

# “Un estudio comparativo en Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNS)”

Tesista: Ernesto Sánchez  
Director (UNSa): Msc. Daniel Arias Figueroa  
Co-Director (UNLP): Lic. Javier Díaz

"Tesis presentada para obtener el grado de Magister en Redes de Datos"

Facultad de Informática  
Universidad Nacional de La Plata  
Abril2017

## **Reconocimientos**

*A mis directores de tesis,*

*Magister Daniel Arias Figueroa y Lic. Javier Díaz, por el apoyo brindado y guía para la culminación del presente trabajo de tesis.*

*A mis padres y abuela,*

*Por los ejemplos de perseverancia y esfuerzo que me han infundado siempre*

*A mi esposa e hijos*

*Por la paciencia, motivación y amor que me brindan día a día*

# Índice de Contenidos:

---

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1. Objetivos.....	1
<b>2. EL SISTEMA DE NOMBRES DE DOMINIO (DNS) .....</b>	<b>2</b>
2.1. Introducción.....	2
2.2. El Espacio de Nombres DNS.....	2
2.3. Sintaxis de nombre de dominio.....	3
2.4. Concepto de Zona de autoridad .....	4
2.5. Delegación de autoridad .....	5
2.6. Servidores de Nombres Raíz.....	6
2.7. Componentes de un Sistema de Nombres de Dominio .....	7
2.8. Tipos de Servidores DNS.....	8
2.8.1. Servidores DNS Maestro y Esclavo.....	8
2.8.2. Servidores de almacenamiento temporal.....	10
2.9. Archivos de Zona y Registros de Recursos .....	10
2.9.1. Contenido de un Archivo de Zona.....	11
2.10. El Protocolo DNS .....	12
2.10.1. Consultas DNS .....	12
Consultas recursivas .....	13
Consultas iterativas .....	13
2.10.2. Mapeo de Direcciones a Nombres (DNS Reverse Mapping).....	14
2.11. Mantenimiento y Transferencias de Zonas.....	16
2.11.1. Transferencia de Zona Completa (AXFR) .....	16
2.11.2. Transferencia de Zona Incremental (IXFR).....	17
2.11.3. Notificación DNS (NOTIFY) .....	17
2.11.4. Actualización Dinámica (Dynamic Update).....	18
2.12. Mensajes DNS .....	19
2.12.1. Formato del Mensaje .....	19
Encabezado DNS.....	20
Sección "QUESTION" .....	22
Secciones "ANSWER", "AUTHORITY" y "ADDITIONAL INFORMATION" .....	23
2.13. Mecanismo de Extensión para DNS (EDNS0) .....	24
2.13.1. Formato del Registro OPT .....	24
2.13.2. TCP o UDP.....	25
<b>3. VULNERABILIDADES DEL SISTEMA DE NOMBRES DE DOMINIO .....</b>	<b>27</b>

<b>3.1.</b>	<b>Clasificación de amenazas al Sistema DNS</b> .....	<b>27</b>
<b>3.2.</b>	<b>Análisis de amenazas al Sistema DNS</b> .....	<b>28</b>
3.2.1.	Corrupción de Archivos .....	28
3.2.2.	Actualizaciones no autorizadas.....	28
3.2.3.	Suplantación de identidad en transferencia de zonas.....	29
3.2.4.	Envenenamiento de Cache.....	29
3.2.5.	Otros tipos de ataques .....	31
<b>4.</b>	<b>CRIPTOGRAFÍA EN EL CONTEXTO DEL SISTEMA DNS</b> .....	<b>33</b>
<b>4.1.</b>	<b>Introducción</b> .....	<b>33</b>
<b>4.2.</b>	<b>Criptografía Simétrica</b> .....	<b>33</b>
<b>4.3.</b>	<b>Criptografía Asimétrica</b> .....	<b>34</b>
<b>4.4.</b>	<b>Digesto de mensajes</b> .....	<b>34</b>
<b>4.5.</b>	<b>Código de Autenticación de Mensaje</b> .....	<b>35</b>
<b>4.6.</b>	<b>Firma Digital</b> .....	<b>35</b>
<b>4.7.</b>	<b>Uso de la criptografía en DNS</b> .....	<b>35</b>
4.7.1.	Autenticación e Integridad en Transferencias de Zonas.....	36
<b>5.</b>	<b>DNSSEC</b> .....	<b>37</b>
<b>5.1.</b>	<b>Introducción</b> .....	<b>37</b>
5.1.1.	Autenticación del origen de los datos.....	37
5.1.2.	Integridad de los datos.....	37
5.1.3.	Autenticación de la Negación de Existencia .....	37
<b>5.2.</b>	<b>Conceptos generales sobre DNSSEC</b> .....	<b>38</b>
5.2.1.	Puntos de Entrada Seguros (SEPs) .....	39
5.2.2.	Compatibilidad con el esquema tradicional de DNS.....	39
5.2.3.	Cadenas de confianza.....	40
5.2.4.	Concepto de Clave de Zona (ZSK) y Clave de Claves (KSK).....	41
Motivaciones para un uso separado de Claves.....		41
Consideraciones sobre revocación de Claves.....		42
5.2.5.	Delegación segura .....	43
<b>5.3.</b>	<b>Como trabaja DNSSEC</b> .....	<b>44</b>
5.3.1.	<i>Proceso de generación y verificación de firmas DNSSEC</i> .....	48
<b>5.4.</b>	<b>Contexto a nivel mundial</b> .....	<b>49</b>
<b>5.5.</b>	<b>Evolución del despliegue a nivel mundial</b> .....	<b>54</b>
<b>5.6.</b>	<b>Topología DNSSEC</b> .....	<b>56</b>
5.6.1.	Introducción .....	56
5.6.2.	Servidor de Nombres Autoritativo .....	57
5.6.3.	Servidores de Nombres Cache .....	57
5.6.4.	Stub Resolver .....	58
<b>5.7.</b>	<b>Especificaciones</b> .....	<b>59</b>



5.7.1. Registros de Recurso de clave pública(DNSKEY) .....	59
Formato de la porción RDATA .....	59
Formato de presentación del Registro DNSKEY .....	61
Ejemplo de un Registro DNSKEY .....	61
5.7.2. Registro de firma digital (RRSIG) .....	62
Formato de la porción RDATA .....	62
Ejemplo de un Registro RRSIG .....	64
5.7.3. Negación de la existencia autenticada(NSEC) .....	65
Formato de la porción RDATA .....	66
5.7.4. Registro NSEC3 .....	68
5.7.5. Registro DS (Delegación de Firma) .....	69
Formato de la porción RDATA .....	70
Ejemplo de un Registro DS .....	71
<b>5.8. Un método de validación alternativo (DLV) .....</b>	<b>72</b>
<b>6. ANÁLISIS DE FUNCIONAMIENTO DE DNSSEC.....</b>	<b>75</b>
<b>6.1. Descripción del ambiente de pruebas utilizado .....</b>	<b>75</b>
6.1.1. Descripción del proceso de configuración DNSSEC .....	76
6.1.2. Comprobaciones sobre configuración DNSSEC.....	77
<b>6.2. Observaciones sobre el proceso de configuración DNSSEC .....</b>	<b>79</b>
<b>6.3. Análisis comparativo resolución de nombres DNS tradicional vs DNSSEC.....</b>	<b>80</b>
<b>6.4. Análisis de falla de proceso de validación.....</b>	<b>84</b>
<b>7. LA ALTERNATIVA DNSCURVE.....</b>	<b>85</b>
<b>7.1. Introducción.....</b>	<b>85</b>
<b>7.2. Conceptos sobre DNSCurve .....</b>	<b>85</b>
7.2.1. Confidencialidad.....	85
7.2.2. Integridad .....	85
7.2.3. Disponibilidad.....	85
7.2.4. Primitivas Criptográficas .....	85
7.2.5. DNSCurve frente a DNSSEC .....	86
<b>8. CONCLUSIONES.....</b>	<b>87</b>
<b>9. APÉNDICE: CONTRIBUCIONES .....</b>	<b>89</b>
<b>9.1. Publicaciones con referato .....</b>	<b>89</b>
<b>9.2. Actividades de transferencia .....</b>	<b>90</b>
<b>9.3. Proyectos relacionados a la investigación .....</b>	<b>90</b>
<b>10. REFERENCIAS .....</b>	<b>91</b>
<b>11. ABREVIATURAS .....</b>	<b>95</b>

# Índice de Figuras:

---

Figura 1: El espacio de Nombres de Dominio.....	3
Figura 2: Concepto de Zona y Dominio.....	5
Figura 3: Relación entre la jerarquía de nombres y la delegación de la autoridad para los nombres.....	6
Figura 4: Proceso de actualización servidores raíz.....	7
Figura 5: Configuración Servidor Maestro, Servidor Esclavo.....	9
Figura 6: Configuración Servidor Maestro oculto.....	9
Figura 7: Fragmento de un archivo de zona.....	11
Figura 8: Proceso típico de consulta DNS desde un PC Cliente.....	12
Figura 9: Consultas recursivas e iterativas.....	14
Figura 10: reverse mapping usando consultas recursivas.....	15
Figura 11: Encabezado mensaje DNS.....	20
Figura 12: Estructura datagrama IP para mensaje DNS.....	25
Figura 13: Vectores de ataque en el flujo de datos DNS.....	27
Figura 14: Ejemplo ataque envenenamiento de cache (Kaminsky).....	30
Figura 15: Ejemplo ataque DoS a DNS.....	32
Figura 16: Creación cadena de confianza DNSSEC.....	40
Figura 17: Proceso de delegación segura en DNSSEC.....	44
Figura 18: Intercambio de mensajes solicitudes/respuestas en DNSSEC.....	46
Figura 19: Proceso de consulta y validación DNSSEC.....	48
Figura 20: Proceso de generación de firma DNSSEC.....	48
Figura 21: Proceso de verificación de firma DNSSEC.....	49
Figura 22: Mapa mundial despliegue DNSSEC ccTLD.....	50
Figura 23: Mapa África despliegue DNSSEC ccTLD.....	51
Figura 24: Mapa América del Norte despliegue DNSSEC ccTLD.....	51
Figura 25: Mapa Latinoamérica y Caribe despliegue DNSSEC ccTLD.....	52
Figura 26: Mapa Asia y Oceanía despliegue DNSSEC ccTLD.....	52
Figura 27: Mapa Europa despliegue DNSSEC ccTLD.....	53
Figura 28: Reporte porcentajes de adopción.....	53
Figura 29: Configuraciones erróneas enfoque descendente.....	55
Figura 30: Configuraciones erróneas enfoque ascendente.....	56
Figura 31: Formato porción RDATA en Registro DNSKEY.....	60
Figura 32: Ejemplo Registro DNSKEY.....	62
Figura 33: Formato porción RDATA en Registro RRSIG.....	63
Figura 34: Ejemplo Registro RRSIG.....	64
Figura 35: Ejemplo Zona DNS (Funcionamiento NSEC).....	65
Figura 36: Ejemplo Zona DNS firmada (Funcionamiento NSEC).....	66
Figura 37: Formato porción RDATA en Registro NSEC.....	66
Figura 38: Formato porción RDATA en Registro NSEC3.....	68
Figura 39: Formato porción RDATA en Registro DS.....	70
Figura 40: Ejemplo Registro DS.....	72
Figura 41: Proceso de validación con DLV.....	73
Figura 42: Ambiente de pruebas utilizado.....	75
Figura 43: Bytes enviados y recibidos. DNS vs DNSSEC.....	81
Figura 44: Tiempos de respuestas. DNS vs DNSSEC.....	81
Figura 45: Captura de tráfico. Resolución DNSSEC.....	83

# Índice de Tablas:

---

Tabla 1: Registros Regionales de Internet (RIRs).....	15
Tabla 2: Formato de mensaje DNS.....	19
Tabla 3: Descripción campos encabezado DNS.....	22
Tabla 4: Formato campos sección QUESTION.....	23
Tabla 5: Formato campos secciones ANSWER, AUTHORITY y ADDITIONAL INFORMATION.....	23
Tabla 6: Formato campo NAME.....	24
Tabla 7: Descripción campos Registro OPT.....	25
Tabla 8: Clasificación amenazas al Sistema DNS.....	28
Tabla 9: Listado algoritmos para DNSSEC.....	61
Tabla 10: Listado algoritmos para digesto Registro DS.....	71
Tabla 11: Configuración de red ambiente de pruebas.....	76
Tabla 12: Resultados comparativos DNS vs DNSSEC.....	80

# 1. Introducción

En esencia, el Sistema de Nombres de Dominio, proporciona un esquema jerárquico de nombres basado en dominios y una base de datos distribuida para implementar este esquema. Su función principal es la de relacionar direcciones de host y servicios de red con sus direcciones IP correspondientes y viceversa, (*Tanenbaum, 2003*).

Desde su creación, el sistema DNS ha carecido de un diseño que asegure la información intercambiada entre Clientes y Servidores y por considerarse parte fundamental para el funcionamiento de Internet, es que particulares y organizaciones han puesto especial énfasis en garantizar la seguridad de los datos intercambiados.

Actualmente la solución que logró imponerse por sobre otras alternativas y que alcanzó adopción a nivel mundial es, “Extensiones de Seguridad para DNS (DNSSEC)”, que tiene por objetivo proveer autenticación del origen e integridad de los datos intercambiados a través del protocolo DNS, mediante una jerarquía de “firmas criptográficas”.

El presente trabajo tiene por objetivo principal, presentar los aspectos teóricos y un caso de implementación de DNSSEC, que permitan la comparación frente a una arquitectura DNS tradicional, es decir no segura, con la finalidad de proporcionar las herramientas necesarias que permitan acompañar el despliegue global de tales extensiones de seguridad en los entornos operacionales de DNS actuales.

Se desarrollarán también los aspectos teóricos de la alternativa “DNSCurve” y se expondrán las justificaciones correspondiente del porque ambas alternativas no son comparables.

