J. Lopez, et al., "Aspectos de Implementación de una Infraestructura de Clave Pública Distribuida", *Simposio Español de Informática Distribuida (SEID00)*, pp. 313-320, 2000.
NICS Lab. Publications: https://www.nics.uma.es/publications

Aspectos de Implementación de una Infraestructura de Clave Pública Distribuida

J. López, A. Maña, J. A. Montenegro y J. J. Ortega
Dpto. Lenguajes y Ciencias de la Computación – E.T.S. Ingeniería Informática
Universidad de Málaga, Campus de Teatinos, 29071 - Málaga
Tel: (95) 2131327, Fax: (95) 2131397
e-mail: {jlm, amg, crypto, juanjose}@lcc.uma.es

Resumen: *La seguridad es uno de los aspectos más conflictivos del uso de Internet. La falta de una política de seguridad global está frenando el desarrollo de Internet en áreas tan interesantes y prometedoras como el comercio electrónico o la interacción con las administraciones públicas. Las técnicas criptográficas actuales proporcionan un alto grado de confidencialidad; no obstante, es difícil garantizar la identificación segura de los usuarios y, además, la gestión de las claves de los mismos es poco eficiente y presenta graves problemas de escalabilidad. Este trabajo describe las características de implementación de una solución a ambos problemas basada en una Infraestructura de Clave Pública (PKI) que proporciona una administración simple y eficiente de las claves de los usuarios y posibilita la autenticación segura de los mismos.*

1. INTRODUCCIÓN

Existen sistemas para la comunicación segura a través de redes cerradas que utilizan sólo criptografía de clave simétrica, como el sistema *Kerberos* del MIT [Kohl89]. Sin embargo, tales sistemas no son escalables para grandes grupos de usuarios pertenecientes a diferentes organizaciones a pesar de las mejoras introducidas con ese objetivo [Davi95] [Gane95].

La criptografía de clave pública [DiHe76] aparece como la herramienta que mejor se adapta para satisfacer los requerimientos de seguridad de Internet, y se está convirtiendo rápidamente en la base de los sistemas de comercio electrónico en línea y otras aplicaciones que requieren seguridad y autenticación en redes abiertas.

Para la utilización global de un criptosistema de clave pública en Internet es crucial utilizar un medio práctico y fiable para la publicación y administración de esas claves: la *Infraestructura de Clave Pública* (PKI, Public Key Infrastructure). Sin una infraestructura que funcione adecuadamente, la criptografía de clave pública es sólo marginalmente más útil que la tradicional criptografía simétrica.

La *certificación* de claves públicas es la función fundamental de todas las PKIs. Así, se define un *certificado* como el medio utilizado por una PKI para comunicar el valor de una clave pública, la información sobre ella, o ambas cosas. Es decir, en su forma más básica un certificado no contiene más que una clave pública, y en términos más generales es una colección de información firmada digitalmente por su emisor.

El usuario de una PKI confía en que las entidades emisoras publiquen certificados fiables que asocien a los sujetos con sus claves públicas (*certificados de identidad*), o que describan propiedades de esos sujetos (*certificados de atributo*). Cada una de esas posibles entidades emisoras se denomina *Autoridad de Certificación* (CA, Certification Authority).

La segunda operación básica de una PKI es la *validación* de certificados. La información del certificado puede cambiar a lo largo del tiempo por lo que el usuario del certificado necesita estar seguro de que los datos contenidos en él son fiables. Existen dos métodos básicos:

- *interactivo*: el usuario solicita directamente a la CA la confirmación de que el certificado es válido cada vez que lo va a usar.
- *diferido*: la CA incluye en el certificado una información relativa al periodo de validez (un par de fechas que definen un rango de vigencia).

Relacionado con el procedimiento de validación está el de *revocación* de certificados, el proceso por el que se le hace saber a los usuarios que la información contenida en el certificado ya no es valida. Esto puede ocurrir cuando la clave privada del sujeto queda comprometida o cuando la información incluida en el certificado ha variado.

En caso de validación interactiva el problema de la revocación es trivial porque la CA al ser consultada indicará que el certificado no es válido. Si se emplean periodos de validez, el método de revocación es crítico. En estos casos, el método más común es utilizar una *Lista de Revocación de Certificados* (CRL, Certificate Revocation List). Una CRL no es más que una lista, firmada y emitida periódicamente por una CA, conteniendo todos los certificados revocados. Durante el proceso de validación el usuario tiene que chequear la última CRL de ese emisor para asegurarse de que el certificado no ha sido revocado.

Actualmente existen diferentes propuestas de infraestructuras de clave pública para Internet, muchas aún en fase de desarrollo. Ninguna ha adquirido un uso generalizado en la red; de hecho, cada vez se extiende más la idea de que en un futuro próximo habrá distintos tipos de PKIs operando conjuntamente en Internet.

