación de los recursos realizada como parte del nivel de gestión 0 tiene como resultado determinar la notación empleada a la hora de codificar los recursos e identificar el conjunto de operaciones que pueden realizarse sobre los mismos. Entre dicho conjunto de operaciones podemos encontrar relaciones de exclusión, inclusión o agrupamiento que son muy útiles a la hora de asignar bloques de privilegios. Concretamente, este nivel 1 del bloque AMS intenta encontrar:

- Conjuntos de operaciones sobre un mismo recurso consideradas como mutuamente excluyentes. Es decir, operaciones que salvo en caso de control total sobre el recurso no suelen estar ligadas de forma conjunta a una misma entidad, como por ejemplo consultar las calificaciones de un examen y establecerlas.

- Conjuntos de operaciones claramente relacionadas, es decir, operaciones que suelen estar asociadas de forma conjunta a una misma entidad, como por ejemplo fichar para iniciar la jornada laboral y para terminarla.

- Rangos válidos de valores para los parámetros involucrados en las operaciones a controlar. Por ejemplo, la delimitación del número máximo de usos de cierto recurso o el número mínimo de unidades de valor necesarias para tener acceso a un documento digital.

- Intervalos temporales válidos para realizar las operaciones. Con esto no se hace referencia al periodo del tiempo durante el cual se puede disfrutar de los privilegios, es decir, al periodo de validez del certificado de credencial asociado, sino al intervalo durante el cual las operaciones pueden solicitarse. Algunos ejemplos de este tipo de intervalos se encuentran en los escenarios de control de acceso físico, los cuales pueden llegar a controlar el horario durante el cual puede abrirse cierta puerta, o en entornos de control de acceso a una red de comunicaciones en los cuales se contrate el acceso durante una determinada franja horaria.

Nivel 2: Asignación de permisos a entidades receptoras

Una vez delimitados y agrupados los permisos a asignar, el siguiente nivel AMS lo constituye el proceso de identificación y propagación de privilegios a entidades receptoras. Para ello, se llevan a cabo los siguientes pasos:

1. *Identificación de los roles y entidades individuales implicadas.* En primer lugar se determina el conjunto de elementos a los que se les desea asignar las autorizaciones pertinentes. Dichos elementos, los cuales deben encontrarse bajo la gestión de la autoridad, pueden hacer referencia tanto a entidades individuales como a roles específicos.

2. *Determinación de los bloques de permisos asociados a cada receptor.* Una vez determinados los elementos receptores, se establece qué parte de los privilegios gestionados por la autoridad son asignados a los elementos identificados mediante el paso anteriormente explicado.

3. *Control de la propagación.* La autoridad debe determinar si los permisos emitidos pueden ser propagados por parte de las entidades receptoras. En el caso del sistema DCMS, este control es booleano por ser éste el único mecanismo de limitación en la propagación que proporciona SPKI.

4. *Determinación del periodo de validez y método de validación.* Además, debe acotarse el periodo temporal durante el cual podrán disfrutarse los privilegios asignados, siendo dicha limitación muy dependiente del entorno de aplicación en cuestión. Dependiendo de la relevancia de los recursos involucrados, puede establecerse también cuál debe ser el sistema de validación de certificados a emplear.

5. *Determinación de los reductores confiables.* Por último, las autoridades deben determinar si delegan en una tercera entidad (un reductor) la capacidad de simplificar

cadenas de delegación relacionadas con el conjunto de privilegios que se están administrando. Como ya se vio en la sección 4.4.3, dicha delegación puede realizarse mediante certificados de autorización o mediante la extensión de las listas de control de acceso de los controladores. La elección de un mecanismo u otro depende del entorno de aplicación.

Una vez seguidos los procedimientos de los dos primeros niveles del bloque AMS, quedan totalmente especificadas las s-expresiones *cert-request* que aparecerán en las políticas de autorización de la autoridades.

Nivel 3: Determinación de solicitantes y periodos de solicitud

Dos son los pasos principales de este nivel de gestión. Por un lado, la especificación de las entidades que están autorizadas a solicitar los certificados de credencial de la autoridad. Dicho conjunto de entidades solicitantes es especialmente importante en los casos en los que las entidades receptoras hagan referencia a roles específicos (ver sección 5.3.4). En el caso de que sea posible que las propias entidades receptoras actúen como solicitantes (situación muy común durante la asignación a entidades individuales), la especificación de los solicitantes puede omitirse.

Por otro lado, debe determinarse el periodo temporal durante el cual podrá solicitarse la emisión de los certificados. Dicho periodo temporal podría coincidir con el periodo de validez del certificado, aunque no es obligatorio.

Al finalizar con los procedimientos de este nivel 3, las listas de control de acceso que codifican las políticas de autorización quedan totalmente especificadas.

Nivel 4: Modos de acceso a la autoridad

El último nivel del bloque AMS hace referencia al modo mediante el cual la autoridad ofrece sus servicios de emisión de certificados al resto de entidades del sistema. Los siguientes parámetros condicionan dicho modo de acceso:

- *Modo de distribución de credenciales.* Como vimos en la sección 5.2.3, el marco AMBAR proporciona varias alternativas en lo que a distribución de información de autorización se refiere. La autoridad debe determinar cuál es el modo que más satisface sus necesidades, atendiendo a parámetros como la disponibilidad de repositorios públicos de credenciales o la sensibilidad de la información contenida en la política.

- *Acceso anónimo o identificado.* Dependiendo de los privilegios gestionados, la autoridad debe determinar si permite la solicitud anónima de los mismos.

- *Acceso directo o a través de punto de acceso.* Ya se ha comentado que en ciertas situaciones puede resultar conveniente ocultar las autoridades a los solicitantes. Para ello, pueden emplearse los puntos de acceso al sistema con el fin de tener controlado el conjunto de entidades que establecen contacto con las autoridades de autorización.

El conjunto de puntos de acceso autorizados, o bien la determinación de un acceso libre por parte de cualquier entidad, es otra de las decisiones que forman parte de este nivel de gestión.

La figura 5.17 resume la misión de cada uno de los niveles de gestión del bloque AMS. Como puede apreciarse, invirtiendo el orden en el cual se muestran dichos niveles respecto a la figura 5.16 es posible plasmar cuál es el objetivo global de todo el bloque AMS. Del mismo modo, se aprecia el paralelismo entre los objetivos parciales de cada nivel y los elementos de información involucrados en el proceso de gestión (s-expresiones, formato de los tags, entradas de la ACL).

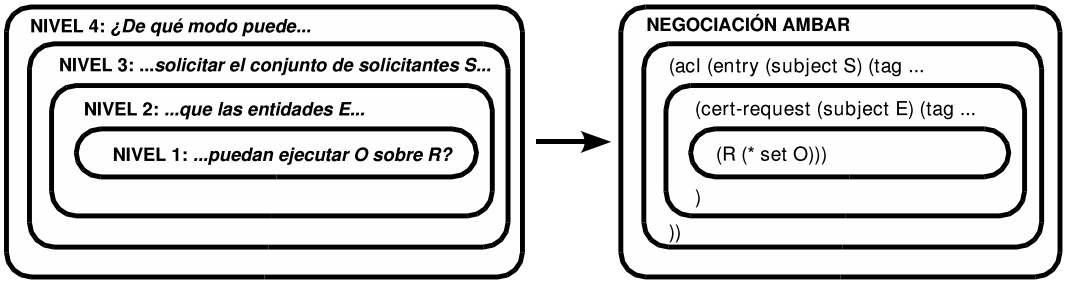

Figura 5.17: Objetivo global del bloque de procedimientos AMS

5.4.3 Procedimientos asociados a las autoridades de nombramiento

El bloque de procedimientos asociados a las autoridades de nombramiento tiene como objetivo construir las políticas de nombramiento asociadas a las mismas. Lo especificado en este bloque debe ser seguido por dichas autoridades cada vez que se les asigna la gestión de un conjunto de identificadores, los cuales suelen hacer referencia a roles concretos dentro de la organización. Esto puede suceder tras la ejecución de los procedimientos de nivel 0, es decir, tras la identificación de nuevos recursos o controladores, o bien tras la identificación de nuevos conjuntos de funciones que puedan ser asociados a un nuevo rol. A continuación se exponen los cometidos de cada uno de los niveles pertenecientes a este bloque.

Nivel 1: Identificación del conjunto de elementos

El primer nivel de gestión hace referencia a la identificación de los elementos que se encuentran dentro del ámbito de la autoridad. Dichos conjunto de elementos está compuesto tanto por entidades individuales como por roles ya definidos dentro del sistema. Otros autores [164] contemplan además la posibilidad de crear agrupaciones de nivel superior a los roles, a las cuales denominan *unidades organizativas*.

Nivel 2: Determinación de los identificadores (pertenencia)

Una vez identificados los elementos a gestionar, el siguiente nivel consiste en la determinación de los identificadores que pueden ser asignados a cada uno de ellos. En el caso de que dichos identificadores se empleen para construir grupos de usuarios (roles), debe determinarse el conjunto de todas las entidades individuales y roles de menor nivel que pertenecen al nuevo rol.

Por otro lado, en el caso de que la autoridad esté realizando una labor de identificación de claves, es decir, de asignación de nombres localmente únicos a cada uno de los elementos identificados, será necesario determinar cuál va a ser la política de asignación a seguir. Tanto la pertenencia como la asignación de nombres son dos procesos totalmente dependientes del entorno de aplicación, lo que implica que no sea posible a este nivel proporcionar detalles más concretos de cómo realizar dichos procedimientos.

Una vez seguidos los procedimientos de los dos primeros niveles del bloque NMS, quedan totalmente especificadas las s-expresiones *cert-request* que aparecerán en las políticas de nombramiento de la autoridades.

Nivel 3: Determinación de solicitantes y periodos de solicitud

Este nivel de gestión es totalmente análogo al nivel 3 del bloque AMS. El mayor énfasis debe ponerse en la identificación de las entidades solicitantes de los certificados de creación de subgrupos (ver sección 5.3.4), donde el conjunto válido de solicitantes no es tan evidente como en el caso de la pertenencia de una entidad final a un rol.

Al igual que sucedía en el bloque AMS, la ejecución de los procedimientos de nivel 3 finaliza la especificación de las listas de control de acceso que codifican las políticas de nombramiento.

Nivel 4: Modos de acceso a la autoridad

El control del acceso a la autoridad debe realizarse siguiendo los mismos criterios que ya fueron especificados para el bloque AMS. Ahora bien, es importante recalcar que, por norma general, las autoridades de nombramiento tienen un carácter más global que las autoridades de autorización. Resulta común a gran cantidad de escenarios de autorización el hecho de que un mismo rol reciba varios privilegios por parte de distintas autoridades. Además, la relación entre usuarios y roles suele ser más dinámica que la existente entre los roles y los permisos, ya que el conjunto de funciones asociadas a un rol suele cambiar menos frecuentemente que el conjunto de usuarios pertenecientes a un determinado rol. En consecuencia, esta mayor dinamicidad y globalidad deben considerarse a la hora de determinar los mejores modos de acceso a las autoridades para cada escenario.

La figura 5.18 resume la misión de cada uno de los niveles de gestión del bloque NMS. Al igual que sucedía con el bloque AMS, invirtiendo el orden en el cual se muestran dichos niveles respecto a la figura 5.16 es posible plasmar cuál es el objetivo global de todo el bloque. Del mismo modo, se aprecia el paralelismo entre los objetivos parciales de cada

nivel y los elementos de información involucrados en el proceso de gestión (s-expresiones, identificadores, entradas de la ACL).

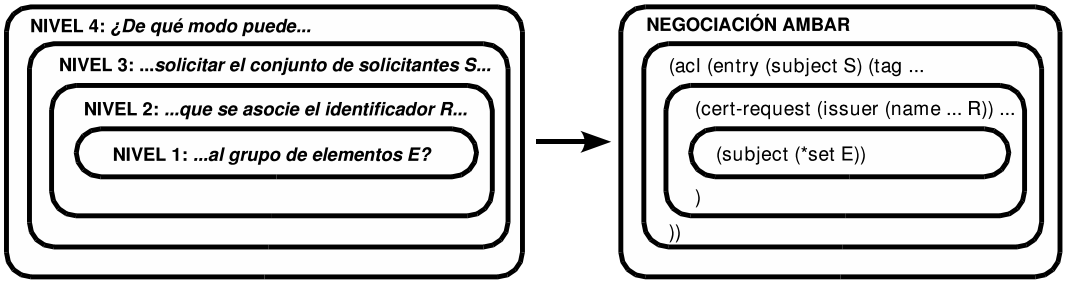

Figura 5.18: Objetivo global del bloque de procedimientos NMS

5.4.4 Reflejo de las estructuras de gestión en los puntos de acceso

Los puntos de acceso pueden operar de dos formas distintas. En primer lugar, pueden actuar como simples interfaces de comunicación entre los solicitantes y las autoridades, es decir, como software de comunicación AMBAR que evita a los usuarios disponer de una implementación del marco y que a la vez controla las direcciones desde las cuales pueden establecerse conexiones con las autoridades. En este modo de operación los puntos de acceso no necesitan disponer de ningún tipo de información especial acerca de las autoridades además de su dirección de red y sus parámetros de funcionamiento del marco AMBAR.

La otra posibilidad de funcionamiento implica la prestación de un servicio adicional de orientación a los solicitantes. Entre las ventajas que se pueden introducir encontramos:

- *Encaminamiento automático de solicitudes a las autoridades apropiadas.* El punto de acceso puede orientar al usuario acerca de la autoridad encargada de emitir los certificados relacionados con cierto tipo de privilegios o con la pertenencia a cierto grupo. De esta forma, a partir de datos como el privilegio o el rol solicitado puede completarse la especificación del emisor de la credencial.

- *Omisión de la solicitud de permisos.* Las peticiones pueden ser abortadas directamente en el punto de acceso en aquellos casos en los que los privilegios solicitados no estén siendo emitidos por ninguna autoridad o bien su validez haya expirado.

- *Comprobación de solicitudes creadas fuera del punto de acceso.* En el caso de que los solicitantes acudan con peticiones ya creadas, es decir, firmadas digitalmente utilizando una aplicación externa, es posible comprobar si los datos incluidos en dicha solicitud son correctos. Por ejemplo, puede verificarse si la entidad especificada como emisora tiene en realidad competencia para gestionar los privilegios solicitados o si éstos han caducado.

Con el fin de proporcionar estos mecanismos adicionales de orientación, los puntos de acceso deben conocer parte de la información contenida en las políticas de autorización y nombramiento de las autoridades DCMS. Tal y como se comentó en la sección 5.3.3, dicho conocimiento podría estar basado en la utilización de políticas similares a las comentadas para el caso de la PKI en la sección 3.4.

El documento que contiene las políticas de las autoridades debe incluir aquella parte de la información que no se considere sensible. Para el caso de las autoridades de autorización, se tratará del conjunto de privilegios gestionados y el intervalo de tiempo durante el cual tendrán vigor. Se omite así la información considerada como confidencial, por ejemplo el conjunto de entidades receptoras o la posibilidad de propagar dichos privilegios. En el caso de las autoridades de nombramiento, la información publicada corresponderá con el conjunto de identificadores que gestionan.

Así pues, durante el proceso de registro de una autoridad en un punto de acceso, cuestión que es totalmente dependiente de la implementación concreta del sistema DCMS, se procede a la especificación de los parámetros de comunicación AMBAR a emplear con dicha autoridad y, opcionalmente, a la provisión de un documento que contenga el conjunto de credenciales gestionados por dicha autoridad. Dicho documento deberá ser actualizado con cada cambio en la política de la autoridad.

5.5 Conclusiones

La infraestructura presentada proporciona un amplio abanico de servicios en lo que a gestión del ciclo de vida y uso de los certificados de credencial SPKI se refiere. Se ha visto cómo, partiendo de una PKI de identidad que proporcione las funciones básicas de gestión de claves de usuarios, es posible derivar un sistema que asigne privilegios a dichas claves y que permita distribuir la información a los controladores que protegen recursos sensibles.

La gestión del ciclo de vida llevada a cabo mediante DCMS proporciona un tratamiento completo a todas las cuestiones relacionadas con la certificación de privilegios. Sus cualidades más relevantes son:

- División del problema en 3 bloques conceptuales principales: gestión de la pertenencia a roles y su jerarquía, gestión de la asignación de privilegios a entidades finales o conjuntos de entidades, y gestión de la reducción de autorizaciones.

- Su diseño totalmente descentralizado y basado en delegación permite adaptarlo correctamente a escenarios en los que la gestión de los privilegios se realiza por parte de entidades con escasa conexión entre sí.

- Las políticas inherentes del sistema son mínimas. Es posible especificar entidades solicitantes distintas a las receptoras de las credenciales, periodos de solicitud distintos a los de disfrute, delegar la posibilidad de solicitud en terceras partes confiables, establecer entidades reductoras distintas de las autoridades raíz, etc.

- Los formatos de las solicitudes y de las políticas están basados en s-expresiones, sin que haya necesidad de introducir nuevos formatos de codificación distintos a los empleados por los controladores a la hora de proteger sus recursos. Además, esto proporciona cierta interoperabilidad al sistema ya que es capaz de ser ejecutado en cualquier plataforma que proporcione soporte a la especificación SPKI.

- Mediante las políticas de autorización es posible especificar conjuntos, posiblemente infinitos, de privilegios que pueden ser asociados a las entidades del sistema sin necesidad de tener que emitir los certificados previamente. El hecho de que la emisión se realice bajo demanda permite solventar algunos de los problemas de escalabilidad presentes en escenarios complejos.

- El mecanismo de reducción automática de certificados, junto con el uso de claves temporales, permite eliminar el enlace que pudiera existir entre la identidad de un usuario y sus privilegios. De esta forma, en escenarios en los que el anonimato está permitido o es un requisito, es posible ocultar la relación existente entre los certificados generados por la PKI y los del sistema DCMS. Adicionalmente, el servicio de reducción introduce mejoras en la eficiencia del tratamiento global de las autorizaciones ya que permite simplificar cadenas largas de certificación.

Por otro lado, el diseño del marco AMBAR se ha realizado considerando la diversidad de escenarios de control de acceso en los cuales resulta necesario llevar a cabo un proceso de distribución de información relativa a autorización. Entre sus características más relevantes encontramos:

- Un mecanismo de negociación de los parámetros de autorización. De esta forma, es posible adaptar el marco a escenarios con distintos requisitos de seguridad en lo que se refiere a confidencialidad, especificaciones a utilizar, anonimato, revelación de políticas y entidades responsables del cálculo de autorizaciones.

- Implementación de técnicas de optimización de sesiones, tanto en lo que respecta a la transmisión de información como al cálculo de autorizaciones.

- AMBAR se integra correctamente dentro del sistema DCMS, proporcionándole a este último un mecanismo de comunicación entre elementos que permite realizar el intercambio de solicitudes de certificación, políticas de seguridad y certificados de credencial.

Finalmente, la metodología de definición de estructuras de gestión presenta un enfoque estructurado que permite construir de forma ordenada sistemas de autorización basados en DCMS. Entre sus propiedades encontramos:

- Se proporcionan procedimientos que permiten la identificación y definición de las operaciones relacionadas con los recursos a proteger. De esta forma, es posible abordar de forma estructurada la definición de los privilegios del sistema.

- Se presentan conjuntos de procedimientos específicos tanto para la gestión de la pertenencia a roles como para la asignación de privilegios a los mismos. Por tanto, es posible abordar el diseño de ambos aspectos de forma independiente, lo que permite dividir las tareas de puesta en marcha o modificación del sistema.

- Los procedimientos se estructuran en niveles de gestión, identificándose además las dependencias entre dichos niveles con el fin de extraer cuáles son las implicaciones de la aplicación de dichos procedimientos.

- Existen una correlación clara entre los pasos de la metodología y los elementos de información del sistema DCMS que se van generando como consecuencia de su aplicación. Esto implica que la definición de las políticas pueda realizarse de forma ordenada, permitiendo la modificación de cualquiera de sus elementos sin más que aplicar los procedimientos relacionados.

- Se distingue claramente entre las políticas de emisión de credenciales, también denominadas políticas de autorización y de nombramiento, y las políticas relacionadas con el reflejo de las estructuras de gestión en los puntos de acceso. En consecuencia, no sólo se proporcionan los medios para especificar el comportamiento de las autoridades sino que también es posible plasmar cuál va a ser la dinámica del sistema, es decir, cómo se van a producir las relaciones entre los distintos elementos participantes.

El capítulo siguiente demostrará las posibilidades reales de la infraestructura a la hora de ser integrada en escenarios de aplicación que abarcan tanto el control de acceso físico a recintos como la subscripción electrónica basada en pagos electrónicos. Se analizará además cómo se ha implementado tanto el marco AMBAR como el sistema DCMS.

Capítulo 6

Implementación de la infraestructura de autorización

El estudio realizado en el capítulo anterior acerca de los componentes de la infraestructura de autorización se centró principalmente en el proceso de diseño. Con el fin de comprobar la viabilidad de dichas propuestas, el capítulo que aquí nos ocupa se centrará en todos aquellos aspectos relacionados con la implementación e integración de la infraestructura en escenarios de aplicación concretos.

Dado que la mayor parte de los desarrollos que se plantearán están basados en la arquitectura CDSA (Common Data Security Architecture) [87], en primer lugar se realizará una introducción de las principales características de dicha arquitectura, introducción que estará enfocada principalmente a identificar los principales módulos de los que se ha hecho uso así como las limitaciones que se han encontrado a lo largo de la evolución de las implementaciones.

Posteriormente, se expondrá cómo se ha desarrollado tanto el marco AMBAR como el sistema DCMS. En ambos casos, el objetivo principal es presentar las librerías de programación y aplicaciones relacionadas, especialmente desde el punto de vista de su uso e integración con otros proyectos.

El siguiente paso consistirá en la descripción de dos escenarios de aplicación concretos en los cuales tanto la PKI basada en X.509 como el sistema DCMS se han utilizado para proporcionar mecanismos de autenticación y autorización. En concreto se trata de un entorno de control de acceso físico y un sistema de suscripción electrónica basada en un protocolo seguro de pagos.

Por último se realizará un análisis del rendimiento ofrecido tanto por la implementación de AMBAR como de DCMS. Dicho análisis tiene como objetivo extraer conclusiones acerca de la sobrecarga que pueden llegar a introducir dichos elementos dentro de cualquier escenario de autorización.

6.1 La arquitectura CDSA

En lo que respecta a arquitecturas de seguridad software, o también denominado *middleware* de seguridad, existen actualmente dos propuestas principales que intentan aglutinar los distintos algoritmos, protocolos y estándares de representación de información en lo que a criptografía y certificación digital se refiere.

Por un lado, el lenguaje de programación Java posee su propia arquitectura denominada JCA (Java Cryptography Architecture) [112], la cual está basada en la definición de un conjunto de interfaces y clases abstractas Java que tienen como objetivo definir una API (Application Programming Interface) de acceso a servicios de seguridad. La API JCA oculta a los programadores de aplicaciones de seguridad los detalles concretos de utilización de cada algoritmo criptográfico, elemento de información (certificados, solicitudes, listas de certificados revocados) o protocolo. Aunque la implementación concreta de cada técnica criptográfica o protocolo no forma parte de la propia arquitectura, sí se define cuál es la interfaz que deben cumplir los distintos proveedores de servicios criptográficos a la hora de poder insertar dicha implementación dentro del sistema.

Actualmente, existen multitud de proveedores que ofrecen conjuntos más o menos amplios de algoritmos y protocolos accesibles a través de la JCA, entre ellos IAIK [78], JCSI [55] o Cryptix [128]. Sin embargo, ninguno de estos proveedores proporciona soporte para las especificaciones de certificados de credencial analizadas en la sección 4.3, a excepción de IAIK que incluye un subconjunto muy reducido de las características de los certificados de atributo X.509. La especificación SPKI tiene asociadas varias implementaciones realizadas en este lenguaje de programación [147, 152, 161], todas ellas caracterizadas por ser totalmente independientes de la arquitectura JCA y carecer de soporte actualmente.

Junto con la falta de proveedores de certificados de credencial, otra de las carencias de JCA es la ausencia de características relacionadas con mecanismos de toma decisiones de autorización y de gestión de políticas. Al no ser parte de la arquitectura, dicha funcionalidad debe ser implementada de forma totalmente independiente, quizá por las propias aplicaciones finales, lo cual acaba propiciando que el sistema final derive en una composición no estructurada de elementos.

Por el contrario, la otra propuesta relevante en lo relativo a middleware de seguridad, la arquitectura CDSA, constituye un enfoque mucho más estructurado y ambicioso que viene a suplir la mayor parte de las carencias enunciadas anteriormente. Dado que todas las implementaciones han sido realizadas haciendo uso de CDSA, en los próximos apartados se introducirán los detalles más importantes de dicha arquitectura.

6.1.1 Visión general de la arquitectura CDSA

La arquitectura CDSA (Common Data Security Architecture) [87, 88] es un conjunto estructurado de servicios e interfaces de programación, expresadas en lenguaje C, encargado de proporcionar diversos mecanismos de seguridad a aplicaciones finales. Las capas inferiores de la arquitectura aportan los componentes fundamentales, tales como algoritmos criptográficos o generadores de números pseudo-aleatorios. Sobre dichos componentes se

construyen módulos relacionados con la implementación de certificados digitales, mecanismos de gestión de claves, certificados de credencial, gestión de políticas o almacenamiento de material criptográfico. Finalmente, en los niveles superiores se desarrollan los protocolos de seguridad de más alto nivel.

El diseño de la arquitectura sigue cinco principios estructurales:

- *Modelo de proveedores de servicio basado en niveles.* CDSA está constituida como un conjunto de niveles horizontales, los cuales proporcionan servicio a los niveles superiores.

- *Proceso abierto de diseño.* Es una propuesta abierta de Open Group [86], y como tal está expuesta a un proceso público de revisión y contribución.

- *Modularidad y extensibilidad.* Los componentes de cada nivel pueden utilizarse como si de módulos independientes se tratara. La arquitectura proporciona un marco mediante el cual introducir nueva funcionalidad o nuevas implementaciones más eficientes de servicios ya existentes.

