Alternativa de Infraestructura de Clave Pública Basada en el uso de DNSSEC

Rolando Chaparro, Pablo Greenwood, Benjamín Barán
Universidad Nacional de Asunción, Centro Nacional de Computación
Asunción, Paraguay, CC1439
{rfox, pgreen, bbaran}@cnc.una.py

Resumen

El modelo de PKI más ampliamente difundido se basa en el uso de certificados digitales emitidos por Autoridades de Certificación o CAs (*Certificate Authorities*). Por lo general, las CAs están desvinculadas de la infraestructura de red sobre la que se necesita validar y utilizar los certificados. Esta disociación presupone algunas importantes limitaciones. En este artículo se define un modelo de PKI en el que las aplicaciones, en lugar de recurrir a las tradicionales CAs, utilizan las extensiones de seguridad del DNS, conocidas como DNSSEC, como base para la provisión de los servicios fundamentales de seguridad.

Palabras clave: Redes, Seguridad de Datos y Criptografía, PKI, DNS, DNSSEC.

Abstract

The most widely spread PKI model is based on digital certificates issued by Certificate Authorities (CA). In general, there is not a strong connection between these CAs and the underlying network infrastructure on which certificates must be validated and used. This dissociation entails an appreciable number of constraints. This paper proposes an alternative PKI model where applications take advantage of DNS security extensions (know as DNSSEC) as a foundation to build security services.

Keywords: Networking, Data Security and Cryptography, PKI, DNS, DNSSEC.

1 Introducción

Una PKI (*Public Key Infraestructure*) es el medio que facilita el acceso a las claves públicas, asegurando a sus usuarios la correspondencia unívoca de las mismas con sus respectivos propietarios. Este concepto se ha establecido en los últimos años y se ha usado como punto de referencia para proyectar una nueva generación de aplicaciones que hagan posible la provisión de servicios de seguridad a gran escala, sobre todo en sistemas abiertos como Internet. Sin embargo, las actuales implementaciones de PKI presentan aún estimables desafíos, lo cual ha dado origen a diversos estudios, propuestas y trabajos de investigación.

La mayoría de los modelos de PKI se basan en el uso de certificados digitales [30,31], los cuales comúnmente utilizan el formato X.509 [20]. Un certificado es en esencia un vínculo entre una clave pública y los atributos que identifican a una determinada entidad. En el modelo de certificación más difundido, este vínculo es mantenido por organizaciones tipo TTP (*Trusted Third Party*)[1], las que por lo general adoptan la forma de Autoridades de Certificación o CAs (*Certificate Authorities*).

Simultáneamente al auge de las PKIs, en los últimos años se ha venido trabajando en el marco de la IETF[2] en el desarrollo de extensiones de seguridad para el Sistema DNS de Internet [1,27]. Estas extensiones, conocidas como DNSSEC [3,9], han sido diseñadas con el fin de proporcionar servicios de autenticación del origen de los datos del DNS mediante verificaciones criptográficas.

DNSSEC posee algunas características significativas que serán analizadas en este documento con el propósito de demostrar que puede contribuir a definir una alternativa de PKI con un enfoque diferente. Este enfoque se basa en el principio de que los servicios de seguridad deben sustentarse en la propia infraestructura de red y en sus organizaciones, en lugar de depender de agentes externos.

En la sección 2 de este artículo se hace una muy breve presentación de DNSSEC. En la sección 3 se resumen las características elementales de las PKIs de Internet basadas en el concepto de TTPs y en la sección 4 se señalan algunas limitaciones importantes de las mismas. En la sección 5 se propone un nuevo modelo en el que las aplicaciones utilizan a DNSSEC como base para la construcción de servicios fundamentales de seguridad. Finalmente, en la sección 6 se ofrecen las conclusiones de este trabajo.

1 *Trusted Third Party:* tercera parte en la que los usuarios deben depositar su confianza
2 IETF: *Internet Engineering Task Force*

2 Fundamentos de DNSSEC

Las extensiones DNSSEC proporcionan servicios de autenticación del origen de los datos del DNS mediante verificaciones criptográficas[3]. Para ello se realizan firmas digitales sobre los registros de la base de datos del DNS. Estas firmas y sus claves públicas asociadas son insertadas en las tablas de zona [28] mediante registros tipo DNSSIG y DNSKEY respectivamente, los cuales son incluidos y enviados en los mensajes de respuesta a las consultas DNS [4]. Los clientes, denominados *resolvers*, finalmente verifican los datos de respuesta empleando las firmas digitales y las claves públicas recibidas [2].

La legitimidad de las claves públicas es corroborada mediante el recorrido del *chain of trust* del DNS [14]. En este modelo de autenticación, las claves públicas de una zona son convalidadas por su zona padre a través de una firma digital denominada DS (*Delegation Signing*) [14]. Esta relación entre una zona y su antecesora se reproduce siguiendo la jerarquía del árbol de DNS hasta llegar a una zona cuyas claves públicas son consideradas confiables (*secure entry keys*).

En resumen, DNSSEC provee una infraestructura que soporta el uso de claves públicas para dar servicios de seguridad, cuenta con un mecanismo de distribución de estas claves (el DNS) y confiere autenticación a las mismas. Aunque estas características se ajustan al concepto de PKI, se debe notar que DNSSEC ha sido ideado únicamente para brindar protección contra los ataques de tipo *spoofing* al DNS [6]. DNSSEC se diferencia además de una PKI tradicional en que el protocolo no utiliza certificados, el directorio es descentralizado y la autenticación se basa en autoridades subdivididas y delegadas [3].

3 Las Infraestructuras de Clave Pública en Internet

De todos los modelos de PKI, el basado en certificados digitales emitidos por TTPs, es el que ha logrado mayor difusión en la Internet global, así como en soluciones corporativas, llegando de forma masiva a una gran cantidad de usuarios. Por consiguiente, de aquí en adelante este artículo se referirá principalmente a este paradigma de PKI, cuyas características elementales se resumen a continuación.

A partir del concepto de certificación de TTPs han sido diseñadas varias propuestas de PKI, pero el esquema adoptado por X.509 es actualmente la base de casi todos los productos existentes. Conforme se ilustra en la Figura 1, el rol central es desempeñado por las CAs, que emiten los certificados a las entidades. Opcionalmente las CAs pueden delegar labores administrativas a otras organizaciones denominadas RAs (*Registration Authorities*), pero son las CAs las que en realidad actúan como TTPs, en cuyos certificados los usuarios deben depositar su confianza. El estándar X.509 ha evolucionado en sucesivas versiones hasta la actual X.509v3 [21], cuyo formato de certificado se aprecia en la Figura 2.