2. CARACTERÍSTICAS Y ANÁLISIS DE LAS INFRAESTRUCTURAS DE CLAVE PÚBLICA

La política de seguridad de una Infraestructura de Clave Pública debe establecer en primer lugar qué entidades pueden ejercer como CAs. Algunas proponen que las CAs sean entidades determinadas que han de cumplir una serie de requisitos. Dentro de las de este grupo, se considera como característica básica la disposición de las diferentes CAs dentro de la infraestructura. Varios sistemas utilizan una *jerarquía general* (figura 1) en la que cada CA certifica al nodo del nivel inmediatamente superior (nodo padre) y a todos los que de ella dependen (nodos hijos), creando de esta forma las *cadenas de certificados*. Asociada a una PKI de este tipo siempre existe una Autoridad que establece una política global para la infraestructura. Esta autoridad debe establecer las líneas de actuación que las demás CAs del sistema y todos los usuarios finales han de seguir.

En la figura 1 también se aparecen *certificados cruzados* (líneas discontinuas); estos son certificados que no siguen la jerarquía básica. El uso de certificados cruzados permite que las cadenas de certificados sean muy pequeñas. Pero si el número de CAs crece la certificación cruzada no produce una arquitectura viable porque el número de cruces es demasiado elevado.

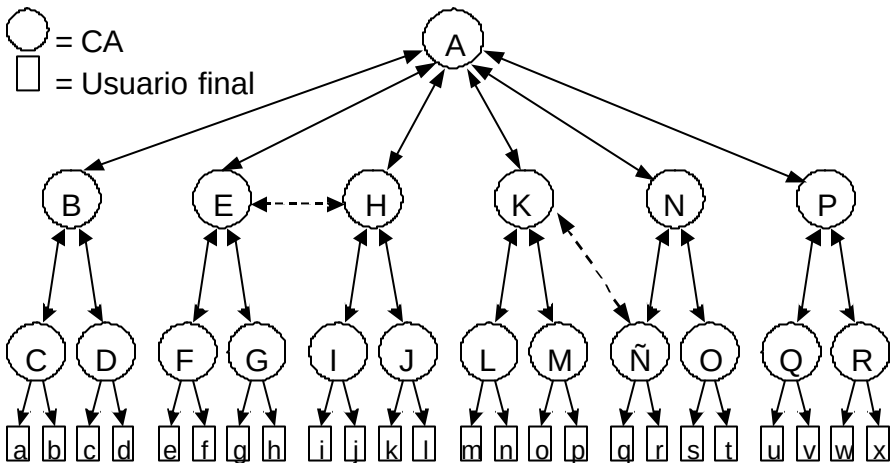

Figura 1. Jerarquía General de Certificación

Algunas propuestas de PKI para Internet, como la fallida *PEM* (Privacy Enhanced Mail [RFC1421][RFC1422][RFC1423][RFC1424]), utilizan una variante de la jerarquía general, la *jerarquía top-down*, en la que las CAs sólo certifican a sus nodos hijos y donde la CA raíz es la fuente de todos los caminos de certificación.

La propuesta PEM se fundamenta en que cada entidad posee un *nombre distinguido* según el estándar X.500 [ISO88]. Una década después esta solución no ha alcanzado su implementación global y, finalmente, la propia *IETF* (Internet Engineering Task Force) ha decidido clasificarla como "no útil" [RFC2316].

Tomando como base el trabajo realizado en el PEM, ha surgido dentro de la IETF el Grupo de Trabajo PKIX cuyo objetivo es el diseño de una infraestructura [PKIX97] que cubra las necesidades de las funciones de identificación automática, autenticación, control de accesos y autorización utilizando certificados con formato X.509 v.3 [ISO96]. Los documentos de trabajo elaborados por este grupo todavía no han sido adoptados como estándares porque no están cerradas muchas cuestiones sobre su implementación en Internet y porque las versiones desarrolladas han resultado incompletas.

Otros importantes ejemplos basados en una jerarquía top-down son las Infraestructuras de Clave Pública gubernamentales de EEUU [Chok94] [NIST96] y Canadá [CSE98].

Las extensiones al *Sistema de Nombres de Dominio* (DNS, Domain Name System) [RFC1101] constituyen otra propuesta que permitiría la autenticación a través de firmas digitales. Esta propuesta es el llamado *DNS-Seguro* (Secure-DNS) [RFC2065]. Tales extensiones describen una infraestructura de clave pública, también jerárquica, integrada en la base de datos del DNS, añadiendo a esta una serie de registros para las claves públicas de los usuarios de cada dominio. Pero, como se verá más tarde, los servidores de nombres plantean varios problemas para albergar las claves públicas.

Existen otras políticas de seguridad, que no contemplan una estructura prefijada de la PKI. En ellas cada usuario puede ejercer el papel de CA con autonomía plena sobre cómo asignar su confianza. El ejemplo más clásico de este tipo de PKIs sin estructura es el de PGP (Pretty Good Privacy) [Zimm95], en el que cada usuario basa su confianza en los certificados de otros usuarios, formando así una *red de confianza* (web of trust).

El hecho de que cada usuario pueda emitir certificados permite una gran flexibilidad y facilidad de implantación porque cada cual certifica a aquellos que conoce de forma personal.