## 1.1. Objetivos

Para alcanzar los objetivos descriptos anteriormente, se definen los siguientes objetivos específicos:

- Presentar una revisión del estudio del arte del Sistema de Nombres de Dominio.
- Presentar las fallas de seguridad del protocolo DNS y métodos de “ataques” comunes al Sistema DNS.
- Presentar una revisión del estudio del arte de las Extensiones de Seguridad para DNS.
- Implementar un ambiente de pruebas para una arquitectura DNS “tradicional”.
- Implementar un ambiente de pruebas para una arquitectura DNS “segura”, basada en DNSSEC.
- Generar casos de pruebas a partir de los ambientes antes citados.
- Analizar y comparar los datos obtenidos, generar informes a partir de los mismos.
- Presentar los aspectos teóricos de la alternativa DNSCurve y justificaciones de porque no puede ser comparada frente a DNSSEC.

# 2. El Sistema de Nombres de Dominio (DNS)

## 2.1. Introducción

El Sistema de Nombres de Dominio básicamente proporciona un mecanismo que permite la administración y el balanceo de carga para la traducción de los nombres de dispositivos (computadoras personales, servidores, routers, etc.) conectados a una red, principalmente a Internet, hacia sus respectivas direcciones IP y viceversa.

La asignación de nombres DNS se utiliza en redes TCP/IP, para localizar equipos y servicios con nombres descriptivos (fáciles de recordar y utilizar). Cuando un usuario escribe un nombre DNS en una aplicación, los servicios DNS podrán traducir el nombre a otra información asociada con el mismo, por ejemplo su dirección IP.

El esquema es jerárquico y basado en dominios utilizando una Base de Datos Distribuida para implementarlo. Se utiliza un mecanismo Cliente/Servidor, donde unos programas llamados servidores de nombres contienen información acerca de un segmento de la base de datos y la ponen a disposición de los clientes. Los clientes a través de rutinas de resolución llamadas *stub resolvers*, generan peticiones hacia un servidor de nombres de dominio. (Por ejemplo: ¿Qué dirección IP corresponde a [www.unsa.edu.ar](http://www.unsa.edu.ar)?)

El Sistema de Nombres de Dominio cubre dos aspectos:

- El primero, especifica la estructura, sintaxis de los nombres y las reglas para delegar autoridad sobre los nombres.
- El segundo especifica la implementación en servidores de la base de datos distribuida que transforma de una manera eficiente los nombres a direcciones IP.

## 2.2. El Espacio de Nombres DNS

La Base de Datos Distribuida de DNS está indexada por nombres de dominio. Cada nombre de dominio es esencialmente una trayectoria en un árbol invertido denominado “espacio de nombres de dominio”. La estructura jerárquica del árbol es similar a la estructura de un sistema de archivos en un sistema operativo.

Cada nodo del árbol tiene una etiqueta de texto, que identifica el nodo en relación con su padre. Esto es más o menos análogo a una ruta “relativa” en un sistema de archivos. Una etiqueta nula o “”, está reservado para el nodo raíz. En el texto, el nodo raíz se escribe como un solo punto (.). En el sistema de archivos Unix, la raíz se escribe como una barra (/).

Los niveles más altos del árbol, por debajo del nodo raíz, son denominados Dominios de Nivel Superior (TLDs), que incluyen: Dominios de Nivel Superior Genéricos (gTLDs), Dominios de Nivel Superior Geográficos (ccTLDs) y Dominios de Nivel

Superior Geográficos Internacionalizados (IDN ccTLDs), más una infraestructura especial llamada ARPA [17].

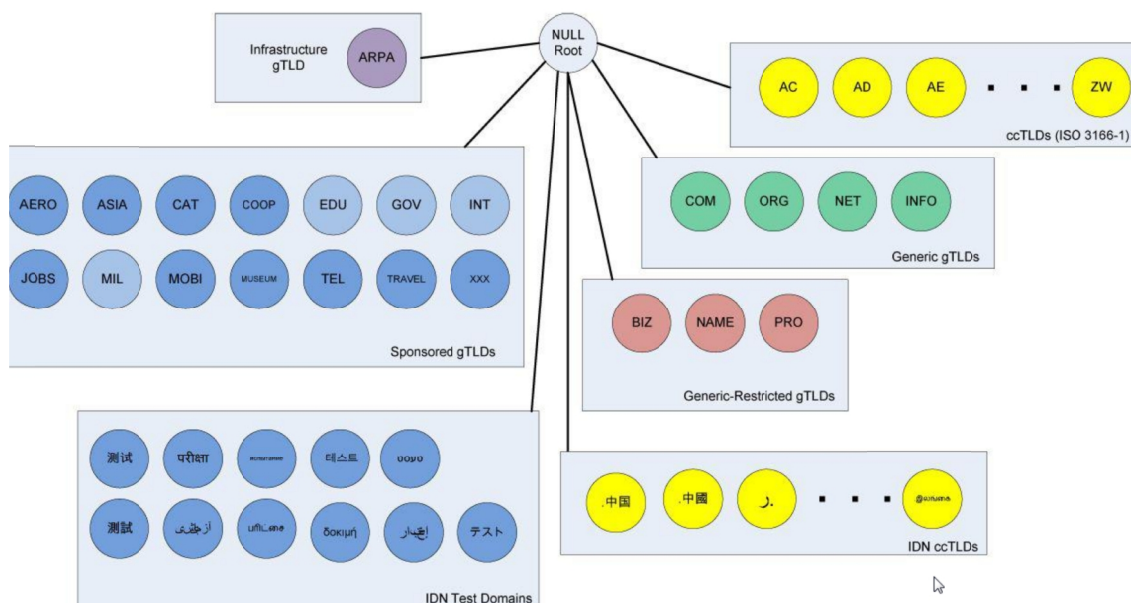


Figura 1: El espacio de Nombres de Dominio.

Como se observa en la figura anterior [1], son cinco los grupos de uso común, correspondientes a los Dominios de Nivel Superior, más un grupo de dominios especializados que se utiliza para los nombres de dominio internacionalizados (IDNs). La creación de este último grupo se debe a la necesidad de incorporar nombres de dominio que puedan contener caracteres con acento diacrítico o caracteres de escrituras no latinas como la árabe y las chinas. El estándar IDN fue propuesto originalmente en 1998 y después de mucho debate y propuestas competidoras, un sistema llamado Internacionalización de Nombres de Dominio en Aplicaciones (*Internationalizing Domain Names in Applications– IDNA*) fue adoptado como el estándar elegido, y en el año 2005 empezó su presentación pública.

En *IDNA*, el término nombre de dominio IDN específicamente denota a cualquier nombre de dominio que consiste solamente en etiquetas en las que el algoritmo IDNA ToASCII puede ser exitosamente aplicado. ToASCII se basa en la codificación *ASCII Punycode* de cadenas *Unicode* normalizadas.[2].

Los Dominios de Nivel Superior Genéricos (gTLDs) se agrupan en categorías: genérico, genérico-restringido y patrocinado. La primera categoría agrupa a los denominados de libre uso, las últimas dos categorías agrupan a los dominios restringidos para un uso en particular. Por ejemplo: el dominio EDU es usado para instituciones educativas, MIL y GOV, son usados por instituciones militares y gubernamentales de los Estados Unidos e INT es usado por organizaciones internacionales. En [3], se encuentra disponible un listado de los Dominios de Nivel Superior coordinados por la IANA.

### 2.3. Sintaxis de nombre de dominio

Un nombre de dominio usualmente consiste en dos o más partes (técnicamente denominadas etiquetas) las cuales se encuentran separadas por puntos. Por ejemplo, *www.unsa.edu.ar*.

- Cada nombre de dominio inicia por la derecha con un punto (“.”), explícito regularmente y que hace referencia al nodo raíz (*root*).
- A la etiqueta ubicada más a la derecha se la llama dominio de nivel superior (Top Level Domain o TLD), por ejemplo la etiqueta “.ar” es el TLD de [www.unsa.edu.ar](http://www.unsa.edu.ar).
- Cada etiqueta a la izquierda especifica una subdivisión o subdominio. En teoría, esta subdivisión puede tener hasta 127 niveles, en donde cada etiqueta puede contener hasta 63 caracteres, pero restringido a que la longitud total del nombre del dominio no exceda los 255 caracteres.
- Finalmente, la parte más a la izquierda del dominio representa el nodo hoja, mejor conocido como el nombre del equipo al que se desea establecer una conexión (*hostname*).

## **2.4. Concepto de Zona de autoridad**

Toda la información acerca de un espacio de nombres de dominio se guarda en una computadora que llamamos Servidor de Nombres. Esto es la implementación física del llamado software DNS. Generalmente, los servidores tienen información completa acerca de una parte del espacio de nombres de dominio, que llamamos zona de autoridad. Más exactamente una zona es la porción del espacio de nombres de dominio de la que es responsable un determinado servidor DNS. La zona de autoridad de estos servidores abarca al menos un dominio y también pueden incluir subdominios, aunque a veces los servidores de un dominio pueden delegar sus dominios en otros servidores.

La diferencia entre una zona y un dominio es que la primera contiene los nombres de dominio y datos que representan a un dominio y un dominio es un nombre que agrupa a otras máquinas o dominios inferiores.

La siguiente figura, ilustra la diferencia entre dominios y zonas, [21]. En la misma, se visualiza el dominio microsoft.com, que contiene nombres de dominio para Microsoft. Cuando el dominio microsoft.com se crea por primera vez en un sólo servidor, se configura como una zona única para todos los espacios de nombres DNS de Microsoft. Sin embargo, si el dominio microsoft.com tiene que utilizar subdominios, estos subdominios deben incluirse en la zona o delegarse a otra zona.

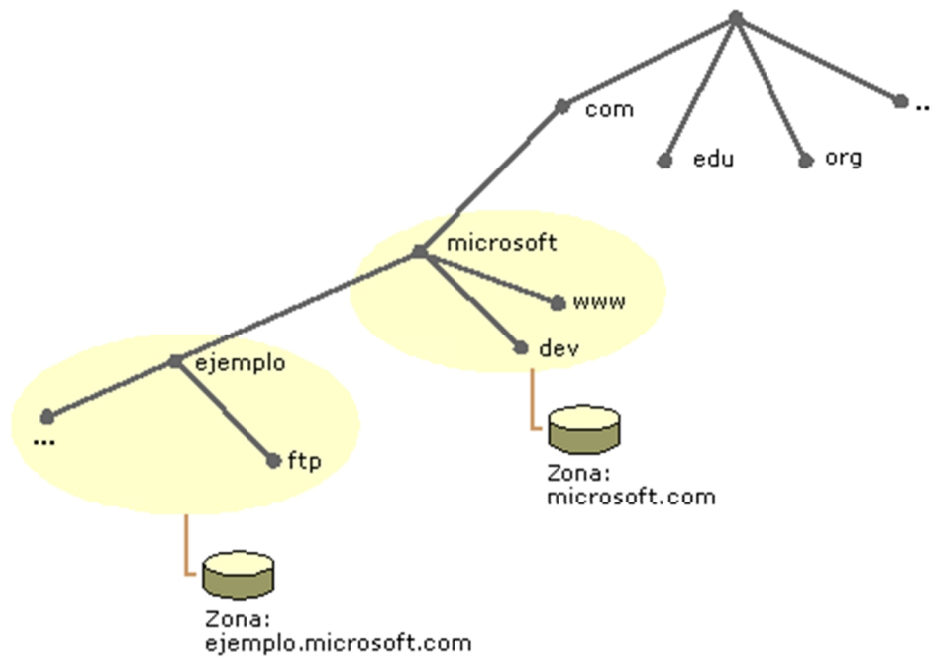


Figura 2: Concepto de Zona y Dominio.

El dominio ejemplo.microsoft.com muestra un subdominio nuevo, el dominio ejemplo.microsoft.com, delegado fuera de la zona microsoft.com y administrado en su propia zona. Sin embargo, la zona microsoft.com debe contener algunos registros de recursos para proporcionar información sobre la delegación que haga referencia a los servidores DNS que están autorizados para el subdominio delegado.

Si la zona microsoft.com no utiliza la delegación en un subdominio, los datos del subdominio continúan formando parte de la zona microsoft.com. Por ejemplo, el subdominio dev.microsoft.com no está delegado fuera de la zona, sino al contrario, está administrado por la zona microsoft.com.

La información de una zona se supone es almacenada en al menos dos lugares diferentes, lo que implica que debe haber por lo menos dos servidores que contienen información para cada zona. Esto se realiza por razones de redundancia, por lo que si un servidor falla, al menos alguno de los otros estará disponible, razón por la cual, todos los servidores deben contener la misma información sobre la zona. La configuración típica incluye un servidor maestro que alberga información de la zona, almacenada en disco; y uno o más servidores esclavos que obtienen una copia completa de la información de zona desde el servidor maestro, en un proceso llamado transferencia de zona, para lo cual se utiliza un protocolo especial, el cual se describe posteriormente.

## 2.5. Delegación de autoridad

Una organización que posee un nombre de dominio, como por ej: *cidia.unsa.edu.ar*, es responsable del funcionamiento y mantenimiento de los servidores de nombres que traducen sus propios nombres a direcciones. Ésta responsabilidad recae sobre un administrador local quien tiene por tareas la operación y administración de los nodos servidores, incluso puede delegar parte de los dominios que caen bajo su responsabilidad en otro administrador.



Considérese el siguiente ejemplo que muestra la relación entre la jerarquía de nombres y la delegación de la autoridad para los nombres.

En Argentina, la ARIU, es la entidad en la que NIC Argentina delegó la responsabilidad de la operación estable y confiable de la base de datos autorizada llamada Sistema de Nombres de Dominios “*edu.ar*”. Del mismo modo, la ARIU, delega la responsabilidad del subdominio “*unsa.edu.ar*” a la Universidad Nacional de Salta (esto es similar a un montaje remoto de sistema de archivos en Unix).