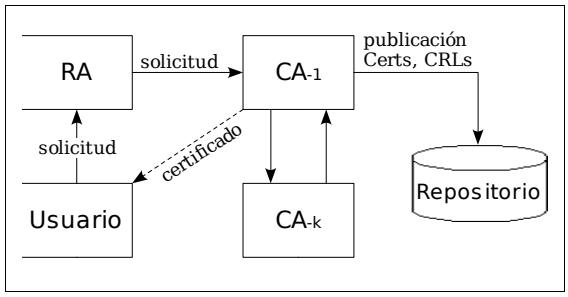

Figura 1: Estructura básica de PKI tipo TTP

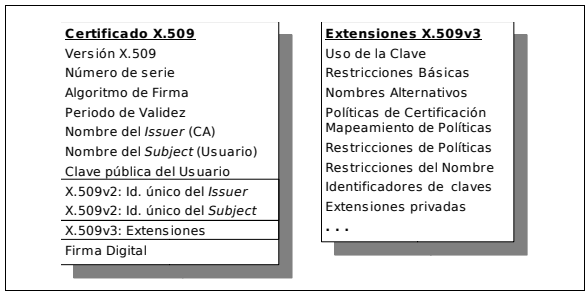

Figura 2: Formato del certificado X.509v3

El requerimiento fundamental de un certificado es establecer y mantener de forma precisa el vínculo entre los atributos que lo constituyen. En la mayoría de los casos, los atributos de interés dominante son una clave pública y los datos que identifican al propietario de esta clave.

X.509 fue diseñado como un esquema de certificación que define una infraestructura de clave pública para autenticar los servicios del directorio X.500 [18]. Los certificados y otras estructuras de datos de la PKI son guardados en el mismo directorio. La identificación de un certificado y su consecuente ubicación en el directorio X.500 se realiza en base a un nombre distintivo o DN (*distinguished name*), que describe un camino jerárquico único en el directorio, entre la raíz y el propietario del certificado [19]. Algunos de los atributos más importantes de los certificados X.509, como el *Subject Name* y el *Issuer Name*, están en formato DN. Un DN consta de un conjunto de atributos abreviados, cada uno asociado a un nivel en la jerarquía X.500. Ejemplo: {C=PY, O=UNA, OU=CNC, CN=Rolando Chaparro Fox}[4].

3 Actualmente se está concluyendo la revisión y corrección de las especificaciones originales del RFC-2535 (1999)
4 C (*country*), O (*organization*), OU (*organizational unit*) y CN (*common name*)

X.509 define un método de revocación de certificados, denominado CRL (*Certificate Revocation List*), consistente en una estructura de datos firmada por la CA y actualizada periódicamente, la cual contiene la lista de certificados revocados. Recientemente han sido desarrollados protocolos como OCSP (*Online Certificate State Protocol*) que permiten realizar consultas en línea del estado de los certificados [26].

4 Algunas Limitaciones de las PKIs Basadas en las Actuales CAs

Es posible diferenciar a las CAs en función de su tipo de organización y a la manera en que están facultadas a emitir certificados. A continuación se presenta una clasificación tentativa[5] basada en [24]:

- *Organizacional*: Típicamente es una universidad o empresa que actúa como su propia CA para emitir certificados a las sub-unidades que la componen y a los integrantes de las mismas.
- *Geopolítica*: El gobierno, o alguna organización que reciba la anuencia de éste, expide los certificados a las entidades en base a un modelo nacional de PKI.
- *Universal*: La autoridad de certificación de una hipotética PKI global que es reconocida por todos.
- *Propietaria*: Una organización que controla un determinado espacio de nombres y que emite certificados que son utilizados en ese contexto.
- *Comercial*: Empresa que actúa como TTP y cuyo negocio es la emisión de certificados en los que las entidades de la PKI deben depositar su confianza.

Una inmensa mayoría de los certificados de sitios de Internet son emitidos por CAs que corresponden a este último grupo. Tomándolo como referencia, los puntos a continuación están orientados a señalar algunos desafíos que enfrentan estas PKIs, en busca de una mejor alternativa desde el punto de vista técnico, con mayores ventajas para los usuarios.

4.1 Relación de confianza desde la perspectiva del usuario.

Es generalmente aceptada la idea de que el usuario es responsable de una importante parte de las evaluaciones de confianza en el proceso de autenticación. Por ejemplo, las denominadas *trusted-CAs* son hipotéticamente aquellas CAs elegidas por el usuario para depositar su confianza. Sin embargo, en la práctica adquieren este rango por el sólo hecho de que sus certificados auto-firmados (certificados raíz) están pre-instalados en el *browser*. Es decir, el usuario promedio no se toma el trabajo de examinar los certificados raíz de las CAs predefinidas para decidir si confía o no en ellas.

De forma similar, se sostiene que los certificados recibidos contienen importantes datos sobre los que se espera que el usuario realice algún tipo de discernimiento o juicio de valor. Este es el caso de los nombres del propietario (*Subject*) y de la CA (*Issuer*). Pero aquí nuevamente se presentan algunas inconsistencias.

Veamos primero el caso del *Issuer*. Basta que los certificados recibidos estén firmados por alguno de los muchos certificados raíz incluidos en el *browser*, para que haya una aceptación implícita de los mismos[6]; pero acabamos de mencionar que las *trusted-CAs* nunca fueron validadas por el usuario. Esto implica que, en la práctica, los vendedores de las aplicaciones intervienen de forma decisiva condicionando los criterios de confiabilidad del usuario. Esta situación de hecho es una deformación del concepto de confianza.

La situación no es muy diferente respecto al *Subject*. Supongamos que existen dos empresas con la misma denominación, digamos "*Multimedia Inc*"; una es una agencia de publicidad y la otra desarrolla software multimedia[7]. Al acceder al sitio web *www.multimedia.com.py*, el usuario difícilmente podrá discernir a partir de los datos incluidos en el *Subject*, a cual de las dos empresas se refiere el certificado recibido (si es que acaso este usuario es de los pocos que optan por examinar certificados y además, coincidentemente, sabe de la existencia de otra empresa con el mismo nombre). Este es en el fondo un problema inherente a la multiplicidad y a la diversidad de los espacios de nombres, que ninguna de las soluciones actuales puede resolver con absoluta precisión [11].

Finalmente, dado que la mayoría de los usuarios ni siquiera revisa el contenido de los certificados recibidos, ni de los certificados raíz, tiene poca importancia que sus atributos sean datos estructurados y legibles por las personas. En la práctica, el usuario se vale de otras señales que le ayudan a identificar a la empresa una vez que haya accedido al sitio web; o bien, tiene algún grado de certeza de que el *host* corresponde a la organización a la que él tiene interés de contactar.

---

5 En esta clasificación, los tipos de CA no son mutuamente excluyentes

6 Este es el comportamiento por defecto del explorador web utilizado por cerca del 90% de los usuarios de Internet

7 Al tratarse de diferentes rubros, este tipo de igualdad de nombres es válida en muchas legislaciones del mundo

4.2 Manejabilidad de las soluciones X.509

Dado que casi la totalidad de los vendedores se basan en X.509, puede tenerse la sensación de que este escenario siempre resulta conveniente a todas las partes. Después de todo, para eso están los estándares. Pero el estándar X.509 es muy general y por lo tanto tiene varias imprecisiones. Esto implica que en la práctica se necesita definir un perfil específico para implementar un sistema real de certificación.

