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.

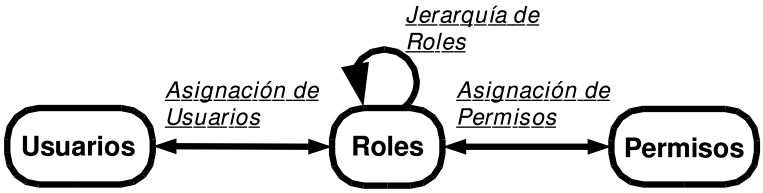

Figura 4.1: Relación entre elementos RBAC

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.