La delegación de autoridad para “*unsa.edu.ar*” a la Universidad de Salta crea una nueva zona, una pieza autónoma administrada del espacio de nombres. La zona de *unsa.edu.ar* ahora es independiente del dominio “*edu.ar*” y contiene todos los nombres de dominio que terminan en *unsa.edu.ar*. La zona del dominio *edu.ar*, por el contrario, sólo contiene los nombres de dominio que terminan en *.edu.ar*. Ahora bien, “*unsa.edu.ar*” puede dividirse en subdominios, como “*cidia.unsa.edu.ar*”, “*naturales.unsa.edu.ar*” y algunos de estos subdominios pueden ser también zonas separadas, si los administradores de “*unsa.edu.ar*” delegan su responsabilidad a otros administradores.

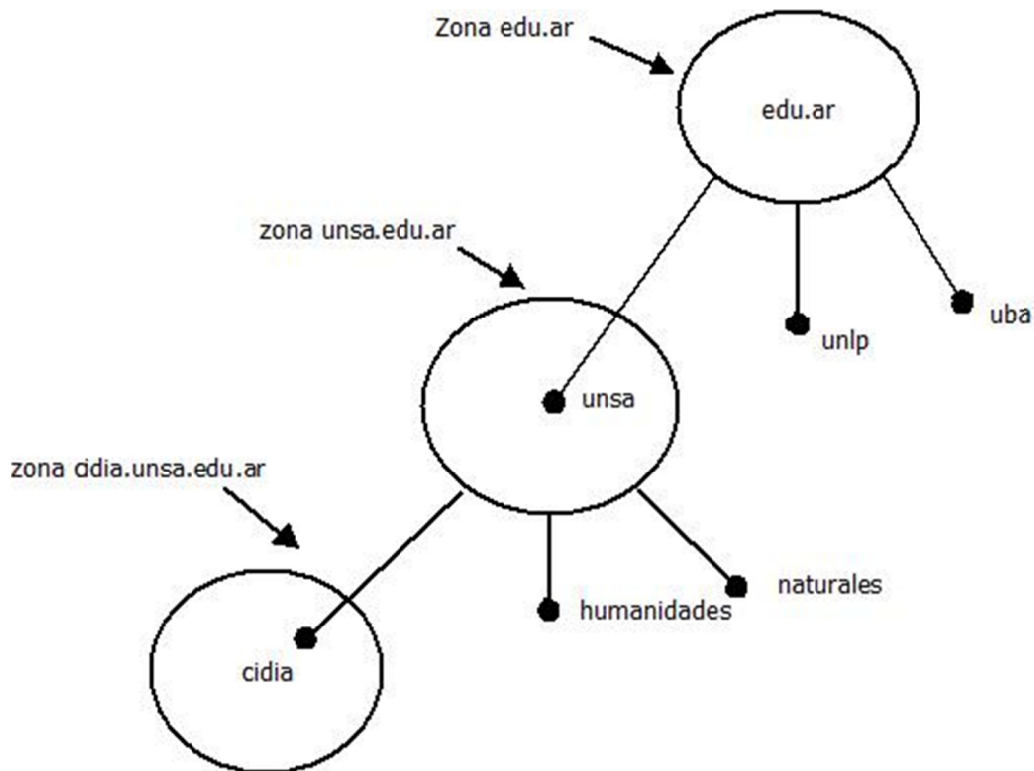


Figura 3: Relación entre la jerarquía de nombres y la delegación de la autoridad para los nombres.

## 2.6. Servidores de Nombres Raíz

Los Servidores Raíz son responsabilidad de la *ICANN*, pero son operados bajo un acuerdo conocido como el Acuerdo de Cooperación en Investigación y Desarrollo (*CRADA*), firmado entre la *ICANN* y el *U.S Department of Commerce* [4]. La *ICANN*, también ha creado el *Root Server System Advisory Committee (RSSAC)*, para proporcionar asesoramiento y orientación en cuanto al funcionamiento y desarrollo de este recurso crítico. Por otro lado, la *IETF*, fue convocada por el *RSSAC* para

desarrollarlos estándares para el funcionamiento de los servidores raíz, lo que dio lugar a la publicación del RFC2870.

Actualmente son trece los servidores de Nombres Raíz en el mundo y son parte fundamental de Internet. Ocupan un nombre de dominio reservado (*root-servers.net*). Cada servidor es en realidad un conjunto de servidores físicos que comparten una dirección IP en común. Son nombrados desde *a.root-servers.net* hasta *m.root-servers.net*. Se puede consultar el listado detallado en [18].

El trabajo de los servidores raíz es proporcionar una referencia a los servidores de nombres autoritativos para los dominios de primer nivel requerido (*gTLD* o *ccTLD*), es decir, cuando cualquier servidor de nombres es consultado para obtener información sobre un nombre de dominio y éste no tuviera tal información, procede en primer lugar a consultar a alguno de los servidores raíz, obteniendo por respuesta un listado de los servidores de nombres autoritativos del dominio de primer nivel que fuera consultado.

En el año 2004, la *ICANN*, asumió la responsabilidad para el mantenimiento del archivo principal para los dominios de primer nivel (*TLD*), este archivo contiene un listado de los servidores autoritativos para cada *TLD*.

La distribución de este archivo a cada uno de los servidores raíz se realiza mediante transacciones seguras y para aumentar aún más la seguridad, el servidor que proporciona las actualizaciones es sólo accesible desde los servidores raíz.

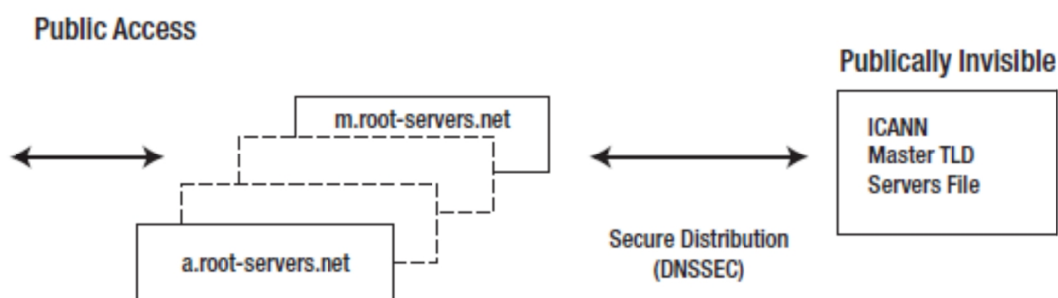


Figura 4: Proceso de actualización servidores raíz.

## 2.7. Componentes de un Sistema de Nombres de Dominio

Un DNS incluye tres componentes principales:

1. Datos que describen el dominio o los dominios, organizados en archivos de texto llamados archivos de zona.
2. Uno o más programas servidores de nombres de dominio.
3. Un programa o librería de resolución (*stub resolver*)

Un único servidor de nombres puede contener información sobre un dominio, muchos o bien ninguno. Los datos para cada dominio o zona se definen en elementos denominados Registros de Recursos, organizados en archivos de zona. El formato de los archivos de zona y sus Registros de Recursos están definidos en el RFC 1035 [19].

Un programa DNS normalmente realiza tres acciones:

1. Lee uno o más archivos de zona, que describen los dominios para los que es responsable.
2. Dependiendo de la funcionalidad del software DNS, se lee un archivo de configuración, que describe diferentes comportamientos requeridos (por ejemplo realizar almacenamiento *cache* o no).
3. Responde a las preguntas (consultas) de clientes locales o remotos (otros servidores de nombres o *resolvers*).

El programa o librería de resolución (*resolver*) es instalado en cada host y proporciona un medio de traducción para solicitudes de usuarios, utilizando en la mayoría de los casos a UDP como protocolo de transporte. Un *resolver* es un programa complejo, pero los estándares permiten una implementación mucho más simple llamada *stub resolver*. Casi todos los *resolvers* instalados en Windows y sistemas UNIX (Linux, UNIX y BSD) son *stub resolvers*. Por ejemplo, un navegador web, utiliza un *stub resolver* para traducir el nombre o URL introducido, a su correspondiente dirección IP.

## **2.8. Tipos de Servidores DNS**

Los servidores DNS pueden asumir una gran variedad de roles, por ejemplo un único servidor de nombres puede ser un servidor “maestro” para algunas zonas, servidor “esclavo” para otras y ofrecer servicios de “reenvío” o “almacenamiento temporal” (*Cache*) para una tercer zona. A continuación se describen las características para cada tipo de servidor antes mencionado.

### **2.8.1. Servidores DNS Maestro y Esclavo**

Por su naturaleza de sistema público, DNS requiere un alto grado de rendimiento y confiabilidad por lo que es necesario en la mayoría de los casos el uso de más de un servidor de nombres para administrar una o varias zonas. No es raro hoy en día ver organizaciones en Internet con cuatro, cinco o más servidores de nombres, incluso ubicados físicamente en sitios diferentes, de manera tal de minimizar la sobrecarga de administración necesaria para la sincronización de los archivos de zonas involucrados. La especificación de DNS permite que un único servidor DNS sea el propietario de una copia maestra del archivo de zona y permitir la transferencia de zona hacia los otros servidores de nombres esclavos. El término “maestro”, está relacionado con la ubicación del archivo de zona, más que con cualquier otra característica operativa. Un servidor maestro puede ser solicitado para transferir archivos de zona, utilizando las operaciones de transferencia de zona, a uno o más servidores esclavos cada vez que el archivo de zona sea modificado.

Un servidor de nombres maestro obtiene los datos de sus archivos de zonas locales y sobre las cuales tiene autoridad. Los cambios en una zona, como la adición de dominios o hosts, se realizan en éste.

Un servidor maestro puede indicar cambios en los archivos de zonas, usando mensajes de notificación, hacia los servidores esclavos. Este mecanismo asegura que tales cambios serán rápidamente propagados hacia los servidores esclavos en lugar de estar esperando a que éstos últimos consulten por cambios en cada intervalo de actualización.

Un servidor de nombres esclavo es aquel que obtiene información de su zona desde un servidor maestro, éste podrá responder con autoridad sobre las zonas en las que fue definido como esclavo. El acto de transferir la zona se puede ver como una delegación de autoridad para la zona, al servidor esclavo por un período de tiempo definido en el Registro *SOA (Start of Authority)*.

El hecho de que un servidor de nombres esclavo pueda responder con autoridad a consultas para un dominio dado, permite mantener al servidor de nombres maestro oculto para el acceso público, solo en caso de ser necesario. Para ilustrar un caso donde esta estrategia puede ser útil considere el siguiente ejemplo: Si la información en un servidor de nombres esclavo se corrompe a través de un ataque malicioso, ésta puede ser rápidamente restaurada desde un servidor maestro, a través de una transferencia de zona. En el caso de que el servidor maestro fuera el comprometido por un ataque similar, los archivos de zona podrán ser restaurados desde un respaldo (*backup*) almacenado seguramente en algún soporte magnético (disco rígido, cinta, etc.), lo que llevaría mucho más tiempo.

En la primera figura se muestra una configuración típica de Servidor Maestro y Esclavo, en la segunda, se muestra el ejemplo de configuración Servidor Maestro oculto.

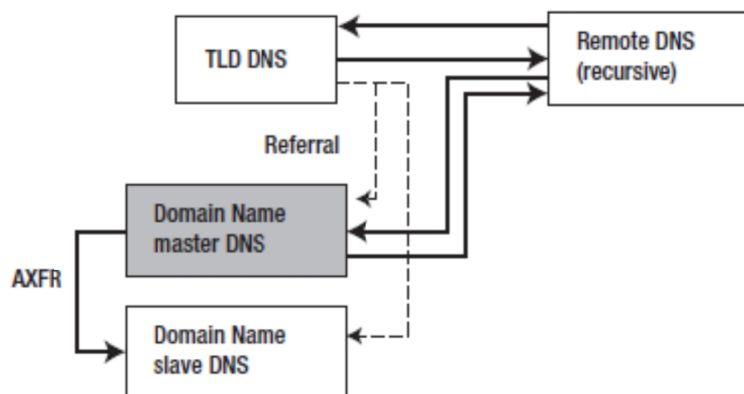


Figura 5: Configuración Servidor Maestro, Servidor Esclavo.

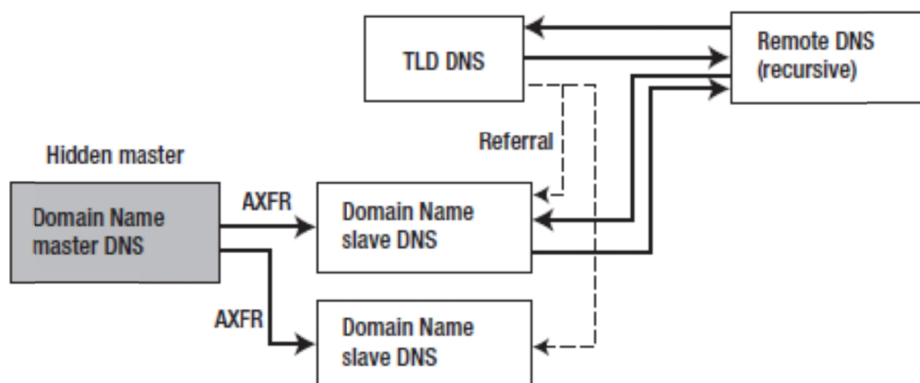


Figura 6: Configuración Servidor Maestro oculto.

## 2.8.2. Servidores de almacenamiento temporal

Un Servidor de almacenamiento temporal, también denominado *DNS Caché*, o comúnmente *Resolver*, obtiene información específica en forma de uno o más Registros de Recursos acerca de un dominio mediante la consulta a un Servidor de Nombres (Maestro o Esclavo) con autoridad sobre una zona, con el fin de responder a una consulta que recibió de un host cliente, procediendo posteriormente a almacenar estos datos de manera local. En una solicitud subsiguiente de los mismos datos, el Servidor *DNS Caché*, responderá con los datos almacenados previamente. Este proceso continuará hasta que el valor indicado en el campo “tiempo de vida” (*TTL*) del Registro de Recurso expire, momento en el que el Registro anterior se descartará del almacenamiento temporal (*Caché*). La siguiente solicitud por este Registro, resultará en una nueva consulta por parte del *Resolver*, a un Servidor de Nombres con autoridad para la zona. El uso de Servidores *Caché*, incrementa considerablemente el rendimiento de las operaciones DNS para los host clientes, se reduce de manera significativa la carga sobre la red, obteniendo una copia única de datos accedidos de manera frecuente, permitiendo que éstos estén disponibles todas las veces que se lo requieran.