Hoy existen múltiples vendedores de PKI que emplean diferentes perfiles de X.509. Como consecuencia, las soluciones difieren en sus implementaciones, incluso en aquellos atributos más importantes de los certificados. Esta multiplicidad de soluciones implica también entornos de operación divergentes, cada uno con sus propias políticas, herramientas y prácticas. En muchos casos se vuelve necesario realizar trabajos personalizados de adecuación para lograr un aceptable grado de interoperabilidad.

Actualmente se están realizando esfuerzos para minimizar los inconvenientes de interoperabilidad, desde diferentes áreas y en diferentes etapas de definición. PKIX [17] es tal vez el más notable de ellos. Sin embargo, no puede perderse de vista que la idea de interoperabilidad comúnmente implica transitividad de la confianza depositada en las CAs y en sus sistemas de certificación. Esto a pesar de que la confianza es generalmente un concepto relativo y no transitivo.

Muchas aplicaciones actuales basadas en la tecnología de clave pública podrían beneficiarse de soluciones más simples, con menor grado de dependencia de las complejidades derivadas del uso de X.509 y de las relaciones de confianza.

4.3 Estructura de costos

Idealmente, la escala de precios de cualquier producto debería estar definida en función y en proporción a los servicios y beneficios obtenidos por los consumidores. En este sentido, la actual estructura de costos de las CAs define niveles de seguridad muy generales para los certificados. Difícilmente estos niveles puedan corresponder de forma adecuada a los requerimientos de todo el rango de aplicaciones en las que hoy son empleados los certificados.

Por otra parte, las entidades deben realizar una considerable inversión si pretenden tener varios certificados compatibles con las rutas de certificación pre-instaladas en las aplicaciones de uso masivo[8]. Por ejemplo, al momento de escribir este artículo, VeriSign[9] estaba ofertando un paquete de 100 (cien) certificados a un precio de US$ 57.000[10]. Si bien a las entidades que adquieren este paquete se les facilita herramientas para administrar sus certificados, los mismos en realidad son firmados por la CA.

A esto hay que agregar que, al estar pagando en esencia por la firma de sus certificados, una entidad que decida cambiar de CA debe realizar una reinversión completa. Esto es así aún cuando sólo haya usado el certificado por un breve tiempo, o no lo haya usado en absoluto. Es posible que a partir de modelos alternativos de PKI, surjan también nuevos modelos de negocios; aunque debe reconocerse que muchas veces es más rentable desarrollar productos a partir de estructuras y tecnologías ya establecidas.

4.4 Repositorios y servicios de directorio

DAP (*Directory Access Protocol*) es el protocolo original de acceso al servicio de directorio X.500. El mismo fue diseñado para soportar la infraestructura de clave pública definida por X.509. Sin embargo, LDAP(*Lightweight Directory Access Protocol*), que es la herramienta utilizada actualmente por la mayoría de las implementaciones de PKI, ha tenido un origen distinto y con diferentes motivaciones [32]. Esto ha originado una serie de desajustes operacionales, los cuales se describen de forma precisa en [8].

Otro aspecto que aún necesita de una solución satisfactoria es que los repositorios de certificados puedan ser desplegados de forma masiva en Internet, brindando mecanismos simples de recuperación. En este sentido, trabajos recientes han explorado la posibilidad de emplear al sistema DNS como repositorio de certificados de una PKI [22].

De forma similar, las propias CAs han planteado utilizar al DNS para facilitar a los usuarios la localización de sus repositorios de certificados y listas de revocación. La propuesta denominada RLS (*Repository Locator Service*) es actualmente un *draft* de la IETF que plantea asociar a un nombre de dominio, el protocolo y la dirección de estos servicios de la PKI [7].

---

8 Navegadores web o el software de correo electrónico

9 VeriSign es actualmente una de las empresas más importantes en el mercado de las CAs comerciales

10 Corresponde a los certificados con soporte de cifrado simétrico de 128 bits

4.5 Nombres e identidad

El nombre de la entidad propietaria es uno de los atributos decisivos y de mayor importancia en un certificado. El propio estándar X.509 establece que es responsabilidad de una CA asegurar la unicidad de los nombres de las entidades para las que emite los certificados.

El diseño original de X.509 presupone que los nombres en el certificado están en formato DN[11]. Para garantizar su unicidad, los DNs están organizados de forma jerárquica y se asume que hay una autoridad que hace que los mismos sean globalmente únicos y significativos. Sin embargo, esta idea nunca se puso en práctica. En consecuencia, al no estar integrados los servicios de directorios, ni estandarizados los criterios de asignación de nombres, las CAs no cumplen con este requisito de unicidad.

Dos organizaciones distintas podrían tener el mismo DN en dos CAs diferentes, también es posible que una misma organización tenga DNs disímiles en ambas CAs. Esta ambigüedad representa un potencial problema de interoperabilidad. Incluso en el ámbito cerrado de una misma CA, los DNs de sus clientes no reflejan verdaderamente ninguna organización jerárquica. El siguiente ejemplo ilustra una de las tantas discordancias que podrían ocurrir a raíz de la permisividad con la que se manejan los DNs en los certificados emitidos por las CAs:

{C=PY, O=Gobierno Central, OU=Secretaría de Economía, ...}
{C=PY, O=Gobierno Nacional, OU=Secretaría de Salud Pública, ...}
{C=PY, O=Poder Ejecutivo, OU=Secretaría de Educación y Cultura, ...}

Como puede notarse, la nomenclatura jerárquica implícita en los DNs pierde significación. En la práctica, las aplicaciones procesan los nombres en los certificados como si fueran simples conjuntos de datos no estructurados. Además, como se ha mencionado en la sección 4.1, tiene escasa relevancia que los mismos sean legibles por una persona, dado que el usuario promedio no examina los certificados.

La situación del ejemplo arriba señalado no se reproduce tan fácilmente en el Sistema DNS de Internet, donde la operación del servicio y la responsabilidad de asignación de los nombres de dominio se distribuye y delega efectivamente de manera jerárquica. El DNS es una verdadera estructura de árbol en la que se asegura la unicidad de los nombres.

El DNS se encuentra en un ámbito externo, desvinculado de las PKIs. A pesar de ello, existe una clara dependencia por parte de las CAs hacia el DNS, dado que éste es el espacio de nombres de la infraestructura de red en la que se pretende desplegar los servicios de seguridad. Esta dependencia se vuelve evidente en servicios abiertos como SSL (*Secure Socke Layer*) [12]. Para hacer uso efectivo de este protocolo, los certificados de sitios web expedidos por las CAs incluyen en el DN del *Subject* el nombre DNS del sitio. Sólo así es posible identificar en el ámbito de Internet al propietario de un certificado.

4.6 Confianza y autoridad

Las CAs son terceras partes en las que los usuarios deben depositar su confianza. Sin embargo, en el caso de las actuales PKIs de uso masivo, se ha visto en la sección 4.1 que esta relación en la práctica no se establece en base a verdaderos criterios de confiabilidad. Aunque así fuera, la confianza es en realidad una relación compleja, no transitiva, de carácter relativo, no cuantificable y culturalmente influenciada.