- *Transparencia de cara al programador.* CDSA gestiona de forma interna la mayor parte de los detalles de seguridad relacionados con los servicios ofrecidos a las aplicaciones. De esta forma, los programadores de aplicaciones de alto nivel no necesitan conocer los pormenores de cada uno de los módulos que aportan funcionalidad al conjunto de la arquitectura.

- *Incorporación de tecnologías emergentes.* El diseño de la arquitectura está continuamente sujeto a revisiones motivadas por el surgimiento de nuevas tecnologías de seguridad a incorporar.

La figura 6.1 muestra los tres niveles básicos de CDSA: servicios de seguridad del sistema, gestor de servicios de seguridad comunes (CSSM, Common Security Services Manager) y módulos de seguridad incorporables (add-in).

El gestor CSSM es el núcleo de CDSA. CSSM gestiona cada una de las clases de servicios de seguridad ofrecidos así como las distintas implementaciones de dichos servicios mediante módulos *add-in*. En concreto, se encarga de la definición de la interfaz de programación (API, Application Programming Interface), la definición de la interfaz de proveedores de servicio (SPI, Service Provider Interface), y de la carga y ejecución dinámica de componentes. Respecto a esto último es importante recalcar que CSSM realiza un control de la integridad de todos los módulos de menor nivel cada vez que debe ejecutarse alguna de las funciones contenidas en dichos módulos. El proceso está basado en la verificación de la firma digital del código y su ejecución dentro de una base de computación confiable (TCB, Trusted Computing Base).

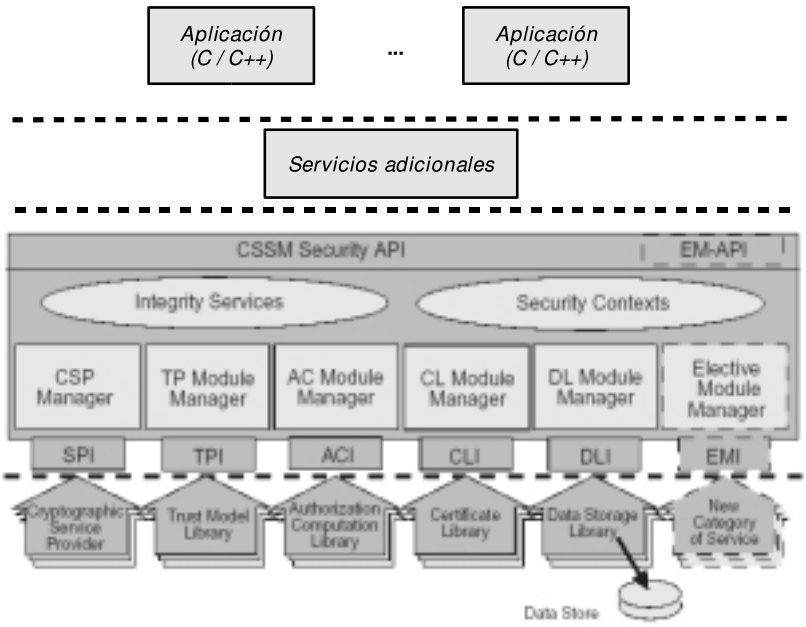

Figura 6.1: Arquitectura CDSA

6.1.2 Módulos de seguridad principales

En CDSA, todas las operaciones de seguridad deben ser realizadas finalmente por alguno de los módulos *add-in* instalados en el sistema. Dichos módulos se encuentran agrupados en cinco categorías de servicio:

- *Proveedores de servicios criptográficos.* Son módulos diseñados para realizar operaciones criptográficas y para almacenar de forma segura claves privadas. Un CSP puede implementar operaciones de criptografía simétrica y asimétrica, firma digital, resumen digital o generación e importación de claves.

- *Módulos de políticas de confianza.* Este tipo de módulos implementan las políticas definidas por las autoridades o instituciones. Dentro de CDSA, las políticas definen el nivel de confianza necesario para que ciertas acciones relacionadas con la certificación puedan llevarse a cabo.

- *Módulos de manejo de certificados.* Estos módulos implementan la manipulación sintáctica de los certificados y CRLs contenidos en memoria. Cada implementación de este tipo de módulos debe ofrecer las operaciones relacionadas con el tipo concreto de certificado al cual dé soporte, sea este X.509, SDSI, SPKI o cualquier otro. Las operaciones más comunes de manejo de certificados son la firma y verificación, la extracción de valores contenidos en campos concretos y la gestión de una CRL.

- *Módulos de almacenamiento de datos.* Este tipo de módulos ofrece operaciones para el almacenamiento persistente de información de seguridad. Dicha información pueden

ser certificados, listas de certificados revocados, claves criptográficas, credenciales, políticas u objetos dependientes de la aplicación.

- *Módulos de toma de decisiones de autorización.* Este módulo toma decisiones de autorización basadas en la política de seguridad del sistema evaluador, la solicitud cuya autorización se está evaluando y las credenciales presentadas por el solicitante.

Cada instancia o implementación de dichos módulos se instala en la arquitectura haciendo uso del servicio de directorio de módulos (MDS, Module Directory Service). Dicho servicio registra el identificador del módulo, una descripción de los servicios que éste proporciona y la información necesaria para cargar de forma dinámica el módulo. De esta manera, las aplicaciones pueden utilizar MDS para consultar qué módulo ofrece alguna de las funcionalidades necesarias para su ejecución.

6.1.3 Integración de CDSA en los desarrollos presentados

La implementación más completa de la arquitectura CDSA ha sido realizada por Intel [56]. Se trata de un proyecto de código abierto [182], periódicamente actualizado, que incorpora un gran número de módulos *add-in* para cada uno de los tipos de servicio ofrecidos por la arquitectura. Entre dichos módulos encontramos:

- *Proveedores de servicios criptográficos.* Hay disponibles dos proveedores distintos, uno de ellos basado en la librería criptográfica OpenSSL [167] y otro en BSAFE [180].

- *Módulos de manejo de certificados.* Tanto X.509 como SPKI disponen de su propio módulo CL (Certificate Library) para la construcción y uso de este tipo de certificados.

- *Módulos de gestión de confianza.* El único módulo incorporado implementa funciones de validación de certificados X.509.

- *Módulos de almacenamiento de datos.* La distribución proporciona una simulación de una base de datos implementada mediante ficheros (flat files).

- *Módulos de toma de decisiones de autorización.* La librería incluye un motor de cálculo de cadenas de certificación basado en la representación de tuplas comentada en la sección 4.3.4. Junto con el motor están disponibles funciones encargadas de realizar la conversión de certificados SPKI a tuplas.

Funcionalidad utilizada en los desarrollos

En los desarrollos que se describen en este capítulo se ha empleado un subconjunto de los módulos anteriormente enunciados. A continuación, se incluye una relación de dichos módulos y de los componentes de la infraestructura que han hecho uso de ellos:

- *Proveedor de servicios criptográficos basado en OpenSSL.* Las rutinas criptográficas han sido utilizadas de forma directa para la implementación del protocolo AMBAR (ver sección 6.2) y para los mecanismos de gestión de claves de las autoridades y claves temporales de usuario del sistema DCMS (ver sección 6.3).

- *Módulo de manejo de certificados X.509.* Este módulo se emplea tanto en la implementación de la fase de negociación SM del protocolo AMBAR como en algunas de las aplicaciones que forman parte del sistema DCMS.

- *Módulo de manejo de certificados SPKI.* Se utiliza en las implementaciones tanto de las aplicaciones de los solicitantes DCMS como de las autoridades de autorización y nombramiento.

- *Módulo de gestión de confianza.* Se emplea para la validación de certificados X.509 realizada durante la fase de negociación de AMBAR.

- *Módulo de toma de decisiones AuthCompute.* El motor de cálculo de autorizaciones se utiliza tanto en el proceso de optimización de solicitudes llevado a cabo por el módulo RM de AMBAR como en la validación de solicitudes de certificación realizada por las autoridades DCMS.

Como se verá en la sección 6.4, dentro del grupo de investigación ANTS se ha realizado también una implementación propia de un proveedor de servicios criptográficos y un módulo de almacenamiento de datos que permite integrar el uso de tarjetas inteligentes dentro de CDSA.

Limitaciones de CDSA

Durante el proceso de desarrollo de la infraestructura se identificaron varias carencias en la implementación de CDSA. Algunas de estas limitaciones eran propias del diseño de CDSA y otras formaban parte de los desarrollos llevados a cabo por Intel. A continuación se presenta una relación de las más importantes y se indica en qué estado se encuentra la corrección de las mismas.

- *Limitación en la generación de certificados.* La versión 3.12 de la implementación de CDSA desarrollada por Intel sólo proporcionaba mecanismos para generar certificados SPKI de autorización. En consecuencia, no era posible generar certificados de pertenencia a grupos (certificados de identidad) ni certificados de asignación de privilegios a nombres (certificados de atributo). Pocos meses después de la notificación que llevamos a cabo acerca del problema a los responsables del núcleo SPKI de CDSA se publicó la versión 3.14, la cual soportaba los tres tipos de certificados.

- *Error en el motor de toma de decisiones.* En Junio de 2002 identificamos un error bastante grave en el funcionamiento del módulo de cálculo de autorizaciones *AuthCompute*. El error permite a un usuario obtener privilegios para realizar cualquier

operación sobre un recurso determinado siempre que demuestre que ha sido autorizado a realizar al menos dos de ellas. A pesar de la notificación del error al equipo de CDSA, en el momento de la redacción de este trabajo, la deficiencia sigue sin subsanarse.

- *Limitación en la gestión de la confianza.* Pese a contar con un servicio de políticas de confianza, a la hora de integrar la gestión y el cumplimiento de los distintos tipos de políticas que forman parte de la infraestructura de autorización, se descubrió que la API correspondiente este tipo de módulos está demasiado enfocada a las operaciones de validación de certificados y listas de certificados revocados. En consecuencia, resulta prácticamente imposible integrar como parte de la arquitectura operaciones tan comunes como la verificación de solicitudes de certificación o la inserción y modificación de políticas. Sería aconsejable que se produjera un replanteamiento de la interfaz y los servicios a ofrecer por este tipo de módulos con el fin de poder adaptarlos a las nuevas tendencias relacionadas con políticas de seguridad.

A pesar de todo, CDSA puede considerarse el mejor *middleware* de seguridad existente para desarrollar servicios de autorización ya que proporciona un tratamiento integral a todos los aspectos que, de una forma u otra, se encuentran relacionados con este servicio. Además, su modelo basado en módulos extensibles favorece la incorporación de nuevas funcionalidades a la infraestructura con un número de modificaciones mucho menor del que se requeriría con otros enfoques menos estructurados.

6.2 Implementación del marco AMBAR

El marco AMBAR, y más concretamente el protocolo del mismo nombre, ha sido implementado como un servicio de seguridad basado en la arquitectura CDSA (versión 3.14 de Intel para Linux). Se trata de una librería de programación, desarrollada en lenguaje C++, que proporciona una interfaz mediante la cual es posible crear sesiones AMBAR tanto de cliente como de servidor.

El mecanismo de sesiones oculta los detalles relativos a la arquitectura interna del protocolo, sin por ello dejar de ofrecer toda la funcionalidad de cada uno de los módulos del marco. Del mismo modo, los programadores que hagan uso de la librería no necesitan tampoco estar familiarizados con los detalles de CDSA ya que todos los aspectos relativos a la criptografía y manejo de certificados forma parte de la implementación interna del protocolo.

Así pues, desde el punto de vista del programador de aplicaciones de alto nivel, sólo es necesario conocer dos aspectos de la implementación: por un lado, la API de acceso al protocolo, compuesta principalmente por los métodos relacionados con las sesiones AMBAR y el contexto de comunicación; por otro lado, la dinámica del protocolo, es decir, la secuencia de estados por los que puede atravesar una sesión. En esta sección, además de los dos detalles enunciados, se analizará también cómo se integra la implementación de AMBAR dentro de CDSA.

6.2.1 Integración de la arquitectura CDSA

La implementación de AMBAR está completamente basada en el uso de los servicios ofrecidos por la arquitectura CDSA, tal y como puede observarse en la figura 6.2.

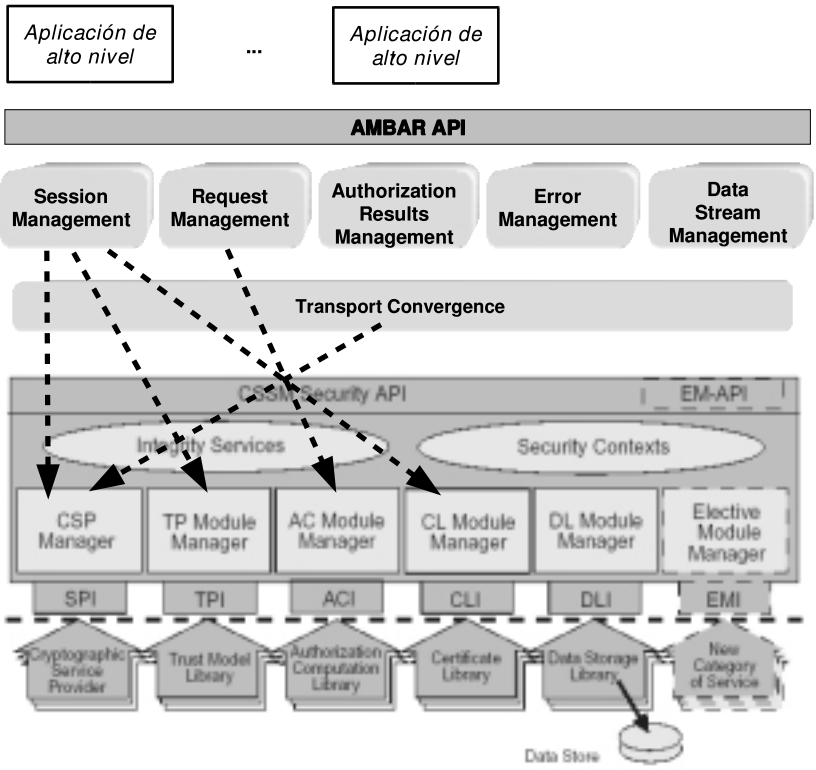

Figura 6.2: Integración de AMBAR con CDSA

Como se aprecia, AMBAR se encuentra ubicado en el nivel de servicios de seguridad construidos sobre la arquitectura. Las aplicaciones de alto nivel se comunican con la librería mediante una API que simplifica el uso del marco y oculta todos aquellos detalles irrelevantes para los programadores de aplicaciones de control de acceso a recursos.

Se observa también que los módulos AMBAR emplean la mayor parte de los proveedores de servicio contenidos en la implementación elegida de CDSA. Más concretamente, los servicios utilizados por cada módulo son:

- *Módulo SM.* Usa el módulo CSP (Cryptographic Service Provider) para realizar las operaciones de criptografía asimétrica relacionadas con los mensajes de autenticación (concretamente el mensaje *ActivateCrypto*), así como para generar las cargas aleatorias intercambiadas al principio de la fase. Además utiliza los módulos CL (Certificate Library) y TP (Trust Policy) para procesar y validar los certificados X.509 intercambiados.

- *Módulo RM.* Emplea el módulo AC (AuthCompute) para construir las reducciones de los certificados intercambiados.

- *Nivel TC*. Mediante el módulo CSP, realiza las transformaciones criptográficas de los mensajes AMBAR generados por los módulos superiores. En concreto, utiliza rutinas de cifrado simétrico y de construcción de códigos de autenticación.

6.2.2 Esquema de funcionamiento del protocolo

A la hora de emplear el protocolo como mecanismo de control de acceso a recursos protegidos, es necesario conocer cuál es la secuencia de estados por la que puede atravesar, es decir, cuáles son todas las posibilidades que pueden llegar a darse en una comunicación entre un controlador y un solicitante. Este conocimiento del funcionamiento del protocolo permite al programador comprender el uso correcto de la interfaz de programación ofrecida, es decir, de las primitivas de servicio ofrecidas por la librería para que esta pueda emplearse como si se tratase de un nivel de abstracción dentro de una arquitectura de red. La secuencia de estados permite conocer la dinámica del protocolo, y por tanto el orden en el cual deben ejecutarse los distintos métodos que forman parte de la API.

En la figura 6.3 se muestra un autómata de estados en el que cada transición representa la recepción o el envío de un mensaje AMBAR por parte de un solicitante. Aunque la representación está enfocada desde el punto de vista del cliente, es lo suficientemente completa como para orientar la programación de aplicaciones AMBAR de servidor.

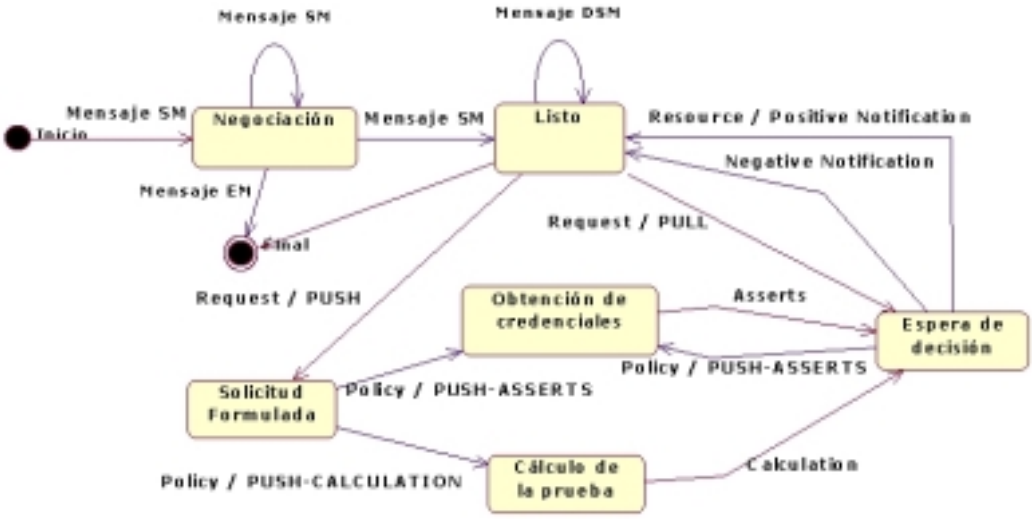

Figura 6.3: Secuencia de estados de AMBAR

En primer lugar se produce el inicio de la fase de negociación, la cual representa el primer estado del autómata. Por simplicidad se han agrupado todos los mensajes de esta fase con el nombre genérico de *mensaje SM*. Dado que la negociación se realiza de forma transparente para las aplicaciones de nivel superior, no es necesario detallar los estados que forman parte de esta fase. En el caso de que se produzca un error en la verificación de un mensaje o se presenten parámetros de autorización incompatibles se llega al final de la ejecución (estado *Final*).

Si la negociación finaliza con éxito, la ejecución se sitúa en el estado *Listo*. A partir de este instante, la posible secuencia de estados varía completamente en función del método de distribución de información negociado. A continuación se desglosan las tres opciones válidas:

- *Distribución pull*. Con esté método, tras el envío del mensaje *Request* el solicitante queda a la espera de que el controlador envíe alguno de los tres tipos de mensajes ARM (recurso y notificación negativa o positiva). En los tres modos de distribución, la recepción de un mensaje ARM implica una vuelta al estado *Listo*.

- *Distribución push-asserts*. Este método de distribución es propio de los controladores que realizan una revelación progresiva de su política de seguridad. Tras el envío de la solicitud se llega al estado *Solicitud formulada* y se produce la recepción del primer mensaje *Policy*, el cual implicará la transmisión de un conjunto de credenciales relacionadas con la política recibida. El número de veces que se repite dicho intercambio de mensajes *Policy* y *Asserts* depende de la estrategia del controlador en cuestión, y finaliza con la transmisión por parte del controlador de alguno de los mensajes ARM.

- *Distribución push-calculation*. Este caso se diferencia del anterior en el hecho de que la recepción del mensaje *Policy* no conlleva el envío de credenciales, sino que lleva al autómata al estado *Cálculo de la prueba*. Dicho estado implica el descubrimiento de una cadena de certificación que demuestre las posibilidades de acceso del solicitante según lo formulado por la política recibida. Tras el envío de dicha cadena al controlador, el solicitante debe quedar a la espera de un mensaje ARM.

Aunque no se muestran en la figura, la recepción o el envío de la mayor parte de mensajes EM implica el fin prematuro de la ejecución del protocolo. De hecho, desde cualquiera de los estados mostrados en la figura 6.3 habría una transición hacia un estado de finalización anormal.

Por último, comentar que la implementación del protocolo aquí presentada controla que la ejecución se ajuste rigurosamente al autómata presentado. Cualquier intento de transición no contemplada, motivada por el envío o recepción de mensajes incorrectos, es detectado e implica la finalización inmediata de la comunicación.

6.2.3 Interfaz de programación (API)

La API de acceso a la librería AMBAR está formada principalmente por 4 clases distintas: *AMBARClientSession*, *AMBARServerDaemon*, *AMBARServerSession* y *AMBARContext*. La relación entre dichas clases puede verse en el diagrama de clases UML [24] que muestra la figura 6.4.

AMBARContext

La construcción del *AMBARContext* es la primera acción que debe llevarse a cabo antes de establecer la comunicación. Esta clase representa el contexto de comunicación AMBAR, es

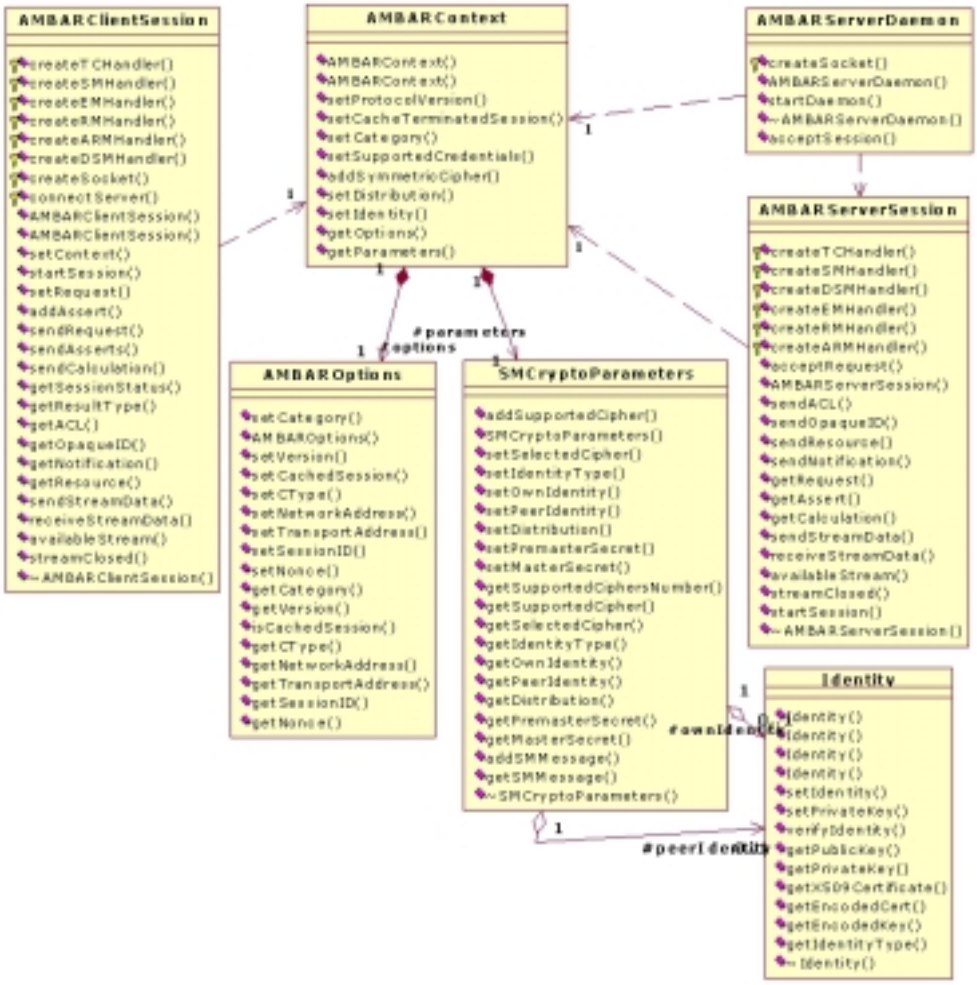

Figura 6.4: Clases de la API de AMBAR

decir, todos aquellos elementos de información relacionados con los parámetros de autorización a negociar, identidad de las entidades participantes y material criptográfico generado durante la sesión. Su implementación está basada en la agregación de dos objetos pertenecientes a las clases *AMBAROptions* y *SMCryptoParameters*, los cuales proporcionan la mayor parte de la funcionalidad de *AMBARContext*.

Si bien la cantidad de elementos de información gestionada por esta clase es muy elevada, el programador de aplicaciones tanto de cliente como de servidor sólo debe conocer los métodos principales relacionados con la especificación de los parámetros de autorización a negociar:

- *setProtocolVersion()*. Establece la versión del protocolo AMBAR que se desea utilizar.

- *setCategory()*. Indica el modo de operación preferido (anónimo o identificado).

- *setSupportedCredentials()*. Se utiliza para especificar el tipo de certificados de credencial que será empleado para determinar los privilegios.
- *addSymmetricCipher()*. Mediante este método se introducen, por orden de prioridad, los algoritmos simétricos preferidos.
- *setDistribution()*. Establece el método de distribución de credenciales a emplear en la sesión (*pull*, *push-asserts* y *push-calculation*).
- *setIdentity()*. Se utiliza tanto para especificar el certificado de identidad del comunicante como para establecer el conjunto de autoridades de certificación confiables.

Una vez creado el contexto, éste debe pasarse a las clases que representan las sesiones para que puedan emplearlo durante la fase de negociación.