Aunque todos los servidores de nombres DNS almacenan temporalmente las consultas que han resuelto, los Servidores de almacenamiento temporal o Servidores *Caché* son servidores de nombres DNS cuyo único trabajo es realizar consultas, almacenar en caché las respuestas y devolver los resultados. Es decir, no tienen autoridad sobre ningún dominio y sólo contienen la información que han acumulado en caché mientras resolvían consultas. En solicitudes posteriores por datos ya consultados, el servidor de almacenamiento temporal responderá con registros ya almacenados localmente.

## 2.9. Archivos de Zona y Registros de Recursos

Los Archivos de zona son archivos de texto que contienen información sobre un espacio de nombres particular. Cada archivo de zona contiene directivas y Registros de Recursos. Las directivas le dicen al servidor de nombres que realice tareas o aplique configuraciones especiales a la zona. Los Registros de Recursos definen los parámetros de la zona y asignan identidades a hosts individuales.

Estos archivos siguen un estándar descrito en el RFC 1035 [19]. Pueden contener tres tipos de entradas:

- Comentarios: Todos los comentarios comienzan en el carácter “;”, continuando hasta el final de la línea.
- Directivas: Todas las directivas comienzan con el carácter “\$” y se utilizan para controlar el procesamiento de los archivos de zona.
- Registros de Recursos (RR): Los Registros de Recursos se utilizan para definir las características y propiedades que figuran dentro del dominio. Los RR están contenidos en una sola línea, con la excepción de aquellas entradas entre paréntesis, las cuales pueden definirse en varias líneas.

La siguiente figura muestra un fragmento de un archivo de zona.

```

; this is a full line comment
$TTL 12h    ; directive - comment terminates the line
$ORIGIN example.com.
; Start of Authority (SOA) record defining the zone (domain)
; illustrates an RR record spread over more than one line
; using the enclosing parentheses
@ IN SOA ns1.example.com. hostmaster.example.com. (
                2003080800 ; se = serial number
                3h        ; ref = refresh
                15m       ; ret = update retry
                3w        ; ex = expiry
                2h20m     ; min = minimum
                )
; single line RR
    IN NS ns1.example.com.
...

```

Figura 7: Fragmento de un archivo de zona.

### 2.9.1. Contenido de un Archivo de Zona

En general un archivo de zona normalmente contiene los siguientes Registros de Recursos y directivas:

- Directiva \$TTL: Define el valor *Time to Live (TTL)* de la zona dominio, es el tiempo que un registro de recursos puede ser almacenado en caché por otro servidor DNS. Esta directiva es obligatoria.
- Directiva \$Origin: Es el nombre de dominio para la zona que se define. Esta directiva es opcional.
- RR *Start of Authority (SOA)*: El Registro de Recurso *SOA* debe ser el primer registro en un archivo de zona, en el cual se describen las características globales de la zona. Solo puede haber un RR *SOA* en un archivo de zona y la información que contiene es utilizada por los servidores DNS esclavos que trabajen bajo esta *SOA*. Este RR es obligatorio.
- RR *Name Server (NS)*: Define los servidores de nombres que tienen autoridad sobre la zona. Debe haber dos o más registros de recursos *NS* en un archivo de zona. Estos registros, pueden hacer referencia a servidores en el dominio o a un dominio externo. Estos RRs son obligatorios.
- RR *The Mail Exchanger (MX)*: Define los servidores de correo para la zona. Si el dominio no ofrece servicios de correo electrónico, no es necesario definir registros *MX*. Este registro es opcional.
- RR *Address (A)*: Se utiliza para definirla dirección IPv4 de todos los hosts (o servicios) que existen en la zona. Permite crear un mapeo nombre-dirección de manera tal que los recursos estén disponibles para ser accedidos. Los registros *AAAA*, permiten definir entradas para direcciones IPv6. Este registro es opcional.
- RR *CNAME*: El Registro de Recurso Nombre Canónico, permite proporcionar un nombre alternativo para un recurso (alias). Estos registros permiten utilizar más de un nombre para señalar a un único host, lo que facilita tareas como alojar un servidor FTP y un servidor Web en un mismo equipo. Es opcional.

## 2.10. El Protocolo DNS

El protocolo DNS consta de dos partes principales: un protocolo de pregunta/respuesta utilizado para realizar consultas para nombres de dominios, y otro protocolo para el intercambio de registros de bases de datos (transferencias de zona) y notificaciones a los Servidores esclavos, como consecuencia de un cambio en la zona principal (*DNS Notify*) y en las actualizaciones dinámicas de la zona (*dynamic updates*).

### 2.10.1. Consultas DNS

La principal tarea llevada a cabo por un Servidor de Nombres Autoritativo, es responder a las consultas de un Servidor *Resolver* local o remoto, o de otro Servidor de Nombres actuando como un *Resolver*. La librería de resolución (*stub resolver*) presente en cada PC Cliente, es utilizada para traducir una solicitud de usuario o aplicación a una consulta DNS. La siguiente figura ilustra el proceso típico de consulta DNS desde un PC Cliente.

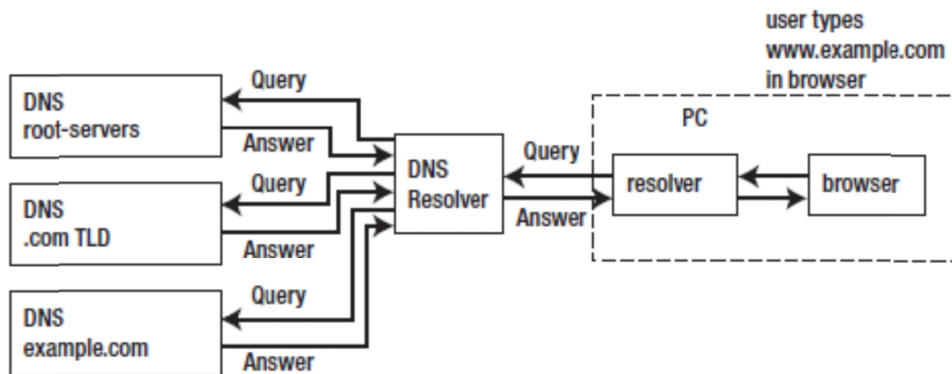


Figura 8: Proceso típico de consulta DNS desde un PC Cliente.

Existen tres tipos de consultas en un Sistema de Nombres de Dominios:

- Consultas recursivas: Una consulta de este tipo es aquella en la que el Servidor *Resolver* hará todo el trabajo necesario para devolver una respuesta completa. Es decir que el *Resolver* tendrá que enviar múltiples consultas a una serie de Servidores de Nombres Autoritativos en la jerarquía DNS, a fin de resolver la solicitud de manera completa.
- Consultas iterativas: En las consultas iterativas, si el Servidor *Resolver* tiene la respuesta o bien ésta se encuentra almacenada en caché, devolverá una respuesta. Caso contrario, devolverá una referencia al siguiente nivel de delegación.
- Consultas inversas: Declaradas obsoletas según RFC 3425.

## **Consultas recursivas**

Cuando un servidor de nombres recibe una consulta recursiva, éste está obligado a responder con los datos solicitados, o con un error que indique que el nombre del dominio especificado o los datos del tipo pedido no existen.

Existen tres tipos de respuestas ante una consulta recursiva:

- La respuesta a la consulta, acompañada de posibles registros CNAME (alias).
- Un error que indica que el dominio o el host no existe (NXDOMAIN). Esta respuesta también puede contener registros CNAME que apuntan al host no existente.
- Una indicación de error temporal, por ejemplo: no es posible acceder a otros servidores de nombres debido a un error en la red, etc.

## **Consultas iterativas**

El servidor debe proveer una respuesta parcial o bien un error ante la consulta que recibe. Existen cuatro tipos de respuestas ante consultas iterativas:

- La respuesta a la consulta, acompañada de posibles registros CNAME.
- Un error que indica que el dominio o el host no existe (NXDOMAIN). Esta respuesta también puede contener registros CNAME que apuntan al host no existente.
- Una indicación de error temporal, por ejemplo: no es posible acceder a otros servidores de nombres debido a un error en la red, etc.
- Una referencia (un listado de dos o más servidores de nombres y direcciones IP más cercanas al siguiente nivel en la jerarquía DNS, al nombre de dominio solicitado). Una referencia es el método de respuesta habitual de los servidores raíz y los servidores de dominios de nivel superior debido a que ambos tipos de servidores sólo admiten consultas iterativas.

En la siguiente figura se muestra un ejemplo de ambos tipos de consultas. En el paso 1, la estación cliente formula una **consulta recursiva** a su servidor DNS. El servidor local es el responsable de resolver la consulta, aunque para ello tenga que reenviar la pregunta a otros servidores. Si se ha solicitado información local, el servidor extrae la respuesta de su propia base de datos. Si es sobre un host externo, el servidor comprueba su caché. Si no tiene dirección IP entonces formulará una **consulta iterativa** al servidor del dominio raíz (paso 2). Ambas consultas indican en la Sección “Question” el tipo de Registro de Recurso solicitado (RR type A). El servidor del dominio raíz no conoce la dirección IP solicitada, pero responde con un conjunto de Registros del tipo NS, donde se indican los servidores de nombres con autoridad sobre el dominio .com (paso 3).

El proceso anterior se repite en otro nivel de la jerarquía, es decir que ahora el servidor local reenvía la consulta iterativa al servidor del dominio .com que tampoco conoce la



dirección IP, aunque sí conoce la dirección del DNS del dominio *.google.com* (pasos 4 y 5). El servidor local vuelve a reenviar la pregunta iterativa al DNS *google.com*, que ahora sí conoce la dirección IP de *www.google.com* y devuelve la IP al DNS local, utilizando en este caso un Registro del tipo A (pasos 6 y 7).

En el último paso, el servidor local se la reenvía a la estación cliente, al mismo tiempo que la almacena en la propia caché. El tiempo de validez de la respuesta en la caché se configura en los servidores remotos y se envía como parte de la respuesta

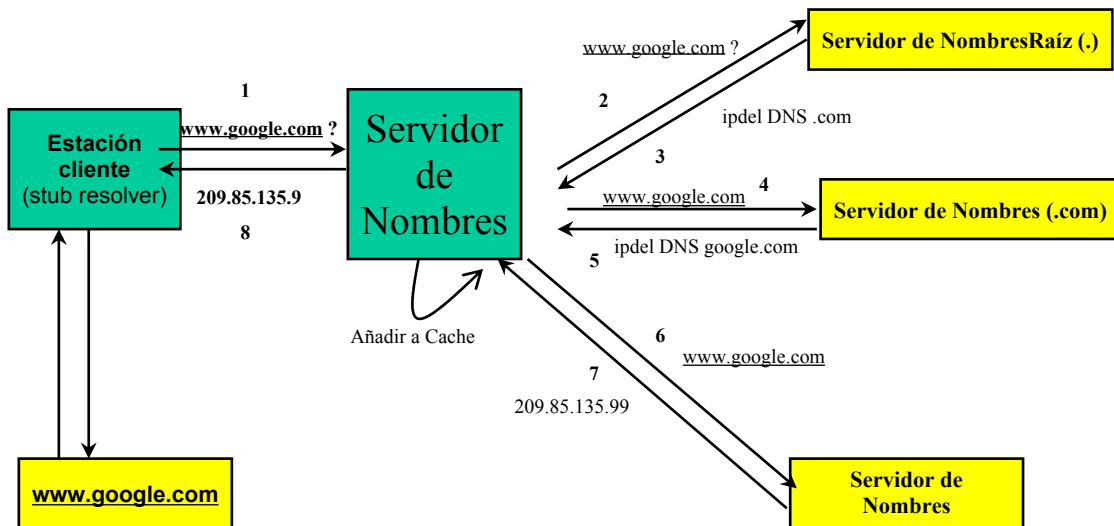


Figura 9: Consultas recursivas e iterativas.

Se debe resaltar que el proceso de interacción ilustrado anteriormente se encuentra simplificado, ya que las respuestas del tipo NS en los pasos 3 y 5 contienen los nombres de servidores autoritativos y no direcciones IP.

## 2.10.2. Mapeo de Direcciones a Nombres (DNS Reverse Mapping)

Dado un nombre de dominio, en una consulta normal de DNS, se trata de determinar la dirección IP asociada a éste. Sin embargo, en algunas oportunidades se necesitará encontrar el nombre de un host dada una dirección IP.

El uso de esta funcionalidad puede observarse como una herramienta de diagnóstico pero fundamentalmente se utiliza por razones de seguridad, por ejemplo, muchos sistemas de correo hacen uso de *reverse mapping* para proporcionar un método de autenticación simple usando políticas de *DNS lookup*, es decir que una resolución (IP a Nombre) y (Nombre a IP) confirman que la dirección IP dada corresponde al host indicado.

Para poder resolver este tipo de situaciones se ha creado un dominio especial (reservado) llamado IN-ADDR.ARPA. Cuando un cliente DNS desea conocer el nombre de dominio asociado a la dirección IP *w.x.y.z* realiza una pregunta inversa a *z.y.x.w.in-addr.arpa*. La inversión de los bytes es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones IP.

La organización que posee una dirección de red es responsable de registrar todas sus traducciones de dirección a nombre en la base de datos del DNS. Esto se hace en una tabla que es independiente de las correspondencias entre nombre y direcciones. El dominio in-addr.arpa se creó para apuntar hacia todas esas tablas de red.

El dominio ARPA es administrado conjuntamente por ICANN/IANA y la IETF/IAB [20] y a diferencia de los nombres de dominio, los cuales usan servidores de nivel superior (*gTLD* o *ccTLD*) como el siguiente nivel de delegación, las direcciones IPv4 son delegadas a través de Registros Regionales de Internet (*RIRs*). La siguiente tabla muestra una lista de estos últimos Registros.