Sin embargo, la autenticación no necesita estar basada en la confianza. Si una entidad es autoritativa[12] sobre un determinado espacio de nombres, la débil y difusa noción de confiabilidad es irrelevante. Uno no se pregunta si se deposita o no confianza en una determinada empresa para entregar carnés de identificación a sus propios empleados, o en un determinado país para expedir pasaportes a sus ciudadanos. En estos ejemplos, cada entidad emisora es autoritativa sobre el espacio de nombres en el que estas credenciales son emitidas, de modo que la confianza es intrínseca [24].

Se sostiene que los usuarios deben confiar en las CAs, pero son muy limitados el aval y la responsabilidad que éstas pueden ofrecer sobre la autenticidad, la integridad o la exactitud de los datos contenidos en los certificados, lo cual se refleja en sus términos contractuales [13]. Esto se debe sobre todo a que las CAs no son autoritativas sobre ningún espacio de nombres, y esto incluye al espacio de nombres de la infraestructura de red en la que los certificados son efectivamente evaluados y utilizados (el Sistema DNS de Internet). En cambio, las entidades autoritativas de un espacio de nombres tienen una responsabilidad igualmente intrínseca. Ellas no pueden eludir las obligaciones que se desprenden de su autoridad.

---

11 Si bien X.509v3 permite otros tipos de nombres, la nomenclatura DN es mantenida por todas las CAs

12 Para la mayoría de los tipos de nombres existen entidades del mundo físico que tienen el control o que manejan el espacio en el cual los nombres son asignados. Se dice de estas entidades que son autoritativas (que tienen autoridad) sobre esos espacios de nombres.

4.7 Mecanismos de revocación

Las listas de revocación de las PKIs son escasamente utilizadas en la práctica. Por ejemplo, SSL, que es la principal aplicación de seguridad de la actualidad, no descarga los CRLs. Esto se debe en parte al gran tamaño de algunos de ellos y a la potencial saturación del servicio, el cual es esencialmente centralizado.

Ante una situación de clave comprometida, las CAs deben desplegar procedimientos adicionales de seguridad que pueden demorar, aumentando la latencia entre el reporte de revocación y la distribución efectiva de la información a todas las partes.

Los protocolos de verificación en línea también tienen sus limitaciones. Si bien OCSP puede reportar el estado de un certificado, en la práctica este método aumenta la complejidad del modelo al imponer un nuevo requerimiento de seguridad. El cliente debe también verificar la integridad y la autenticidad del origen de los datos de este nuevo servicio de validación.

5 Infraestructura de Clave Pública Basada en DNSSEC

Un certificado digital es en esencia un vínculo entre una clave pública y los atributos que identifican a una determinada entidad. El modelo de certificación más difundido plantea que este vínculo es mantenido por CAs comerciales, en las cuales los usuarios deben depositar su confianza. Sin embargo, en la sección 4 se ha visto que este enfoque trae consigo una serie de limitaciones[13].

En esta sección se plantea una alternativa de PKI desde una perspectiva diferente que no depende de agentes externos. Se propone un nuevo modelo en el que la propia infraestructura de red de Internet contribuye a la provisión de los servicios fundamentales de seguridad. Para ello se vale de la jerarquía de autenticación implícita en el DNS y sus extensiones de seguridad.

5.1 Aspectos generales del modelo

Además de la información estrictamente necesaria para realizar la resolución de nombres, el diseño original del DNS faculta a una organización a incluir en su base de datos otros atributos[14] asociados a sus entidades (*hosts*, sub-dominios, personas, roles), permitiendo a las aplicaciones hacer uso de los mismos.

Basado en este mismo principio, el modelo aquí expuesto propone la creación de un RR destinado a publicar a través del DNS las claves públicas asociadas a una entidad. Este nuevo registro, denominado APPKEY, permitirá a las aplicaciones localizar, recuperar y hacer uso de estas claves. Dada su comprobada eficacia como servicio de directorio simple, su escalabilidad, su naturaleza distribuida y su uso extendido en toda la red Internet, el sistema DNS resulta ideal para este trabajo.

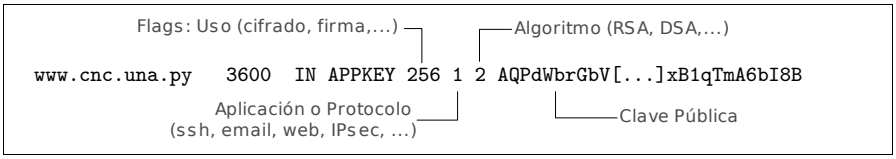

Figura 3: Ejemplo de registro APPKEY con su posible conformación de RDATA

La integridad del registro APPKEY y la legitimidad de su origen, estarán dadas por DNSSEC y su *chain-of-trust*. La posibilidad de obtener las claves públicas ya autenticadas representa muy claramente una alternativa a la utilización de certificados emitidos por CAs externas.

Una aplicación que ha recuperado una clave pública a través de DNS/DNSSEC puede pasar directamente a comprobar la autenticidad de la otra parte con la que pretende establecer comunicación[15]. Igualmente, valiéndose de esta clave pública, puede proceder a negociar y a establecer otros servicios de seguridad como la confidencialidad.

5.2 La infraestructura de red como sustento de los servicios de seguridad

El DNS tiene como finalidad principal facilitar a los usuarios la identificación y localización de entidades, típicamente de *hosts*, mediante el mapeo de nombres de dominio a números IP. Si bien estos dos atributos, Nombre DNS + Dirección IP, eran suficientes para cumplir con este cometido en los inicios de Internet, los actuales requerimientos de seguridad de una red global demandan mayor esfuerzo.

13 Información más completa y detallada puede encontrase en [10,13,29].
14 Mediante *Resource-Records* como MX, SRV, HINFO, TXT o CERT.
15 Para ello, la otra parte debe demostrar la posesión de la correspondiente clave privada.

En este contexto fueron diseñadas las extensiones DNSSEC, las cuales permiten confirmar la asociación Nombre DNS + Dirección IP. No obstante, un intruso puede aún apropiarse indebidamente de un número IP. En tal sentido, este trabajo plantea que una entidad puede ser positivamente identificada a partir de sus atributos Nombre DNS + Dirección IP + Clave Pública, y que este último dato también debería poder obtenerse del servicio de infraestructura DNS. Esta es justamente la finalidad el registro APPKEY.

En Internet, cada organización tiene autoridad sobre sus nombres de dominio. En base a esta autoridad definen los nombres particulares que tendrán sus entidades. La presente propuesta refleja fielmente este importante principio, pues cada organización es libre de firmar y publicar las claves de sus entidades.