AMBARClientSession

La recepción y envío de mensajes por parte de los solicitantes se hace a través de *AMBARClientSession*. Esta clase encapsula todos los aspectos relacionados con el establecimiento de la comunicación TCP/IP, la negociación de sesiones, el envío y optimización de solicitudes, y la recepción de notificaciones procedentes del controlador. Sus métodos se pueden agrupar en los siguientes bloques funcionales:

- Métodos relacionados con el establecimiento de los parámetros de comunicación y de autorización:
  - *AMBARClientSession()*. Mediante el constructor es posible especifica la dirección IP y el puerto TCP del controlador elegido.
  - *setContext()*. Establece el contexto AMBAR de la sesión.
- Métodos relacionados con la fase de negociación de sesiones:
  - *startSession()*. La llamada a este método implica el inicio de la fase de negociación AMBAR.
- Métodos relacionados con el envío de solicitudes, políticas y credenciales:
  - *setRequest()*. Establece una nueva solicitud de acceso.
  - *addAssert()*. Mediante este método es posible incluir credenciales en el próximo mensaje a enviar por el solicitante.
  - *sendRequest()*. Envía la solicitud de acceso y, si se produjo alguna llamada a *addAssert*, las posibles credenciales asociadas.
  - *sendAsserts()*. Envía un conjunto de certificados de credencial.

  - *sendCalculation()*. Envía una prueba de autorización calculada previamente por el solicitante.
  - *getSessionStatus()*. Este método indica si el controlador ha tomado una decisión definitiva acerca de la solicitud en curso o si por el contrario ha enviado información relacionada con su política de control de acceso, es decir, si le está pidiendo al solicitante más evidencias.
  - *getACL()*. Con este método es posible obtener la última política enviada por el controlador durante la transacción en curso.

- Métodos relacionados con la obtención de respuestas del controlador:

  - *getResultType()*. Este método indica si la respuesta es una notificación (positiva o negativa) o bien el recurso solicitado.
  - *getNotification()*. En el caso de que el controlador haya enviado una notificación, este método devuelve el contenido de la misma.
  - *getResource()*. Devuelve el contenido del recurso solicitado.

- Métodos relacionados con la gestión de flujos de datos:

  - *sendStreamData()*. Una vez que ha sido autorizada una solicitud relacionada con el establecimiento de un flujo de datos entre controlador y solicitante, este método se emplea para enviar información que será transportada sin modificación alguna hasta el controlador. Dicha información puede tratarse de datos de un protocolo de nivel superior.
  - *receiveStreamData()*. Este método devuelve parte de los datos enviados por el controlador a través del flujo.
  - *availableStream()*. Indica el número de bytes pendientes de lectura que están almacenados en el buffer de recepción del flujo.
  - *streamClosed()*. Mediante este método es posible conocer si el flujo sigue activo o si ha sido cancelado por la otra entidad.

Como puede apreciarse estudiando el API ofrecida, el programador de aplicaciones cliente sólo debe tener conocimiento de la secuencia de estados por la que puede atravesar el protocolo (ver figura 6.3). El resto de detalles relacionados con el funcionamiento interno del protocolo, como el intercambio concreto de mensajes o la optimización de las solicitudes, no son visibles al programador. El resultado es una API bastante intuitiva que permite desarrollar de forma sencilla aplicaciones que hagan uso de AMBAR.

AMBARServerDaemon

Con el fin de que un mismo controlador pueda atender de forma simultánea peticiones de distintos solicitantes, la librería proporciona una clase con funcionalidad similar a la de un *daemon* de comunicaciones. La clase *AMBARServerDaemon* permite que los controladores puedan establecer varias sesiones simultáneas caracterizadas por los mismos parámetros de autorización, es decir, por el mismo contexto AMBAR. Una vez creado dicho contexto, es posible iniciar un *AMBARServerDaemon* y establecer sesiones haciendo uso de los siguientes métodos:

- *AMBARServerDaemon()*. Mediante el constructor es posible especificar tanto el puerto TCP asociado al controlador como el contexto AMBAR.
- *startDaemon()*. Inicia la ejecución del *daemon*.
- *acceptSession()*. Tras ejecutar este método el controlador queda a la espera del establecimiento de una nueva sesión.

Una conexión entrante realizada por parte de un solicitante implica la generación de una nueva sesión AMBAR de servidor mediante la cual será posible llevar a cabo el intercambio de la información relativa a autorización.

AMBARServerSession

Los objetos de esta clase no debe generarlos el programador ya que es el método *acceptSession()* de *AMBARServerDaemon* el encargado de generar las nuevas sesiones de servidor. La consecuencia es que el contexto de cada sesión ya está creado tomando como base el contexto del *daemon*. Por tanto, todos los métodos de esta clase están relacionados sólo con el intercambio de información entre solicitante y controlador:

- Métodos relacionados con la fase de negociación de sesiones:
  - *startSession()*. La llamada a este método implica el inicio de la fase de negociación AMBAR.
- Métodos relacionados con el intercambio de solicitudes, políticas y credenciales:
  - *acceptRequest()*. La ejecución de este método implica la espera de una petición de acceso procedente del solicitante.
  - *getRequest()*. Se utiliza para obtener la solicitud recibida.
  - *getAssert()*. Mediante este método es posible ir obteniendo las credenciales que ha enviado el solicitante, tanto en la solicitud como de forma independiente.
  - *sendACL()*. Envía la política relacionada con la solicitud recibida. El contenido de dicha política depende de la estrategia de revelación seguida por el controlador.

  - *getCalculation()*. Recibe la prueba de autorización hallada por el solicitante.

- Métodos relacionados con el envío de respuestas a las solicitudes:
  - *sendNotification()*. Devuelve una notificación, negativa o positiva, acerca del resultado del procesamiento de la petición formulada por el solicitante.
  - *sendResource()*. Envía el recurso protegido que se había solicitado.

- Los métodos relacionados con la gestión de flujos de datos son los mismos que ofrece la clase *AMBARClientSession*.

Como se verá en la siguiente sección, el protocolo AMBAR se ha integrado como parte de las aplicaciones de las autoridades y de los solicitantes del sistema DCMS. La librería ofrece toda la funcionalidad necesaria para implementar el intercambio de información entre las dos aplicaciones.

6.3 Implementación de DCMS

El primer paso en el desarrollo de DCMS fue la elección del tipo de implementación que se deseaba realizar, ya que había dos alternativas claras en lo que a la concepción final del sistema se refería. Por un lado, cabía la posibilidad de desarrollar aplicaciones que intentaran ocultar todos los detalles internos del sistema a los usuarios de las mismas, lo cual representaría la creación de herramientas de alto nivel totalmente independientes de cuestiones como el formato de las políticas, solicitudes, tags, etc. Por otro lado, existía la alternativa de crear herramientas más cercanas a la estructura del sistema, que permitieran ver y configurar todos los parámetros y elementos de información que forman parte de DCMS. Tomando como referencia la idoneidad para poder realizar un análisis y evaluación de las posibilidades de DCMS, finalmente se tomó la decisión de desarrollar aplicaciones altamente configurables que mostraran el mayor número posible de las características propias del sistema, como la monitorización de las comunicaciones realizadas, la visualización del contenido de todos los elementos de información o la configuración de los parámetros criptográficos.

Por otro lado, el conjunto de aplicaciones debía estructurarse de acuerdo con la metodología analizada en la sección 5.4. El objetivo era que la aplicación de la metodología tuviera una correspondencia directa con los distintos componentes del sistema, lo cual permitiría que la aplicación de los procedimientos contenidos en la metodología pudiera realizarse de forma estructurada y bien definida. Así pues, el resultado final fue que parte de las aplicaciones desarrolladas puede interpretarse como una materialización práctica de los distintos niveles de procedimientos. Dichas aplicaciones están claramente relacionadas entre sí y con las dos herramientas principales del sistema: la aplicación de las autoridades y la de los solicitantes. A lo largo de esta sección, se verá cuál es la funcionalidad de cada uno de los componentes del sistema así como algunos detalles concretos de la implementación del mismo.

6.3.1 Visión general

Todas las aplicaciones desarrolladas poseen su propia interfaz gráfica de usuario (GUI, Graphic User Interface), lo cual permite interactuar con ellas de forma amigable y visualizar la información ordenadamente. En concreto, su implementación se ha llevado a cabo en Linux, utilizando el lenguaje de programación C++, la arquitectura CDSA y las librerías gráficas QT [185]. La figura 6.5 muestra tanto la relación existente entre todas las aplicaciones del sistema DCMS que se han desarrollado como su interconexión con la infraestructura de clave pública y el marco AMBAR. Por un lado, dicha interconexión está basada en el uso de los servicios de generación y validación de claves proporcionados por la PKI, los cuales representan el proceso de gestión básico de las claves a las cuales se les asignarán las autorizaciones. Por otra parte, la implementación del marco AMBAR permite realizar el intercambio de información entre las autoridades y los solicitantes.

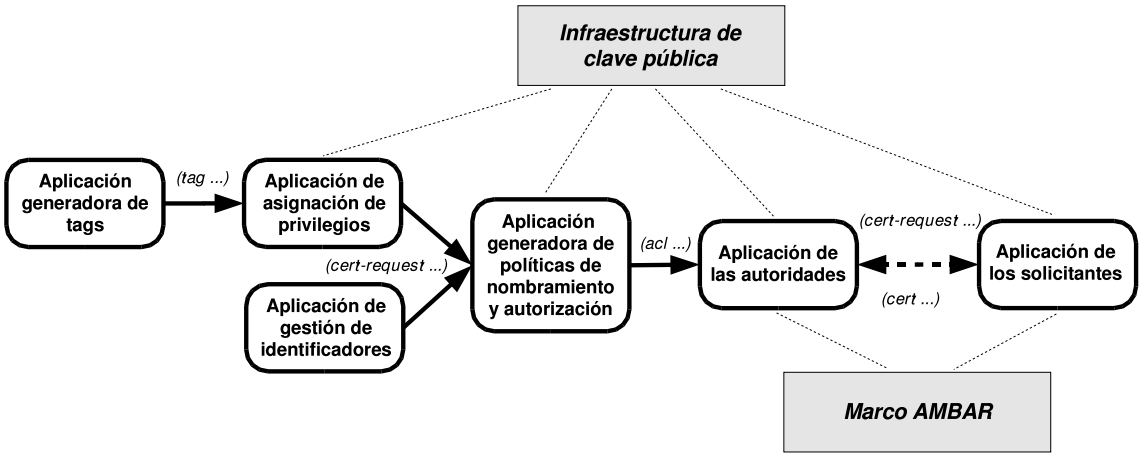

Figura 6.5: Conjunto de aplicaciones DCMS

Como se observa, los resultados producidos por parte de las aplicaciones del sistema DCMS suponen el punto de partida de otras, lo cual desemboca en una composición secuencial de elementos claramente relacionados entre sí. La figura muestra dos bloques de aplicaciones distintos ya que, por un lado, tenemos un conjunto de herramientas encargadas de producir las políticas de autorización y de nombramiento de las autoridades, y por otro lado se aprecian las dos aplicaciones que interactúan para generar los certificados de credencial SPKI (autoridades y solicitantes). En los siguientes apartados se realizará una descripción concreta de la funcionalidad de cada uno de los componentes de la figura.

6.3.2 Aplicación generadora de tags

La primera aplicación está relacionada con parte de los procedimientos contenidos en el nivel 0 de la metodología, concretamente con aquellos asociados a la especificación de los recursos que se desea proteger. Como se vio en la sección 5.4.1, dicha especificación

comprende la definición de un esquema de identificación de recursos y la identificación de las operaciones realizadas sobre dichos recursos.

La aplicación generadora de tags es totalmente dependiente del entorno de aplicación al cual se esté aplicando el sistema. Su objetivo principal es la creación de s-expresiones que representen tags de autorización válidos, los cuales serán posteriormente utilizados tanto para la definición de políticas de autorización como para la construcción de solicitudes de certificación. La sección 6.4 mostrará un ejemplo concreto de este tipo de aplicación.

Dejar constancia de que se trata del único componente del sistema DCMS que varía de un entorno de aplicación a otro. El resto de componentes, al estar basados en la composición de las s-expresiones que cada uno de ellos produce, no necesita ningún tipo de modificación para ser adaptado a otro escenario. De esta forma se minimiza el trabajo de desarrollo necesario para integrar DCMS en distintos sistemas.

6.3.3 Aplicación de asignación de privilegios

La aplicación de asignación de privilegios permite plasmar los procedimientos de nivel 2 del bloque AMS de la metodología. Se trata del componente mediante el cual es posible especificar quienes son las entidades autorizadas a obtener los privilegios previamente especificados mediante la aplicación comentada en el apartado anterior. La figura 6.6 muestra parte de la interfaz gráfica de este componente.

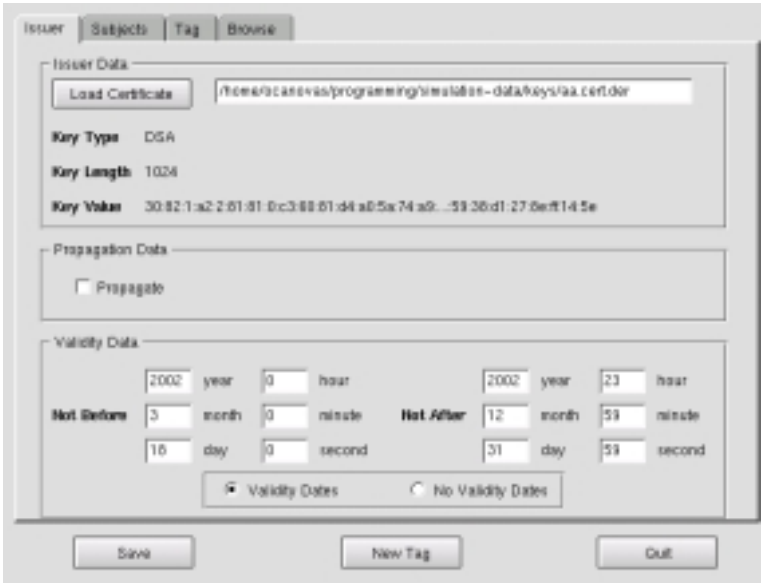

Figura 6.6: Aplicación de asignación de privilegios

La aplicación permite construir las s-expresiones del tipo *cert-request* que posteriormente formarán parte de la política de autorización de una autoridad concreta. Para ello, es necesario especificar:

- *Emisor de los certificados.* Se trata de la clave pública de la autoridad encargada de emitir los privilegios.
- *Receptores de los certificados.* Son todas aquellas claves públicas y nombres SDSI que están autorizados a recibir un conjunto determinado de privilegios.
- *Posibilidad de propagación de los privilegios.* Es posible especificar si el conjunto de entidades anterior tiene concedido el derecho a poder propagar a su vez los privilegios recibidos, es decir, a actuar como autoridades de autorización.
- *Privilegios asignados.* Se trata del resultado de la aplicación del apartado anterior, es decir, de s-expresiones que contienen tags de autorización concretos relacionados con el entorno de aplicación que se está modelando.
- *Periodo de disfrute de los privilegios.* Por último, se especifica el periodo de tiempo durante el cual los receptores podrían hacer uso de los privilegios concedidos.

Como resultado final se obtiene una s-expresión que codifica los criterios de asignación de privilegios seguidos por la entidad emisora. La s-expresión, cuya estructura fue analizada en la sección 5.3.4, es uno de los elementos de información necesarios para la aplicación generadora de políticas.

6.3.4 Aplicación de gestión de identificadores

La aplicación de gestión de identificadores está ligada al nivel 2 del bloque NMS de la metodología. Una vez que el primer nivel determina los elementos que se encuentran dentro del ámbito de la autoridad, es decir, el subconjunto de entidades individuales y los roles ya definidos del sistema, el nivel 2 debe determinar los identificadores que pueden ser asignados a cada uno de ellos. Esta asignación de identificadores se lleva a cabo mediante la aplicación mostrada en la figura 6.7.

La aplicación permite construir las s-expresiones del tipo *cert-request* que posteriormente formarán parte de la política de nombramiento de una autoridad concreta. Para ello, es necesario especificar:

- *Emisor de los certificados.* Se trata de la clave pública de la autoridad encargada de emitir los certificados de identidad SPKI.
- *Receptores de los certificados.* Son todas aquellas claves públicas y nombres SDSI que están autorizados a recibir un conjunto determinado de privilegios. En el caso de que se desee asignar nombres localmente únicos a entidades, el conjunto de receptores estará formado por una única clave pública. Si lo que se desea es gestionar la pertenencia a un rol, el conjunto de receptores puede estar formado tanto por las claves públicas de las entidades pertenecientes al mismo como por el nombre SDSI de los roles que se encuentran contenidos dentro del que se está definiendo (lo que conlleva la generación de jerarquías).

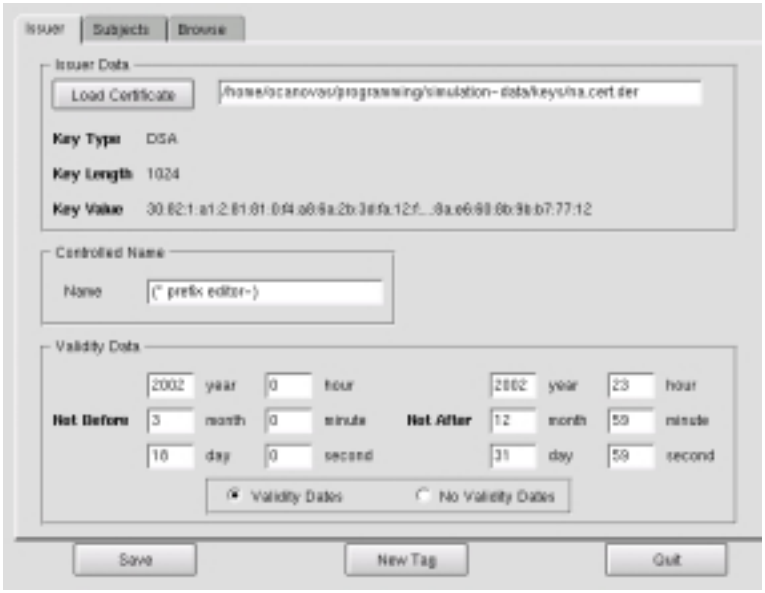

Figura 6.7: Aplicación de gestión de identificadores

- *Identificador a gestionar.* Se trata del identificador que será asociado a los receptores, el cual puede tratarse de un nombre de rol o de un identificador de usuario localmente único.

- *Periodo de asociación.* Por último, se especifica el periodo de tiempo durante el cual los receptores podrían permanecer ligados al identificador.

6.3.5 Aplicación generadora de políticas

La aplicación generadora de políticas es la aplicación práctica de los procedimientos de nivel 3 tanto del bloque AMS como NMS. Es decir, una vez realizada la asignación de privilegios e identificadores, mediante esta aplicación es posible finalizar la especificación de la política mediante el establecimiento del conjunto de solicitantes válidos y del periodo de solicitud. La figura 6.8 muestra parte de la interfaz gráfica de este componente.

Partiendo de las s-expresiones generadas mediante la aplicación de asignación de privilegios y la aplicación de gestión de identificadores, es posible especificar los siguientes parámetros:

- *Conjunto de solicitantes autorizados.* Se trata de las claves públicas o nombres SDSI de las entidades que están autorizadas a solicitar las credenciales. En el caso de que se omita se asumirá que coincide con el conjunto de entidades receptoras.

- *Posibilidad de delegación.* Es posible especificar si el conjunto de entidades solicitantes puede delegar el privilegio de solicitar las credenciales.

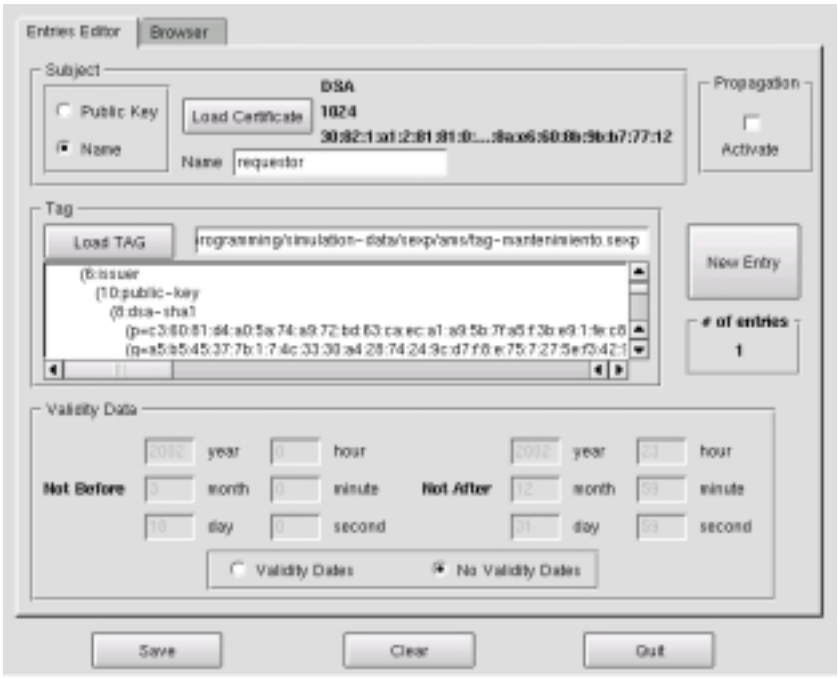

Figura 6.8: Aplicación generadora de políticas

- *Credenciales a emitir.* S-expresiones que denotan qué privilegios o qué identificadores pueden ligarse a un conjunto de entidades finales. En función del tipo de s-expresión, obtenidas como resultado de las aplicaciones de asignación de privilegios o de gestión de identificadores, se estará definiendo una política de autorización o una de nombramiento.

- *Periodo de solicitud.* Especifica el intervalo de tiempo durante el cual se puede llevar a cabo la solicitud de las credenciales. En caso de omisión se asumirá que coincide con el periodo de validez especificado en las s-expresiones anteriores.

El resultado principal de esta aplicación es la construcción de elementos de información muy similares a las listas de control de acceso SPKI que codifican todos los criterios de autorización derivados a partir de la aplicación de la metodología. Como se verá en el siguiente apartado, dichos elementos de información condicionan el comportamiento que tendrán las autoridades frente a las solicitudes de certificación recibidas de forma directa o bien a través de los puntos de acceso.

6.3.6 Aplicación de las autoridades

La aplicación de las autoridades constituye la herramienta más completa del sistema DCMS ya que puede actuar como autoridad de autorización, autoridad de nombramiento y reductor confiable. Además está diseñada para operar tanto en línea como desconectada de la

red. Sus usuarios serán todas aquellas entidades que tras la aplicación de los procedimientos de nivel 0 de la metodología deban asumir las funciones de gestión de un conjunto de credenciales. En los siguientes apartados se analizará de forma independiente cada uno de los grupos de operaciones que es posible realizar haciendo uso de ella.

Configuración de las propiedades de la autoridad

El primer bloque está relacionado con el establecimiento de las propiedades de la autoridad, entendiendo como propiedades los siguientes elementos:

- *Tipo de autoridad.* Para que la aplicación ofrezca sólo las funcionalidades asociadas a cada tipo de autoridad, en primer lugar es necesario especificar si actuará como autoridad de autorización, autoridad de nombramiento o reductor confiable.

- *Valores criptográficos.* Para poder emitir certificados es necesario especificar el par de claves asimétricas pertenecientes a la autoridad. Dicha configuración puede realizarse de tres formas distintas: en primer lugar es posible solicitar que la aplicación genere un nuevo par de claves criptográficas y un certificado X.509 autofirmado para la autoridad; por otro lado, también es posible generar el par de claves y construir una solicitud PKCS#10 con el fin de que el certificado de la autoridad sea generado por parte de la PKI; por último, se puede especificar la localización de una clave privada y certificado ya existentes.

- *Certificados digitales de la interfaz AMBAR.* Tal y como se comentó en la sección 5.3.6, no resulta conveniente que las autoridades utilicen sus claves privadas de firma para establecer conexiones AMBAR. Constituye una alternativa más correcta que éstas generen pares de claves temporales destinadas a proteger las comunicaciones. En consecuencia, la aplicación permite generar el par de claves temporales que será empleado para establecer sesiones AMBAR con los solicitantes y puntos de acceso. Para que dichas claves sean consideradas como válidas por el resto de entidades del sistema, la aplicación emite también un certificado de autorización que les confiere el privilegio de actuar como su interfaz AMBAR.

Configuración de los parámetros AMBAR

En relación con el modo de operación en línea de la autoridad, es necesario especificar las preferencias relacionadas con el establecimiento de sesiones AMBAR. En concreto, es necesario especificar:

- *Parámetros a negociar.* La aplicación permite especificar las preferencias de la autoridad en lo que respecta al tipo de certificados de identidad a emplear, tipo de certificados de credencial, método de distribución de credenciales, modo de comunicación y algoritmo de cifrado simétrico preferido. Estas preferencias están relacionadas con la aplicación de los procedimientos del nivel 4 de los bloques AMS y NMS de la metodología.

- *Parámetros de red.* Es posible especificar tanto el puerto TCP mediante el cual se realizará la comunicación con las entidades externas como el número máximo de sesiones simultáneas activas que se atenderán.

Especificación de las políticas

Uno de los modos de operación de las autoridades es la tramitación de solicitudes de certificación que conforman con la política de la autoridad. Por tanto, una de las opciones que ofrece la aplicación de las autoridades es la especificación de un conjunto indefinido de políticas que deben cumplirse, conjunto que podrá ser modificado en cualquier instante.

Operaciones en modo desconectado

La aplicación permite realizar varias operaciones de gestión en modo *desconectado*, es decir, no propiciadas por la recepción a través de un canal AMBAR de una solicitud de servicio. Esta opción es especialmente útil para gestionar privilegios que necesitan una protección especial. Más concretamente, las operaciones de este tipo son:

- *Generación de certificados.* Esta operación pueden realizarla tanto las autoridades de nombramiento como las de autorización. Se trata de un formulario que permite introducir todos los campos que forman parte de alguno de los tres tipos de certificados SPKI. De esta forma es posible emitir manualmente certificados que no estén contemplados en la política.