Esta es la mejor opción para la comunicación entre un círculo cerrado de personas, como un grupo de amigos o de empleados de una empresa. No obstante, la inexistencia de una entidad que ejerza de responsable ante situaciones problemáticas impide que este esquema se adapte bien al uso con fines comerciales al no proteger de una forma adecuada los intereses de empresas y consumidores. Además, varios aspectos de la administración de claves y en especial, la revocación necesitan mecanismos específicos que introducen un factor importante de ineficiencia e inseguridad en el diseño.

Este esquema no es escalable en tamaño, pues el número de certificados necesarios para lograr una comunicación global es muy grande y se generan cadenas de certificados de gran longitud. Tampoco es escalable en tiempo debido, principalmente, a los problemas asociados con el mantenimiento de las CRLs. Más aún, si se utilizan valores de confianza asociados a los certificados, entonces la tarea de encontrar el mejor camino de certificación (el del valor de confianza más alto) entre dos usuarios es un problema NP-completo.

Otras propuestas como la del *SPKI* (Simple Public Key Infrastructure) [SPKI98a], otro Grupo de Trabajo de la IETF, y la del *SDSI* (Simple Distributed Security Infrastructure) [RiLa97] son similares a la anterior en cuanto a que en su filosofía se abandona el uso de infraestructuras globales de clave pública. El objetivo perseguido en ambas propuestas es definir un mecanismo que proporcione seguridad para un conjunto amplio de aplicaciones de Internet, incluyendo el cifrado de correo electrónico y de documentos WWW, protocolos de pago, etc. El grupo pretende producir una estructura de certificados y un procedimiento de operación para permitir la administración de confiabilidad de los usuarios de Internet.

Estos dos esquemas comparten la idea de que cada usuario ha de estar identificado inequívocamente por un número, su clave pública, y no por un nombre común (como es el caso de los esquemas que se aproximan a la filosofía de X.500). Puesto que no existen Autoridades de Certificación propiamente dichas, un certificado puede ser creado y firmado por cualquier usuario. Las dos propuestas han sufrido un proceso de fusión [SPKI98b].

3. CARACTERÍSTICAS BÁSICAS DE LA PKI DISTRIBUIDA

En el diseño de Cert'eM se han realizado una serie de decisiones previas en cuanto a su filosofía de diseño, estableciendo un balance entre los objetivos perseguidos. Estas decisiones condicionan la flexibilidad, la generalidad, la facilidad de uso e implantación y, sobre todo, la eficiencia y la seguridad. Cert'eM se basa en los siguientes objetivos:

- proporcionar un medio seguro para la identificación de los usuarios y la difusión de sus claves públicas;
- utilizar una arquitectura de CAs que permita que los usuarios sean certificados por autoridades cercanas, de modo que esta certificación pueda basarse en los mismos elementos en los que basamos la confianza en el mundo real;
- diseñar una arquitectura que no plantee problemas de escalabilidad;
- evitar los problemas de sincronización propios de los esquemas que mantienen

múltiples copias de claves o certificados;

- minimizar el tráfico originado por el sistema, especialmente en operaciones de mantenimiento; y
- eliminar los problemas asociados con la revocación de certificados.

Según se puede deducir del análisis del apartado segundo, para obtener un grado aceptable de seguridad es necesario que las claves estén certificadas por autoridades de certificación específicas y no por un usuario cualquiera. En Cert'eM se propone la existencia de múltiples CAs que operan de forma independiente.

En la figura 2 se muestra la estructura básica del sistema y los elementos que lo componen. La unidad básica dentro de la jerarquía es la *Unidad de Servicio de Claves* (USC), donde se almacenan y mantienen las claves de los usuarios de la estafeta de correo electrónico asociada a un dominio de Internet. Al contrario que en otros sistemas, en Cert'eM la jerarquía está preestablecida: existe una USC por cada estafeta de correo electrónico. Así el conjunto de USCs forma una jerarquía de nodos basada en la jerarquía de dominios.

Además, por cada una de estas USCs existe una Autoridad de Certificación que es responsable de la certificación y mantenimiento de las claves y de la integridad del sistema. Las claves certificadas son gestionadas sólo por la correspondiente CA, por lo que tanto la actualización como la revocación de tales claves son operaciones locales.