Por otra parte, fácilmente se infiere que en este modelo, las organizaciones que administran los nombres de dominio correspondientes a cada país (*ccTLD*[16] *Authorities*) desempeñarán de forma espontánea un rol equivalente al que hoy cumplen las CAs. Esta idea tiene sólidos fundamentos:

- Existe gran similitud entre las operaciones realizadas en el registro de nombres de dominio bajo un ccTLD y las solicitudes de certificados a una CA. Actualmente se realizan de forma redundante trámites que son equiparables, y en la práctica es un requisito tener antes un nombre de dominio.
- Las *ccTLD Authorities* tienen un vínculo más cercano con las empresas y organizaciones nacionales. Esto les brinda mayor autoridad para los procesos de verificación respecto a las CAs comerciales. Incluso hay muchos países en los que las CAs ni siquiera tienen presencia.
- Las CAs recurren a las bases de datos de las *ccTLD Authorities* como práctica normal para validar parte de la información suministrada por sus clientes.
- Las *ccTLD Authorities* son las entidades autoritativas para la delegación de los nombres DNS a las empresas y organizaciones con presencia en Internet en cada país. Considérese que el DNS es el espacio de nombres de la infraestructura de red sobre la que se pretende desplegar los servicios de seguridad.
- Al estar estrechamente vinculadas al funcionamiento de Internet en sus respectivos países, las *ccTLD Authorities* siempre están presentes; pueden cambiar de administración, pero no desaparecer.
- Son organizaciones sin fines comerciales, ampliamente reconocidas y que tienen el respaldo de la Comunidad Internet local de cada país. Por lo general son asociaciones de ISPs y usuarios, o universidades, o comités multi-sectoriales, en ocasiones con participación de agencias gubernamentales.

En resumen, esta propuesta defiende el principio de que los servicios de seguridad deben sustentarse en la propia infraestructura de red y en sus organizaciones, en lugar de depender de agentes externos.

5.3 La autenticación como base del modelo propuesto

En toda infraestructura de red, la autenticación es el servicio de seguridad fundamental, del que dependen y a partir del cual se *construyen* todos los demás servicios, por ejemplo la confidencialidad. En Internet, y en cualquier red TCP/IP, se plantean siempre tres preguntas centrales desde el punto de vista de la autenticación. Para ilustrar cómo responde el modelo propuesto a estas interrogantes, retomemos el ejemplo del usuario tratando de acceder al sitio web de *Multimedia Inc.* Las preguntas en cuestión son:

a. Es en verdad *www.multimedia.com.py* un nombre DNS perteneciente a la organización *Multimedia Inc,* con la que el usuario quiere establecer contacto ?

b. Es la dirección IP obtenida a través del DNS la que corresponde realmente a *www.multimedia.com.py* ?

c. Se está realizando la comunicación con el *host* verdadero o con otro que se ha apropiado indebidamente de su número IP ?

Respecto a la primera pregunta, si bien es posible argumentar que el *Subject* es un dato suficientemente descriptivo, en la sección 4.1 se han mencionado algunas razones de peso que cuestionan el uso real que se da actualmente a este y a otros atributos de los certificados. Se ha visto además que existen espacios de nombres que no garantizan unicidad, y que aún las actuales soluciones basadas en X.509 no responden con absoluta precisión a esta interrogante.

Sin embargo, como se indica en la sección anterior, los administradores de los ccTLDs son tanto o más confiables que las actuales CAs para identificar a las organizaciones en el ámbito de un determinado país. Además, tienen autoridad sobre el espacio de nombres de la infraestructura de red subyacente.

No obstante, hay alternativas para aquellos usuarios en situación de duda. Las aplicaciones podrían por ejemplo disponer de una operación opcional para recuperar, a partir del servicio *whois* de los ccTLDs, los datos requeridos para identificar a una entidad. Las bases de datos *whois* contienen información generalmente más rica y completa que la que puede encontrarse actualmente en un certificado.

16 ccTLD: *Country Code Top Level Domain.* Son los nombres de dominio de los países (.ar, .br, .cl, etc.)

En cuanto a la segunda pregunta, la autenticidad de los datos del DNS estará dada por DNSSEC, eliminando la necesidad de certificados de CAs externas debido a que las claves públicas estarán legitimadas por las propias organizaciones y por la jerarquía de autenticación de DNSSEC.

Finalmente, la tercera pregunta sólo puede responderse demostrando en términos criptográficos la posesión de la correspondiente clave privada. Este es un paso indispensable en cualquier protocolo o aplicación basada en la criptografía de clave pública.

5.4 Obtención y autenticación de las claves públicas

La Figura 4 ilustra de forma simplificada cómo una aplicación cliente, consultando a su servidor DNS local, recupera la dirección IP y la clave pública del *host* destino.

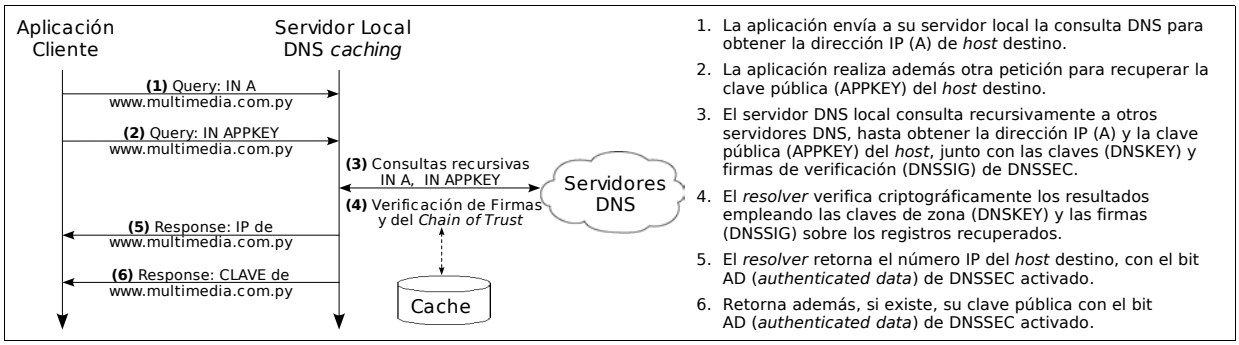

Figura 4: Pasos abreviados de la interacción con DNS/DNSSEC para recuperar una clave pública

Gran parte de la complejidad del protocolo DNSSEC reside en el *resolver*. En la práctica los pasos 3 y 4 de la Figura 4 no se efectúan necesariamente de forma secuencial. Aquí se hace una abstracción respecto a los algoritmos concretos que el *resolver* emplea para obtener los resultados parciales a sus consultas y para realizar las verificaciones. De hecho, estos procedimientos se encuentran aún en estudio [3].

A partir de la obtención y validación de la clave pública del registro APPKEY, la aplicación está ahora en condiciones de autenticar a la entidad destino. Luego podrá intercambiar claves secretas y negociar los algoritmos de cifrado simétrico y de MAC (*message authentication code*) para establecer servicios de seguridad como la integridad de los datos y la privacidad, tal como se hace en protocolos como SSL, TLS.