RIR Name	Coverage	Web
APNIC	Asia Pacific	www.apnic.net
ARIN	North America, Southern Africa, parts of the Caribbean	www.arin.net
LACNIC	South America, parts of the Caribbean	www.lacnic.net
RIPE	Europe, Middle East, Northern Africa, parts of Asia	www.ripe.net
AFRINIC	Africa (This RIR is planned to be fully accredited by ICANN in late 2005/2006 and at that time will assume responsibilities for African registrations that are presently handled by ARIN and RIPE.)	www.afrinic.net

Tabla 1: Registros Regionales de Internet (RIRs).

Los *RIRs* operan bajo procedimientos definidos en el RFC 2050. Las direcciones IPv4 se asignan en bloques de red (*net blocks*) por los *RIRs* a un Registro Local de Internet (*LIR*), por lo general un ISP, o a un Registro Nacional de Internet (*NIR*), que a su vez los asignará a algún *LIR*. A cada Registro se le delega la responsabilidad de realizar un “mapeo de direcciones a nombres”, para las direcciones que se le han asignado. Del mismo modo un *LIR* puede delegar la responsabilidad de realizar el mapeo a usuarios finales en el caso de que se les haya asignado direcciones IPv4 fijas. La siguiente figura muestra un ejemplo de *reverse mapping* usando consultas recursivas. El uso de la dirección IPv4 192.168.250.15 es a modo ilustrativo ya que se trata de una dirección IP privada, careciendo de sentido su uso en una red pública.

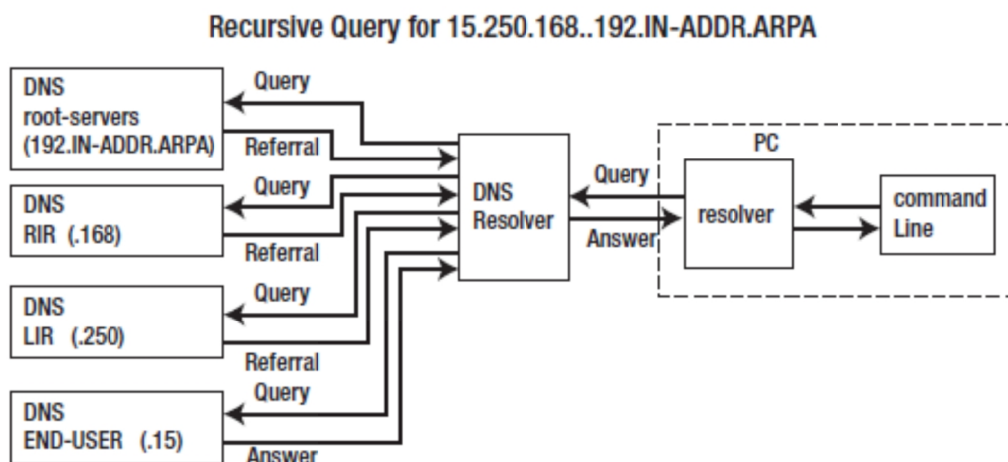


Figura 10: reverse mapping usando consultas recursivas.

## 2.11. Mantenimiento y Transferencias de Zonas

Debido al importante papel que desempeñan las zonas en DNS, se pretende que éstas estén disponibles desde varios servidores DNS en la red para proporcionar disponibilidad y tolerancia a fallos al resolver consultas de nombres. En caso contrario, si sólo se utiliza un servidor y éste no responde, se pueden producir errores en las consultas de nombres de la zona. Para que otros servidores alojen una zona, son necesarias transferencias de zona que repliquen y sincronicen todas las copias de la zona utilizadas en cada servidor configurado para alojar la zona. Con el fin de simplificar estas operaciones que involucran a múltiples servidores, es conveniente que un único origen pueda actualizar a múltiples servidores, [21].

El tiempo transcurrido entre la transferencia de los cambios de los archivo de zona es un factor determinante de la velocidad con la que los cambios en la información de la zona son propagados a través de Internet. El diseño inicial del Sistema de Nombres de Dominio permitía la propagación de cambios usando operaciones de transferencia de zona completa (*AXFR*). Pero desde aquellos comienzos (año 1987) a la actualidad, Internet ha crecido exponencialmente, por lo que el deseo de acelerar el proceso de propagación de actualizaciones en una zona minimizando el uso de recursos, ha dado como resultado un número de cambios en este aspecto al diseño e implementación del DNS, desde simples pero efectivos retoques como la Transferencia de Zona Incremental (*IXFR*) y mensajes de Notificación (*NOTIFY messages*), hasta conceptos más complejos como actualizaciones dinámicas (*DDNS*).

### 2.11.1. Transferencia de Zona Completa (AXFR)

Cuando se agrega un nuevo servidor DNS a la red y se configura como un nuevo servidor esclavo en una zona existente, dicho servidor realiza una transferencia inicial completa de la zona para obtener y replicar una copia total de los registros de recursos de la zona.

En las especificaciones originales de DNS [22], se define que un servidor de nombres esclavo, de forma periódica, debe contactar al servidor maestro de la zona, para comprobar si hubo cambios en tales archivos de zona y en consecuencia solicitar una transferencia de zona. El periodo de tiempo para contactar al servidor maestro se encuentra presente en un valor definido en el Registro de Recurso *SOA* para el dominio.

Para detectar cambios, los servidores esclavos comprueban el campo número de serie del Registro de Recurso *SOA* de la zona. De forma adicional, el campo número de serie del *SOA* de la zona siempre avanza cuando se produce cualquier cambio en la zona. El avance es un incremento simple, o puede estar basado en la fecha y la hora del archivo maestro. El propósito es hacer posible la determinación de cuál de las dos copias de una zona es más reciente comparando sus números de serie. Para el caso de los números de serie, se utilizan una secuencia aritmética, por tanto hay un límite teórico en cuanto a la rapidez de actualización de una zona, básicamente las copias antiguas mueren antes de que el número de serie llegue a la mitad de su rango de 32 bits.

La comprobación periódica de los servidores esclavos se controla con los parámetros del RR *SOA* de la zona, que establece el intervalo mínimo aceptable de comprobación. Los parámetros se llaman *REFRESH*, *RETRY*, y *EXPIRE*. Incluso cuando se carga una

nueva zona secundaria, ésta espera los segundos de refresco (*REFRESH*) antes de comprobar un nuevo número de serie de la zona primaria. Si no se completa esta comprobación, se comprueba de nuevo cada (*RETRY*) segundos. La comprobación es una consulta simple al RR *SOA* de la zona primaria. Si el campo del número de serie de la zona secundaria es igual al número de serie devuelto por la zona primaria, quiere decir que no se han realizado cambios, y el intervalo de refresco espera a reiniciarse. Si la zona secundaria ve que es imposible realizar una comprobación del número de serie en el intervalo de expiración (*EXPIRE*), asumirá que su copia de la zona es obsoleta y la descartará.

Cuando la comprobación dice que la zona ha cambiado, el servidor esclavo solicitará una transferencia de zona mediante una solicitud *AXFR* para la zona. El *AXFR* puede provocar un error, como un rechazo, pero normalmente es respondido por una secuencia de mensajes de respuesta.

La transferencia de Zona Completa usa como protocolo de Transporte a TCP.

### **2.11.2. Transferencia de Zona Incremental (*IXFR*)**

Transferir archivos de zona muy grandes puede llevar mucho tiempo, consumo de ancho de banda e inclusive otros recursos, sobre todo si solo un único registro ha sido modificado.

Las transferencias de zonas incrementales se describen en [23] como un estándar DNS adicional para replicar zonas DNS. Cuando un servidor DNS que actúa como origen para una zona y los servidores que copian la zona de él admiten las transferencias incrementales, se obtiene un método más eficiente para la propagación de los cambios y las actualizaciones de la zona.

Con las transferencias de zona incrementales (*IXFR*), primero se determinan las diferencias entre el origen y las versiones replicadas de la zona. Si se determina que las zonas tienen la misma versión, en función del valor indicado en el campo de número de serie del registro de recursos de inicio de autoridad (*SOA*) de cada zona, no se realiza ninguna transferencia.

Si el número de serie de la zona de origen es mayor que el del servidor esclavo solicitante, solamente se realiza una transferencia de los cambios efectuados en los Registros de Recursos de cada versión incremental de la zona. Para realizar una consulta *IXFR* efectiva y enviar los cambios, el servidor DNS de origen de la zona debe mantener un historial de los cambios incrementales de la zona para utilizarlo al responder a estas consultas. El proceso de transferencia incremental requiere bastante menos tráfico en la red y las transferencias de zona se completan mucho más rápidamente.

### **2.11.3. Notificación DNS (*NOTIFY*)**

Definido en [24], como un mecanismo mediante el cual un servidor maestro enviará un mensaje *NOTIFY* a un servidor de nombres esclavo cada vez que la zona es cargada o actualizada. Este mensaje indica que pueden haber ocurrido cambios en los registros del dominio. El servidor esclavo, al recibir el mensaje *NOTIFY* solicitará el Registro de

Recurso *SOA* al servidor maestro y comprobará el campo número de serie, si éste resulta mayor que el almacenado por el servidor esclavo, se solicitará una transferencia de zona utilizando transferencia completa (*AXFR*) o bien transferencia incremental (*IXFR*), para extraer los cambios de zona y actualizar sus réplicas locales de la zona.

Mediante el uso de mensajes *NOTIFY* se puede reducir considerablemente los tiempos de propagación de cambios en una zona, hacia servidores esclavos.

#### **2.11.4. Actualización Dinámica (*Dynamic Update*)**

El método clásico de actualización de Registros de Recursos de una zona, es mediante la edición manual del archivo de zona y luego detener e iniciar el servidor de nombres para que éste lea el archivo de zona actualizado y propague los cambios. Cuando el volumen de cambios alcanza un cierto nivel, esto puede volverse operacionalmente inaceptable, especialmente considerando que en una organización se manejan un gran número de archivos de zona, tal es el caso de los Proveedores de Servicios de Internet. A un servidor ejecutando a BIND como software DNS, puede llevarle bastante tiempo reiniciar ya que inicializa un gran número de archivos de zona.

Muchos usuarios de DNS buscan un método que rápidamente cambie los registros de una zona mientras el servidor de nombres continúa respondiendo consultas de usuarios. Existen dos enfoques que permiten resolver este problema:

1. Permitir actualizaciones en tiempo de ejecución de Registros de Recursos de la zona desde un origen externo o aplicación.
2. Directamente alimentar los Registros de Recursos de la zona desde una base de datos, la cual puede ser actualizada dinámicamente.

En base al primer enfoque, en el RFC 2136,[10], se define un proceso llamado DNS Dinámico (*DDNS*), por el cual los registros de una zona pueden ser actualizados desde uno o más orígenes externos. La limitación clave en ésta especificación es que un nuevo dominio o zona no puede ser agregado o eliminado dinámicamente. Todos los registros dentro de una zona existente pueden ser agregados, modificados o eliminados con la excepción del Registro de Recurso *SOA*.

Como parte del RFC citado en el párrafo anterior, se encuentra la definición del término “Maestro Primario”, que describe el nombre del Servidor definido en el Registro de Recurso *SOA* para la zona. Cuando se actualizan dinámicamente Registros de Recursos de una zona, se hace necesario actualizar solo un servidor aún cuando existan múltiples servidores maestros para la zona. Para solucionar esta situación, se debe seleccionar un *boss server*. El servidor maestro primario que actúa como *boss server*, no tiene ninguna característica especial, más que la de estar definido en el servidor de nombres en el registro de recursos *SOA* y puede aparecer en una sentencia que permita actualización de la configuración del archivo de configuración del software DNS (*BIND named.conf*) para controlar el proceso de actualización dinámico.

La definición de DNS Dinámico (*DDNS*) aparece también en conjunto con las características de DNS Seguro, especialmente *Transaction Signature*[9] y *Transaction Key*[8]. Sin embargo *DDNS*, no requiere ni depende de tales características.

La razón de que estas dos características antes citadas estén estrechamente relacionadas, es que al permitir DNS Dinámico, los archivos de zona quedan expuestos a la posibilidad de corrupción, por ejemplo mediante la técnica de “envenenamiento”, por orígenes mal intencionados. Sin embargo, la principal ventaja del uso de DDNS es que usuarios remotos están habilitados a realizar actualizaciones y controlar la configuración de sus dominios de manera semiautomática. Bajo estas circunstancias, un uso responsable de DNS Dinámico implica el uso de procedimientos provistos por el protocolo *TSIG*, que permitan autenticar las solicitudes entrantes.

Un enfoque alternativo a DNS Dinámico es la solución de software para BIND DNS, *DLZ (Dynamically Loadable Zones)*, mediante el cual se reemplazan todos los archivos de zona con un único archivo de zona que describe una base de datos. Mediante BIND-DLZ, todas las consultas entrantes, son dirigidas a rutinas de acceso a la base de datos, de modo que cualquier dato de zona, ya sea nuevo, modificado o eliminado se refleja inmediatamente en la respuesta del servidor de nombres. Están soportadas la mayoría de las bases de datos *Open Source* tales como *MySQL*, *PostgreSQL*, *BDB* y *LDAP*.

## 2.12. Mensajes DNS

Los mensajes DNS son los paquetes de datos intercambiados entre Servidores de Nombres (Maestro – Esclavo) o Servidores y *Resolvers* (Cliente). Comprender como se encuentran conformados es muy importante ya que corresponden a la base del protocolo de comunicación en DNS (preguntas y respuestas). Como se describe en [5], existe un formato único, el cual es usado para todas las operaciones DNS (consultas, respuestas, transferencia de zonas, notificaciones y actualizaciones dinámicas).

Por lo general para el intercambio de preguntas y respuestas en DNS, se utiliza a UDP como protocolo de transporte. Sin embargo si éstas sobrepasan el límite de 512 bytes, se reemplaza el uso de UDP por TCP, dado el eventual truncado de mensajes que se produce al utilizar UDP con una cantidad de datos superior a este número.