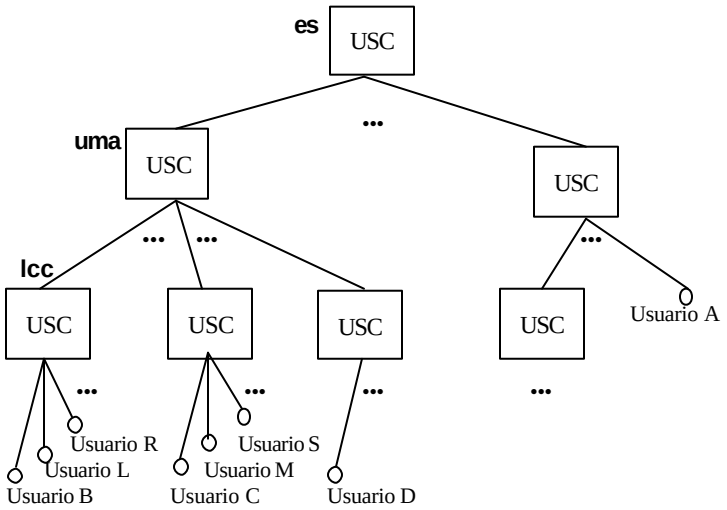

Figura 2. Jerarquía de nodos de Cert'eM

Respecto a esto último es necesario señalar que el uso de CRLs lleva a adoptar soluciones un tanto extremas con objeto de minimizar el número de accesos [Rive98]. Por ello, en el diseño de Cert'eM se ha priorizado la eliminación de estas listas. En Cert'eM los usuarios no distribuyen sus certificados, sino que estos se almacenan en una base de datos vinculada a la CA emisora, la cual puede modificar, eliminar o añadir nuevos certificados de las claves de los usuarios de forma directa.

Una USC se compone de tres elementos, una Autortidad de certificación y dos bases de datos. La Autoridad de Certificación se subdivide, a su vez, en otros dos elementos, el Núcleo Certificador (NC) y el Servidor de Certificados (SC).

El NC gestiona algunas de las tareas relativas a la administración de los certificados dentro de la CA, como la generación del par de claves (o bien la comprobación de la fortaleza de las mismas en caso de que hayan sido generadas por el propio usuario), la fabricación del certificado y el almacenamientro del mismo. Además, incorpora funciones independientes de la administración de los certificados como, por ejemplo, aquellas necesarias para la elaboración de estadísticas de acceso a los certificados.

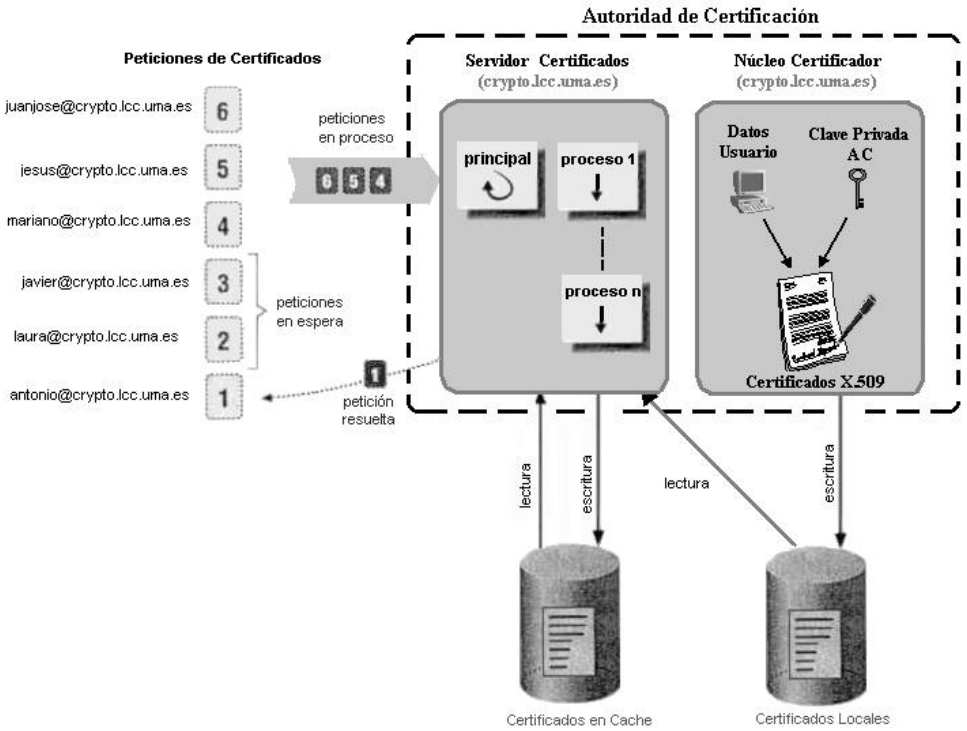

Figura 3. Estructura Interna de una USC

El SC tiene dos funciones principales: en primer lugar, aceptar las peticiones de certificados realizadas por los usuarios locales, y buscar tales certificados dentro de la jerarquía de nodos de Cert'eM; y, en segundo lugar, aceptar las peticiones remotas de certificados relativas a sus propios usuarios por parte de otras USCs, así como dar servicio a esas peticiones. El funcionamiento de un SC puede ajustarse según las caracterísiticas del nodo en el que se encuentre. Esta configuración es gestionada por el NC. El servicio proporcionado por el SC, tanto en plataformas Unix como en MS Windows NT [Becker97] [Kathtan99], acepta las peticiones a través del puerto 850. En las plataformas Unix, el servidor puede ser instalado o bien como proceso de servicio del *inetd*, o bien como proceso independiente siguiendo un modelo standalone.