Está fuera del alcance de este artículo definir los mecanismos específicos para la realización de estas operaciones. Sin embargo, en la siguiente sección se mencionan algunas posibles alternativas con el fin de realizar estimaciones teóricas de rendimiento.

5.5 Algunas consideraciones acerca del rendimiento

Con el fin de facilitar su apreciación, en esta sección se ha elegido como ejemplo el acceso a un servidor web/SSL para realizar comparaciones de referencia. Sin embargo, el modelo propuesto puede también ajustarse a otros tipos de aplicaciones, como el intercambio seguro de mensajes de correo electrónico.

5.5.1 Tamaño y cantidad de paquetes

Al realizar una primera comparación, se observa en la Figura 5 que el servidor SSL envía su clave pública en el certificado X.509, el cual es transferido en un único mensaje del sub-protocolo *Handshake.* Sin embargo, en las Tablas 1 y 2 se aprecia que ambos enfoques emplean en realidad un *round-trip* completo para la entrega de la clave pública[17]. Por un lado, el mensaje *Certificate* de SSL y su necesario TCP *Acknowledge*; y por el otro, los mensajes *Query* y *Response* de DNS[18] (pasos 2 y 6 de la Figura 4).

Sin embargo, es factible que la aplicación cliente obtenga con una sola consulta DNS la dirección IP y la clave pública. Para ello, este último dato puede incluirse en la sección adicional del mensaje de respuesta a la consulta DNS tipo A[19]. Esto eliminaría los mensajes vinculados a la petición explícita de APPKEY, aunque aún podría realizarse de ser necesario. Es decir, el *round-trip* para obtener la clave pública puede reducirse a cero, dado que de todas formas hay que recuperar del DNS la dirección IP.

17 En la Tabla 2, el tamaño estimado de *Query* está basado en el tamaño de un típico mensaje DNS de 800 bits [1]. Para el mensaje *Response* se asume el peor escenario y se utiliza el tamaño máximo del *payload.*

18 Se consideran solo los mensajes intercambiados entre la aplicación y el servidor DNS local para obtener la clave pública. No se toman en cuenta los mensajes recursivos originados por este último.

19 Esta misma idea se encuentra en la sección 3.5 de la especificación original de DNSSEC [9].

Mensaje	*Encabezado TCP*	*Encabezado HS SSL*	*Payload (X.509)*	*Total*
Certificate	20	4	1160	1184
(TCP ACK)	20	-	-	20
Total				**1204**

Tabla 1: Cantidad aproximada de bytes para la obtención del Certificado X.509 en SSL v3.0

Mensaje	*Encabezado UDP*	*Tamaño Estimado de Mensaje DNS*	*Total*
Query	8	100	108
Response	8	512	520
Total			**628**

Tabla 2: Cantidad aproximada de bytes para la obtención de la clave pública en el modelo propuesto

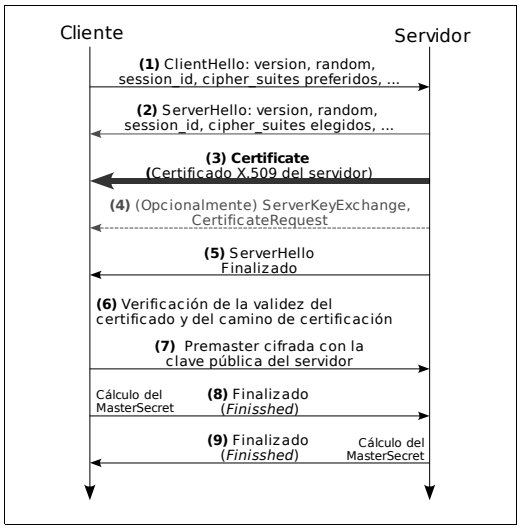

Figura 5: HandShake abreviado de SSL

También en las Tablas 1 y 2 puede verse que la cantidad teórica[20] de bytes enviados y recibidos para obtener la clave pública es mayor en SSL. Esto se debe al tamaño del certificado[21] y al mayor *overhead* de TCP (usado por SSL) respecto a UDP. Este último es el servicio de transporte por defecto de DNS.

No obstante, los valores de la Tabla 2 pueden verse aumentados si se contemplan los mensajes DNS recursivos disparados por el servidor DNS local (paso 3 de la Figura 4). La cantidad de consultas recursivas que lleguen a hacerse depende de varios factores, entre ellos la información ya guardada en el *cache* del servidor local y en los *caches* de los servidores consultados por este.

Esto significa que el *cache* del DNS desempeña un papel muy importante en el modelo propuesto, en particular desde el punto de vista de las cantidades totales de paquetes y bytes transmitidos. El *cache* permite que las consultas recursivas sean realizadas únicamente cuando el TTL del RR APPKEY haya expirado. El resto del tiempo, la clave estará disponible para las aplicaciones en el *cache* del servidor DNS local.

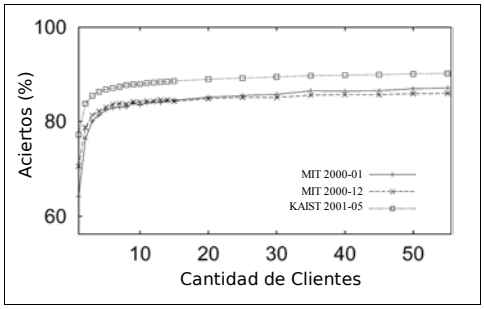

Figura 6: Porcentaje de Aciertos de los *caches* de varios servidores DNS en función a la cantidad de clientes

La Figura 6 muestra uno de los resultados de un reciente trabajo de análisis de la efectividad del *cache* del DNS [23]. En ella se observa que el índice de aciertos del *cache* mejora al aumentar la cantidad de clientes que están consultando, y que los mayores beneficios se notan ya con los primeros 10 a 20 usuarios. A partir de la misma gráfica se puede presumir que, en general, la probabilidad de encontrar un registro en el *cache* de un servidor local aumenta en proporción a la cantidad de consultas que este recibe. Por lo tanto, la mayor parte del tiempo, aquellas claves más requeridas serán recuperadas localmente. Esto representa un mejor aprovechamiento del ancho de banda y es una importante ventaja respecto a SSL.

La inclusión del APPKEY en la sección adicional de un mensaje de respuesta puede hacer que se exceda el tamaño máximo del *payload* de DNS (512 bytes). En tal caso, se emplea el mecanismo de extensión EDNS0 para no realizar un *fall-back* a TCP, como se define en [15]. Además, el tamaño máximo de paquetes UDP en IPv6 es cercano a los 1500 bytes, lo que casi elimina la posibilidad de *overflow*.

Existe también una trabajo denominado SK-DNSSEC [5] que sugiere el uso de criptografía simétrica en DNSSEC. Dicha propuesta plantea una reducción del tamaño de los mensajes, entre otras ventajas.

Además, cabe destacar que, cuando DNSSEC sea parte de la infraestructura de red, el *chain-of-trust* será indefectiblemente verificado para establecer cualquier comunicación TCP/IP normal, por lo que la presente propuesta prácticamente no agregaría *overhead* adicional. En cambio, las aplicaciones que en ese entonces sigan usando certificados de las CAs externas estarían realizando operaciones de forma redundante y a la vez desaprovechando los recursos disponibles.