- *Reducción de certificados.* La operación de reducción está disponible tanto para los reductores confiables como para las autoridades de autorización. Como muestra la figura 6.9, la aplicación permite simplificar cadenas de certificación que tengan como raíz la clave pública de la autoridad o del reductor y como nodo final la clave pública de la entidad especificada en el formulario. Además, la reducción se puede restringir a un conjunto determinado de privilegios.

- *Gestión de solicitudes.* Esta operación, disponible para los tres tipos de autoridades, permite tramitar una solicitud existente. Dicha solicitud puede ser tanto de certificación como de reducción de una cadena de certificación. Para su tramitación se comprueba si cumple con la política de la autoridad, en cuyo caso se procede a la generación del certificado correspondiente.

Operaciones en línea

La aplicación permite activar un proceso independiente encargado de atender de forma concurrente, mediante una ejecución multihilo, las peticiones formuladas por los usuarios a través de la red, bien de forma directa o a través de los puntos de acceso. En todo momento, tal y como se muestra en la figura 6.10, es posible monitorizar cuál es la evolución de dichas comunicaciones. En concreto, la aplicación permite:

Figura 6.9: Reducción de certificados en modo desconectado

- Visualizar en qué instante se produjo el inicio de una negociación AMBAR o la tramitación de un solicitud para cada una de las conexiones entrantes activas.

- Conocer los parámetros de autorización negociados en cada una de las sesiones establecidas.

- Conocer el número total de solicitudes tramitadas correctamente y los datos relativos a los certificados generados en consecuencia.

En este modo, la emisión de certificados se realiza siempre tras la validación de que las solicitudes correspondientes cumplen la política especificada por la autoridad.

6.3.7 Aplicación de los solicitantes

Este componente recibe el nombre genérico de *aplicación de los solicitantes* porque aglutina tanto la funcionalidad de un punto de acceso como la de una aplicación de gestión de autorizaciones diseñada para usuarios finales. Su objetivo general es la provisión de mecanismos que permitan a las entidades finales crear y transmitir solicitudes de certificación y de reducción. En los siguientes apartados se analizará de forma independiente cada uno de los grupos de operaciones que es posible realizar haciendo uso de ella.

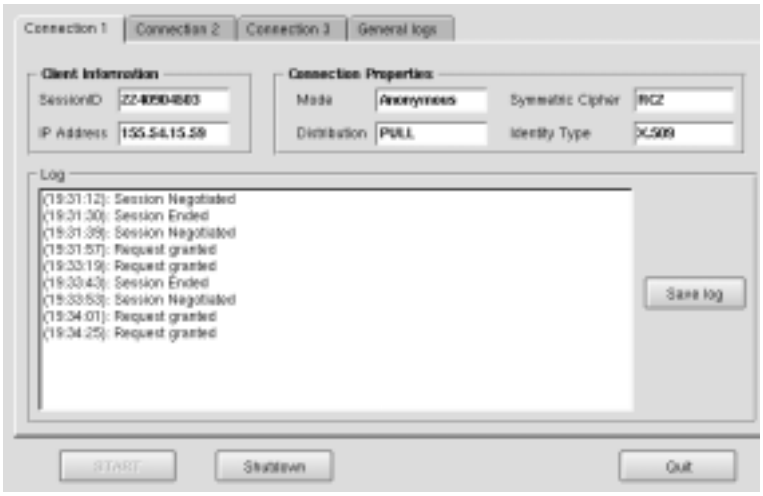

Figura 6.10: Monitorización de las conexiones de la autoridad

Gestión de claves temporales

Tal y como se comentó en la sección 4.4.3, el uso de claves temporales por parte de los usuarios finales proporciona dos servicios adicionales de especial utilidad. Por un lado, un usuario puede crear claves temporales distintas para cada una de las tareas que realiza, limitando así el alcance de cualquier posible incidente de seguridad, ya que el compromiso del par de claves asociado a una tarea no afectaría al resto. Por otro lado, el uso de claves temporales y la reducción de las cadenas de delegación permite ocultar la identidad del usuario.

En relación con lo anterior, la aplicación de los solicitantes permite la:

- *Generación de claves temporales.* Es posible crear un número indefinido de claves, tanto RSA como DSA.

- *Delegación de privilegios a claves temporales.* La aplicación permite emitir certificados de autorización SPKI que asocien un subconjunto de los privilegios de los solicitantes a alguna de las claves temporales generadas. De esta forma el usuario puede determinar qué tareas asignar a cada una de dichas claves.

Configuración de autoridades

La aplicación puede almacenar datos relacionados con las distintas autoridades del sistema, los cuales son empleados para generar las solicitudes y encaminarlas hacia la aplicación de la autoridad. En el caso de que la aplicación actúe como un punto de acceso, es posible registrar la información relacionada con el tipo de credenciales que emite cada autoridad y con los parámetros de autorización utilizados para establecer sesiones AMBAR.

Gestión de solicitudes

Al igual que sucedía con la aplicación de las autoridades, aquí también es posible tramitar solicitudes tanto en modo desconectado como en línea. La gestión de las solicitudes se divide en dos operaciones principales:

- *Generación de solicitudes.* Mediante esta opción es posible generar solicitudes tanto de certificación como de reducción. Para ello, tal y como se aprecia en la figura 6.11, el solicitante selecciona la autoridad que deberá emitir el certificado, la clave que firmará la solicitud (es posible crear solicitudes de certificación utilizando alguna de las claves temporales previamente generadas), la credencial que se está solicitando (tanto en el caso de solicitar la pertenencia a un rol como la asignación de un privilegio concreto), la posibilidad de propagar el permiso si éste es obtenido y el periodo durante el cual quiere ejercerse la credencial. En el caso de que la aplicación actúe como punto de acceso, es posible controlar que sólo se soliciten aquellas credenciales que la autoridad seleccionada está autorizada a emitir. Una vez generada la solicitud hay dos posibilidades a la hora de tramitarla: por un lado, en el caso de que la autoridad opere siempre en modo desconectado debido a la sensibilidad de los permisos tratados, será necesario presentar la solicitud utilizando un mecanismo fuera de línea; por otro lado, si la solicitud puede presentarse mediante una conexión AMBAR, la aplicación ofrece la posibilidad de establecer una conexión directa con la autoridad emisora.

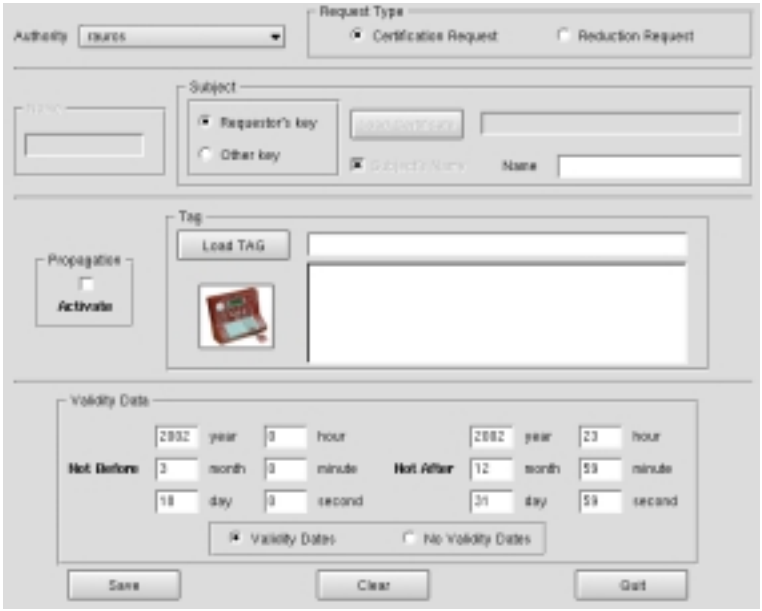

Figura 6.11: Creación de solicitudes

- *Envío de solicitudes.* Para tramitar en línea las peticiones creadas por los solicitantes, la operación de envío de solicitudes permite contactar mediante AMBAR con

cualquiera de las autoridades dadas de alta en la aplicación. Una vez negociada la sesión, los solicitantes pueden enviar cualquier número de solicitudes y de credenciales de apoyo. A continuación, la aplicación mostrará el resultado generado por la autoridad, tanto si se trata de un certificado como de una notificación negativa. En todo momento, es posible monitorizar los mensajes AMBAR que están siendo intercambiados con la autoridad y conocer de esa forma cómo se encapsula la información dentro del protocolo.

6.3.8 Conclusiones

A partir de la descripción realizada en esta sección de la implementación del sistema DCMS, se puede comprobar que su estructura está totalmente condicionada por la definición de la metodología. Esto hace que el conjunto de aplicaciones pueda emplearse de forma ordenada y que los distintos niveles de la metodología tengan una correspondencia con la funcionalidad ofrecida por dichas aplicaciones. Además, el hecho de que se trate de componentes independientes favorece la aplicación selectiva de los procedimientos necesarios y favorece la reutilización de la información generada por cada una de las aplicaciones.

Tanto la aplicación de las autoridades como de los solicitantes ofrecen un amplio abanico de posibilidades, pudiendo además configurarse para que actúen como si de distintos tipos de aplicaciones se tratara. Incorporan todas la funcionalidades criptográficas, de gestión de certificados y de comunicación necesarias para no depender de ninguna otra aplicación externa, excepto de la infraestructura de clave pública tomada como punto de partida.

Por último, dejar constancia de la integración lograda entre el marco AMBAR y el sistema DCMS. Como se ha visto, dicha integración se produce en lo que respecta la configuración de las sesiones, al uso de las primitivas de negociación, envío y recepción, y a la monitorización de los mensajes intercambiados.

A continuación, se mostrará cómo se ha integrado el sistema DCMS en dos entornos de aplicación reales. En primer lugar, se analizará cuáles son las ventajas que puede introducir el uso de certificados de credencial en un escenario tan clásico como es el control de acceso físico a recintos. Posteriormente, se verá cómo se ha diseñado un sistema de suscripción electrónica, o fidelización, que hace uso de un protocolo de pago electrónico desarrollado en el seno del grupo de investigación.

6.4 Integración en un entorno de control de acceso físico

El control de acceso físico implica la provisión de mecanismos que impidan la entrada a determinados recintos, tales como laboratorios de investigación, despachos o almacenes. Este tipo de control es especialmente relevante en aquellos casos en los que los recintos a proteger contienen equipamiento informático, puesto que la manipulación malintencionada de servidores, enrutadores u otros equipos de telecomunicaciones puede llegar a tener un gran impacto sobre algunas comunidades de individuos.

Estamos hablando de un problema clásico, en el cual hay un conjunto de aspectos muy definidos a los que debe aportarse solución. En primer lugar, encontramos el problema de la distribución de claves, es decir, cómo proporcionar las claves apropiadas a los usuarios autorizados que pueden hacer uso de ellas. En relación con esto, las claves pueden ser canceladas o revocadas como respuesta a ciertas situaciones que impliquen una amenaza de seguridad. Por otro lado, es necesario especificar cómo será gestionada la información de los usuarios, es decir, sus datos personales y sus privilegios de acceso. Como ya se ha comentado en apartados anteriores, los usuarios suelen desempeñar ciertos roles dentro del sistema, por lo que resulta muy importante poder reflejar dichos roles en el proceso de gestión de privilegios de acceso.

Tradicionalmente, las principales propuestas se han basado en el uso de bases de datos centralizadas que contenían información acerca de los usuarios autorizados. Por ejemplo, una situación típica es la de un usuario que dispone de datos identificativos, probablemente únicos para cada usuario, que presenta a un dispositivo especial localizado a la entrada de un recinto concreto. En estos entornos clásicos, el dispositivo no conoce qué identificadores son válidos por lo que debe realizar una consulta a la base de datos central para obtener los privilegios del usuario en cuestión. El tipo de identificador puede ser una clave pública, un nombre X.500, un certificado de identidad, datos biométricos o simplemente un nombre de usuario.

Por otro lado, a lo largo de esta tesis se ha demostrado que los certificados de credencial constituyen un mecanismo muy apropiado para codificar información relativa a cualquier tipo de privilegio, lo cual incluye privilegios relacionados con operaciones de control de acceso físico. Además, el uso de este tipo de certificados permite plasmar de forma sencilla las estructuras de roles de un sistema, lo cual simplifica la gestión del mismo.

A lo largo de esta sección, se presentarán dos propuestas [54] que resuelven el problema del control de acceso físico de forma muy distinta. La primera de ellas es el sistema básico que se está empleando en la Universidad de Murcia para controlar el acceso a ciertas instalaciones. La segunda es una modificación que toma como base la propuesta anterior y que proporciona algunas ventajas adicionales desde el punto de vista del no repudio, escalabilidad y la conectividad.

Ambas propuestas hacen uso de un dispositivo especial llamado TICA (Terminal Inteligente de Control de Acceso), el cual ha sido desarrollado completamente en la Universidad de Murcia [136, 160]. Los TICAs contienen lectores de tarjetas inteligentes y son capaces de intercambiar información con un servidor de aplicación a través de una red de comunicaciones ya que están equipados con una tarjeta Ethernet. Además de las operaciones de control de acceso, también son capaces de registrar la hora de entrada y de salida de los trabajadores, así como de realizar operaciones de mantenimiento como el control de iluminación o control de alarmas. Están desarrollados sobre el sistema operativo Linux y su módulo principal de procesamiento está basado en un SBC (Single Board Computer) con una frecuencia de reloj de 133 MHz.

Otro de los elementos comunes a ambas propuestas son las tarjetas inteligentes, las cuales se emplean como repositorios de certificados y como dispositivos habilitados para realizar operaciones criptográficas. En ellas está contenida la información relacionada con

los privilegios de los usuarios, que dependiendo de la propuesta estará constituida por identificadores únicos o por certificados digitales.

En los siguientes apartados se describirán los detalles de las dos propuestas y se mostrará la integración del sistema DCMS así como la aplicación de la metodología a la hora de poner en marcha la propuesta basada en certificados SPKI.

6.4.1 Propuesta centralizada

El sistema actual de control de acceso físico instalado en la universidad sigue un enfoque claramente centralizado. La intención de este análisis es mostrar las carencias asociadas a esta propuesta y cómo pueden ser éstas solventadas empleando otra filosofía en el diseño del sistema. Sin embargo, es importante dejar constancia de que la implementación centralizada actual lleva varios años utilizándose de forma satisfactoria dentro de la comunidad universitaria y en otros escenarios.

La distribución de claves se realiza utilizando identificadores únicos de usuario. Cada persona dispone de su propio identificador (normalmente su número de DNI), el cual se encuentra almacenado en su tarjeta inteligente (recordar que todos los miembros de la Universidad de Murcia disponen de su propia tarjeta inteligente). Por otro lado, existe una base de datos central que contiene una tabla por cada uno de los TICAs que se encuentran instalados. Dichas tablas están compuestas por distintos registros que contienen, entre otros campos, un identificador de usuario y el conjunto de permisos asignados al mismo. Cuando un usuario es autorizado a realizar una acción determinada se introduce un nuevo registro en la tabla asociada al TICA correspondiente. Sin embargo, el sistema no tiene soporte para roles ya que esta solución asigna directamente permisos a usuarios.

Por otro lado, la revocación de claves se implementa de forma bastante sencilla. En el caso de que tengan que anularse los permisos asignados a un usuario, basta con eliminar de la base de datos el registro correspondiente.

En lo que respecta a la verificación de claves, ésta se realiza mediante una consulta remota a la base de datos. Con el fin de garantizar la integridad de los datos intercambiados, se realizan conexiones SSL entre cada uno de los TICAs y los servidores de aplicación que acceden a la base de datos. Cada vez que se instala un nuevo TICA, es necesario generar un par de claves asociadas al dispositivo y solicitar un certificado X.509 a la autoridad de certificación de la universidad, ya que los servidores de aplicación realizan una autenticación de cliente con el fin de verificar que la consulta proviene de un TICA válido. En caso de que se produzca una caída temporal de la red, el dispositivo dispone de una base de datos local que contiene información relacionada con solicitudes de acceso ya tramitadas. Haciendo uso de dicha información local, el TICA puede seguir ofreciendo servicio a aquellos usuarios que ya fueron autorizados en alguna ocasión.

El esquema general de esta propuesta aparece ilustrado en la figura 6.12. Durante la fase de registro inicial, el usuario obtiene su identificador y la correspondiente tarjeta inteligente. En este momento, se puede solicitar también la creación de los registros necesarios para conceder el acceso a determinados recintos. Posteriormente, el usuario introducirá su tarjeta inteligente en el TICA y solicitará una de las operaciones ofrecidas por el dispositivo

(apertura de puertas, control horario del personal, etc.). A continuación, el TICA realizará una consulta a la base de datos central para averiguar si el usuario en cuestión tiene permiso para realizar la acción solicitada. En función de la respuesta, el dispositivo llevará a cabo el servicio o bien notificará al usuario la negativa correspondiente.

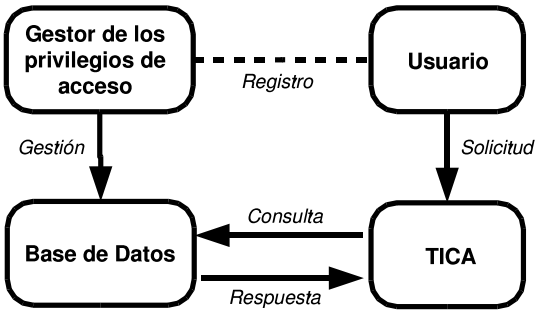

Figura 6.12: Visión general de la propuesta centralizada

6.4.2 Propuesta descentralizada

Aunque la propuesta centralizada se está utilizando de forma satisfactoria y proporciona soluciones a la mayoría de los problemas relacionados con el control de acceso, presenta algunas carencias que pueden ser solventadas siguiendo un enfoque descentralizado basado en el uso de certificados de credencial.

Motivación

El enfoque centralizado requiere una conectividad permanente con la base de datos central. Cuando ésta se rompe, los dispositivos continúan proporcionando servicio basándose en las copias locales de la información de autorización. Por tanto, cualquier modificación posterior de la información contenida en la base de datos será vista sólo por aquellos TICAs que permanecen conectados. En consecuencia, el sistema permanece durante un cierto periodo de tiempo en un estado inconsistente ya que algunos dispositivos rechazarán solicitudes que sin embargo otros aceptarán.

De hecho, en algunos entornos no resulta sencillo disponer de conectividad a la red de comunicaciones, lo cual impide poder utilizar un enfoque de este tipo. Sería aconsejable que el sistema de control de acceso fuera realmente distribuido, no por el hecho de estar basado en dispositivos distribuidos geográficamente sino por la posibilidad de ejercer sus funciones sin depender de un punto central. En la propuesta descentralizada que aquí se presenta los terminales pueden operar en modo desconectado y pueden determinar si un usuario está autorizado a realizar la acción solicitada sin necesidad de consultar a ninguna entidad externa.

Por otro lado, si el escenario en el cual se desarrolla en sistema de control de acceso está constituido por una comunidad de usuarios extensa, la tarea de gestionar cada TICA,

la lista de usuarios autorizados o la copia local de la información de autorización puede resultar muy compleja. Una solución más acertada es definir grupos de usuarios a los cuales asignar conjuntos de permisos en lugar de gestionar cada usuario de forma individual. No obstante, realizar dicha definición haciendo uso de una base de datos central presenta los mismos problemas derivados de la falta de conectividad que aparecían en el caso de las autorizaciones individuales. Por tanto, el mecanismo de definición de grupos o roles debe tener también un enfoque distribuido.

Por último, la propuesta centralizada carece de mecanismos robustos de no repudio de solicitante. Es cierto que tanto la base de datos como los TICAs realizan apuntes de las acciones que han sido solicitadas por los usuarios. Sin embargo, dichos apuntes no constituyen un mecanismo robusto de cara a ser utilizados como prueba de no repudio. La información registrada es susceptible de ser modificada por cualquier intruso capaz de acceder a parte de la información almacenada en la base de datos. Aunque es posible utilizar algunas soluciones basadas en el cifrado de datos para almacenar información confidencial en sistemas no confiables, lo realmente necesario es poder disponer de información generada por el solicitante, y no por los TICAs o la propia base de datos, que pueda ser utilizada como una evidencia irrefutable de las peticiones realizadas. Más concretamente, se está haciendo referencia al uso de solicitudes de servicio firmadas digitalmente por los usuarios, las cuales podrán ser empleadas, junto con las decisiones de autorización, para adoptar las medidas pertinentes tras un incidente de seguridad.

Diseño de la propuesta

La propuesta descentralizada está completamente basada en el uso del sistema DCMS, lo que implica que la gestión de las autorizaciones se realice según el esquema RBAC (ver sección 4.2.3) y el mecanismo de delegación de privilegios.

Por un lado, el esquema RBAC permitirá separar la gestión de la pertenencia de los usuarios a roles de la asignación de privilegios concretos a dichos roles, lo cual simplifica enormemente la gestión de las autorizaciones. La pertenencia se implementará mediante la emisión de certificados SPKI de identidad mientras que la asignación de privilegios a los roles se realizará mediante certificados SPKI de atributo. Los certificados de pertenencia se almacenarán siempre en la tarjeta inteligente del usuario asociado, el cual dispondrá normalmente de un certificado de identidad X.509 emitido por la PKI presentada en el capítulo 3. Los certificados de atributo podrán ser almacenados tanto en las tarjetas de los usuarios como en los propios TICAs. Esto es debido al hecho de que los permisos asociados a un rol suelen cambiar menos frecuentemente que el conjunto de personas que lo componen, por lo que la opción de almacenarlo en los TICAs no presentará normalmente problemas de actualización y liberará a los usuarios de agotar el escaso espacio de almacenamiento proporcionado por las tarjetas inteligentes.

Por otro lado, la delegación de privilegios permite que los TICAs asignen la responsabilidad de la gestión de los mismos a entidades externas que actuarán como autoridades. De esta forma, el TICA no es sólo el punto de cumplimiento de la política de control de acceso sino también el inicio de la cadena de autorización. El dispositivo es capaz de tomar

decisiones de autorización analizando tanto los certificados de credencial que mantiene almacenados como los presentados por los usuarios. El esquema general utilizado, mostrado en la figura 6.13, se corresponde con el modelo de gestión propio del sistema DCMS.

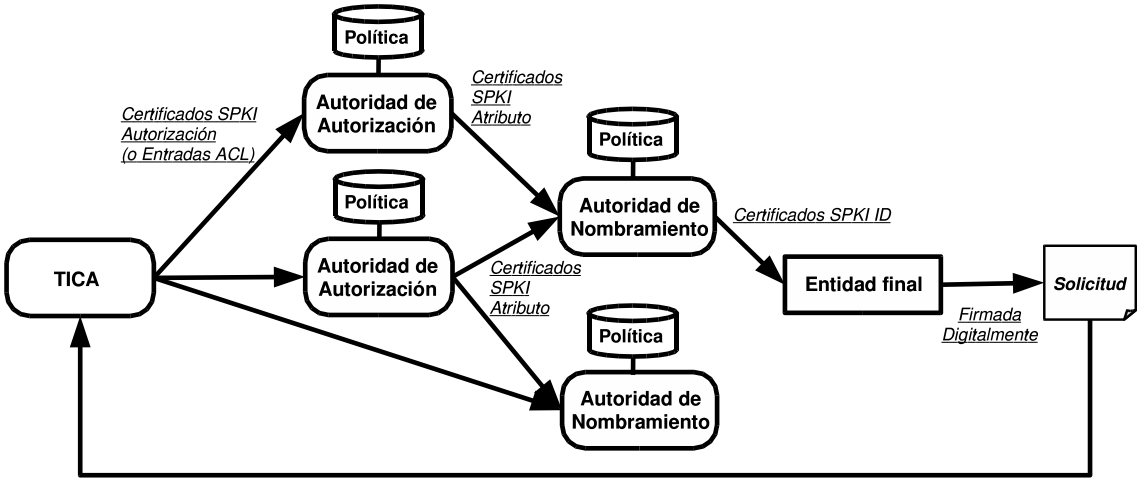

Figura 6.13: Arquitectura del sistema de control de acceso descentralizado

Como se puede apreciar, el usuario presenta al TICA solicitudes de acceso firmadas digitalmente mediante su clave privada. A continuación, el TICA intenta construir una cadena de certificados que partiendo de él mismo sea capaz de llegar hasta la clave pública del usuario pasando por las distintas autoridades del sistema. En el caso de que logre hallar una prueba de autorización, el dispositivo lleva a cabo la acción correspondiente y almacena dicha prueba con el propósito de registrar evidencias que puedan ser utilizadas en caso de incidencia.

6.4.3 Implementación de la propuesta distribuida

La aplicación de control de acceso del TICA está también basada en la arquitectura CDSA. En concreto, el grupo de investigación ANTS desarrolló una librería multimódulo [173] que actuaba como proveedor de servicios criptográficos (CSP) y como módulo de almacenamiento de datos (DL).

El módulo CSP proporciona soporte para el uso de las tarjetas inteligentes y está encargado de la realización de las operaciones de firma digital que implican el uso de la clave privada contenida en la tarjeta. La gran ventaja de este enfoque es que la aplicación del TICA accede a la tarjeta de forma transparente a través de la API proporcionada por CSSM, sin que tenga que conocer ninguna interfaz de programación específica desarrollada para tal efecto.

Por otro lado, dado que los certificados de credencial SPKI se encuentran almacenados en la tarjeta, era necesario desarrollar un módulo DL que ofreciera la posibilidad de recuperar esos datos a través de la API CSSM.

6.4.4 Aplicación de la metodología e integración con DCMS

Para mostrar cómo es posible hacer uso de la metodología y de DCMS para poner en marcha el sistema de control de acceso físico anteriormente descrito, se planteará aquí un escenario concreto formado por un conjunto de TICAs, roles y usuarios. A continuación, se aplicará la metodología para determinar tanto las políticas de control de acceso del sistema como las entidades encargadas de su cumplimiento. Una vez finalizados los procedimientos, el sistema DCMS podrá ser utilizado para generar los certificados en base a los cuales tomarán sus decisiones los TICAs del sistema.