Respecto a las dos bases de datos mencionadas anteriormente, la primera almacena los certificados digitales de los usuarios pertenecientes al dominio gestionado por la AC; por ello la denominamos *base de datos de certificados locales*. Como se puede observar en la figura, estos certificados son creados por el NC a partir de la información proporcionada por el usuario y de otra información que el NC crea especificamente para cada certificado. Ejemplos de campos de

tales certificados son la clave pública, el identificador del usario, la fecha de expedición, la fecha de expiración, etc. Es decir, todos los campos propios del estándar de certificación X.509 v3.

La segunda de las bases de datos almacena los certificados de los usuarios de otros dominios diferentes al suyo, es decir, de los usuarios externos. Tales certificados habrán sido solicitados por los usuarios locales con el objetivo de realizar las operaciones de cifrado de unos documentos y/o autenticación de otros. En realidad, esta base de datos funciona a modo de cache del sistema para evitar la continua petición de certificados del SC a otros dominios. Es por ello por lo que la denominamos *base de datos de certificados de cache*. Esta base de datos, de tamaño configurable, se va completando con los certificados externos solicitados por los usuarios locales, de tal forma que mediante el uso de la política LRU (Menos Recientemente Usado) se van extrayendo para dejar espacio a otros certificados solicitados con posterioridad. Por lo tanto, cuando un usuario local solicita un certificado externo, el SC comprueba si éste existe en la cache, y de ser así sólo demanda a la USC externa una confirmación de su validez, obteniéndose un menor tráfico de información entre las USC que si se solicitara el certificado completo en cada comunicación.

4. LOCALIZACIÓN FÍSICA DE LAS USCS

Para describir cómo una USC llega a saber a donde dirigir la petición del certificado que su usuario local necesita, se hace uso de la información contenida en el DNS (Servidor de Nombres de Dominio), ya que este es un elemento esencial de consulta dentro de nuestro esquema de localización de USCs. Esta es la secuencia de pasos:

1. Una vez que un usuario del dominio *<upm.es>* ha comunicado a su USC el deseo de obtener el certificado del usuario cuya dirección de correo electrónico es *<jlopez@lcc.uma.es>*, la USC intenta resolver la dirección IP del dominio *<lcc.uma.es>*, para lo cual debería existir en el DNS una entrada del tipo:

   lcc.uma.es *IN A* *111.111.222.222*

2. En el caso de que no exista una entrada de ese tipo, la USC realiza una segunda petición al DNS. Esta petición está vinculada a la naturaleza intrínseca de Cert'eM. Más concretamente, debido a que el identificador del certificado es la dirección de correo del propietario del mismo, Cert'eM establece un vínculo entre la máquina donde reside la estafeta de correo de un dominio y el SC del mismo. Por lo tanto, esa segunda petición al DNS va referida al registro MX (Mail Exchanger) [RFC1034] de la estafeta. Es decir, la USC local busca en el DNS una entrada del tipo:

   lcc.uma.es *MX 5* *111.111.222.222*

   Hay que hacer notar que en el caso de existir más de un registro MX para un mismo dominio el contenido del campo de preferencia (5 en el ejemplo anterior) indicará la entrada a elegir en MX, basándonos en la política de mayor preferencia (menor valor del campo).

3. Por último, en el caso de no encontrar el registro MX en el paso 2, la USC realiza una petición al DNS referida al dominio del mismo nombre con el prefijo "certem-tcp". Por ejemplo:

certem-tcp.lcc.uma.es *IN A* *111.111.222.222*

Esta alternativa obliga al gestor del DNS a incluir explícitamente esta entrada. Su inclusión se debe al gran flujo de información que soportan las estafetas de correo de ciertos sistemas, lo cual no aconseja la instalación del servidor de claves en la misma máquina, evitándose de esta forma la disminución en la velocidad de respuesta del SC.
Además, otra de las causas que ha impulsado este mecanismo de búsqueda de la USC es la posibilidad de utilización de host virtuales; es decir, ubicar distintas USCs en una misma máquina, lo cual permite que sistemas que tienen que ser divididos por motivos administrativos en dominios pequeños puedan centralizarse en una misma máquina. Esta operación se consigue haciendo que, en el servidor de nombres, todos los dominios involucrados apunten a la misma dirección física. Por ejemplo:

certem-tcp.lcc.uma.es *IN A* *111.111.222.222*

certem-tcp.crypto.lcc.uma.es *IN A* *111.111.222.222*

La secuencia de pasos queda representada en la siguiente figura:

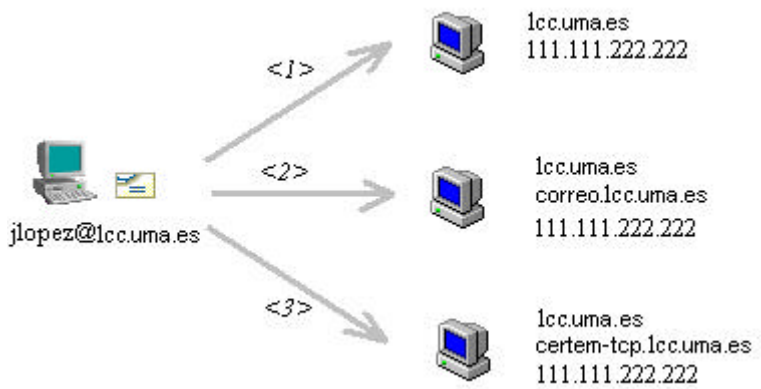

Figura 4. Algoritmo de localización de una USC

4.1 Diagrama de Flujo del Protocolo

En este apartado se describe el protocolo utilizado para la obtención de un certificado. La figura 5 muestra la parte en la que se determina si la localización del certificado es local o externa. La figura 6 representa, para el caso de localización externa, el protocolo entre las USCs.