### 2.12.1. Formato del Mensaje

Cada mensaje tiene el mismo formato genérico, donde se distinguen cinco secciones:

Sección	Significado/Usó
Sección 1	<i>HEADER</i> : Encabezado del mensaje
Sección 2	<i>QUESTION SECTION</i> : La consulta DNS para el que se solicita una respuesta.
Sección 3	<i>ANSWER SECTION</i> : Contiene el o los Registros de Recursos correspondientes a la consulta.
Sección 4	<i>AUTHORITY SECTION</i> : Detalla de forma explícita los Registros de Recursos de los Servidores de Nombres (Registros NS) referentes a la petición solicitada.
Sección 5	<i>ADDITIONAL SECTION</i> : Contiene la información que permite completar el resto de las secciones. Por ejemplo en el caso de existir varios servidores de nombres se explicitan sus direcciones IP para que puedan ser contactados

Tabla 2: Formato de mensaje DNS.

El encabezado del mensaje tiene una longitud fija de 12 bytes e incluye campos que especifican que secciones están presentes, si el mensaje se trata de una consulta o una respuesta, si es una consulta estándar u otro tipo de operación.

### Encabezado DNS

El mismo se encuentra definido en [19] y a continuación se describen los bits de bandera (*flags*) y campos que lo conforman. Los valores presentes permiten controlar una transacción. La siguiente figura ilustra el formato del encabezado de un paquete DNS, la misma se organiza en filas con una longitud de 2 bytes, es decir que cada fila puede almacenar 16 bits de datos. Es así que por ejemplo el campo Identificador puede contener un valor dentro del rango  $(0 - 2^{16} - 1)$

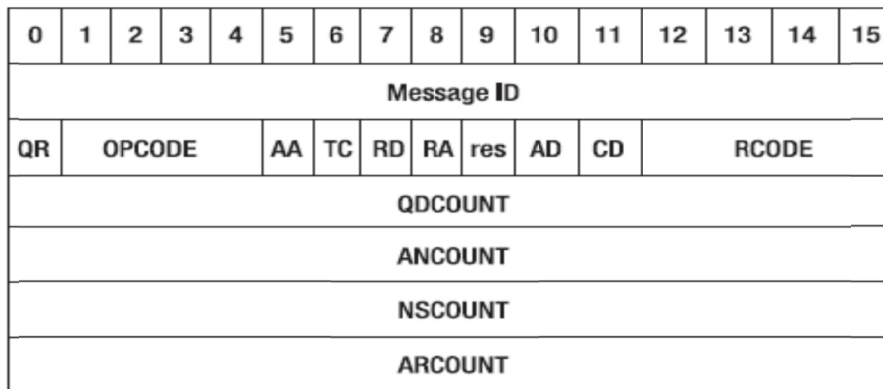


Figura 11: Encabezado mensaje DNS.

Sección	Bits	Significado/Uso
ID Mensaje	16	Identificador de 16 bits de petición que crea el cliente cada vez que desea realizar una resolución de nombres. De esta forma el cliente puede solicitar simultáneamente varias peticiones para nombres distintos (cada una con un <i>Query ID</i> diferente) y no confundir las respuestas.
QR	1	Campo de un bit que especifica si este mensaje es una consulta (0), o una respuesta (1).
OPCODE	4	Campo de 4 bits de longitud que identifica la solicitud/tipo de operación. Puede contener los siguientes valores: 0 = QUERY. Consulta estándar. 1 = IQUERY. Consulta Inversa. Obsoleta por RFC 3425. 2 = STATUS. Estado de la solicitud DNS 3 = Solicitud NSID [25] 4 = NOTIFY. 5 = DDNS. Actualización 6-15 = Reservado para uso futuro.  Nota: Actualmente este campo siempre tiene por valor 0, lo que indica que es una consulta estándar.
AA	1	<i>Flag</i> de respuesta autoritativa ( <i>Authoritative Answer</i> ). Solo válido en respuestas, si está activo, especifica que el servidor de nombres que responde tiene autoridad para el nombre de

		dominio enviado en la consulta.
TC	1	<i>Flag</i> de truncado. Especifica que el mensaje fue truncado debido a que su longitud es mayor de lo que permite el canal de transacción. Cuando un mensaje DNS es enviado usando UDP como protocolo de transporte, la longitud máxima permitida es de 512 bytes (p32 [19]). Las consultas no pueden exceder este límite, debido a la longitud máxima fijada para un nombre de dominio, es decir que solo las respuestas podrán tener el bit TC puesto a 1. Por lo tanto un Cliente DNS realiza una consulta, la cual se limita a 512 bytes como máximo a través de UDP. A continuación la respuesta se envía a través de UDP; si la respuesta no se ajusta a un paquete UDP, ésta es truncada (se envía un paquete UDP con el bit TC a 1). Un Cliente DNS que observa el bit puesto a 1, ahora reenvía su consulta sobre TCP.
RD	1	<i>Flag</i> de recursividad. Este bit indica al Servidor de Nombres que se pide resolución recursiva. Si el Servidor soporta recursión (por ejemplo un Servidor Cache), éste responde con ambos bits, el RD y el RA puestos a 1. El soporte para resolución recursiva es opcional.
RA	1	<i>Flag</i> de recursividad disponible. Solo válido en una respuesta. Indica si el servidor de nombres soporta resolución recursiva (1).
Res	1	Reservado por la IANA para uso futuro, según se especifica en [19].
AD	1	Dato autenticado. Usado por DNSSEC (Extensiones de seguridad para DNS). Indica que el dato fue autenticado de manera confiable.
CD	1	Comprobación Desactivada. Usado por DNSSEC. Si esta seteado, significa que el que inicia la solicitud (Servidor de Nombres o Resolver) tomará la responsabilidad para el procesamiento de la seguridad. Tanto para el bit AD como para el CD, se extenderán detalles en el capítulo sobre DNSSEC.
RCODE	4	Código de Respuesta. Identifica el tipo de respuesta ante una consulta. Por lo tanto los cuatro bits deben ser 0 en el caso de una consulta. A continuación se listan los valores asignados actualmente: 0 = NOERR. Ningún Error. 1 = FORMERR. Error de formato. El servidor fue incapaz de interpretar el mensaje, es decir que la consulta no se pudo analizar. 2 = SERVFAIL. Fallo en el servidor. El mensaje no fue procesado debido a un problema con el servidor. 3 = NXDOMAIN. Error en nombre. El nombre de dominio de la consulta no existe, es decir que el Servidor no dispone de datos para responder. Sólo válido si el bit AA está activo en la respuesta. 4 = NOTIMP. No implementado. El tipo solicitado de consulta no está implementado en el servidor de nombres.



		<p>5 = REFUSED. El Servidor de Nombres rechaza responder a la operación solicitada debido a las políticas configuradas, por ejemplo: solo se permite recursión para Clientes locales.</p> <p>6 = YXDOMAIN. Un nombre que no debe existir, existe, [10]</p> <p>7 = YXRRSet. Un conjunto de Registros de Recursos que no debe existir, existe,[10].</p> <p>8 = NXRRSet. Conjunto de Registros de Recursos que debe existir, no existe,[10].</p> <p>9 = NotAuth. El Servidor no tiene autoridad para la zona que aparece en la sección de zona,[10].</p> <p>10 = NotZone. El nombre usado en la Sección Actualización no se encuentra denotado en la Sección Zona,[10].</p> <p>11-15 = Reservado para uso futuro. Actualmente usado con EDNS0.</p> <p>16 = BADVERS. Para uso con Mecanismos de Extensión de DNS, [26].</p> <p>16 = BADSIG. Fallo de firma <i>TSIG</i>, [9].</p> <p>17 = BADKEY. <i>KEY</i> no reconocida, [9].</p> <p>18 = BADTIME. Firma fuera de ventana de tiempo, [9].</p> <p>19 = BADMODE. Modo <i>TKEY</i> incorrecto o nombre de <i>KEY</i> inválido, [8].</p> <p>20 = BADNAME. Nombre de llave duplicado, [8].</p> <p>21 = BADALG. Algoritmo no soportado, [8].</p> <p>22-3840 = Disponible para uso futuro.</p> <p>3841-4095 = Para uso privado.</p> <p>4096-65535 = Disponible para uso futuro.</p>
QDCOUNT	16	Especifica el número de entradas en la sección "QUESTION". Normalmente se especifica el valor 1.
ANCOUNT	16	Especifica el número de Registros de Recursos (RRs) en la sección "ANSWER".
NSCOUNT	16	Especifica el número de RRs en la sección "AUTHORITY".
ARCOUNT	16	Especifica el número de RRs en la sección "ADDITIONAL RECORDS".

Tabla 3: Descripción campos encabezado DNS.

### Sección "QUESTION"

Se especifican los parámetros que definen lo que se solicita. El formato es el siguiente:

Nombre del campo	Significado/Uso
QNAME	El nombre de dominio que está siendo consultado
QTYPE	El tipo de Registro de Recurso que está siendo solicitado. La lista de valores son definidos por la <i>IANA</i> , [27]. Ejemplo: A, ANY, NAPTR, etc.
QCLASS	La clase de Registro de Recurso que está siendo solicitado. Ejemplo: Internet, CHAOS, etc.

### Formato de los campos

Nombre	Explicación
QNAME	El nombre de dominio se especifica como una secuencia de etiquetas. Cada etiqueta podrá tener una longitud máxima de 63 bytes y es representada como una longitud (un octeto), seguida de una cadena de caracteres (definida por la longitud). El nombre de dominio termina con un octeto de longitud cero para la etiqueta nula de la raíz. Ejemplo: <pre> 03 77 77 77 07 65 78 61 6D 70 6D 65 03 63 6F 6D 00 w w w   e x a m p l e   c o m </pre>
QTYPE	Tipo de Registro de Recurso que está siendo solicitado. Por ejemplo el Registro tipo A tiene el valor 1, el Registro tipo MX tiene el valor 15. Los valores posibles son asignados por la IANA, [27].
QCLASS	Valor de 16 bits que indica la clase de Registro de Recurso que se solicita. Algunos de los valores posibles son: 1 = IN (Internet) 2 = Obsoleto 3 = CH (CHAOSNet, usado en los Laboratorios del MIT) 4 = HS (HESIOD) Este sistema de clasificación fue diseñado en un principio para soportar el uso de DNS en diferentes protocolos de red. Actualmente solo el valor 1 es usado.

Tabla 4: Formato campos sección QUESTION.

### Secciones "ANSWER", "AUTHORITY" y "ADDITIONAL INFORMATION"

Estas secciones comparten el mismo formato, contienen un número variable de Registros de Recursos, número que es especificado en su correspondiente campo en el encabezado del mensaje DNS. Cada Registro de Recurso tiene el siguiente formato:

Nombre del campo	Significado
NAME	El nombre que se devuelve, por ejemplo: www o ns2.ejemplo.net. Si el nombre está en el mismo dominio de la consulta, se devuelve solo la parte de host ( <i>label</i> ) y un registro puntero el cual es usado para construir un ( <i>FQDN</i> ), caso contrario se devuelve el nombre completo.
TYPE	El tipo de Registro de Recurso retornado, por ejemplo: NS o AAAA.
CLASS	La Clase de Registro de Recurso que se devuelve, por ejemplo: Internet, CHAOS, etc.
TTL	Define cuánto tiempo se puede considerar válido el Registro de Recurso retornado, expresado en segundo, por ejemplo: 2800.
RDLENGTH	La longitud en octetos del campo RDATA que se devuelve.
RDATA	Una cadena de longitud variable de octetos que describe el recurso. El formato de esta información varía de acuerdo a los campos TYPE y CLASS del registro de recurso. Por ejemplo si TYPE es A y CLASS es IN, el campo RDATA es una dirección de Internet con una longitud de cuatro octetos

Tabla 5: Formato campos secciones ANSWER, AUTHORITY y ADDITIONAL INFORMATION.

## Formato del campo NAME

EL formato del campo *NAME* se indica en un valor de dos bits que se encuentran al principio del contenido del campo. La siguiente tabla muestra el significado y diseño de cada uno de los tres posibles tipos.

Valor	Long. campo	Significado
00	6	Indica que se trata del formato de etiqueta ( <i>label</i> ).
11	14	Indica que se trata del formato puntero. Los siguientes 14 bits se asumen como el desplazamiento ( <i>offset</i> ) desde el comienzo del mensaje de un nombre que se encuentra en un formato de etiqueta estándar.
01	6	Denota un formato de mensaje con soporte para uso de Mecanismos de Extensión para DNS ( <i>EDNS0</i> ), [26].

Tabla 6: Formato campo NAME

### 2.13. Mecanismo de Extensión para DNS (EDNS0)

El formato básico de un mensaje DNS descrito anteriormente, está limitado a una longitud fija de 512 bytes con UDP como protocolo de transporte (no incluye encabezado UDP ni IP), el campo RCODE presente en el encabezado del mensaje, también está limitado a una longitud de 4 bits. Para dar soporte a las extensiones de seguridad para DNS (DNSSEC) y así permitir el uso de criptografía, en [26], se describe el mecanismo de extensión EDNS0, el cual especifica un tipo particular de Registro de Recurso llamado “pseudo-RR OPT”, el cual es agregado a la Sección Adicional de una respuesta, para indicar el uso de EDNS0. Por lo tanto, un mensaje DNS que incluye este pseudo RR, puede exceder el límite de 512 bytes de longitud y puede contener un conjunto de código de errores expandido.

Al Registro de Recurso OPT, se lo denomina “pseudo-RR”, porque se relaciona con un determinado nivel de transporte de mensajes y no a los datos reales de DNS. Éste nunca será almacenado en caché, reenviado o cargado desde un archivo de zona, [26].

Un servidor configurado con soporte para EDNS0, anunciará su capacidad de participar en transacciones EDNS, enviando un Registro de Recurso OPT en el campo *ADDITIONAL SECTION* de una consulta. Si el servidor que recibe la consulta anterior, no tiene soporte para EDNS o no reconoce el Registro OPT, éste responderá con un mensaje de fallo, indicado en el campo RCODE del encabezado del mensaje (NOTIMP, FORMERR, o SERVFAIL). En tal caso, el servidor que envió la consulta puede continuar con la comunicación, sin el soporte de EDNS.