Escenario a gestionar

El escenario de partida está formado por cuatro TICAs distintos, los cuales pueden realizar operaciones relacionadas con la apertura de puertas, control horario, control de iluminación y gestión de alarmas. Todos los TICAs están localizados en la Facultad de Informática, concretamente en la puerta principal, puerta posterior, puerta del laboratorio de investigación del departamento DITEC y puerta del laboratorio de investigación del departamento DIIC.

Por otro lado, los usuarios se agrupan en siete roles distintos. Con el fin de acotar la extensión de este apartado, supondremos que la jerarquía formada por dichos roles ya ha sido definida mediante la aplicación de los procedimientos correspondientes al bloque NMS de la metodología. Dicho bloque de procedimientos será utilizado aquí sólo para gestionar la pertenencia de los usuarios a los roles. La figura 6.14 muestra la jerarquía concreta de roles.

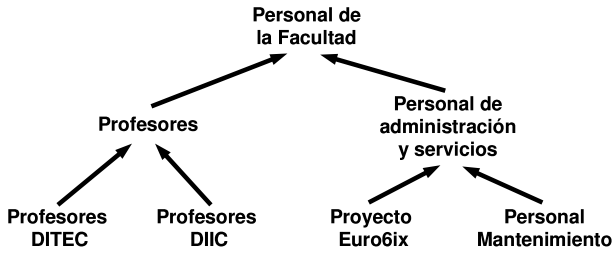

Figura 6.14: Roles de la jerarquía

Aplicación de los procedimientos de nivel 0

A continuación se desglosa cómo se han aplicado los procedimientos contenidos en este nivel:

- *Especificación de los recursos a proteger y sus operaciones.* Los recursos que se desea gestionar son los propios TICAs del sistema. Para especificarlos se llevarán a cabo los siguientes pasos:

  - *Establecimiento del esquema de identificación de recursos.* Para identificar a los dispositivos se utilizará una notación basada en la concatenación del nombre de la facultad en la cual se encuentra el TICA y de la localización física del mismo. Por ejemplo, el dispositivo situado en la puerta principal se denotará con la expresión `(informatica/principal)`. El resto de dispositivos tienen los identificadores `(informatica/posterior)`, `(informatica/sala-diic)` e `(informatica/sala-ditec)`. Esta notación nos permite hacer referencia a todos los TICAs de una misma facultad mediante el uso de expresiones del tipo `(* prefix informatica/)`.
  - *Establecimiento de las operaciones sobre los recursos.* En este caso concreto podemos identificar cinco operaciones a controlar. La apertura de las puertas se representa con la expresión `(open-door (* range time (hh:mm:ss) (hh:mm:ss)))`, la cual permite especificar el intervalo horario durante el cual podrá realizarse la apertura de la puerta controlada por el TICA. El control horario de los trabajadores tiene asociadas dos operaciones `(start-work)` y `(finish-work)`, utilizadas respectivamente para fichar al inicio y al final de la jornada laboral. Por último, las operaciones de mantenimiento son representadas mediante las expresiones `(lighting-control)` y `(alarm-management)`.
  - *Publicación de la especificación.* La representación de los recursos y sus operaciones se hará pública mediante la implementación de una aplicación generadora de tags, tal y como se comentó en la sección 6.3.2. Dichos tags contendrán siempre la cadena *dkronos*, la cual hace referencia al tipo de privilegios que se están gestionando (el nombre *dkronos* está derivado a partir del nombre KRONOS con el cual se denominó al proyecto mediante el cual se construyeron los TICAs).

- *Determinación de los controladores.* En este caso concreto, los controladores son los propios TICAs ya que serán los encargados de verificar las solicitudes presentadas por los usuarios.

- *Determinación de las autoridades.* El siguiente paso es la delegación de la gestión de privilegios por parte de cada uno de los TICAs. Para ello hay que determinar las entidades que actuarán como autoridades y generar la correspondiente lista de control de acceso o certificado de autorización, generación que puede realizarse respectivamente con la aplicación generadora de políticas (ver sección 6.3.5) o con la aplicación de las autoridades (ver sección 6.3.6). En este caso concreto, la determinación de autoridades se ha realizado de la siguiente forma: los TICAs de la puerta principal y posterior delegan la gestión de todas las operaciones en una autoridad que llamaremos *AA-Facultad*; el TICA del laboratorio de DITEC delega en *AA-DITEC* la gestión de la operación de apertura de puertas ; por último el TICA del laboratorio de DIIC delega también la gestión de la apertura en la autoridad *AA-DIIC*.

En el caso de que la delegación se realizara mediante la construcción de listas de control de acceso, estas tendrían una estructura similar a la especificada en la figura 6.15.

```
(acl
  (subject AA − Facultad)
  (tag (dkronos
         (* set (informatica/principal) (informatica/posterior))
         (*)))
)
(acl
  (subject AA − DITEC)
  (tag (dkronos (informatica/sala-ditec) (open-door (*))))
)
(acl
  (subject AA − DIIC)
  (tag (dkronos (informatica/sala-diic) (open-door (*))))
)
```

Figura 6.15: Delegación de los controladores mediante ACLs

Aplicación de los procedimientos del bloque AMS

El siguiente paso es la determinación de la política de autorización por parte de las autoridades derivadas de la aplicación de los procedimientos de nivel 0. Para ello se emplean los siguientes niveles del bloque AMS:

- *Nivel 1. Identificación de las relaciones entre las operaciones.* En el caso concreto del entorno que se está diseñando, las cinco operaciones posibles pueden agruparse en tres grupos distintos: apertura de puertas, control horario y mantenimiento. Además, en el caso de la operación de apertura hay que determinar las franjas horarias más comunes a emplear, por ejemplo la correspondiente con el horario de apertura de la facultad.

- *Nivel 2. Asignación de permisos a entidades receptoras.* Cada una de las autoridades seguirá un criterio de asignación distinto en función de los permisos que esté gestionando. En este caso concreto, se realizarán las siguientes asignaciones (los periodos de validez se omiten por simplicidad):

  - *AA-Facultad.* Establecerá la siguiente política de autorización: el personal de la facultad podrá realizar las operaciones de control horario en cualquiera de los TICAs; el personal de mantenimiento podrá solicitar las operaciones de control de iluminación y de gestión de alarmas tanto mediante el dispositivo de la puerta principal como mediante el de la puerta posterior; por último, el personal de la facultad podrá abrir la puerta principal a cualquier hora. Las s-expresiones que codifican estas condiciones se muestran en la figura 6.16
  - *AA-DITEC.* La política de autorización de esta autoridad está compuesta de una única autorización, la que permite a los profesores de DITEC abrir la puerta de su laboratorio de investigación a cualquier hora.

– *AA-DIIC.* Esta autoridad autoriza tanto a los profesores de DIIC como a los miembros del proyecto Euro6ix a acceder a su laboratorio a cualquier hora del día.

```
(cert-request
  (issuer AA − Facultad)
  (subject Personal)
  (tag (dkronos (*) (* set (start-work) (finish-work))))
)
(cert-request
  (issuer AA − Facultad)
  (subject Personal)
  (tag (dkronos (informatica/principal) (open-door)))
)
(cert-request
  (issuer AA − Facultad)
  (subject Mantenimiento)
  (tag (dkronos
         (* set (informatica/principal) (informatica/posterior))
         (* set (lighting-control) (alarm-management))))
)
```

Figura 6.16: S-expresiones de la autoridad AA-Facultad

- *Nivel 3. Determinación de los solicitantes y periodos de solicitud.* Al tratarse de privilegios asignados a roles, es necesario especificar qué entidad, o conjunto de entidades, estará autorizada a solicitar los certificados asociados a los privilegios. En este caso, se supondrá que todas las autoridades consideran que los solicitantes válidos son las autoridades de nombramiento encargadas de gestionar la pertenencia a los roles autorizados. Se permitirá además que dichas autoridades puedan delegar dicha posibilidad de solicitud a otras entidades.

- *Nivel 4. Modos de acceso a la autoridad.* En este caso supondremos que todas las autoridades han configurado su aplicación de forma que acepte conexiones AMBAR procedentes de cualquier origen. Además, las sesiones deben ser identificadas y deben utilizar el método de distribución *push-asserts*.

Aplicación de los procedimientos del bloque NMS

Por último, deben aplicarse los procedimientos del bloque NMS con el fin de establecer las políticas de las distintas autoridades de nombramiento del sistema. Se supone que dentro del sistema se han definido tres autoridades de nombramiento distintas: *NA-Facultad* se encarga de gestionar la jerarquía de roles mostrada en la figura 6.14 y la pertenencia al grupo *Personal de Mantenimiento*; por otro lado, *NA-DITEC* tiene la responsabilidad de

especificar y emitir los certificados de pertenencia al rol *Profesores DITEC*; finalmente, la autoridad *NA-DIIC* gestiona la pertenencia tanto al rol *Profesores DIIC* como al rol *Proyecto Euro6ix*. Todas ellas llevarán a cabo los procedimientos de los siguientes niveles para especificar sus políticas:

- *Nivel 1. Identificación del conjunto de elementos.* En primer lugar, es necesario delimitar el conjunto de entidades que se encuentra dentro de la influencia de cada autoridad. Para el caso de *NA-Facultad* serán todas aquellas personas que pertenecen al personal de administración y servicios, como los administradores del centro de cálculo, personal de secretaría y conserjes. En el caso de las otras dos autoridades dicho conjunto abarca tanto a los profesores de cada departamento como al conjunto de becarios, contratados con cargo a proyectos de investigación, profesores visitantes y alumnos de proyecto fin de carrera.

- *Nivel 2. Determinación de la pertenencia.* Durante esta etapa, las autoridades deben decidir qué usuarios pertenecerán a cada uno de los roles que gestionan. Para este ejemplo, se supondrá que el rol *Personal de Mantenimiento* estará formado por los conserjes de la facultad, y que los roles de los profesores estarán constituidos exclusivamente por los profesores de cada departamento. Por último, formarán parte del *Proyecto Euro6ix* aquellos contratados y becarios que se encuentren implicados en el proyecto. Todas estas decisiones podrán ser plasmadas en s-expresiones haciendo uso de la aplicación de gestión de identificadores (ver sección 6.3.4).

- *Nivel 3. Determinación de los solicitantes y periodos de solicitud.* Se supondrá que las tres autoridades han decidido que el conjunto de solicitantes válidos esté formado por las entidades pertenecientes a cada uno de los roles, es decir, los usuarios podrán solicitar personalmente sus certificados. En consecuencia, durante la construcción de la política mediante la aplicación generadora, no será necesario especificar el conjunto de solicitantes válidos por ser equivalente al conjunto de receptores válidos especificados en las s-expresiones de tipo *cert-request*.

- *Nivel 4. Modos de acceso a la autoridad.* En este caso supondremos que todas las autoridades han configurado su aplicación para que sólo acepte conexiones AMBAR que procedan de puntos de acceso. Además, las sesiones deben ser identificadas y deben utilizar el método de distribución. *push-asserts*.

6.4.5 Conclusiones obtenidas

En vista de las características de las dos propuestas presentadas, y del uso que se ha realizado tanto de la metodología como del sistema DCMS a la hora de poner en marcha la propuesta descentralizada, es posible extraer varias conclusiones:

- La propuesta basada en el uso de certificados de credencial genera datos que pueden ser utilizados como evidencias a la hora de garantizar el no repudio del solicitante.

- La escalabilidad del sistema se ve mejorada mediante la aplicación del modelo RBAC y el uso de certificados SPKI, ya que es posible dividir las responsabilidades de gestión de forma natural entre las distintas entidades del sistema.

- La versión distribuida es capaz de proporcionar servicio en aquellos entornos en los cuales la conectividad a una red de comunicaciones no es posible.

- La aplicación de la metodología permite poner en marcha el sistema de forma estructurada y clara. Además, la aplicación de los procedimientos se puede realizar de forma inmediata mediante el uso de las aplicaciones que forman parte del sistema DCMS.

En resumen, el uso de la infraestructura de autorización amplía las posibilidades de los sistemas de control de acceso físico centralizados, todo ello desde un punto de vista estructurado y metodológico que simplifica además el proceso de diseño del sistema.

6.5 Integración en un entorno de suscripción electrónica

Además del entorno analizado en la sección anterior, se creyó conveniente mostrar cómo la infraestructura de autorización puede integrarse también en escenarios más enfocados al comercio electrónico, como es el caso de la fidelización de clientes.

La fidelización de clientes es uno de los principales objetivos de la mayoría de las empresas de cara a asegurar una determinada cuota de mercado. Las empresas buscan consolidar en el tiempo sus comunidades de clientes, lo cual suele repercutir positivamente en ambas partes. Para ello, se han diseñado multitud de fórmulas que tienen como objetivo poner al alcance de los clientes ventajas adicionales respecto al cliente eventual, casi siempre enfocadas al plano económico (descuentos especiales, ofertas, facilidades de pago, etc.). Uno de los mecanismos que más se ha empleado para este tipo de propósitos es la posibilidad de pagar una cuota de suscripción que asocie al cliente a un determinado grupo beneficiario. Dependiendo de la cuota, el cliente disfrutará de un conjunto de ventajas adicionales, las cuales serán mejores cuanto más alta sea la cuota de suscripción.

En esta sección, se expondrá una propuesta que proporciona los mecanismos necesarios tanto para realizar el pago de la cuota de suscripción como para hacer uso de la misma a la hora de comprar de forma electrónica a través de una red de comunicaciones. El sistema está basado en el uso del protocolo SPEED [174], desarrollado en el marco del proyecto PISCIS. SPEED proporciona los medios necesarios para realizar de forma segura pagos basados en el uso del monedero electrónico, distribuir de forma electrónica el producto solicitado, generar toda la información necesaria para resolver posibles disputas que pudieran surgir en el futuro e intercambiar información relativa a atributos o credenciales de las entidades participantes. Como se verá, el uso de dicho protocolo, a través de su integración con la infraestructura de clave pública y el sistema DCMS, permite diseñar un sistema de suscripción electrónica especialmente enfocado a sistemas distribuidos en los cuales no es apropiado realizar una gestión centralizada de las suscripciones.

6.5.1 El protocolo SPEED

En los últimos años, la comunidad científica ha ido tomando conciencia de la necesidad de diseñar e implementar nuevas formas de pago adaptadas al comercio electrónico que hagan un buen uso de la tecnología existente y proporcionen al usuario un cierto grado de percepción de seguridad. En general, cada uno de estos sistemas propuestos intenta satisfacer las necesidades del entorno en el cual está definido, y por tanto no podemos considerar que haya un sistema válido para cualquier entorno. Algunas de estas propuestas [142] han demostrado ser lo suficientemente seguras y flexibles, si bien no han alcanzado un alto grado de adopción en mercados reales.

Durante el desarrollo del proyecto PISCIS [53] se identificó el conjunto de características de seguridad que debería reunir un protocolo de pago de cara a poder ser utilizado en un escenario real de comercio electrónico. Entre dichos requisitos se encontraba la siguiente lista:

- El sistema de pagos debe ser capaz de ofrecer medios para negociar el precio de los productos o servicios ofrecidos por los comerciantes.
- La entrega electrónica de los bienes adquiridos debe formar parte del sistema.
- El protocolo debe estar basado en estándares de seguridad reconocidos. Este requisito incrementa la sensación de seguridad percibido por los usuarios, y garantiza un diseño basado en propuestas debatidas y probadas por la comunidad científica.
- Se debe disponer de elementos de arbitraje capaces de ejercer como mediadores y como entidades de confianza a la hora de mediar en conflictos y situaciones excepcionales.

Aunque algunos de estos requisitos ya habían sido satisfechos por algunas de las propuestas ya existentes, con el fin de proporcionar una respuesta común a todos ellos dentro del marco de trabajo del Proyecto PISCIS, se definió un nuevo sistema de pago llamado SPEED (Smartcard-based Payment with Encrypted Electronic Delivery), el cual proporciona, como su propio nombre indica, un sistema de pago basado en monedero electrónico para tarjeta inteligente con entrega cifrada de bienes. Aunque la información en detalle del protocolo puede encontrarse en [174], en esta sección haremos una breve descripción de sus características principales, participantes y modelo de compra.

Visión general

Una transacción SPEED transfiere bienes electrónicos desde un vendedor a un cliente, debitando el monedero electrónico del cliente e incrementando el saldo de la cuenta del vendedor por el valor de producto. El diseño de SPEED consiste en una serie de fases que incluyen la negociación del precio, la entrega del producto y su pago. Además, hay dos modos posibles de operación: el modo normal incluye la capacidad de negociación del precio del producto, y ha sido diseñado para proporcionar el mayor número de características de

seguridad (como por ejemplo la prevención de ataques de denegación de servicio y la autenticación completa de las partes participantes antes del suministro del producto); el modo rápido de operación está compuesto por un número menor de mensajes que el modo normal, y está pensado para la venta de bienes de menor tamaño o escenarios con menores requisitos de seguridad.

Su diseño se basa en el uso de estándares como ASN.1 [105] para la especificación de la estructura de los mensajes, PKCS#7 [118] como formato criptográfico para el intercambio de información protegida, certificados X.509v3 [99] para la identificación de los participantes en el escenario de compra y WG10 [39] como sistema estándar de monedero electrónico.

Participantes

El modelo de negocio de SPEED está compuesto por tres entidades principales: el cliente, el comerciante y el intermediario (broker). El broker gestiona las cuentas de los comerciantes (y opcionalmente las de los clientes) y mantiene el conjunto de módulos de seguridad que realizan las operaciones de decremento sobre el monedero electrónico del cliente. Esta entidad no interviene hasta la fase de pago, una vez que el cliente envía la solicitud de transacción.

En una primera instancia, el cliente y el vendedor acuerdan el producto a comprar y su precio, lo cual puede ser llevado a cabo después de una fase de negociación de ofertas que es opcional. El producto se transmite al cliente cifrado con una clave simétrica, que sólo le es proporcionada una vez que el pago correspondiente se ha materializado. Cuando dicho pago se ha realizado, tanto el cliente como el vendedor obtienen una prueba del resultado de la transacción (denominada recibo).

Modelo de compra

La figura 6.17 muestra un esquema global de las comunicaciones que componen una secuencia de compra de SPEED en el modo normal de operación. Los mensajes 1, 2 y 3, intercambiados entre el cliente y el comerciante, constituyen la fase de negociación del producto. El mensaje 4 contiene el producto cifrado con una clave simétrica generada aleatoriamente por el comerciante, y que será proporcionada al cliente una vez que el pago se haya realizado (mensajes 5 y 6). El broker y el cliente intercambian una serie de mensajes adicionales destinados a realizar el proceso de decremento del monedero electrónico (en la figura estos mensajes están representados por la línea punteada). Todas las comunicaciones están protegidas frente a ataques de entidades externas haciendo uso de criptografía simétrica principalmente.

En relación con el mecanismo de suscripción electrónica, es importante analizar en detalle el formato del primer mensaje del protocolo (*NegotiationRequest*), el cual está encuadrado dentro de la fase de negociación del precio del producto. Se trata de un mensaje firmado digitalmente que el cliente envía al vendedor para preguntar o proponer el precio de un determinado producto. Su estructura es la siguiente:

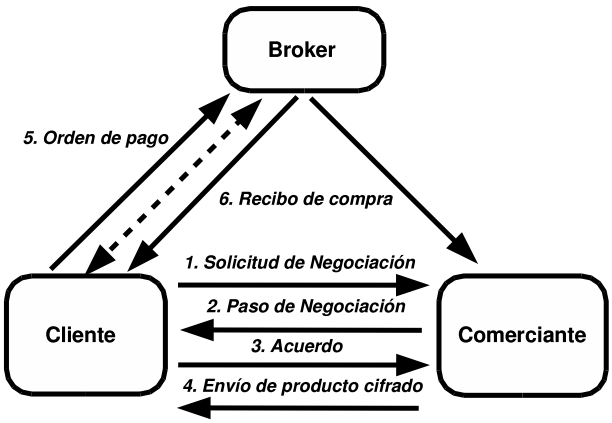

Figura 6.17: Modelo de compra de SPEED

$$1 \quad \mathrm{C} \Rightarrow \mathrm{V} \quad \mathit{NegotiationRequest} \quad \{\{NID, SeqN, ProductID, [Price], VendorID, EnKey, SignKey, Flag, Credential\}_{C^{-1}}\}_V$$

Entre todos los campos que forman este mensaje, son de especial interés para esta sección los relacionados con la descripción del producto (*ProductID*), el precio (*Price*) y las credenciales (*Credential*):

- *ProductID (Product Identifier).* Se trata de una cadena de octetos que representa al producto a comprar. Es totalmente dependiente del entorno de aplicación en el cual se encuentre ubicado el protocolo, y en este caso concreto servirá tanto para designar a los certificados de pertenencia como a los productos finales a comprar.

- *Price.* Es el precio que el cliente está dispuesto a pagar por el producto. Se trata de un elemento opcional ya que es posible que el cliente no conozca de antemano el precio.

- *Credential.* Durante el proceso de diseño del protocolo SPEED ya se consideró la posibilidad de proporcionar un soporte especial que permitiera realizar la transmisión de información relativa a credenciales como parte del protocolo. El propósito genérico de este tipo de información era la modificación de la estrategia de negociación del proveedor, un cambio de estrategia que implicara ciertas ventajas para el cliente. Dicha información se incluye dentro del campo *Credential* y su formato concreto es totalmente transparente para el protocolo ya que éste sólo se encarga de transmitir la información sin interpretar su estructura.

Tal y como se verá más adelante, el uso de estos campos del protocolo permitirá diseñar tanto el escenario de solicitud de suscripción como el de disfrute de la misma. El resto de los campos así como de los mensajes del protocolo son totalmente independientes del mecanismo aquí diseñado.

6.5.2 Integración de la PKI

Cada participante de SPEED (clientes, comerciantes y broker) poseerá una clave privada RSA almacenada en su tarjeta inteligente y un certificado X.509 emitido por la infraestructura de clave pública presentada en el capítulo 3. De hecho, SPEED asume la existencia de relaciones de confianza entre las entidades participantes. Los brokers son considerados las entidades de mayor confianza, seguidos de los comerciantes y por último de los clientes (en los cuales podría no tenerse ningún tipo de confianza). Los brokers juegan el rol de participar como entidades intermediarias y los comerciantes poseen relaciones a largo plazo con los brokers de la misma forma que lo harían con un banco. La reputación del broker dentro del sistema es un punto importante, ya que resulta vital que asuman su papel según lo establecido con el fin de no perder la confianza del resto de las entidades participantes.

Por tanto, la relación con la PKI tiene dos puntos de unión claramente diferenciados. Por un lado, todas las entidades participantes deben obtener un certificado digital X.509 que les permita establecer comunicaciones confidenciales y autenticadas entre ellos. La identidad digital contenida en dichos certificados será vital a la hora de resolver posibles disputas que pudieran surgir en el futuro. Además, en el caso de los clientes, este proceso de certificación implica el uso de tarjetas inteligentes como elementos tanto contenedores de información sensible como capaces de realizar operaciones criptográficas.

Por otro lado, la infraestructura de clave pública proporciona los mecanismos de validación en línea del estado de los certificados implicados en las transacciones. Estos servicios son de vital importancia a la hora de evitar posibles fraudes derivados del uso malintencionado de información criptográfica ajena.

6.5.3 Implementación de la suscripción electrónica mediante certificados de credencial

El escenario de suscripción electrónica realizado está basado en la existencia de varias categorías de suscripción, gestionadas posiblemente por entidades distintas, y varios proveedores de servicios que ofrecen ventajas económicas a los suscriptores, las cuales variarán dependiendo del tipo de suscripción. Tal y como se muestra en la figura 6.18, los distintos proveedores de contenidos y servicios mantienen una relación de colaboración con los diferentes gestores de suscripciones. Los términos de dicha colaboración especificarán qué tipo de ventajas aplicará el proveedor a aquellos miembros de alguno de los grupos que controla el gestor (descuentos en algunos de sus productos, facilidades de pago, regalos, etc.).

Los usuarios finales tienen dos puntos de conexión con el sistema de suscripción. Por un lado deberán pagar la cuota correspondiente y obtener el justificante de suscripción. Por otro lado, deberán transmitir dicho justificante junto con cada solicitud de compra realizada a alguno de los proveedores relacionados con el grupo de suscripción.

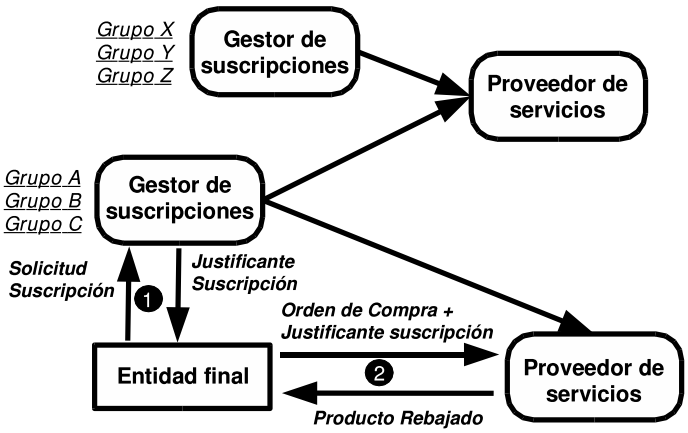

Figura 6.18: Modelo de suscripción electrónica

Solicitud de suscripción

La solicitud de suscripción se realiza a través del protocolo SPEED. Dado que dicho protocolo fue diseñado especialmente para el pago de productos digitales, es posible especificar cuál es el grupo concreto al cual se desea pertenecer, realizar el pago correspondiente y obtener el justificante de suscripción.

En relación con lo visto en el apartado 6.5.1, la especificación del grupo está contenida en el campo *ProductID* del mensaje *NegotiationRequest*, el cual contendrá una s-expresión de tipo *issuer* [68] con la clave pública del gestor de suscriptores y el identificador del grupo. La suscripción, cuyo precio está especificado en el campo *Price*, podría estar sujeta al cumplimiento de ciertas condiciones de autorización (por ejemplo, ser mayor de edad, tener crédito disponible o cualquier otro criterio dependiente del entorno). En dicho caso, las evidencias necesarias para demostrar la posibilidad de ingreso en el grupo podrían transmitirse en el campo *Credential*.