La obtención de certificados es iniciada con una etapa de conexión, de tal forma que el cliente se identifica ante el servidor, y, dependiendo de la política de seguridad establecida por la

USC, ésta, o bien le permite la conexión, o bien le niega el acceso, ya sea porque el host del cliente no está autorizado, o porque el propio cliente no lo esté.

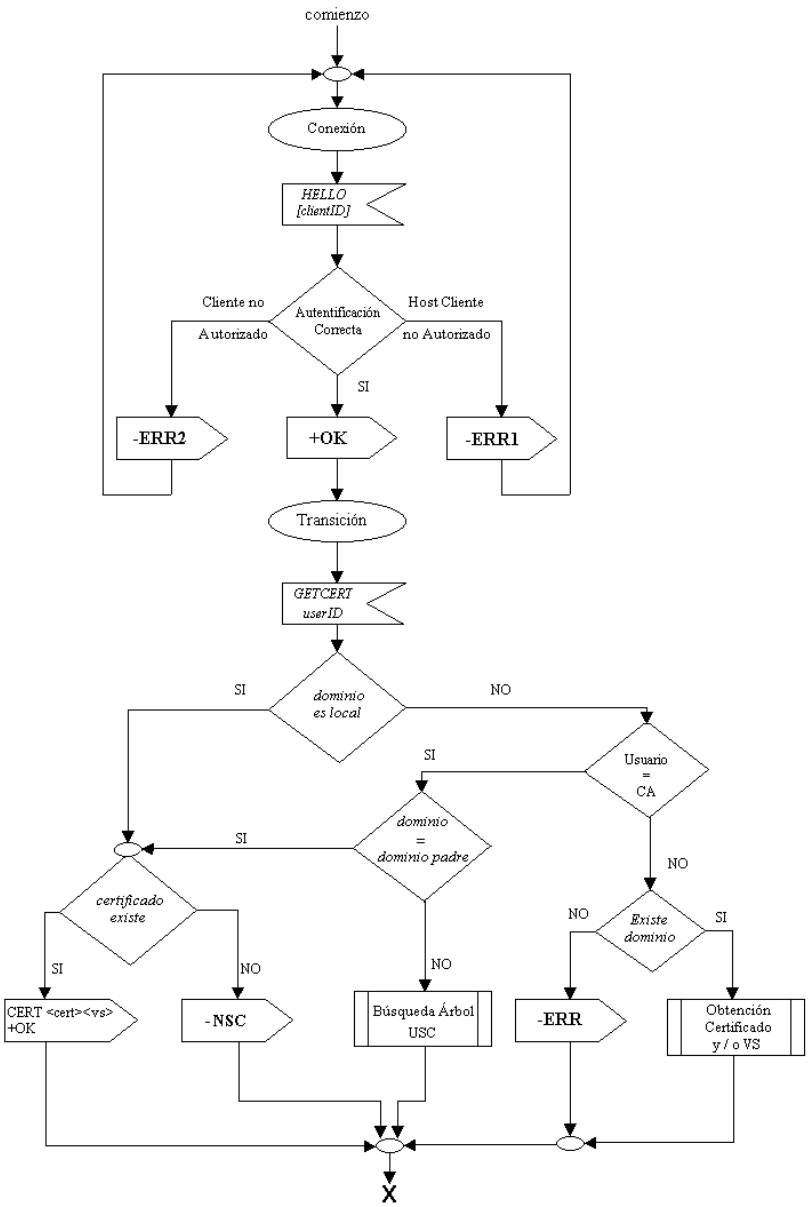

Figura 5. ***Protocolo localización de un Certificado***

En caso de que se le permita la conexión, el cliente realiza la petición del certificado y su SC se encarga de localizarlo en en el propio dominio local o en uno remoto. Si el dominio del certificado es local, el SC actua de forma simple pues se limita a comprobar su existencia mediante un acceso a la base de datos de certificados locales. Si el dominio es remoto entonces se accede en primer lugar a la cache del sistema, y en caso de que el certificado esté almacenado en ese lugar se procede a comprobar su vigencia, solicitando a la USC remota una declaración de validez (VS). Por el contrario, si el certificado no está en la cache, se solicita a la USC remota tanto el certificado digital como la declaración de validez. Para la localización de la USC remota se hace uso de los pasos de localización descritos en el apartado anterior.