#### 2.13.1. Formato del Registro OTP

Como se explicó anteriormente, el Registro OPT es creado dinámicamente por el servidor, por lo que no aparece en un archivo de zona. Usa el formato estándar indicado en la tabla 5, pero redefine el uso de cada campo, según se muestra a continuación.

Nombre del campo	Significado
NAME	Siempre se indica 00 (Dominio raíz)

TYPE	Tipo de Registro (OPT)
CLASS	Tamaño máximo del mensaje UDP que puede ser aceptado por el servidor.
TTL	Este campo se presenta de la siguiente manera: Campo 1: Extended RCODE. (valores definidos en tabla “encabezado DNS”). Campo 2: Versión. Campo 3: Bits Banderas: Bit 0: DO (DNSSEC OK) Bits 1-15: Sin uso.
RDLENGTH	La longitud en octetos del campo RDATA que está siendo retornado.
RDATA	Parte variable del Registro OPT, responde a la siguiente estructura:  OPTION-CODE: Asignado por IANA OPTION-LENGTH: Longitud en octetos de la porción OPTION-DATA. OPTION-DATA: Varía según OPTION-CODE.

Tabla 7: Descripción campos Registro OPT.

### 2.13.2. TCP o UDP

Los mensajes DNS por lo general son encapsulados en UDP, el cual a su vez se encapsula en un datagrama IPv4, sin embargo, cuando se quiere pasar el límite de 512 bytes, se utiliza TCP como protocolo de transporte junto a las extensiones EDNS0. El formato y estructura de datagrama se muestra en la siguiente figura:

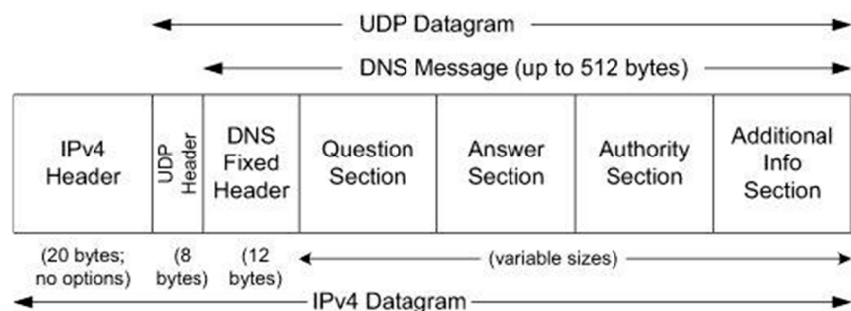


Figura 12: Estructura datagrama IP para mensaje DNS.

Cuando un Servidor Resolver, envía una consulta y la respuesta que recibe contiene el bit TC (bit de truncado) activo, el tamaño de la respuesta completa excede los 512 bytes, por lo que solo los primeros 512 bytes son retornados por el Servidor que generó la respuesta. Ahora el Resolver, puede enviar nuevamente la consulta, pero usando TCP como protocolo de transporte [6]. Las transferencias de zona completa, también hacen uso de TCP, ya que éstas pueden exceder los 512 bytes de longitud.

Cuando UDP es usado en una conexión entre Resolver y un Servidor Autoritativo, las aplicaciones de software deben encargarse de gestionar tiempos de retransmisión. En [7] se sugiere un tiempo inicial de retransmisión de al menos 4 segundos y las siguientes retransmisiones en tiempos resultantes de un incremento exponencial.

# 3. Vulnerabilidades del Sistema de Nombres de Dominio

A nivel macro, el servicio DNS es esencial para el funcionamiento de Internet. En un nivel micro o local, el servicio DNS podría ser esencial para el funcionamiento de una pequeña empresa con presencia en Internet a través de su sitio web. En todos los casos se debe poner especial atención en la seguridad, para garantizarla eficacia y la seguridad del sistema DNS.

Por su naturaleza de sistema público, desde sus comienzos el sistema DNS, fue susceptible a “ataques” de los más diversos, por parte de usuarios mal intencionados, por lo que un punto crítico en la definición de políticas y procedimientos de seguridad es primordial conocer las vulnerabilidades del sistema DNS, representadas en las posibles fuentes de amenazas en los flujos de datos de dicho sistema

En la siguiente figura se muestran los flujos de datos y ataques más conocidos en cada uno de ellos, [28].

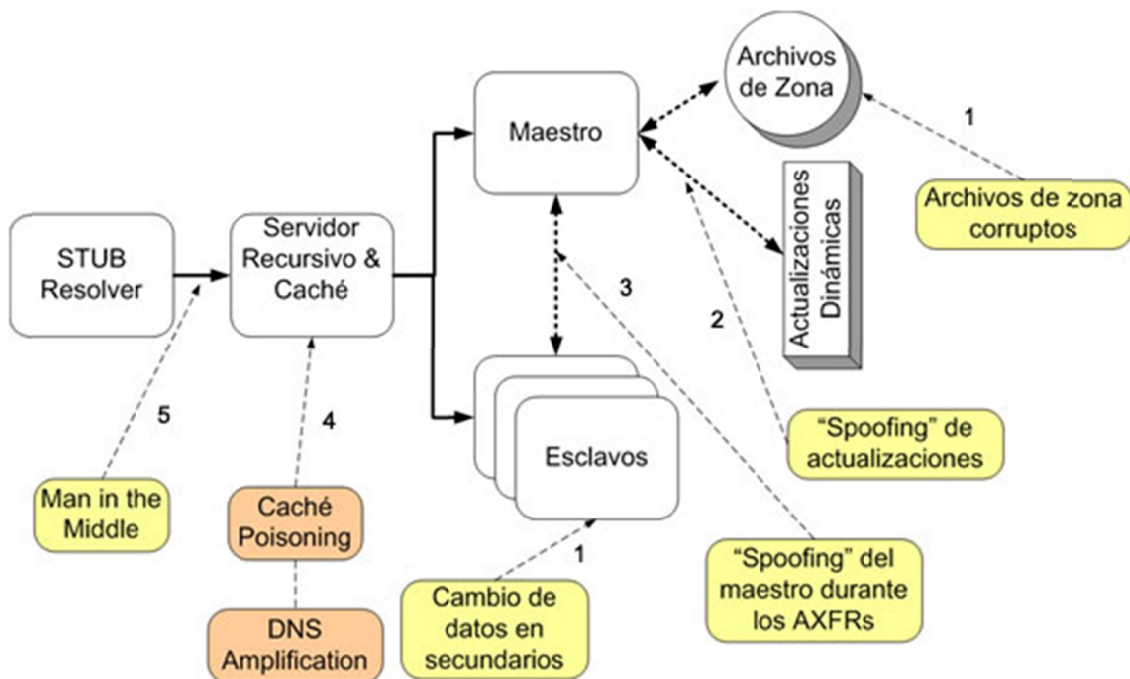


Figura 13: Vectores de ataque en el flujo de datos DNS.

## 3.1. Clasificación de amenazas al Sistema DNS

La clasificación de las amenazas a la seguridad de un sistema DNS, es un medio para permitir la selección de los recursos y estrategias adecuadas para mitigar las mismas. La siguiente tabla presenta un resumen de las amenazas más conocidas clasificadas según a que componentes afectan en un flujo de datos normal en un Sistema DNS.

Etiqueta	Área	Amenaza
1	Archivos de Zona	Corrupción de archivos (accidental o mal intencionada)
2	Actualizaciones dinámicas	Actualizaciones no autorizadas mediante la técnica de “ <i>IP Spoofing</i> ”
3	Transferencias de Zonas	Suplantación de identidad del origen en la actualización de zonas mediante la técnica de “ <i>IP Spoofing</i> ”
4	Servidor Recursivo y Caché	Ataque de amplificación
5	Consultas a un Resolver	Envenenamiento de Cache usando “ <i>IP Spoofing</i> ”, interceptación de mensajes mediante la técnica “ <i>Man in The Middle</i> ”.

Tabla 8: Clasificación amenazas al Sistema DNS.

## 3.2. Análisis de amenazas al Sistema DNS

### 3.2.1. Corrupción de Archivos

De manera general se puede definir este tipo de amenaza como todo tipo de incidentes relacionados con la modificación no autorizada de datos en el Sistema de Nombres de Dominio. Estos incidentes pueden suceder en cualquier momento y en cualquier parte de la propagación de flujos de datos del sistema.

En el supuesto caso de que solo se permita la administración de los datos en los archivos de zona de manera local, un atacante podría valerse de la técnica conocida como “ingeniería social” para obtener información de acceso a datos almacenados en un servidor DNS. En estos casos es altamente recomendable establecer procedimientos administrativos claros y seguros y por sobre todo una correcta educación sobre amenazas a la seguridad de la información a las personas involucradas directa e indirectamente en la administración de la misma.

### 3.2.2. Actualizaciones no autorizadas

Permitir que la actualización de los registros asociados a una zona se realice desde uno o más orígenes externos [10], deja abierta la posibilidad a que un atacante pueda “robar” la identidad de una fuente de origen confiable mediante la técnica de *IP Spoofing* o bien simplemente haciendo uso de Ingeniería Social, para modificar con datos fraudulentos la información almacenada en la zona.

Por todo lo anterior es que la configuración por defecto del software DNS BIND no permite las actualizaciones dinámicas desde ningún tipo de origen. En el caso de que una organización requiera esta característica, BIND proporciona parámetros de configuración que permiten minimizar el riesgo asociado, a partir de definir un perímetro de confianza donde se encuentran todos los sistemas implicados. [29]

Del mismo modo, para reducir aún más el nivel de exposición, se puede hacer uso de los procedimientos provistos por los protocolos *Transaction Key*(TKKEY) [8] y *Transaction Signature* (TSIG) [9] lo que permitirá la autenticación de las solicitudes entrantes.

### **3.2.3. Suplantación de identidad en transferencia de zonas**

El comportamiento por defecto para la transferencia de zona en DNS permite que cualquier host pueda solicitar y recibir una transferencia de zona completa para un dominio dado. Esta característica representa entonces un problema de seguridad, ya que los datos obtenidos pueden ser usados para descifrar toda la información que almacena un servidor de nombres de una organización.

Por todo esto, es que los dos métodos de uso más frecuente para asegurar la transferencia de zona entre servidores de nombres son, la configuración de Listas de Control de Acceso a través del software BIND y el uso de características provistas por el protocolo *Transaction Signatures* (TSIG). [29]

El proceso que permite restringir la transferencia de zonas mediante la utilización de Listas de Control de Acceso solo permitirá aceptar solicitudes de hosts o un grupo de hosts que fueron configurados en esta lista, rechazando cualquier solicitud que viniera de otros orígenes.

El método anteriormente descrito no se ocupa de aspectos como la suplantación de identidad del origen, ya que un atacante, haciendo uso de la técnica “hombre en el medio” (*Man in the Middle*), podría capturar el tráfico intercambiado entre Servidores Maestro y Esclavos y asumir la identidad del origen en la actualización, provocando que los Servidores Esclavos almacenen información falsa.

Es así que el siguiente nivel en el proceso de asegurar la transferencia de zonas es implementar el protocolo *TSIG*, lo que permite la autenticación entre las partes basada en el uso de técnicas criptográficas, que aseguran que Maestro y Esclavo son quienes dicen ser y por otro lado, asegurar la integridad de los datos (los datos recibidos por el servidor esclavo son los que envió el servidor maestro).

### **3.2.4. Envenenamiento de Cache**

Existen muchas variantes basadas en este tipo de ataque, donde todas ellas tienen en común que involucran Registros de Recursos cuya porción RDATA contiene información manipulada por un atacante y destinada a ser almacenada en algún Servidor Cache víctima.

Básicamente en un ataque de envenenamiento de cache, el host “víctima” puede ser redirigido a un host malicioso a través de la inyección de un registro del tipo A falsificado. Para ello el “atacante” puede hacer uso de la técnica “hombre en el medio” para capturar y modificar respuestas DNS.

La idea principal en ataques de este tipo es que un Servidor Cache no tiene forma de autenticar los datos recibidos, ante una consulta, éste aceptará cualquier respuesta que “pareciera correcta”.



Una extensión del ataque antes mencionado fue presentada (Julio 2008) por el experto en seguridad Dan Kaminsky. La principal observación de Kaminsky fue que, en lugar de “falsificar la identidad” de un registro del tipo A, un atacante podría falsificar cualquier respuesta a una consulta involucrada en un proceso de búsqueda. A través de la modificación de un registro NS, un atacante podría “capturar” un dominio entero, incluso un dominio de nivel superior como un .com.

En su forma básica, en un ataque de envenenamiento de cache, un atacante debía esperar a que una entrada almacenada en cache expirara antes de intentar reemplazarla. Sin embargo, Kaminsky demostró también que mediante la consulta a un subdominio falso no almacenado en cache, un atacante podría forzar una búsqueda. La búsqueda iría directamente para el servidor de nombres con autoridad sobre el dominio en cuestión, asumiendo que éste fue almacenado en cache. La acción anterior le da una oportunidad al atacante de falsificar una respuesta sin que haya existido una consulta legítima al subdominio falso. El servidor de nombres “víctima” tomará como válida la respuesta fraudulenta ya ésta llegó antes que la respuesta legítima que proviene del servidor de nombres con autoridad sobre el dominio.