20 Se omiten las capas inferiores y se presume que no ocurre fragmentación a causa de paquetes muy grandes.
21 Tamaño habitual de un certificado de VeriSign con clave RSA de 1024 bits

5.5.2 Procedimiento abreviado de negociación

En la Figura 5 se observa que en SSL el cliente envía primero la lista de los algoritmos de cifrado y de control de integridad que soporta, luego el servidor responde indicando sus elecciones.

En el modelo propuesto es posible disminuir el *round-trip* de esta fase inicial de negociación. En el mismo RR APPKEY podría incluirse las preferencias del servidor respecto a estos algoritmos. Al obtener con una sola consulta DNS la clave y los algoritmos preferidos del servidor, el cliente podría pre-ajustar sus parámetros y enviar en un único mensaje los datos incluidos en los paso 1 y 7 de la Figura 5. Si el cliente no soporta los algoritmos indicados por el servidor, entonces estas operaciones de *handshake* pueden realizarse de la manera habitual.

5.5.3 Carga computacional de las verificaciones criptográficas

En el modelo propuesto, la carga computacional de la verificación de la clave pública se transfiere de la aplicación del cliente al *resolver* del servidor DNS local. Por ello es importante estimar la sobrecarga potencial que este componente de software podría experimentar.

Además de la verificación de la firma sobre su RRset[22], un simple registro como A o APPKEY requiere del recorrido del *chain-of-trust* para verificar todas las delegaciones. En las nuevas especificaciones de DNSSEC, la carga computacional de las verificaciones para consultas sobre un determinado nivel de dominio n, están dadas por $C = f(n) = 2n+1$ [14].

En el modelo propuesto, la consulta DNS para APPKEY se hace de forma casi simultánea a la consulta de la dirección IP. Esto hace improbable que la verificación de la delegación (el primer término en la función anterior) se realice dos veces. Entonces, si se conoce la proporción de sitios web seguros, puede calcularse la sobrecarga computacional SC, respecto a la carga base C:

$$SC = f(n, p) = \frac{1}{2n+1} p$$

Donde p es la proporción de sitios web seguros para ese nivel

Estadísticas recientes[23] muestran que 1.53% es la proporción de sitios web seguros en Internet. Asumiendo que aproximadamente esa misma proporción se mantiene para todos los dominios de tercer nivel, la sobrecarga computacional SC es apenas de 0.2% para ese nivel.

Además, aquí el *cache* juega nuevamente un importante papel. La verificación criptográfica se realiza sólo si ha transcurrido el TTL de APPKEY en el servidor DNS local. Esto se aplica para todos los usuarios que accedan a su *cache*. En consecuencia se logra un mejor desempeño computacional colectivo.

5.5.4 Otros recursos computacionales

La inclusión de APPKEY tiene escasa incidencia en los recursos de cómputo del proceso de firma de las zonas. La mencionada proporción de 1.53% de servidores web seguros, es abismalmente pequeña en relación al 98.47% restante que, de igual manera, debe ser firmado conforme lo requiere DNSSEC.

Por el mismo motivo, los servidores DNS, sus *caches* y sus requerimientos de almacenamiento y de memoria RAM tampoco tendrían que verse en general sensiblemente afectados. Debe tenerse en cuenta que las zonas *super gigantes* como .com y .net sólo alojan datos de delegación.

5.6 Cambio y revocación de las claves

Ante una situación de clave comprometida, las CAs externas deben desplegar procedimientos adicionales de seguridad que pueden demorar, aumentando la latencia entre el reporte de revocación y la distribución de la información a todas las partes. En el modelo propuesto, en cambio, es más sencilla y natural esta verificación, así como la comprobación del origen de la nueva clave, dado que estas operaciones se realizan en el ámbito interno de la misma organización.

Los mecanismos de remoción de una clave y de publicación y distribución segura de la nueva clave a los usuarios, están implícitos en el DNS y en las extensiones DNSSEC, y no dependen de la intervención de una tercera parte. Además, la frecuencia de renovación de las claves y su tiempo de vida en los *caches* son controlados por la propia organización mediante la actualización de los datos de zona y la definición de TTLs, sin depender de políticas particulares de las CAs.

---

22 *RRset*: Conjunto de *resource-records* con el mismo nombre, clase y tipo

23 Fuente: Reportes de abril de 2004 de *Security Space (E-Soft Inc)*, *http://www.securityspace.com/*

5.7 Consideraciones acerca de IPv6

En IPv6, los servicios de cifrado, autenticación e integridad de los datos a nivel de red son proveídos por IPsec, empleando para ello criptografía simétrica. El intercambio de claves secretas se realiza en base al algoritmo *Diffie-Hellman*, con algunas adiciones para contrarrestar ataques tipo *man-in-the-middle* [30].

APPKEY no solo representa una alternativa para la determinación y distribución de claves secretas en IPv6, si no que además su uso implica que la provisión de servicios de seguridad puede ahora situarse por completo en la propia infraestructura de red. Una aplicación simplemente tendría que obtener del DNS la dirección IP y la clave pública del destinatario, para luego iniciar transmisiones seguras empleando IPv6, el cual puede hacer uso de la clave pública APPKEY para negociar e intercambiar las claves simétricas.

5.8 Acerca del registro APPKEY

La versión original de DNSSEC [9] establecía que, además del uso convencional del RR KEY como clave asociada a la firma de una zona, el mismo podía ser empleado por otras aplicaciones, de forma equivalente a la presente propuesta APPKEY. Sin embargo, en [25] se señalan acertadamente las inconveniencias de usar el mismo RR para estos disímiles propósitos, y se prohíbe usar el RR KEY para publicar en el DNS claves de otras aplicaciones. En los actuales *drafts* de la nueva versión del protocolo, se reemplaza el RR KEY por DNSKEY, quedando restringido su uso exclusivamente para la firma de zonas [2-4].

Actualmente, APPKEY es todavía un concepto. Queda por describirse sus características particulares, por ejemplo el formato y las especificaciones de interacción con el protocolo DNSSEC. En este documento han sido omitidos estos y otros detalles a fin de enfatizar los principios de arquitectura de la propuesta.

6 Conclusiones

Las actuales implementaciones de PKI basadas en X.509/PKIX y en el tradicional enfoque de CAs, presentan aún estimables desafíos. Algunos de estos desafíos se deben a que X.509 ha sido extendido para cubrir muchos requerimientos que no formaban parte de su diseño y de su campo de acción original. En este proceso, las especificaciones han crecido en tamaño y complejidad, motivando a su vez nuevas exigencias. Esto ocurre sobre todo cuando se pretende reflejar, en el entorno inmaterial e impersonal de las comunicaciones electrónicas, las relaciones de confianza inherentemente complejas del mundo real.