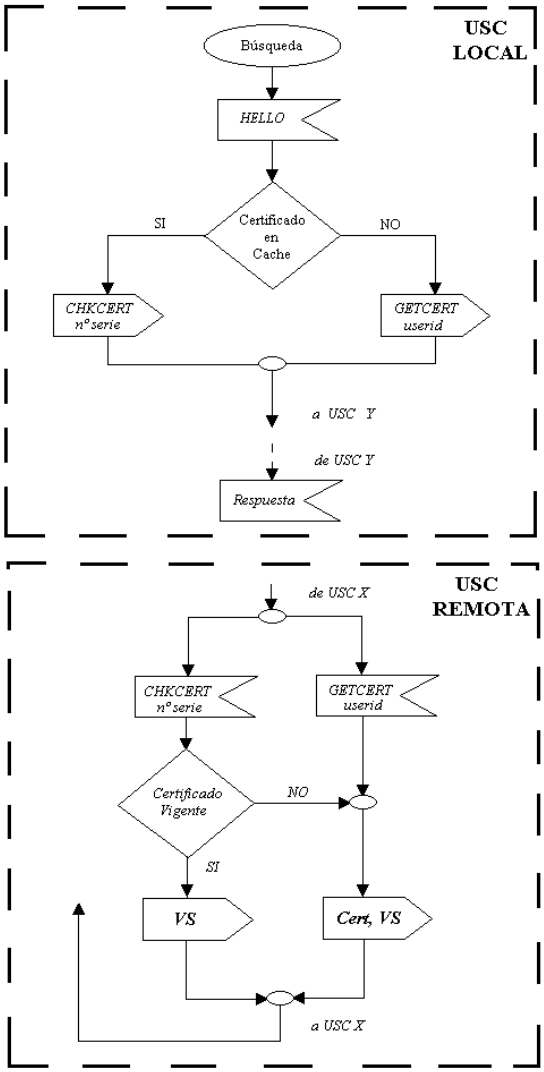

Figura 6. Protocolo localización externa de un Certificado

5. Implementación del Servidor de Claves

El SC es un servicio más dentro de un sistema global, es decir, da soporte a aplicaciones cliente que requieren la información criptográfica proporcionada por el SC. Por lo tanto, un factor clave en su diseño ha sido la eficiencia, de tal forma que no represente una sobrecarga a la aplicación final correspondiente.

Teniendo en cuenta la flexibilidad que nos brinda la plataforma Unix respecto a Windows NT nos hemos centrado en la primera para la implementación de los distintos modelos. La diferencia fundamental de las implementaciones está centrada en la forma en que el proceso principal delega el trabajo en los procesos subordinados:

- **Modelo fork**: Esta solución es la más robusta, ya que cada proceso tiene su propio espacio de memoria y no influye en los demás procesos. Este modelo presenta como inconveniente que la llamada al sistema para crear un proceso hijo es costosa desde el punto de vista temporal, por lo que puede ralentizar el servidor de claves [Robb96] [Brow94] [Stev90].

- **Modelo de hebras**: En este modelo la creación de una hebra por parte del sistema es menos costosa que la creación de un proceso hijo, pero, por el contrario, es menos robusta porque los procesos subordinados comparten el espacio de memoria con el proceso padre y cualquier violación de segmento que se produzca puede dejar inoperable al sistema entero [Rute97] [Mare96].

Independientemente del modelo, tanto procesos como hebras se pueden crear de forma inicial (técnicas de muestreo), y, con posterioridad, se van asignando a las peticiones a medida que es necesario, con lo que se elimina el tiempo correspondiente a la creación de procesos/hebras en el momento de cada petición. Esta solución limita la carga del sistema al número de procesos que se crean durante la puesta en marcha del sistema, por lo que es poco adaptable en aquellos sistemas donde no se hayan realizado estadísticas previas de carga. Mediante la realización de tales estadísticas se pueden evitar situaciones en las que se consuman más recursos de los realmente necesiarios, mientras que en otros casos se evitan situaciones en las que se deniega el servicio a usuarios por una infravaloración previa.

Por lo tanto, se ha optado por una solución intermedia, es decir, inicialmente no se utilizan técnicas de muestreo hasta que se realizan unas estadísticas completas del sistema y, según los resultados obtenidos se configura el sistema para utilizarlas de forma apropiada.

En la siguiente figura se muestra la vida de un subproceso, según los pasos del protocolo anteriormente descrito:

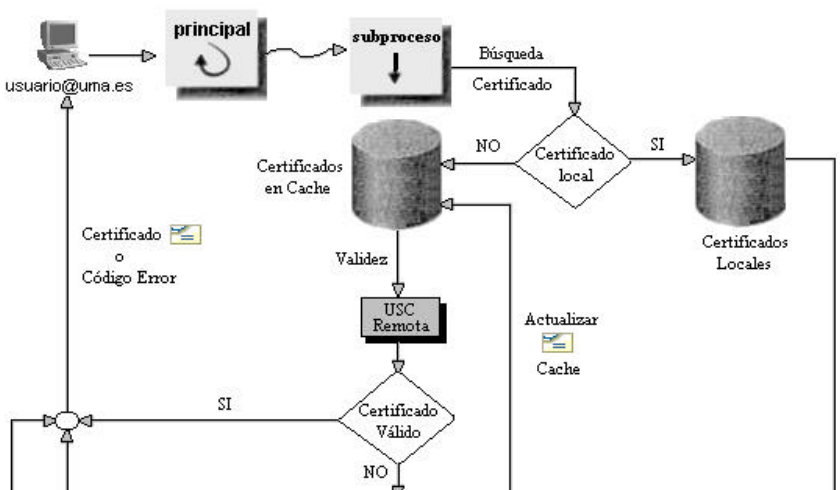

Figura 7. Pasos a seguir por el subproceso de servicio.