Una vez que el broker transmite el recibo, el usuario final puede descifrar el justificante de suscripción, el cual recibió como parte del cuarto mensaje del protocolo. Dicho justificante estará representado mediante un certificado SPKI de identidad, el cual ligará la clave pública del usuario con el grupo de suscripción, todo ello durante el periodo de tiempo en el cual sea efectiva dicha asociación.

El uso de certificados digitales permite descentralizar el sistema puesto que no es necesario mantener almacenadas de forma central todas las relaciones existentes entre los usuarios del sistema y los grupos. En contraposición, la información de suscripción puede ser transmitida por los propios usuarios a la hora de realizar sus compras y será considerada válida por aquellas entidades que tienen una relación de confianza con la entidad emisora.

Presentación de justificantes

Tras la adquisición del justificante, el usuario final puede proceder a la compra de productos de alguno de los proveedores de servicios con los cuales tenga acuerdos el gestor de

suscripciones. La descripción de dichos productos se incluirá en el campo *ProductID* del primer mensaje.

Por otro lado, haciendo uso del campo *Credential* del mensaje *NegotiationRequest*, el cliente puede transmitir al proveedor el justificante de suscripción, es decir, el certificado de identidad SPKI que le asocia a un determinado grupo. Una vez que el proveedor recibe el certificado, verifica su estado y determina la reducción en el precio del producto que ha sido solicitado. Dicha reducción depende del acuerdo establecido entre el proveedor y la entidad gestora de las suscripciones, el cual especificará el conjunto de productos afectados, los porcentajes de descuento u otras medidas favorecedoras, y el periodo de tiempo durante el cual permanecerá en vigor. En concreto, en el prototipo desarrollado, los proveedores de contenido actúan como autoridades de autorización, emitiendo certificados SPKI de atributo que especifican los acuerdos de los cuales se pueden beneficiar aquellos usuarios pertenecientes al grupo de suscriptores al que hace referencia el certificado. Dichos certificados pueden emplearse como prueba de los acuerdos alcanzados entre las entidades del sistema, lo cual es especialmente útil a la hora de decidir el grupo de suscriptores al cual se desea pertenecer.

6.5.4 Conclusiones obtenidas

Uno de los principales criterios de diseño del protocolo SPEED fue la posibilidad de ofrecer un mecanismo de negociación del precio de los productos que formara parte del propio protocolo, lo cual permitiría emplearlo en escenarios basados en entidades mediadoras y gestionar comunidades con distintas prioridades. Respecto a este último aspecto, el uso de certificados de credencial ha permitido introducir de forma estructurada un servicio de fidelización que aprovecha parte de la funcionalidad ofrecida por el protocolo. En consecuencia, el uso de la PKI y de la infraestructura de autorización aportan los mecanismos necesarios de confidencialidad, integridad, no repudio, autenticación y autorización.

6.6 Evaluación de los componentes de la infraestructura de autorización

Con el fin de evaluar el rendimiento proporcionado por los dos componentes fundamentales de la infraestructura, la implementación del marco AMBAR y el sistema DCMS, se realizó un conjunto de pruebas destinadas a obtener conclusiones respecto a los siguientes tres aspectos:

- Comparar el rendimiento ofrecido por el protocolo AMBAR frente a uno de los protocolos de seguridad más utilizados, el protocolo SSL. Dicho análisis abarca tanto la fase de negociación o *handshake* como el envío posterior de datos.

- Averiguar cuál es la sobrecarga que introduce la arquitectura CDSA frente a implementaciones que hagan uso de la funcionalidad criptográfica de forma directa.

- Averiguar cuál es el peso tanto de las comunicaciones AMBAR como del procesamiento DCMS a la hora de tramitar las solicitudes de certificación o reducción, es decir, determinar la contribución de cada elemento al tiempo total.

En los siguiente apartados se presenta tanto el entorno en el cual se realizaron las pruebas como los resultados obtenidos para cada uno de los objetivos fijados.

6.6.1 Entorno de evaluación

Los elementos participantes en las pruebas son un cliente que solicita el acceso a un recurso y el controlador correspondiente. Salvo en las pruebas que involucran al sistema DCMS, las implementaciones de tanto el solicitante como el controlador son básicas, es decir, incluyen exclusivamente la funcionalidad necesaria para establecer canales de comunicación seguros e intercambiar información de autorización.

Las pruebas se han desarrollado en una red de área local Ethernet (10 Mbps) conmutada. Los ordenadores donde se ejecutan los procesos cliente y servidor son Intel Pentium III con 192 MB de memoria principal. El sistema operativo empleado fue Linux (Kernel 2.4.2-2) y todas las aplicaciones fueron compiladas mediante GCC 2.96. Por último, el software criptográfico utilizado fue la implementación de Intel de la arquitectura CDSA (versión 3.14) y las librerías OpenSSL (versión 0.9.6).

Por otro lado, los parámetros relacionados con la configuración de sesiones AMBAR y con las características de la información de autorización intercambiada pueden observarse en la tabla 6.1.

Parámetro	**Valor**
Cifrador simétrico	RC2 (128 bits)
Cifrador asimétrico	RSA (1024 bits)
Algoritmo firma digital	DSA (1024 bits)
Tamaño de los certificados SPKI	1350 bytes
Tamaño de los certificados X.509	700 bytes
Tamaño de la solicitud	1700 bytes
Tamaño de la política	600 bytes
Método de distribución	PUSH-ASSERTS
Modo de conexión	Identificado

Tabla 6.1: Parámetros de la evaluación

6.6.2 Evaluación de la fase de negociación

La primera prueba realizada fue la evaluación del tiempo de ejecución asociado a la fase de negociación de sesiones AMBAR. Para ello, se realizó una comparativa respecto a la fase *handshake* del protocolo SSL. La figura 6.19 muestra los resultados obtenidos.

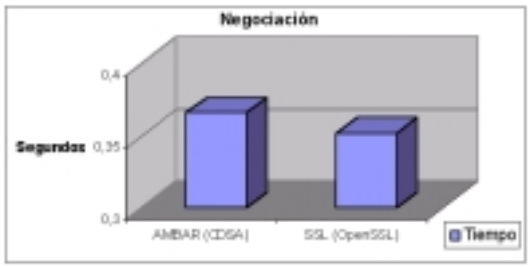

Figura 6.19: Evaluación de la fase de negociación

Como puede apreciarse, el tiempo de ejecución de la negociación de sesiones AMBAR es un 4% superior al empleado por SSL para negociar una sesión de similares características. Hay varias cuestiones que justifican esta ligera diferencia:

- A pesar de que AMBAR tiene una carga criptográfica menor en la fase de negociación que SSL, el tiempo reflejado en la gráfica incluye el intervalo necesario para inicializar los módulos que forman parte de la arquitectura CDSA. Dicho proceso de inicialización implica tanto la carga como la verificación de cada uno de los módulos funcionales de la arquitectura.

- Las llamadas a las funciones de la arquitectura CDSA introducen cierta sobrecarga a la hora de acceder a la funcionalidad ofrecida por alguno de sus módulos. Debe tenerse en cuenta que toda solicitud de servicio pasa por el gestor CSSM, el cual encamina la llamada al gestor del módulo correspondiente y éste a su vez a la implementación concreta del módulo.

A pesar de lo anterior, el resultado obtenido puede considerarse muy destacable ya que es muy similar al rendimiento ofrecido por la implementación de OpenSSL, sobre todo considerando que dicha librería ha estado sometida durante varios años a un proceso continuo de optimización y mejora.

6.6.3 Evaluación de la fase de solicitud y respuesta

Una vez analizada la fase de negociación, el siguiente paso es evaluar el rendimiento relacionado con el intercambio de información de autorización. Para ello, se analizarán dos situaciones distintas que permiten extraer conclusiones acerca de dos aspectos diferentes: por un lado, el rendimiento obtenido en función del número de certificados de credencial transmitidos; por otro lado, el tiempo total asociado a la transmisión de recursos de distinto tamaño. Los datos ofrecidos no incluyen la porción de tiempo necesaria para realizar el cálculo de autorización sino que pretenden ilustrar cuál es la sobrecarga introducida por la fase de intercambio. Más adelante se contrastará este tiempo de transmisión con el tiempo de tramitación de la solicitud.

La figura 6.20 muestra los tiempos de acceso a un recurso, en este caso un certificado SPKI de 1350 bytes, tras el envío de un número variable de credenciales que oscila entre

1 y 16. El intercambio mediante SSL simula el funcionamiento de los módulo RM y ARM del marco AMBAR, puesto que SSL no realiza ningún tipo de distinción respecto al tipo de información que transmite. A partir de la figura puede deducirse que los resultados obtenidos son muy similares (la variación máxima son 10 ms) y que las diferencias existentes están asociadas de nuevo a la sobrecarga introducida por las llamadas CDSA.

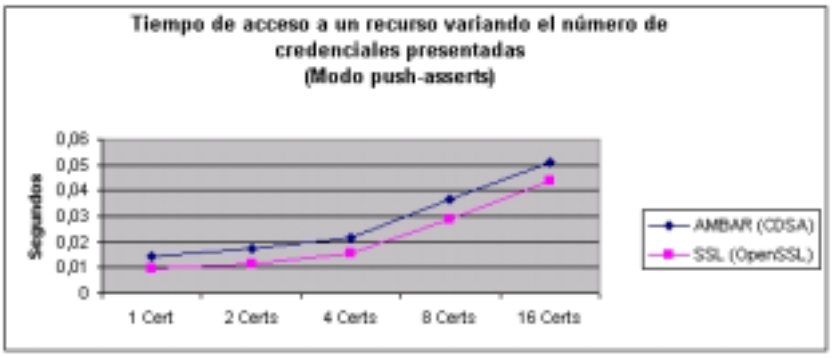

Figura 6.20: Tiempo de acceso en función del número de credenciales

Por otro lado, la figura 6.21 nos permite obtener información acerca de dos cuestiones distintas: en primer lugar, contrasta el rendimiento obtenido por la implementación de AMBAR realizada sobre CDSA frente a una implementación del marco que fue realizada directamente sobre OpenSSL, lo cual permitirá cuantificar la sobrecarga introducida por el *middleware*; en segundo lugar, analiza el tiempo asociado a la transmisión de recursos de mayor tamaño, permitiendo así averiguar si el rendimiento del protocolo se degrada cuando la cantidad de información a transmitir se incrementa.

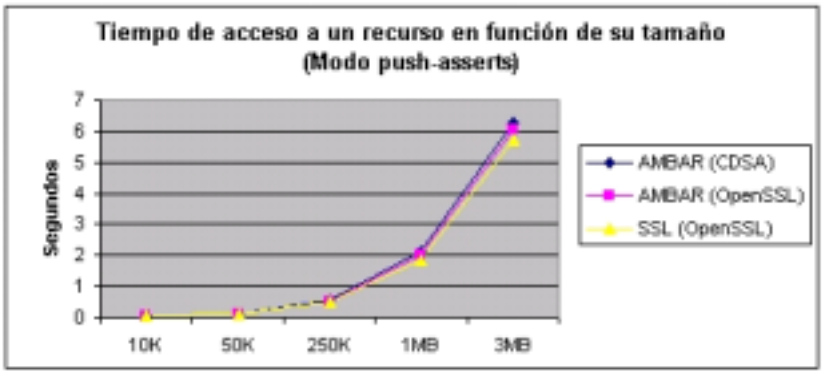

Figura 6.21: Tiempo de acceso en función del tamaño del recurso

En relación con la sobrecarga introducida por CDSA respecto a la implementación de AMBAR basada en OpenSSL, es posible observar que ésta es prácticamente inapreciable ya que apenas supera el 4 % en el peor de los casos. Este hecho demuestra que las ventajas

introducidas por la integración de arquitecturas como CDSA no se ven ensombrecidas por un decremento sustancial en el rendimiento ofrecido por las aplicaciones.

En lo que respecta al comportamiento del protocolo a la hora de transmitir recursos de gran tamaño, la gráfica muestra que el marco AMBAR escala bien, lo que puede ser deducido analizando el incremento lineal del tiempo de transmisión al aumentar también el tamaño del recurso. Al mismo tiempo, puede observarse que los resultados obtenidos son muy similares a los ofrecidos por SSL (la diferencia máxima es de un 6 %).

6.6.4 Evaluación de la tramitación de solicitudes con DCMS

Por último, se analizará el tiempo total de tramitación de una solicitud de certificación realizada mediante las aplicaciones que constituyen el sistema DCMS. Concretamente, las pruebas realizadas implican la participación de la aplicación de las autoridades y de la aplicación de los solicitantes.

El objetivo principal es determinar qué porcentaje de tiempo corresponde al proceso de intercambio de información realizado por el protocolo AMBAR y qué proporción está asociada a la tramitación de la solicitud. A continuación, se detalla cada una de las tareas implicadas en dicho proceso de tramitación, indicando si se engloban dentro de las acciones del protocolo AMBAR o del sistema DCMS:

- Envío de la solicitud y las credenciales relacionadas (AMBAR).
- Verificación de la firma digital de la solicitud (DCMS).
- Verificación de la firma digital de cada una de las credenciales presentadas (DCMS).
- Ejecución del algoritmo que comprueba si la solicitud conforma con la política de la autoridad (DCMS).
- Construcción del certificado de credencial solicitado por el cliente (DCMS).
- Transmisión, por parte de la autoridad, del certificado solicitado (AMBAR).

La figura 6.22 muestra el tiempo total de tramitación obtenido en función del número de credenciales necesarias para satisfacer la política de la autoridad. En ella aparece tanto la porción de tiempo asociada al intercambio AMBAR como al procesamiento DCMS. Como puede apreciarse, el tiempo asociado a los intercambios se incrementa de forma menos severa que el asociado a la tramitación, el cual constituye el 65 % del tiempo total de aquellas solicitudes que necesiten de hasta 16 credenciales distintas para ser satisfechas. Por tanto, el porcentaje ligado al proceso de intercambio va teniendo menor relevancia en cuanto las solicitudes son más complejas, y más peso cuando éstas son más sencillas (cercano al 50 %). Por otro lado, analizando los resultados desde un punto de vista absoluto, es posible comprobar que el tiempo total de tramitación de solicitudes es bastante aceptable al no superar los 150 ms para el caso extremo de 16 credenciales.

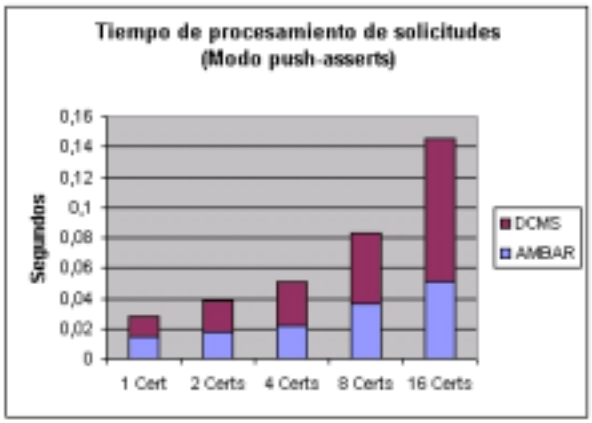

Figura 6.22: Tiempo de tramitación de solicitudes

6.6.5 Conclusiones obtenidas a partir de la evaluación

Las pruebas realizadas permitieron conocer cuáles eran las posibilidades reales de la infraestructura en cuanto a rendimiento se refiere. Como se ha podido comprobar, el protocolo AMBAR ofrece unos resultados muy similares a los aportados por SSL, tanto en su fase de negociación como en la de transmisión de información. Además, se comprobó que la sobrecarga derivada del uso de arquitecturas de seguridad multinivel como CDSA apenas repercutía negativamente sobre el rendimiento total del sistema. Por último, las mediciones relativas al tiempo de tramitación de solicitudes mediante las aplicaciones del sistema DCMS permitieron verificar que el tiempo de respuesta alcanzado se encuentra dentro unos límites que demuestran la aplicabilidad real del sistema, no sólo en cuanto a su diseño sino también en lo que respecta a las prestaciones ofrecidas.

6.7 Conclusiones

Como se puede comprobar a partir de lo descrito en este capítulo, la infraestructura de autorización ha sido desarrollada y verificada con éxito en su papel de middleware de seguridad. En primer lugar, se ha presentado la implementación del marco AMBAR y del sistema distribuido de gestión de credenciales. Tanto la librería AMBAR como el conjunto de aplicaciones DCMS han sido diseñados con el fin de proporcionar servicios de autorización de forma sencilla y estructurada, cuidando que el acceso a su funcionalidad sea lo más completo e intuitivo posible. Para ello, se ha tomado como base la metodología de definición de estructuras de gestión, la cual tiene un fiel reflejo en la forma en la que se articula la implementación de DCMS. Los componentes de la infraestructura se han desarrollado completamente, lo cual implica que toda la funcionalidad descrita en el capítulo anterior (gestión de roles, autorización, reducción, intercambio confidencial de información) se encuentra disponible a la hora de ser aplicada a entornos concretos. Además, se han analizado las prestaciones ofrecidas por dicha implementación, y a la vista de los resulta-

dos obtenidos, es posible afirmar que la infraestructura en su conjunto realiza de forma eficiente las distintas operaciones disponibles, introduciendo una sobrecarga mínima.

En segundo lugar, se ha constatado la viabilidad de estas propuestas a la hora de ser aplicadas a dos escenarios de aplicación completamente distintos. Dicha aplicación de la infraestructura ofrece un conjunto de ventajas adicionales frente al uso de enfoques más tradicionales. Además, la conveniencia de la metodología como directriz a la hora de diseñar sistemas de control de acceso ha sido puesta de manifiesto al modelar con éxito ejemplos concretos de autorización.

Capítulo 7

Conclusiones y líneas futuras

7.1 Conclusiones

La evolución de los sistemas distribuidos ha dejado patente la necesidad de proporcionar mecanismos de seguridad de carácter descentralizado. No estamos frente al reto de proteger un gran sistema central de información administrado por un conjunto reducido de personas. Tampoco se trata de manejar comunidades reducidas y estáticas de usuarios, ni de controlar simplemente un sistema de ficheros compartido o una serie de periféricos. Por el contrario, en el entorno actual se encuentran involucradas grandes comunidades de usuarios, quizá geográficamente dispersas, que desean acceder a un número creciente de recursos de índole muy diversa, administrados de forma local por entidades que tienen la difícil tarea de enfrentarse a un problema de magnitudes muy superiores al existente años atrás.

En el fondo, todo podría considerarse como un problema de gestión de confianza, es decir, la necesidad de tener que plasmar en un sistema informático las distintas relaciones existentes tanto entre las entidades que conforman dicho sistema como entre el propio sistema y otros elementos externos al mismo. Como ya se comentó en el capítulo de introducción, para poder proporcionar los servicios básicos de seguridad es necesario realizar un proceso de identificación, autenticación y autorización que satisfaga los requisitos del sistema al cual pretende aplicarse. A lo largo de este trabajo de tesis, se ha mostrado cómo la certificación digital es una de las tecnologías capaces de proporcionar los mecanismos necesarios para poder llevar a cabo gran parte de las operaciones de seguridad relacionadas con la identificación y la gestión de privilegios.

Hemos visto cómo gran parte de las innovaciones científicas realizadas en torno a los certificados digitales han tenido como especial foco de actuación la provisión de técnicas ligadas a la gestión del ciclo de vida. Se ha comprobado que la aplicación de la certificación digital recae completamente en la existencia de sistemas que proporcionen las herramientas necesarias para su gestión, distribución e integración en entornos de aplicación concretos. Su éxito como herramienta de seguridad depende tanto de su fortaleza criptográfica y expresiva como de la disponibilidad y adecuación del sistema que los gestiona.

El análisis que se ha realizado acerca del estado del arte de las propuestas relacionadas

con la certificación digital, tanto de identidad como de autorización, ha mostrado múltiples aspectos a los que todavía no se les ha proporcionado una solución ampliamente aceptada. Se podría afirmar que tales carencias están asociadas a dos hechos muy concretos: por un lado, la falta de entornos reales de pruebas que permitan conocer las verdaderas necesidades del mercado y validar las distintas propuestas que se van formulando; por otro lado, la obsesión por mantener enfoques tecnológicos obsoletos e intentar adaptar los escenarios reales a las especificaciones existentes en lugar de buscar enfoques alternativos que se adecuen a la realidad en la cual nos encontramos inmersos.

El trabajo de tesis aquí presentado tiene en mente estos dos factores a la hora de proporcionar una solución completa al problema de la gestión y uso de los certificados digitales como herramienta básica de seguridad. En primer lugar, se ha realizado un esfuerzo analítico para encontrar enfoques alternativos al problema de la gestión de la confianza, y más concretamente a la gestión distribuida de la misma. Por otra parte, las propuestas realizadas han sido puestas en marcha y validadas en entornos de aplicación concretos con requisitos muy diversos.

La consecuencia es la especificación de una solución global formada por la composición de una infraestructura de clave pública basada en el estándar X.509 y de una infraestructura de autorización centrada en el uso de certificados SPKI. Ambos sistemas se encuentran claramente conectados, ya que el segundo toma al primero de ellos como punto de partida, y se complementan en la tarea general de etiquetar (identificar) y calificar (autorizar) a las entidades del sistema.

Una cuestión que ha resultado transversal durante el desarrollo de este trabajo ha sido mantener las políticas inherentes del sistema al mínimo, es decir, no definir una solución que conllevará la adopción de una serie de supuestos que pudieran no ser aconsejables en ciertos entornos y que por tanto limitaran su adopción. Por ejemplo, tanto la PKI como el sistema distribuido de gestión de credenciales han sido diseñados para soportar una amplia gama de prácticas de certificación y políticas de autorización. Esta filosofía también se puede apreciar en el diseño modular de muchas de las aportaciones realizadas, el cual tiene como objetivo clave favorecer la inserción o modificación de funcionalidad con el menor impacto posible sobre el resto del sistema.

Como conclusiones concretas en lo que respecta a las aportaciones realizadas en materia de certificación de identidad es posible afirmar que:

- La infraestructura proporciona una amplia gama de modos de acceso a los servicios proporcionados ya que la interacción con las entidades finales puede realizarse de forma directa o a través de las autoridades de registro. Además, al estar basado el diseño en estándares comúnmente aceptados, se favorece la interoperabilidad y se proporciona soporte a la mayor parte de sistemas operativos y aplicaciones.

- La gestión de los distintos componentes se realiza desde un punto de vista unificado. Varios de los aspectos técnicos de las prácticas de certificación se materializan en las denominadas políticas de PKI, las cuales constituyen un mecanismo extensible de especificación basado en la existencia de administradores especiales encargados

de introducir los criterios a seguir. Además, este mecanismo permite variar dinámicamente parte del comportamiento de la PKI, lo cual simplifica la adaptación de la infraestructura a entornos muy dinámicos.

- Se han introducido servicios avanzados en lo que respecta a la revocación y validación de certificados. Por un lado, al igual que sucede con otros esquemas de certificación de identidad, se proporcionan varias soluciones destinadas a permitir la autorrevocación de certificados digitales, lo cual permite agilizar el proceso de notificación de incidencias. Por otro lado, se ha introducido un sistema de validación de certificados basado en el refirmado de los mismos. Este último sistema está basado en la creación de sentencias positivas que permiten a cualquier aplicación validar el estado de un certificado en un instante dado sin tener que realizar ningún tipo de consulta externa ni conocer otras especificaciones en materia de validación. Además, se ha mostrado que, cuando el número de comprobaciones por usuario es elevado, el sistema ofrece mejor rendimiento que las propuestas basadas simplemente en OCSP.

Como ya se ha comentado en numerosas ocasiones, la PKI diseñada constituye el punto de partida del sistema de autorización, el mecanismo empleado para realizar la gestión de claves criptográficas y la generación de identificadores. Con el fin de definir el enfoque concreto de gestión de privilegios que se iba a seguir, se realizó un análisis exhaustivo de los distintos modelos de control de acceso surgidos en los últimos años. Fruto de dicho análisis, se determinó que tanto el control de acceso basado en roles (RBAC) como el modelo distribuido basado en delegación constituían las alternativas más apropiadas para el tipo de escenarios que se deseaba modelar, y que por tanto era necesario contrastar cuáles eran las posibilidades ofrecidas por las distintas especificaciones existentes en materia de certificados de credencial a la hora de implementar dichos modelos. La conclusión obtenida situaba a la especificación SPKI como las más acertada debido, entre otros factores, a su riqueza expresiva, su capacidad para absorber los modelos RBAC, la posibilidad de emplear delegación, la provisión de métodos genéricos de cálculo de autorizaciones y la existencia de implementaciones completas de la especificación.

Si bien hay gran multitud de propuestas que hacen uso de este tipo de certificados a la hora de implementar escenarios de aplicación concretos, la mayoría de estas iniciativas carecen de un sistema genérico de gestión de los certificados. Para subsanar esta carencia, se ha presentado el sistema DCMS, el cual proporciona un tratamiento completo a todas las cuestiones relacionadas con certificación de privilegios. Las principales conclusiones que podemos extraer a partir del diseño de DCMS son:

- El problema queda claramente dividido en 3 bloques conceptuales distintos: gestión de la pertenencia a roles y su jerarquía, gestión de la asignación de privilegios a entidades finales o conjuntos de entidades, y gestión de la reducción de autorizaciones. Esta división permite ver el sistema como una composición de subsistemas, los cuales pueden ser empleados de forma aislada dependiendo de las necesidades de autorización del escenario a modelar.

- Su diseño totalmente descentralizado, basado en delegación, permite adaptarlo correctamente a escenarios en los que la gestión de los privilegios se realiza por parte de entidades con escasa conexión entre sí.

- Se han definido todas las estructuras de datos necesarias para llevar a cabo el proceso completo de generación de credenciales. Por un lado, se ha especificado el formato de las solicitudes de nombramiento, autorización y reducción. Por otra parte, se ha definido el formato que deben tener las políticas de seguridad de las distintas autoridades participantes. Tanto el formato de las solicitudes como de las políticas está basado en s-expresiones, sin que haya necesidad de introducir nuevos formatos de codificación distintos a los empleados por los controladores a la hora de proteger sus recursos.