Sin embargo, esta complejidad no necesariamente debe ser reducida a una serie de formatos y estándares igualmente sofisticados. La búsqueda de la PKI perfecta no debería imponer límites al desarrollo de una buena PKI. Existen muchos usuarios y aplicaciones que precisan de un menor grado de complejidad a cambio de mayor versatilidad. Un importante criterio de diseño de arquitectura aplicado a las redes define que cuando la complejidad no puede ser eliminada, esta debe ser confinada a aquellas partes de la red que sean capaces de soportarla [16].

En este sentido, el presente trabajo propone un modelo alternativo de PKI que, valiéndose del uso de los servicios de infraestructura de Internet, reduce la complejidad inherente al esquema de relaciones de confianza en terceras partes. Se define para ello un nuevo registro DNS, denominado APPKEY, que permite a las organizaciones publicar las claves asociadas a sus entidades. Las aplicaciones obtienen a través de DNSSEC las claves públicas legítimas de cada *host,* con la misma facilidad con la que obtienen sus direcciones IP, para luego establecer los diferentes servicios de seguridad. Se elimina la necesidad de certificados de CAs externas debido a que las claves públicas son autenticadas por las propias organizaciones autoritativas sobre el espacio de nombres de la infraestructura de red en la que se despliegan estos servicios de seguridad.

El modelo propuesto es relativamente sencillo y cubre los requerimientos fundamentales de un importante número de aplicaciones de seguridad. Brinda a las partes la posibilidad de localizar y recuperar las claves públicas de las entidades, confiere autenticación a las mismas y otorga a las organizaciones el control de los criterios de seguridad a ser aplicados, sin depender de políticas internas de las CAs.

Una PKI que pretenda un emplazamiento extendido en toda la red Internet debe cumplir además con otros requerimientos (servicio de directorio simple y eficaz, escalabilidad, naturaleza distribuida, alta disponibilidad y balance de carga). El DNS tiene todas estas características y el modelo propuesto las aprovecha al máximo, sin agregar una sobrecarga significativa al sistema bajo las actuales condiciones.

Se han señalado además algunas ventajas respecto a aplicaciones como SSL, desde el punto de vista de la latencia en la comunicación, medida en *round-trips*, y del tamaño de los mensajes. El aprovechamiento de *caching* del DNS, es además un importante factor que posibilita un mejor aprovechamiento del ancho de banda, así como un mejor desempeño computacional a nivel colectivo.

Referencias

1. Albitz, P. y Liu, C. *DNS and Bind - 4th Edition.* O'Really, (2001).
2. Arends, R., Austein, R., Larson, M., Massey, D. y Rose, S. *DNS Security Introduction and Requirements.* IETF Draft, draft-ietf-dnsext-dnssec-intro-09, (Febrero 2004).
3. Arends, R., Larson, M., Austein, R., Massey, D. y Rose, S. *Protocol Modifications for the DNS Security Extensions.* IETF Draft, draft-ietf-dnsext-dnssec-protocol-05.txt, (Febrero 2004).
4. Arends, R., Austein, R., Larson, M., Massey, D. y Rose, S. *Resource Records for the DNS Security Extensions.* IETF Draft, draft-ietf-dnsext-dnssec-records-07.txt, (Febrero 2004).
5. Ateniese, G. y Mangard, S. *A New Approach to DNS Security (DNSSEC).* Proceedings of the 8th ACM Conference on Computer and Communications Security, (Noviembre 2001).
6. Atkins, D. y Austein, R. *Threat Analysis Of The Domain Name System.* IETF Draft, draft-ietf-dnsext-dns-threats-07.txt, (Abril 2004).
7. Boeyen, S. *Internet X.509 Public Key Infrastructure Repository Locator Service.* IETF Draft, draft-ietf-pkix-pkixrep-02.txt, (Septiembre 2003).
8. Chadwick, D. *Deficiencies in LDAP when used to support PKI.* Communications of the ACM, Vol. 46, No. 3, (Marzo 2003).
9. Eastlake, D. *Domain Name System Security Extensions.* IETF RFC-2535, (1999).
10. Ellison, C. y Schneier, B. *Ten Risks of PKI.* Computer Security Journal, Vol. XVI, No. 1, (2000).
11. Ellison, C. *Naming and certificates.* Proceedings of the tenth conference on Computers, freedom and privacy, ACM Press, (2000).
12. Freier, A., Karlton, P. y Kocher, P. *The SSL Protocol - Version 3.0.* Netscape Communications, (1996)
13. Gerck, E. *Overview of Certification Systems.* The Bell, Vol. 1, No. 3, (Julio 2000).
14. Gudmundsson, O. *Delegation Signer Resource Record.* IETF Draft, draft-ietf-dnsext-delegation-signer-15.txt, (Junio 2003).
15. Gudmundsson, O. *DNSSEC and IPv6 A6 aware server/resolver message size requirements.* IETF RFC-3226, (Diciembre 2001).
16. Hallam-Baker, P. *Trust Assertion XML Infrastructure.* Proceedings of the 1st Annual PKI Research Workshop, (Abril 2002).
17. Housley, R., Polk, W., Ford, W. y Solo, D. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile.* IETF RFC-3280, (Abril 2002).
18. ITU-T. *Recommendation X.500: Information Technology - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Service.* ITU-T, (1993).
19. ITU-T. *Recommendation X.501: Information Technology Open - Systems Interconnection - The Directory: Models.* ITU-T, (1993).
20. ITU-T. *Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Autentication Framework.* ITU-T, (1988).
21. ITU-T. *Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Autentication Framework.* ITU-T, (Junio 1997).
22. Josefsson, S. *Network Application Security Using The Domain Name System.* Royal Institute of Technology - Stockholm, Sweden, (2001)
23. Jung, J., Sit, E., Balakrishnan, H. y Morris, R. *DNS Performance and the Effectiveness of Caching.* IEEE/ACM Transactions on Networking, (Octubre 2002).
24. Kent, S. *How Many Certification Authorities Are Enough?.* Proceedings of MILCOM (unclassified papers), (Noviembre 1997).
25. Massey, D. y Rose, S. *Limiting the Scope of the KEY Resource Record (RR).* IETF RFC-3445,(2002).
26. Myers, M., Ankney, R., Malpani, A., Galperin, S. y Adams C. *Online Certificate Status Protocol - OCSP.* IETF RFC-2560, (Junio 1999).
27. Mockapetris, P. *Domain Names - Concepts and facilities.* IETF RFC-1034, (1987).
28. Mockapetris, P. *Domain Names - Implementation and Specification.* IETF RFC-1035, (1987).
29. Renfro, S. *VeriSign CZAG: Privacy Leak in X.509 Certificates.* Proceedings of the 11th USENIX Security Symposium, (Agosto 2002).
30. Stallings, W. *Cryptography and Network Security.* Prentice Hall, (1998).
31. Tanenbaum, A. *Computer Networks.* Prentice Hall, (2002).
32. Wahl, M., Howes, T. y Kille, S. *Lightweight Directory Access Protocol (v3).* IETF RFC-2251, (1997).