El protocolo en su fase de conexión, nos presenta la posibilidad de identificarnos, limitándo de esta forma las peticiones a usuarios y host eliminándo la posibilidad de ataques de denegación de servicio.

6. Conclusiones

En este trabajo se han descrito las caracteristicas de un sistema de administración de claves públicas y autenticación de usuarios para Internet. Se han expuesto las deficiencias de los sistemas actuales y se ha mostrado cómo Cert'eM puede corregirlas, aportando otras ventajas como son la revocación directa (en tiempo real) sin necesidad de CRLs y la facilidad de actualización de claves de forma transparente a todos los usuarios.

Actualmente se está estudiando la viabilidad de la introducción de un método que permita certificaciones cruzadas con el objeto de permitir mayor flexibilidad y conseguir verificaciones más rápidas. La solución que se estudia pretende evitar los problemas asociados con el cruce de certificados tales como la inviabilidad de la arquitectura cuando el número de cruces es demasiado elevado, la aparición de caminos de certificación cíclicos y la corrupción de la convención jerárquica de nombres como base del sistema.

Referencías

[Beck97] Becker, Thomas. "Porting Server Applications from UNIX to Windows NT". C/C++ Users Journal, Octubre 1997

[Brow94] Brown, Chris. *UNIX Distributed Programming*. Prentice-Hall, 1994

[Chok94] Chokhani S. "Toward a National Public Key Infrastructure". IEEE Communications Magazine, 1994, pp. 70-74

[CSE98] Communications Security Establishment. "Government of Canada Public Key Infrastructure - White Paper", 1998.

[Davi95] Davis, D. "Kerberos Plus RSA for World Wide Web Security". First USENIX Workshop on Electronic Commerce, 1995, pp. 185-188.

[DiHe76] Diffie, W. y Hellman, M. "New Directions in Cryptography". IEEE Transactions on Information Theory. IT-22, n. 6. 1976, pp. 644-654.

[Gane95] Ganesan, R. "Yaksha: Augmenting Kerberos with Public Key Cryptography". Internet Society Symposium on Network and Distributed Systems Security. IEEE Press, 1995, pp. 132-143.

[ISO88] ISO International Standard 9594. Information Technology - Open Systems Interconnection Reference Model: The Directory, 1988.

[ISO96] ISO/IEC JTC1/SC 21. Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, 1996.

[Kath99] Kathtan, Joseph. "Portable Control of Multiple Daemon Processes". C/C++ Users Journal, Mayo 1999

[Kohl89] Kohl, J. “The Use of Encryption in Kerberos for Network Authentication”. Advances in Cryptology - CRYPTO'89. LNCS 435. Springer, 1989, pp. 35-43.

[Mare96] Marejka, Richard. “Multi-threaded Programming”. Sun Microsystems,1996

[PKIX97] PKIX Working Group Internet Draft. Internet Public Key Infrastructure Certificate Policy and Certification Practices Framework, 1997.

[RFC1034] Mockapetris, P. “Domain names - concepts and facilities”, Noviembre 1987

[RFC1421] Linn, J., “Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures”, Febrero 1993.

[RFC1422] Kent, S. “Privacy Enhancement for Internet Electronic Mail. Part II: Certificate-Based Key Management”, Febrero 1993.

[RFC1423] Balenson, D., “Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers”, Febrero 1993.

[RFC1424] Kaliski, B. “Privacy Enhancement for Internet Electronic Mail. Part IV: Key Certification and Related Services”, Febrero 1993.

[RFC2065] Eastlake, D. y Kaufman, C. “Domain Name System Security Extensions”, Enero 1997.

[RFC2316] Bellovin, S. “Report of the IAB Security Architecture Workshop”, Abril 1998.

[NIST96] National Institute of Standards and Technology. “A Public Key Infrastructure for U.S. Government Unclassified but Sensitive Applications”, 1996.

[RiLa96] Rivest, R. y Lampson, B. “SDSI – A Simple Distributed Security Infrastructure”, 1996.

[Robb96] Robbins, Kay A & Steven. *Practical UNIX Programming. A guide to Concurrency, Comunication, and Multithreading*. Prentice-Hall, 1996

[Rute97] Rutenhof, David R. *Programming with POSIX Threads.* Addison-Wesley Professional Computing Series, 1997

[SPKI98a] SPKI Working Group Internet Draft. “Simple Public Key Certificate”, 1998.

[Stev90] Stevens, Richard W. *UNIX Network Programming*. Prentice-Hall,1990

[Zimm95] Zimmerman, P. *The Official PGP User’s Guide*. MIT Press, 1995.