- Mediante las políticas de autorización es posible especificar conjuntos, posiblemente infinitos, de privilegios que pueden ser asociados a las entidades del sistema sin necesidad de tener que emitir los certificados previamente. El hecho de que la emisión se realice bajo demanda permite solventar algunos de los problemas de escalabilidad presentes en escenarios complejos.

- El mecanismo de reducción automática de certificados, junto con el uso de claves temporales, permite eliminar el enlace que pudiera existir entre la identidad de un usuario y sus privilegios. De esta forma, en escenarios en los que el anonimato está permitido o es un requisito, es posible ocultar la relación existente entre los certificados generados por la PKI y los del sistema DCMS. Adicionalmente, el servicio de reducción introduce mejoras en la eficiencia del tratamiento global de las autorizaciones ya que permite simplificar cadenas largas de certificación.

Las comunicaciones entre las entidades de DCMS se han basado en otra de las aportaciones principales de este trabajo de tesis, el marco AMBAR de intercambio de información relativa a autorización. Este marco fue diseñado para suplir la falta de soporte de certificados de credencial en los actuales protocolos de seguridad, lo cual implicaba que la responsabilidad de la transmisión de las solicitudes, políticas y credenciales debía recaer completamente en las aplicaciones finales. Mediante un diseño modular y estructurado, el marco aporta mecanismos de negociación de parámetros de autorización que le permiten adaptarse a distintos entornos, técnicas de optimización de solicitudes basadas en el establecimiento de sesiones, medidas para proteger la confidencialidad de la información que se está transmitiendo y una interfaz de programación que permite ocultar al desarrollador de aplicaciones muchos de los detalles de funcionamiento interno que le son irrelevantes.

Finalmente, dado que la puesta en marcha de un sistema de control de acceso basado en roles y delegación requiere una identificación muy concisa de los elementos participantes y de la relación entre ellos, la especificación de la metodología de definición de estructuras de gestión permitió averiguar que:

- La metodología ofrece un enfoque estructurado para resolver el problema de la puesta en marcha de un sistema basado en autorización. Se presentan conjuntos de

procedimientos específicos tanto para la gestión de pertenencia a roles como para la asignación de privilegios.

- Los procedimientos de la metodología se estructuran en niveles de gestión que tienen una relación directa con los mecanismos ofrecidos por el sistema DCMS, y más concretamente por las aplicaciones que conforman dicho sistema.

- Se distingue claramente entre las políticas de emisión de credenciales, también denominadas políticas de autorización y de nombramiento, y las políticas relacionadas con el reflejo de las estructuras de gestión en los puntos de acceso. En consecuencia, no sólo se proporcionan los medios para especificar el comportamiento de las autoridades sino que también es posible plasmar cuál va a ser la dinámica del sistema, es decir, cómo se van a producir las relaciones entre los distintos elementos participantes.

- Tanto la metodología como la infraestructura de autorización han sido aplicadas con éxito a escenarios concretos de aplicación. Esto ha permitido conocer la adecuación de estas dos propuestas a la hora de modelar dos entornos de autorización tan diversos como el control de acceso físico y la suscripción electrónica. Además, se ha mostrado cómo el uso de credenciales aporta ventajas adicionales respecto a la adopción de enfoques de carácter más centralizado.

Como conclusión global, es posible afirmar que el trabajo aquí presentado constituye un paso importante hacia la utilización de la certificación digital como herramienta fundamental en el diseño de servicios de autenticación y autorización. La composición de la PKI y de la infraestructura de autorización demuestra que la extensión y la mejora de tecnologías ya consolidadas mediante nuevos enfoques tecnológicos da lugar a la creación de nuevos modelos de gestión de sistemas distribuidos, los cuales se ajustan mejor al marco tecnológico actual.

7.2 Líneas futuras

Las aportaciones realizadas en el presente trabajo abren varias líneas futuras de actuación destinadas tanto a la extensión de las soluciones propuestas como a su integración en otros escenarios. Esto es especialmente importante en el campo de la certificación de privilegios, el cual presenta un gran abanico de posibilidades y desafíos que pueden ser resueltos.

En lo que respecta a la extensión de las soluciones aquí presentadas, algunas líneas de investigación futuras son:

- Analizar la problemática de la revocación de privilegios con el fin de dotar a las especificaciones actuales de certificados de credencial de las herramientas necesarias para poder llevar a cabo dicho proceso con las máximas garantías de seguridad y disponibilidad. Los esfuerzos realizados hasta el momento simplemente han intentado extrapolar las soluciones aplicadas en el campo de la identidad digital. Sin embargo,

la revocación de autorizaciones posee sus propias particularidades en lo que respecta a delegación, invalidación y propagación. Consecuencia de este análisis sería la integración en DCMS de un servicio de revocación de certificados SPKI.

- Estudiar lenguajes alternativos de especificación de políticas de autorización con el fin de poder modelar de forma más precisa tanto los requisitos impuestos por los controladores de recursos como por las autoridades. Por ejemplo, la notación basada en las s-expresiones SPKI podría extenderse para reflejar un conjunto más rico de condicionantes.

- Proporcionar al sistema DCMS de un mecanismo de almacenamiento y distribución de credenciales que satisfaga los requisitos de confidencialidad, disponibilidad y eficiencia vistos en la sección 4.4.4.

- Proporcionar implementaciones alternativas del marco AMBAR que sean independientes del mecanismo de transporte, por ejemplo haciendo uso de XML como lenguaje de especificación.

- Extender la metodología de autorización de forma que pueda reflejar los procedimientos necesarios para construir sistemas de control de acceso más complejos, como por ejemplo los basados en los modelos $RBAC_2$ y $RBAC_3$.

Adicionalmente, es importante recalcar la importancia de futuras líneas de actuación que tengan como objetivo tanto la comparación de las aportaciones realizadas respecto a los nuevos modelos que puedan ir surgiendo como su integración en otros escenarios de aplicación. En este sentido, se pueden identificar las siguientes vías:

- Revisión y comparación de los modelos propuestos por la ITU-T en relación con las infraestructuras de gestión de privilegios (PMI). El objetivo de esta línea es contrastar las posibilidades ofrecidas por la nueva versión del estándar X.509 respecto a la especificación SPKI. Para ello, sería necesario modelar un mismo sistema mediante ambas técnicas con el fin de extraer conclusiones a partir de los resultados obtenidos.

- Integración de la infraestructura de autorización en escenarios de acceso a recursos gestionados por controladores ampliamente distribuidos y con políticas de autorización específicas. Evaluar los resultados que pueden obtenerse de la aplicación en entornos de comercio electrónico inteligente (basado en agentes) o sistemas de computación distribuida basados en la composición de componentes software.

- Otra línea de investigación ya iniciada es la relacionada con el control de acceso a redes públicas, especialmente redes inalámbricas. Mediante este entorno, se pretende evaluar las ventajas que introducen las propuestas aquí realizadas a la hora de gestionar un aspecto tan de actualidad como es la movilidad. El objetivo principal será determinar los modelos de autorización que mejor satisfacen los requisitos relacionados con el uso de la red por parte de usuarios no habituales.

- Finalmente, y aprovechando los trabajos ya realizados por otros miembros del grupo de investigación, se está trabajando en la definición de un sistema de políticas de seguridad enfocado a la gestión de servicios telemáticos, como por ejemplo el establecimiento de redes privadas virtuales. La introducción del mecanismo de autorización tiene como finalidad mejorar el proceso de creación, actualización y revocación de políticas que llevan a cabo los administradores de los servicios.

En resumen, las aportaciones realizadas por este trabajo constituyen un punto de partida sólido que permitirá la definición de nuevas líneas de trabajo en la construcción de sistemas distribuidos. La extensión de los servicios de seguridad obtenidos como resultado de esta tesis representa el punto de partida de trabajos posteriores que forman parte del mañana más cercano.

Bibliografía

[1] M. Abadi. On SDSI's linked local name spaces. *Journal of Computer Security*, 6(1-2):3–21, 1998.

[2] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 1(22):6–15, January 1996.

[3] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *New Security Paradigms*, pages 48–60, 1997.

[4] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schiller, and B. Schneier. The Risks Of Key Recovery, Key Escrow, And Trusted Third-Party Encryption, 1998.

[5] M. D. Abrams. Renewed understanding of access control policies. In *Proceedings 16th National Computer Security Conference*, pages 87–95, 1993.

[6] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. *Time-Stamp Protocol (TSP)*, 2001. Request For Comments (RFC) 3161.

[7] C. Adams and S. Farrell. *Internet X.509 Public Key Infrastructure Certificate Management Protocols*, 1999. Request For Comments (RFC) 2510.

[8] S. Ajmani, D. E. Clarke, C. Moh, and S. Richman. ConChord: Cooperative SDSI Certificate Storage and Name Resolution. In *Proceedings of 1st International Workshop on Peer-to-peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 141–154. Springer, March 2002.

[9] A. O. Alan, P. Freier, and P. C. Kocher. *The SSL Protocol Version 3.0*, 1996. Internet Draft.

[10] P. Alterman, R. Weiser, M. Gettes, K. Stillson, D. Blanchard, J. Fisher, R. Brentrup, and E. Norman. Report: EDUCAUSE NIH PKI Interoperability Pilot Project. In *Proceedings of 1st Annual PKI Research Workshop*, pages 177–193, April 2002.

[11] American Bankers Association. *X9.55-199x: Enhanced management controls using digital signatures and attribute certificates*, June 1997.

[12] American National Standards Institute. *ANSI X9.57: American National Standard, Public Key Cryptography for the Financial Services Industry: Certificate Management*, 1997.

[13] R. Anderson and R. Needham. Robustness principles for public key protocols. In *Proceedings International Conference on Advances in Cryptology (CRYPTO 95)*, volume 963 of *Lecture Notes in Computer Science*, pages 236–247. Springer, 1995.

[14] T. Aura. Fast access control decisions from delegation certificate databases. In *Proceedings of 3rd Australasian Conference on Information Security and Privacy ACISP'98*, volume 1428 of *Lecture Notes in Computer Science*, pages 284–295. Springer, July 1998.

[15] T. Aura. On the structure of delegation networks. In *Proc. 11th IEEE Computer Security Foundations Workshop*, pages 14–26, 1998.

[16] T. Aura. Distributed access-rights managements with delegations certificates. In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, volume 1603 of *Lecture Notes in Computer Science*, pages 211–235. Springer, 1999.

[17] T. Aura and C. Ellison. Privacy and Accountability in Certificate Systems. Technical Report HUT-TCS-A61, Helsinki University of Technology, 2000.

[18] T. Austin. *PKI: A Wiley Tech Brief*. John Wiley and Sons, 2001.

[19] O. Bandmann, M. Dam, and B. Sadighi. Constrained Delegations. In *Proceedings of 2002 IEEE Symposium on Security and Privacy*, pages 131–142, 2002.

[20] O. Bandmann, B. Sadighi, and O. Olsson. *Decentralized management of access control*. Swedish Institute of Computer Science, 2001. Internal Project Report.

[21] M. Barbacci, M. Klein, T. H. Longstaff, and C. B. Weinstock. Quality attributes. Technical Report CMU/SEI-95-TR-021, Carnegie Mellon University, 1995.

[22] D. Bell and L. LaPadula. Secure computer systems: unified exposition and multics interpretation. Technical Report ESD-TR-75-306, The Mitre Corporation, 1976.

[23] A. Belokosztolszki and K. Moody. Meta-Policies for Distributed Role-Based Access Control Systems. In *Proceedings of 3rd International Workshop on Policies for Distributed Systems and Networks*. IEEE Press, June 2002.

[24] S. Bennett, S. McRobb, and R. Farmer. *Object-Oriented Systems Analysis and Design*. McGraw Hill, 1999.

[25] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. *The KeyNote Trust Management System Version 2*, September 1999. Request For Comments (RFC) 2704.

[26] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, volume 1603 of *Lecture Notes in Computer Science*, pages 185–210. Springer, 1999.

[27] M. Blaze, J. Feigenbaum, and J. Lacy. Decentraliced Trust Management. In *Proceedings of the Symposium on Security and Privacy*, pages 164–173, 1996.

[28] M. Blaze, J. Feigenbaum, and P. Resnick. Managing Trust in an Information Labeling System. In *Proceedings of European Transactions on Telecommunications*, pages 491–501, 1997.

[29] M. Blaze, J. Feigenbaum, and M. Strauss. Compliance Checking in the PolicyMaker Trust Management System. In *Proceedings of the Financial Cryptography'98*, volume 1465 of *Lecture Notes in Computer Science*, pages 254–274. Springer, 1998.

[30] M. Blaze, J. Ioannidis, and A. D. Keromytis. Trust management and network layer security protocols. In *Security Protocols Workshop*, pages 103–118, 1999.

[31] T. Bray, J. Paoli, and C. M. Sperberg. *Extensible Markup Language (XML) 1.0*, February 1998. W3C Recommendation.

[32] A. Buldas and P. Laud. New Linking Schemes for Digital Timestamping. In *Proceedings of First International Conference on Information Security and Cryptology*, pages 3–14, 1998.

[33] J. A. Bull, L. Gong, and K. R. Sollins. Towards security in an open systems federation. In *European Symposium on Research in Computer Security (ESORICS)*, pages 3–20, 1992.

[34] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 1(8):18–36, February 1990.

[35] A. Caja, O. Cánovas, F. J. García, J. Gil, A. F. Gómez, E. Martínez, and G. Martínez. Experiencia piloto de certificación en la Universidad de Murcia. *Boletín de la Red Nacional de I+D, RedIris*, (46):39–45, December 1998.

[36] A. Caja, O. Cánovas, F. J. García, J. Gil, A. F. Gómez, E. Martínez, and G. Martínez. Providing security to university environment communications. In *Proceedings of the TERENA NORDUnet Networking Conference '99*, Lund (Sweden), June 1999.

[37] J. Callas, L. Donnerhacke, H. Finney, and R.Thayer. *OpenPGP Message Format*, 1998. Request For Comments (RFC) 2440.

[38] CCITT. *Recommendation X.500: The directory-overview of concepts, models and services*, 1988.

[39] CEN. *Inter-sector Electronic Purse, Part 2: Security Architecture*, 1546 edition, January 1996.

[40] D. W. Chadwick and A. Otenko. RBAC Policies in XML for X.509 Based Privilege Management. In *Proceedings of IFIP SEC 2002*, pages 39–53, May 2002.

[41] D. W. Chadwick and A. Otenko. The PERMIS X.509 Role Based Privilege Management Infrastructure. In *Proceedings of SACMAT 2002*, pages 135–140. ACM, June 2002.

[42] P. Cheng and R. Glenn. *Tests Cases for HMAC-MD5 and HMAC-SHA-1*, September 1997. Request For Comments (RFC) 2202.

[43] S. Chokhani and W. Ford. *Certificate Policy and Certification Practices Framework*, March 1999. Request For Comments (RFC) 2527.

[44] Y. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems*, 29(8–13):953–964, 1997.

[45] Dwaine Clarke. SPKI/SDSI HTTP Server and Certificate Chain Discovery in SPKI/SDSI . Master's thesis, M.I.T., September 2001.

[46] O. Cánovas, F. J. García, A. F. Gómez, and G. Martínez. Aplicación de la firma dual para la corrección de exámenes en el entorno web. *Boletín de la red nacional de I+D, Rediris*, (54):65–69, November 2000.

[47] O. Cánovas and A. F. Gómez. AMBAR Protocol: Access Management Based on Authorization Reduction. In *Proceedings of the International Conference on Information and Communications security (ICICS 2001)*, volume 2229 of *Lecture Notes in Computer Science*, pages 376–380. Springer Verlag, November 2001.

[48] O. Cánovas and A. F. Gómez. A Distributed Credential Management System for SPKI-Based Delegation Systems. In *Proceedings of 1st Annual PKI Research Workshop*, pages 65–76, 2002.

[49] O. Cánovas and A. F. Gómez. Gestión Distribuida de Certificados Digitales SPKI. In *Actas de la VII Reunión Española de Criptología y Seguridad de la Información*, pages 137–150, September 2002.

[50] O. Cánovas, A. F. Gómez, G. López, and G. Martínez. Dynamic Virtual Private Networks. In *Proceedings of EUROMEDIA 2000*, pages 317–321, 2000.

[51] O. Cánovas, A. F. Gómez, and G. Martínez. A PKI Scenario for High-Security Communications: Re-issued Certificates. In *Proceedings of the e-Business and e-Work 2000 Conferences*, pages 225–231, 2000.

[52] O. Cánovas, A. F. Gómez, and G. Martínez. A system for self-revocation of digital certificates. In *Proceedings of the Second International Network Conference*, pages 289–296, 2000.

[53] O. Cánovas, A. F. Gómez, and G. Martínez. PISCIS: Comercio Electrónico Basado en Infraestructuras de Certificación Avanzadas y Sistemas de Tarjeta Inteligente. In *Actas del I Simposio Español de Negocio Electrónico*, pages 201–216, 2001.

[54] O. Cánovas, A. F. Gómez, H. Martínez, and G. Martínez. Different Smartcard-based Approaches to Physical Access Control. In *Proceedings of Infrastructure Security Conference 2002*, volume 2437 of *Lecture Notes in Computer Science*, pages 214–226. Springer Verlag, October 2002.

[55] Wedgetail Communications. *JCSI - Java Crypto and Security Implementation*. World Wide Web, http://www.wedgetail.com/jcsi/index.html, 2002.

[56] Intel Corporation. *Common Data Security Architecture (CDSA)*. World Wide Web, http://developer.intel.com/ial/security, 2002.

[57] E. Dawson, J. López, J. A. Montenegro, and E. Okamoto. A new design of Privilege Management Infrastructure for organizations using outsourced PKI. In *Proceedings of Information Security Conference 2002*, volume 2433 of *Lecture Notes in Computer Science*, pages 136–149. Springer, 2002.

[58] Fábrica Nacional de Moneda y Timbre. *Proyecto CERES*. World Wide Web, http://www.cert.fnmt.es, 2002.

[59] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*, January 1999. Request For Comments (RFC) 2246.

[60] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[61] D. D. Downs, J. R. Rub, K. C. Kung, and C. S. Jordan. Issues in discretionary access control. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 208–218. IEEE Press, April 1985.

[62] J. Dávila and L. F. Pardo. Diseño y realización de un servicio de sellado digital de tiempo. *Boletin de la Red Nacional de I+D, Rediris*, (46):31–38, November 1998.

[63] J. G. Dyer, M. Lindemann, R. Perez, R. Sailer, L. van Doorn, S. W. Smith, and S. Weingart. Building the IBM 4758 Secure Coprocessor. *IEEE Computer*, 34(10):57–66, 2001.

[64] D. Eastlake and O. Gudmundsson. *Storing Certificates in the Domain Name System (DNS)*, March 1999. Request For Comments (RFC) 2538.

[65] G. Elcock. Web-based user interface for a Simple Distribute Security Infrastructure (SDSI). Master's thesis, M.I.T., June 1997.

[66] J. E. Elien. Certificate discovery using SPKI/SDSI 2.0 certificates. Master's thesis, M.I.T., May 1998.

[67] C. Ellison. Improvements on Conventional PKI Wisdom. In *Proceedings of 1st Annual PKI Research Workshop*, pages 165–176, April 2002.

[68] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. *Simple Public Key Certificate*. IETF Internet Draft, draft-ietf-spki-cert-structure-06.txt edition, July 1999.

[69] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. *SPKI certificate theory*, September 1999. Request For Comments (RFC) 2693.

[70] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. *SPKI Examples*. IETF Internet Draft, draft-ietf-spki-cert-examples-01.txt edition, March 1999.

[71] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about public key infrastructure. *Computer Security Journal*, 16(1):1–7, 2000.

[72] EuroPKI. *EuroPKI Top Level Certification Authority*. World Wide Web, http://www.europki.org, 2002.

[73] S. Farrel. *TLS extensions for AttributeCertificate based authorization*. IETF Internet Draft, draft-ietf-tls-attr-cert-00.txt edition, February 1998.

[74] S. Farrel and R. Housley. *An Internet Attribute Certificate Profile for Authorization*, April 2002. Request for Comments (RFC) 3281.

[75] D. F. Ferraiolo, J. A. Cugini, and D. R. Kuhn. Role-Based Access Control (RBAC): Features and Motivations. In *Proceedings Annual Security Aplications Conference*, pages 241–248. IEEE Press, 1995.

[76] J. Ferrari, S. Poh, R. Mackinnon, and L. Yatawara. *Smart Cards: A Case Study*. IBM RedBooks, http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg245239.pdf, October 1998.

[77] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. *Hypertext Transfer Protocol – HTTP/1.1*, January 1997. Request for Comments (RFC) 2068.

[78] IAIK Institute for applied information processing and communication. *IAIK JCE*. World Wide Web, http://jce.iaik.tugraz.at/, 2002.

[79] W. Ford and M. S. Baum. *Secure Electronic Commerce*. Prentice Hall, second edition, 2001.

[80] B. Fox and B. LaMacchia. Online certificate status checking in financial transactions: the case for re-issuance. In *Proceedings of Financial Cryptography '98*, volume 1648 of *Lecture Notes in Computer Science*, pages 104–117. Springer-Verlag, 1998.

[81] M. Fredette. An implementation of SDSI-the Simple Distributed Security Infrastructure. Master's thesis, M.I.T., May 1997.

[82] D. Gambetta. Can we trust trust? In *Trust: Making and Braking Cooperative Relations*, pages 213–237, 1990.

[83] B. Gassend, G. E. Suh, D. Clarke, M. van Dijk, and S. Devadas. Caches and merkle trees for efficient memory authentication. In *Proceedings of Ninth International Symposium on High Performance Computer Architecture*, 2003. To be published.

[84] M. Gil and F. Pereñiguez. Extension de una PKI para dar soporte a certificados de atributo X.509. Master's thesis, Universidad de Murcia, June 2002.

[85] A. F. Gómez, G. Martínez, and O. Cánovas. New Security Services based on PKI. *Future Generations Computer Systems*, 2003. To be published.

[86] Open Group. *The Open Group Home Page.* World Wide Web, http://www.opengroup.org, 2002.

[87] The Open Group. *Common Security: CDSA and CSSM, Version 2*, c914 edition, May 2000.

[88] The Open Group. *CDSA Explained.* World Wide Web, http://www.opengroup.org/products/publications/catalog/g905.htm, 2001.

[89] C. A. Gunter and T. Jim. Policy-directed certificate retrieval. In *Proceedings of Software - Practice and Experience*, pages 1609–1640, 2000.

[90] P. Gutmann. PKI: It's not dead, just resting. *IEEE Computer*, 35(8):41–49, 2002.

[91] K. Gutzmann. Access Control and Session Management in the HTTP Environment. *IEEE Internet Computing*, 5(1):26–35, 2001.

[92] A. Hagstrom, S. Jajodia, F. Parisi, and D. Wijesekera. Revocation: a classification. In *Proceedings of the 14th IEEE Computer Security Foundation Workshop*, pages 44–58. IEEE Press, 2001.

[93] J. Y. Halpern and R. van der Meyden. A logic for SDSI's linked local name spaces. In *Proceedings of the 12th Computer Security Foundations Workshop*, pages 111–122. IEEE Computer Society Press, 1999.

[94] D. Harkins and D. Carrel. *The Internet Key Exchange (IKE)*, 1998. Request For Comments (RFC) 2409.

[95] T. Hasu and Y. Kortesniemi. *Implementing an SPKI Certificate Repository within the DNS*, Poster Paper Collection of the Theory and Practice in Public Key Cryptography (PKC 200) edition, January 2000.

[96] M. Hendry. *Smart Card Security and Applications*. Artech House, April 2001.

[97] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid. Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 2–14, April 2000.

[98] R. Housley. *Cryptographic Message Syntax*, July 1999. Request for Comments (RFC) 2630.

[99] R. Housley, T. Polk, W. Ford, and D. Solo. *Internet Public Key Infrastructure, Part I: X.509 Certificate and CRL Profile*, April 2002. Request for Comments (RFC) 3280.

[100] ICE-CAR. *European ICE-CAR Project*. World Wide Web, http://ice-car.darmstadt.gmd.de/, 1999.

[101] ICE-TEL. *European ICE-TEL Project*. World Wide Web, http://www.darmstadt.gmd.de/ice-tel/, 1997.

[102] Identrus. *Identrus Home Page*. World Wide Web, http://www.identrus.com, 2002.

[103] IETF. *PKIX Working Group*. World Wide Web, http://www.ietf.org/html.charters/pkix-charter.html, 2002.

[104] S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith. Implementing a distributed firewall. In *ACM Conference on Computer and Communications Security*, pages 190–199, 2000.

[105] ITU-T. *ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, 1995. Recommendation X.690.

[106] ITU-T. *ISO/IEC 9594-8, The Directory: Authentication Framework*, 2001. Recommendation X.509.

[107] A. Jain, R. Bolle, and S. Pankanti, editors. *BIOMETRICS: Personal Identification in Networked Society*. Kluwer Academic, 1999.

[108] A. Josang. The right type of trust for distributed systems. In *Proceedings of New Security Paradigms '96*, pages 119–131, 1996.

[109] S. Kent. *Privacy enhancement for Internet electronic mail - part II: Certificate-based key management*. Request For Comments (RFC) 1422, February 1993.

[110] S. Kent and R. Atkinson. *IP Authentication Header*. Request For Comments (RFC) 2402, November 1998.

[111] S. Kent and R. Atkinson. *IP Encapsulating Security Payload*. Request For Comments (RFC) 2406, November 1998.

[112] J. Knudsen. *Java Cryptography*. O' Reilly, 1998.

[113] P. Kocher. On certificate validation and revocation. In *Proceedings of Financial Cryptography 98*, volume 1465 of *Lecture Notes in Computer Science*, pages 172–177. Springer-Verlag, 1998.

[114] J. Kohl and C. Neumann. *The Kerberos network authentication service*. Request For Comments (RFC) 1510, September 1993.

[115] L. Kohnfelder. *Toward a Practical Public-Key Cryptosystem*. M.I.T., 1978. Bachelor's Thesis.

[116] J. Koponen, P. Nikander, J. Rasanen, and J. Paajarvi. Internet access through WLAN with XML encoded SPKI certificates. In *Proceedings of NordSec'00*, October 2000.

[117] Y. Kortesniemi, T. Hasu, and J. Sars. Validity Management in SPKI. In *Proceedings of 1st Annual PKI Research Workshop*, pages 27–36, April 2002.

[118] RSA Laboratories. *PKCS#7: Cryptographic Message Syntax Standard Ver 1.5*, May 1997.

[119] RSA Laboratories. *PKCS#5: Password-Based Cryptography Standard*, March 1999.

[120] RSA Laboratories. *PKCS#10: Certification Request Syntax Standard Ver 1.7*, May 2000.

[121] RSA Laboratories. *PKCS#11: Cryptographic Token Interface Standard Ver 2.10*, December 2000.

[122] X. Lai. *On the design and security of block ciphers*, volume 2. ETH Series in Information Processing, 1992.

[123] T. Lampinen. Using SPKI Certificates for Authorization in CORBA based Distributed Object-Oriented Systems. In *Proceedings of NordSec'99*, pages 61–81, November 1999.

[124] B. W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, October 1973.

[125] B. W. Lampson. Protection. *Operating Systems Review*, 8(1):18–24, January 1974.

[126] N. Li. Local Names in SPKI/SDSI. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, pages 2–15. IEEE Press, June 2000.

[127] D. Lie, C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, and M. Horowitz. Architectural support for copy and tamper resistant software. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX)*, pages 169–177, November 2000.

[128] The Cryptix Foundation Limited. *Cryptix*. World Wide Web, http://www.cryptix.org, 2002.

[129] J. Liu and Y. Ye, editors. *E-Commerce Agents, Marketplace Solutions, Security Issues, and Supply and Demand*, volume 2033 of *Lecture Notes in Computer Science*. Springer, 2001.

[130] V. Lortz, M. Wischy, M. Hondo, and T. Nixon. *Universal Plug-and-Play Security Requirement*. UPnP Forum, July 2001.

[131] G. López and O. Cánovas. *PISCIS PKI: Documentación técnica*. Universidad de Murcia, December 2001.

[132] J. López. Infraestructuras de autenticación y autorización. *Seguridad en Informatica y Comunicaciones (SIC)*, (50):56–60, June 2002.

[133] H. S. Madhusudhana and V. R. Ramachandran. *SPKI Certificate Integration with Transport Layer Security*. IETF Internet Draft, draft-ietf-tls-spki-00.txt edition, July 2001.

[134] A. Malpani, R. Housley, and T. Freeman. *Simple Certificate Validation Protocol (SCVP)*, 2002. IETF PKIX Work in progress.

[135] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[136] H. Martínez, F. García, G. López, J. Tavira, J. P. Cánovas, and B. Ubeda. TICA: Dispositivo de Control de Acceso mediante Java y Tarjetas Inteligentes. *Boletín de la Red Nacional de I+D, RedIris*, (55):73–76, January 2001.

[137] MasterCard and Visa. *SET secure electronic transaction specification, version 1.0. Book 1: Business Description, Book 2: Programmer's Guide, Book 3: Formal Protocol Definition*, May 1997.

[138] A. Maywah. An implementation of a secure web client using SPKI/SDSI Certificates. Master's thesis, M.I.T., May 2000.

[139] C. J. McCollum, J. R. Messing, and L. A. Nortagiacomo. Beyond the pale of MAC and DAC: Defining new forms of access control. In *Proceedings IEEE Symposium on Research in Security and Privacy*, pages 190–200. IEEE Press, May 1990.

[140] S. Micali. Enhanced certificate revocation. Technical Report MIT/LCS/TM-542, MIT Laboratory for Computer Science, 1995.

[141] S. Micali. NOVOMODO: Scalable Certificate Validation and Simplified PKI Management. In *Proceedings of 1st Annual PKI Research Workshop*, pages 15–26, 2002.

[142] S. Micali and R. L. Rivest. Micropayments revisited. In *Proceedings of the Cryptographer's Track at the RSA Conference 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 149–163. Springer, 2002.

[143] Sun Microsystems. *XDR: External Data Representation Standard*, June 1987. Request for Comments (RFC) 1014.

[144] Sun Microsystems. *Java Card 2.1.1 Application Programming Interface*. World Wide Web, http://java.sun.com/products/javacard/javacard21.html, May 2000.

[145] J. C. Mitchell, V. Shmatikov, and U. Stern. Finite-State Analysis of SSL 3.0. In *7th USENIX Security Symposium*, pages 201–215, 1998.

[146] D. Mitton, S. Barkley, D. Nelson, B. Patil, M. Stevens, and B. Wolff. *Authentication, Authorization, and Accounting: Protocol Evaluation*. Request For Comments (RFC) 2904, June 2001.

[147] A. Morcos. A Java implementation of Simple Distributed Security Infrastructure. Master's thesis, M.I.T., 1998.

[148] M. Myers. Revocation: Options and challenges. In *Proceedings of Financial Cryptography 98*, volume 1465 of *Lecture Notes in Computer Science*, pages 165–171. Springer-Verlag, 1998.

[149] M. Myers, C. Adams, D. Solo, and D. Kemp. *Internet X.509 Certificate Request Message Format*, March 1999. Request For Comments (RFC) 2511.

[150] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. *OCSP: Online Certificate Status Protocol*, June 1999. Request For Comments (RFC) 2560.

[151] M. Myers, X. Liu, J. Schaad, and J. Weinstein. *Certificate Management Messages over CMS*, 2000. Request For Comments (RFC) 2797.

[152] P. H. Myrvang. *An Infrastructure for Authentication, Authorization and Delegation*. PhD thesis, Department of Computer Science, University of Tromso, May 2000.

[153] M. Naor and K. Nissim. Certificate revocation and certificate update. In *Proceedings 7th USENIX Security Symposium*, pages 561–570, January 1998.

[154] G. Navarro, S. Robles, and J. Borrell. SPKI para el control de acceso a recursos en entornos de agentes móviles. In *Actas de la VII Reunión Española sobre Criptología y Seguridad de la Información*, pages 671–683, September 2002.

[155] Netscape. *Certificate Management System.* World Wide Web, http://wp.netscape.com/cms/v4.0/, 2002.

[156] P. Nikander. *An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems.* PhD thesis, Helsinki University of Technology, March 1999.

[157] H. Nwana, J. Rosenschein, T. Sandholm, C. Sierra, P. Maes, and R. Guttman. Agent-mediated electronic commerce: Issues, challenges, and some viewpoints. In *Proceedings of the Second International Conference on Autonomous Agents (Agents'98)*, pages 189–196, May 1998.

[158] OASIS. *XML-Based Security Services TC (SSTC).* World Wide Web, http://www.oasis-open.org/committees/security/index.shtml, 2002.

[159] Institute of Electrical and Electronic Engineers. *IEEE Standard Computer Dictionary: A compilation of IEEE Standard Computer Glossaries.* IEEE Press, 1990.

[160] University of Murcia. *KRONOS Project.* World Wide Web, http://ants.dif.um.es/kronos, 2002.

[161] Helsinki University of Technology. *Telecommunications Software Security Architecture.* World Wide Web, http://www.tcm.hut.fi/Research/TeSSA/, 2002.

[162] R. Oppliger, G. Pernul, and C. Strauss. Using attribute certificates to implement role-based authorization and access control. In *Proceedings of the 4th Fachtagung Sicherheit in Informationssystemen (SIS 200)*, pages 169–184, October 2000.

[163] J. Partanen and P. Nikander. Adding SPKI certificates to JDK 1.2. In *Third Nordic Workshop on Secure IT Systems (Nordsec'98)*, Trondheim, Norway, 1998.

[164] N. Perwaiz and I. Sommerville. Structured management of role-permission relationships. In *Proceedings of SACMAT 2001*, pages 163–169. ACM, 2001.

[165] D. Pinkas and R. Housley. *Delegated Path Validation and Delegated Path Discovery Protocol Requirements.*

[166] O. Prnjat, I. Liabotis, T. Olukemi, L. Sacksand M. Fisher, P. McKee, K. Carlberg, and G. Martinez. Policy-based Management for ALAN-Enabled Networks. In *Proceedings of 3rd International Workshop on Policies for Distributed Systems and Networks.* IEEE Press, June 2002.

[167] OpenSSL Project. *OpenSSL library.* World Wide Web, http://www.openssl.org, 2002.

[168] B. Ramsdell. *S/MIME Version 3 Message Specification*, 1999. Request For Comments (RFC) 2633.

[169] P. Resnick and J. Miller. PICS: Internet Access Controls Without Censorship. *Communications of the ACM*, 39(10):87–93, 1996.

[170] R. Rivest and B. Lampson. *SDSI: A simple distributed security infrastructure.*

[171] R. L. Rivest. *The MD5 Message-Digest Algorithm*, April 1992. Request For Comments (RFC) 1321.

[172] R. L. Rivest. Can We Eliminate Certificate Revocations Lists. In *Proceedings of Financial Cryptography '98*, volume 1465 of *Lecture Notes in Computer Science*, pages 178–183. Springer-Verlag, 1998.

[173] J. P. Rubio and F. J. Sáez. DKRONOS: Gestión distribuida de Control de Acceso Físico. Master's thesis, Universidad de Murcia, December 2001.

[174] A. Ruiz, G. Martinez, O. Canovas, and A. F. Gomez. SPEED Protocol: Smartcard-Based Payment with Encrypted Electronic Delivery. In *Proceedings of 4th Information Security Conference*, volume 2200 of *Lecture Notes in Computer Science*, pages 446–461. Springer, 2001.

[175] B. Sadighi and M. Sergot. Revocation Schemes for Delegated Authorities. In *Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.

[176] B. Sadighi, M. Sergot, and O. Bandmann. Using Authority Certificates to Create Management Structures. In *Proceeding of Security Protocols, 9th International Workshop*, April 2001.

[177] B. Sadighi and L. van der Torre. Towards a formal analysis of control systems. In *Proceedings of 18th European Conference on Artificial Intelligence*, pages 317–318. John Wiley and Sons, 1998.

[178] R. S. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[179] K. Seamons, M. Winslett, and T. Yu. Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation. In *Proceedings of Network and Distributed System Security Symposium*, April 2001.

[180] RSA Security. *RSA BSAFE.* World Wide Web, http://www.rsasecurity.com/products/bsafe/, 2002.

[181] R. E. Smith. *Internet Cryptography.* Addison Wesley, 1997.

[182] Sourceforge.net. *Project: Common Data Security Architecture.* World Wide Web, http://sourceforge.net/projects/cdsa, 2002.

[183] F. Stajano. *Security for Ubiquitous Computing.* John Wiley and Sons, 2002.

[184] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari. Certificate-Based Access Control for Widely Distributed Resources. In *Proceedings of the 8th USENIX Security Symposium*, pages 215–227, August 1999.

[185] Trolltech. *QT. The cross-platform GUI Toolkit.* World Wide Web, http://www.trolltech.com, 2002.

[186] J. Vollbrecht, P. Calhoun, S. Farrell, L Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. *AAA Authorization Framework.* Request For Comments (RFC) 2904, August 2000.

[187] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. *2nd USENIX Workshop on Electronic Commerce*, pages 29–40, November 1996.

[188] M. Wahl, T. Howes, and S. Kille. *Lightweight Directory Access Protocol (v3)*, December 1997. Request for Comments (RFC) 2251.

[189] P. Zimmermann. *The Official PGP User's Guide.* MIT Press, 1995.

Apéndice A

Definición de las políticas de PKI

Este apéndice incluye la especificación ASN.1 [105] de los elementos de información que componen el sistema de políticas presentado en el capítulo 3. Parte de las definiciones están basadas en estructuras ya definidas en el estándar X.509 [106].

A.1 Estructura general de la política

Se compone de un número de serie, una fecha de emisión, una fecha de próxima emisión y un conjunto de elementos de política. A su vez, cada elemento de política incluye un campo que especifica el conjunto de entidades afectadas por dicho elemento (la especificación se realiza utilizando la estructura GeneralName, la cual abarca los Distinguished Names X.500), y uno o más valores relacionados con el parámetro que está siendo controlado.

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

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

Rule ::= SEQUENCE {
  ruleID        RULE.&id({RuleSet}), --Identificador de la regla
  ruleValue     OCTET STRING         --Codificación DER de la regla
}
```

```
RuleSet RULE ::= {..}
```

A.2 Reglas de la política

KeyType. Tipo de clave que contendrá el certificado.

```
KeyType RULE ::= SEQUENCE OF OBJECT IDENTIFIER --DSA,RSA
```

RSAKeyLength. Longitud máxima y mínima que debe tener la clave RSA contenida en la solicitud.

```
RSAKeyLength RULE ::= SEQUENCE {
  minLength     INTEGER,
  maxLength     INTEGER
}
```

DSAKeyLength. Longitud máxima y mínima que debe tener la clave DSA contenida en la solicitud.

```
DSAKeyLength RULE ::= SEQUENCE {
  minLength     INTEGER,
  maxLength     INTEGER
}
```

AlternativeSubject. Especifica el tipo de nombres alternativos que pueden emplearse y los criterios que deben seguir.

```
AlternativeSubject RULE ::= SEQUENCE OF AltFilteredName

AltFilteredName ::= SEQUENCE {
  mandatory     BOOLEAN,  --Indica si es obligatorio u opcional
  name          OBJECT IDENTIFIER,    --Tipo de nombre
  filter        OCTET STRING OPTIONAL --Filtro a cumplir
}
```

UniqueIdentifier. Indica si es obligatoria la utilización de un campo de identificador único de usuario.

```
UniqueIdentifier RULE ::= BOOLEAN
```

CertNetscape. Extensiones de tipo Netscape que puede contener el certificado a generar.

```
CertNetscape RULE ::= SEQUENCE OF NetscapeCertType
```

KeyUsage. Usos que se le puede dar a la clave a certificar.

```
KeyExtUsage RULE ::= SEQUENCE OF KeyUsage
```

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

```
ValidityDates RULE ::SEQUENCE {
  validityCA    INTEGER,  --Validez en caso de ser CA
  validityUser  INTEGER   --Validez en caso contrario
}
```

RenewalValidity. Contiene tres clases de información relacionada con la renovación de certificados. 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.

```
RenewalValidity RULE ::= SEQUENCE {
  threshold     INTEGER,
  newInterval   INTEGER,
  maxInterval   INTEGER
}
```

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

```
CRLIssuance RULE ::= SEQUENCE {
  immediate     BOOLEAN DEFAULT TRUE,
  periodically  BOOLEAN DEFAULT TRUE,
  interval      INTEGER
}
```

Apéndice B

Estructuras de datos del protocolo AMBAR

Este apéndice contiene la especificación completa del protocolo AMBAR. La notación empleada está basada en el estándar de representación de datos XDR [143]. Las diferentes secciones están estructuradas en función de los distintos módulos del marco AMBAR, y contienen tanto la estructura de datos de los distintos mensajes AMBAR como los detalles relativos a la generación de los valores criptográficos.

B.1 Transport Convergence

TransportData

```
struct {
    ContentType type;
    uint32      length;
    opaque      data[AMBARMessage.length];
} AMBARMessage;
```

ContentType

```
enum {
    session_management(1), request_management(2),
    authorization_results_management(3), error_management(4),
    data_stream_management(5)
} ContentType;
```

Ciphertext

```
struct {
    select (CipherSpec.cipherType) {
        case null: GenericPlain;
        default:   GenericBlockCipher;
    } data;
} Ciphertext
```

GenericPlain

```
struct {
    opaque content[Ciphertext.length];
} GenericPlain;
```

GenericBlockCipher

```
cbc-block-ciphered struct {
    opaque  content[];
    opaque  MAC[HMAC_SIZE];
} GenericBlockCipher;
```

B.2 Error Management

ErrorLevel

```
enum {
    warning(1), fatal(2)
} ErrorLevel;
```

ErrorDescription

```
enum {
    invalid_signature(1), unknown_message(2), illegal_parameter(3),
    out_of_sequence(4), close_notify(5), sm_failure(6),
    bad_certificate(7), bad_credential(8), bad_calculation(9),
    no_common_suite(10), unspecified_error(11), ciphertext_error(12)
} ErrorDescription;
```

ErrorMessage

```
struct {
    ErrorLevel       level;
    ErrorDescription description;
    uint16           errorNumber;
} ErrorMessage;
```

B.3 Authorization Results Management

ResultType

```
enum {
    negative_notification(1), positive_notification(2), resource(3)
} ResultType;
```

AuthorizationResult

```
struct {
    ResultType  type;
    uint16      transaction_number;
    select (type) {
        case negative_notification: NegativeNotificationContent;
        case positive_notification: PositiveNotificationContent;
        case resource:  ResourceContent;
    } content;
} AuthorizationResult;
```

NegativeNotificationContent

```
struct {
    uint8   notification_number;
    opaque  notification<0..2^8>;
    bool    partial;
} NegativeNotificationContent;
```

PositiveNotificationContent

```
struct {
    opaque  notification<0..2^8>;
} PositiveNotificationContent;
```

ResourceContent

```
struct {
    opaque  resource<uint32>;
} ResourceContent;
```

B.4 Request Management

AuthorizationDataType

```
enum {
    request_and_asserts(1), asserts(2), policy(3), calculation(4)
} AuthorizationDatatype;
```

AuthorizationData

```
struct {
    AuthorizationDataType   type;
    uint16  transaction_number;
    uint16  transaction_step;
    select (type) {
        requests_and_asserts: RequestAndAssertsContent;
        asserts:              AssertsContent;
        policy:               PolicyContent;
        calculation:          CalculationContent;
    } content;
} AuthorizationData;
```

AssertItem

```
struct {
    opaque  assert<0..2^16>;
} AssertItem;
```

StreamFlag

```
enum {
    switch_stream (1), unrelated_stream (2)
} StreamFlag;
```

RequestAndAssertsContent

```
struct {
    StreamFlag  sflag;
    opaque      request<0..2^16>;
    AssertItem  asserts<0..2^16>;
} RequestAndAssertsContent;
```

AssertsContent

```
struct {
    AssertItem asserts<0..2^16>;
} AssertsContent;
```

PolicyContent

```
struct {
    opaque  policy<0..2^16>;
} PolicyContent;
```

CalculationContent

```
struct {
    opaque  calculation<0..2^16>;
} CalculationContent;
```

B.5 Data Stream Management

DSMMessage

```
struct {
    uint16  stream_id;
    uint32  sequence_number;
    opaque  stream<0..2^16>;
} DSMMessage;
```

B.6 Session Management

SMType

```
enum {
    client_init(1), server_init(2), pk_value(3), activate_crypto(4),
    init_session(5)
} SMType;
```

SMMessage

```
struct {
    SMType  type;
    select (type) {
        case client_init:       ClientInit;
        case server_init:       ServerInit;
        case pk_value:          PKValue;
        case activate_crypto:   ActivateCrypto;
        case init_session:      InitSession;
    } content;
} SMMessage;
```

ClientInit

```
struct {
    ProtocolVersion version;
    Random          nonce;
    AssertType      atype;
    Category        category;
    SymmetricCipher ciphers<3..2^8>;
    IdentityType    itype;
    AssertsDistribution     distribution;
} ClientInit;
```

AssertType

```
enum {
    spki(1), keynote(2), ac(3)
} AssertType;
```

SymmetricCipher

```
struct {
    SymmetricAlgorithm  algorithm;
    uint8               length;
} SymmetricCipher;
```

SymmetricAlgorithm

```
enum {
    idea(1), des(2), triple_des(3), cast(4), aes(5), null(255)
} SymmetricAlgorithm;
```

IdentityType

```
enum {
    x509(1), pgp(2), sdsi(3), rsa_public_key(4)
} IdentityType;
```

AssertsDistribution

```
enum {
    push_calculation(1), push_asserts(2), push_both(3), pull(4)
} AssertsDistribution;
```

ServerInit

```
struct {
    ProtocolVersion version;
    Random          nonce;
    AssertType      atype;
    SessionID       session_id;
    Category        category;
    SymmetricCipher cipher;
    IdentityType    itype;
    AssertsDistribution     distribution;
} ServerInit;
```

Certificate

```
opaque CodedCert<1..2^16>;

struct {
    CodedCert   certificate_list<1..2^16>;
} Certificate;
```

RSAParams

```
struct {
    opaque  rsa_modulus<1..2^16>;
    opaque  rsa_exponent<1..2^16>;
} RSAParams;
```

PKValue

```
struct {
    IdentityType pktype;
    select (pktype) {
        case rsa: RSAParams;
        default:  Certificate;
    } value;
} PKValue;
```

Signature

```
private-key-encrypted struct {
    opaque  md5_hash[16];
    opaque  sha_hash[20];
} Signature;
```

PreMasterSecret

```
struct {
    opaque  random[PREMASTER_LENGTH];
} PreMasterSecret;
```

EncryptedPreMasterSecret

```
struct {
    public-key-encrypted PreMasterSecret pre_master_secret;
} EncryptedPreMasterSecret;
```

ActivateCrypto

```
struct {
    EncryptedPreMasterSecret material;
    Signature                pk_verification;
} ActivateCrypto;
```

Identifier

```
enum {
    principal(0x50524E43), guard(0x4752444E)
} Identifier;
```

InitSession

```
struct {
    opaque  md5_hash[16];
    opaque  sha_hash[20];
} InitSession;
```

B.7 Valores criptográficos

Esta sección muestra cómo se realiza el cálculo de los valores criptográficos utilizados tanto por el módulo SM como por el nivel TC.

B.7.1 Valores relacionados con el mensaje ActivateCrypto

```
md5_hash = MD5(ClientInit.nonce + MasterSecret + ServerInit.nonce)

sha_hash = SHA1(ClientInit.nonce + MasterSecret + ServerInit.nonce)

MasterSecret = MD5(PreMasterSecret + SHA1(ClientInit.nonce)) +
               MD5(ServerInit.nonce + SHA1(PreMasterSecret))
```

B.7.2 Valores relacionados con el mensaje InitSession

```
md5_hash = MD5(MasterSecret + Identifier + SM_Messages)

sha_hash = SHA1(MasterSecret + Identifier + SM_Messages)
```

B.7.3 Derivación de claves simétricas a partir del MasterSecret

```
clientKey = SHA1(MasterSecret + "AMBAR" + ClientInit.nonce) +
            SHA1(MasterSecret + "AMBAR" + ServerInit.nonce)

serverKey = SHA1(MasterSecret + "AMBAR" +  MD5(ServerInit.nonce)) +
            SHA1(MasterSecret + "AMBAR" + MD5(ClientInit.nonce))

MACKey = MD5(ServerInit.nonce + SHA1(MasterSecret) +
             ClientInit.nonce + "AMBAR")

IV = MD5(ClientInit.nonce + ServerInit.nonce + MACKey)
```

Apéndice C

Elementos de información de DCMS

Este apéndice incluye la especificación en BNF de los elementos de información correspondientes al sistema de gestión distribuida de credenciales (DCMS). Parte de las estructuras que se han empleado, aquellas que contienen dos asteriscos, están definidas en el documento de especificación de SPKI [68]

C.1 Naming Management System

C.1.1 Solicitudes NMS

```
<request-nms>:: "(" "cert-request" <issuer-name> <subject-req>
                <valid>? ")" ;

<issuer-name>:: "(" "issuer" "(" "name" <principal> <id> ")" ")" ;

<principal>:: <*pub-key*> | <*hash-of-key*> ;

<id>:: <byte-string> ;

<subject-req>:: "(" "subject" <subj-req> ")" ;

<subj-req>:: <principal> | <name> ;

<name>:: "(" "name" <principal> <id> ")" ;

<valid>:: "(" "valid" <valid-basic> ")" ;

<valid-basic>:: <not-before>? <not-after>? ;

<not-after>:: "(" "not-after" <date> ")" ;
```

```
<not-before>:: "(" "not-before" <date> ")" ;

<date>:: <byte-string> ;
```

C.1.2 Políticas de nombramiento

```
<acl-nms>:: "(" "acl" <acl-nms-entry>* ")" ;

<acl-nms-entry>:: "(" "entry" <subject-acl>? <deleg>? <tag-nms>
                  <valid>? ")" ;

<subject-acl>:: "(" "subject" <subj-acl> ")" ;

<subj-acl>:: <principal> | <name> | <name-prefix> | <set-subj>;

<name-prefix>:: "(" "name" <principal> <prefix> ")" ;

<prefix>:: "(" "*" "prefix" <id> ")" ;

<set-subj>:: "(" "*" "set" <subj-acl>* ")" ;

<deleg>:: "(" "propagate" ")" ;

<tag-nms>:: "(" "cert-request" <issuer-acl> <subject-acl>
            <valid>? ")" ;

<issuer-acl>:: "(" "issuer" <iss-acl> ")" ;

<iss-acl>:: <name> | <name-prefix>;
```

C.2 Authorization Management System

C.2.1 Solicitudes AMS

```
<request-ams>:: "(" "cert-request" <issuer> <subject-req> <deleg>?
                <*tag*> <valid>? ")" ;

<issuer>:: "(" "issuer" <principal> ")" ;
```

C.2.2 Políticas de autorización

```
<acl-ams>:: "(" "acl" <acl-ams-entry>* ")" ;

<acl-ams-entry>:: "(" "entry" <subject-acl>? <deleg>? <tag-ams>
                  <valid>? ")" ;

<tag-ams>:: "(" "cert-request" <issuer> <subject-acl> <deleg>?
            <*tag*> <valid>? ")" ;
```

C.3 Reduction Management System

C.3.1 Solicitudes RMS

```
<request-rms>:: "(" "sequence" <chain-reduction> <*cert*>* ")"

<chain-reduction>:: "(" "chain-reduction" <issuer> <subject-principal>
                    <*tag*> ")" ;

<subject-principal>:: "(" "subject" <principal> ")" ;
```

C.3.2 Políticas de reducción

```
<acl-rms>:: "(" "acl" <acl-rms-entry>* ")" ;

<acl-rms-entry>:: "(" "entry" <subject-principal> <deleg> <*tag*>
                  <valid>? ")" ;
```