El siguiente gráfico muestra en detalle el proceso antes descrito: [30]

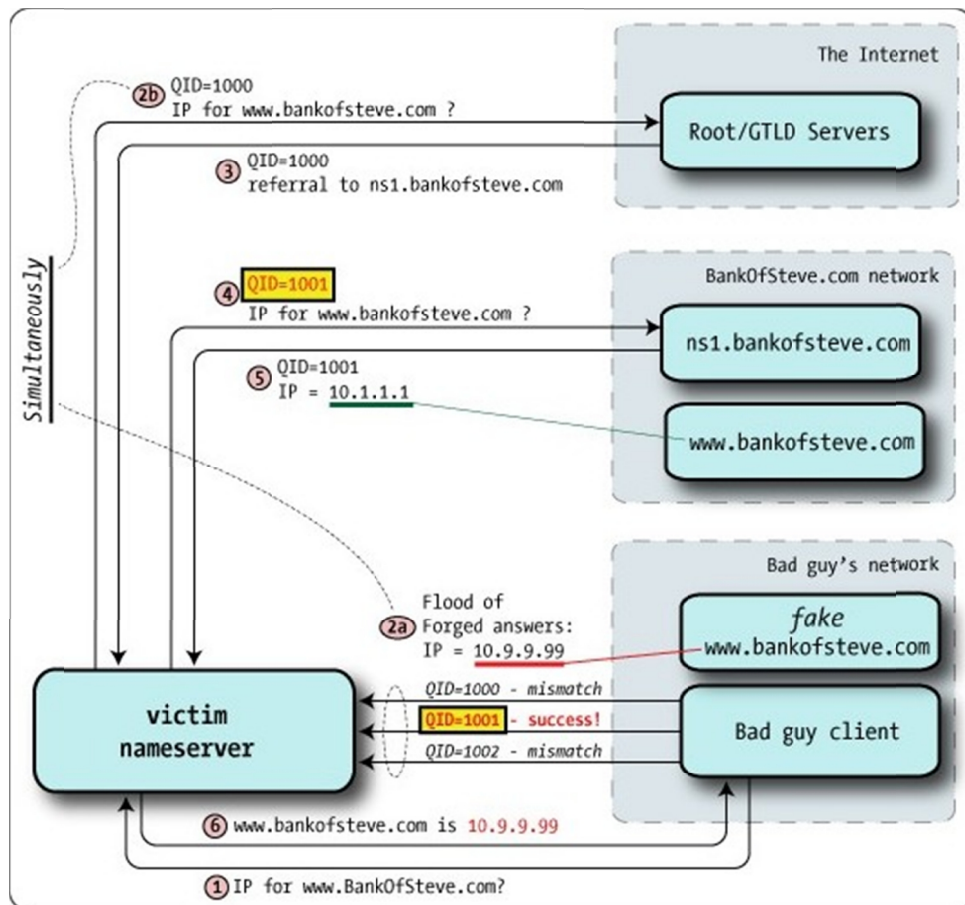


Figura 14: Ejemplo ataque envenenamiento de cache (Kaminsky).

- **1** – El atacante manda una petición DNS al servidor víctima para un nombre de host que quiere comprometer. En este ejemplo se asume que el servidor de nombres permite peticiones recursivas desde el mundo exterior.

- **2a** – Sabiendo que el servidor víctima pedirá en breve a [ns1.bankofsteve.com](http://ns1.bankofsteve.com) (tal como se lo indicarán los servidores raíz/GTLD) una dirección IP para [www.bankofsteve.com](http://www.bankofsteve.com), el atacante comienza a inundar con paquetes de respuesta DNS con el fin de que [ns1.bankofsteve.com](http://ns1.bankofsteve.com) tome por buena la respuesta fraudulenta y asigne a [www.bankofsteve.com](http://www.bankofsteve.com) una IP falsa.
- **2b & 3** – Los servidores raíz/GTLD indican que la petición debe redirigirse a [ns1.bankofsteve.com](http://ns1.bankofsteve.com).
- **4** – El servidor de nombres víctima pide a [ns1.bankofsteve.com](http://ns1.bankofsteve.com) la dirección IP de [www.bankofsteve.com](http://www.bankofsteve.com) y utiliza el query ID 1001 (uno más que la petición anterior).
- **5** – El servidor de nombres real ([ns1.bankofsteve.com](http://ns1.bankofsteve.com)) proporciona una respuesta legítima a la petición con un Query ID de 1001, pero si el atacante ha enviado una respuesta con el mismo Query ID en el paso 2a, esta respuesta legítima llega demasiado tarde y es ignorada.
- **6** – Con la dirección IP falsa (del servidor web del atacante) en el cache el servidor ha quedado comprometido y todas las peticiones al servidor DNS para [www.bankofsteve.com](http://www.bankofsteve.com) (la actual y futuras) redirigen al servidor del atacante.

Expuesta la vulnerabilidad demostrada por Kaminsky, como medida inmediata se implementaron mecanismos para lograr que las respuestas sean más difíciles de predecir a través del agregado de aleatoriedad al campo Query ID y ampliar el espacio de valores para tal campo.

Es de destacar que incluso con estas mejoras, todavía es posible llevar a cabo ataques de envenenamiento de caché DNS, por lo que es necesario algún tipo de garantía de autenticación proporcionado por mecanismos criptográficos. Precisamente es DNSSEC el que fuera diseñado para proporcionar estas garantías.

### 3.2.5. Otros tipos de ataques

Las técnicas utilizadas en los tipos de ataques vistos hasta ahora están dirigidas al protocolo DNS de manera directa. Sin embargo el sistema DNS también puede verse comprometido si se atacara sobre aspectos como la disponibilidad.

Ataques a la disponibilidad son los clasificados como Denegación de Servicio (DoS) y específicamente el ataque conocido como Inundación UDP (*UDP floods*). Por lo tanto, un Servidor Autoritativo o de Cache víctima de éste tipo de ataque quedaría imposibilitado de responder a consultas de clientes legítimos.

Un ataque por inundación UDP está basado justamente en la naturaleza del protocolo UDP, ya que no existe un proceso de sincronización en la comunicación (protocolo sin estado), lo que permite poder alterar datos de encabezado (por ejemplo dirección IP de origen).

Es así que un atacante podría enviar una gran cantidad de peticiones con una dirección IP falsificada (*spoof*) a un Servidor DNS que permitiera recursión, el Servidor procesaría estas peticiones como si éstas fueran válidas enviando una respuesta al sistema al cual se quiere atacar, es decir al sistema víctima. Cuando estas peticiones alcanzan un volumen importante pueden inundar al sistema víctima de repuestas del DNS a las peticiones que se le realizan. Este ataque se vale también de errores en la

configuración en un Servidor DNS (Recursión habilitada) y es llamado de amplificación puesto que los DNS al reflejar el ataque, potencian el nivel de éste hacia un blanco en específico a causa de que las respuestas del servidor de nombres son de un tamaño considerablemente mayor que las peticiones que se le realizan causando con esto, en ocasiones, la caída del servicio y con ello la negación del mismo. La siguiente figura ilustra las acciones antes mencionadas:

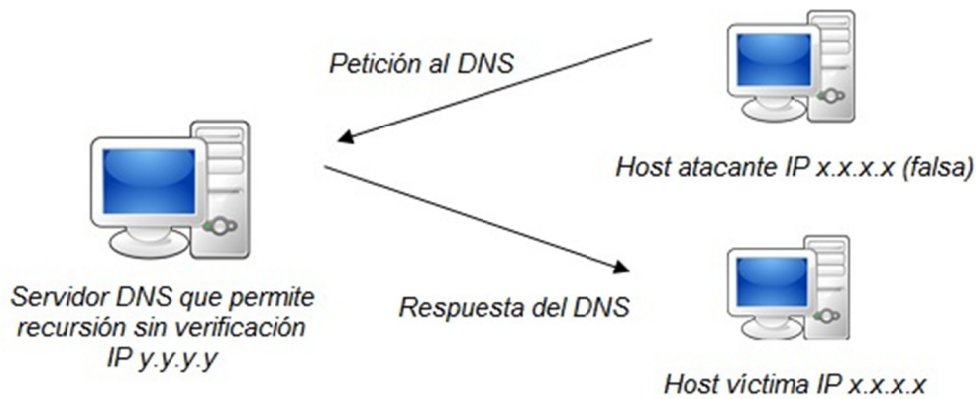


Figura 15: Ejemplo ataque DoS a DNS.

Expuestas las características de la amenaza, surge la siguiente lista de recomendaciones a fin de mitigar la posibilidad de ser víctima de una práctica maliciosa de este tipo:

1. Una práctica de configuración ampliamente recomendable sería el permitir la realización de consultas recursivas solo a un grupo específico de direcciones IP. El documento publicado por la IETF (*BCP 38 – Defeating Denial of Service*, [31]) introduce un conjunto de buenas prácticas destinado a ISPs, el cual puede ser consultado para ampliar detalles.
2. Por otra parte una regla que debe aplicarse para minimizar el riesgo de un ataque de amplificación es la del mínimo privilegio, esto es, un Servidor DNS no tiene que permitir la recursión a nadie más que a los host estrictamente indispensables. Si un Servidor DNS no tiene la necesidad de realizar consultas recursivas, puesto que no tiene un dominio de menor jerarquía al cual permitirle la recursión, no existe la necesidad de permitirla o, en el caso de que tenga ciertos dominios de jerarquía más baja, se debe permitir la recursión solo a los hosts en los cuales se confíe.
3. Por último y como regla general, es altamente recomendable consultar periódicamente reportes de seguridad, tales como los publicados por la *Internet Systems Consortium (ISC)*, [32] y fundamentalmente mantener actualizado el software de servicios DNS (ej: BIND).

# 4. Criptografía en el contexto del Sistema DNS

## 4.1. Introducción

El presente capítulo tiene por objeto introducir de manera general los conceptos fundamentales sobre las técnicas criptográficas utilizadas por las extensiones de seguridad para el Sistema DNS. Se revisarán aspectos como funcionalidades y limitaciones asociados con cada técnica, dejando de lado los procesos matemáticos utilizados ya que no se consideran necesarios en ésta instancia.

Los procesos criptográficos pueden ser usados para tres propósitos fundamentales:

- **Confidencialidad:** Solo las partes intervinientes en una comunicación, pueden comprender los mensajes intercambiados. Un tercero que entra en posesión de la información intercambiada entre el remitente y el destinatario no es capaz de extraer cualquier contenido inteligible.
- **Autenticación:** Propiedad que permite identificar a la fuente que genera la información. Lo que permite garantizar que los datos recibidos provienen de una fuente conocida y por lo tanto son confiables.
- **Integridad:** Es la propiedad que busca mantener los datos libres de modificaciones no autorizadas, lo que garantiza que la información recibida por una de las partes, se encuentra tal cual fue generada, sin ser manipulada o alterada por terceros o procesos no autorizados.

En el contexto de los estándares de DNS, solo la autenticación y la integridad de los datos son de interés. En el caso de que se requiera proveer confidencialidad, se asume que ésta podría ser provista por procesos de comunicación tales como *SSL* o su sucesor *TLS*.

## 4.2. Criptografía Simétrica

Los algoritmos de encriptación simétrica, también llamados de clave única, o de secreto compartido, hacen uso de una única clave para encriptar y desencriptar los datos. Las partes intervinientes en la comunicación deben intercambiar la clave única que usarán para cifrar la información, utilizando un canal seguro.

Las limitaciones que se observan en los sistemas de secreto compartido son dos fundamentalmente. La primera es que la clave debe ser distribuida de manera segura, usando un proceso de gestión de claves, al que se considera un proceso no trivial. En segundo lugar, una vez que la clave ha sido distribuida entre las partes intervinientes, la responsabilidad de mantener la clave de manera segura, recae en todas y cada una de ellas.

Ejemplos de algoritmos de clave simétrica son DES, AES, IDEA y RC4, con tamaños de clave de 64, 128 o 192 bits.

Los algoritmos de clave compartida son usados en aquellas operaciones que requieran autenticación entre las partes (por ejemplo: servidores maestros y esclavos) lo que involucra operaciones de *Transaction Signature (TSIG)*, [9]. El proceso de distribución de claves no está definido en los estándares de DNS y puede llevarse a cabo mediante mecanismos acordados previamente tales como: vía telefónica, fax, email seguro, etc.)

### **4.3. Criptografía Asimétrica**

Los algoritmos de cifrado asimétrico usan un par de claves, son llamados por lo general “sistemas criptográficos de clave pública”. En estos sistemas, los datos son encriptados con una clave dada y solo pueden ser descifrados con su par correspondiente, se dice que dado una clave, es computacionalmente inviable derivar su par a partir de la primera. Estos sistemas trabajan de manera tal que la clave que va a estar disponible se denomina “clave pública”, mientras que su par se mantiene como un secreto y se denomina “clave privada”. Los sistemas de claves “pública/privada” más ampliamente usados son RSA y de Curva elíptica, con longitudes de clave de 1024 bits, 2048 bits y superiores.

Se puede considerar como una limitación existente en los sistemas de clave pública, el conocimiento o confianza de la clave pública utilizada en una comunicación, dado que no existe la manera de verificar si es realmente la clave pública de la entidad original y la misma no ha sido suplantada. Para tal caso se hace necesario el uso del método llamado Infraestructura de Clave Pública (*PKI*), una entidad que administra de manera segura las claves públicas y en la que las partes confían. De esta forma si se necesitara la clave pública de X, la misma puede ser solicitada a una *PKI*, con lo que se asegura que realmente la clave pública pertenece a X.

### **4.4. Digesto de mensajes**

Como se ha descrito anteriormente, para proveer integridad a los datos, el mensaje deberá ser encriptado y es entonces que, solo podrá ser descifrado por aquel que posea la clave única (sistema simétrico) o la clave pública (sistema asimétrico). Sin embargo los sistemas de encriptación hacen uso de funciones matemáticas complejas, lo que conlleva a un uso elevado de procesador para llevar a cabo sus tareas. Es así que para reducir la carga de procesamiento es posible el uso de funciones hash. Las funciones de hash, toman como entrada una cadena de bits o texto de longitud variable llamada pre-imagen y la convierten a una cadena de salida de longitud fija llamada resumen o digesto.

Los mensajes se envían en texto plano junto a su digesto, el receptor aplica el algoritmo de hash al texto plano y el resultado se compara con el digesto recibido. Si ambos coinciden, implica que el mensaje no ha sido alterado en el camino. Algunos de los algoritmos de funciones hash más comunes son MD5, SHA-1 y SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)

## **4.5. Código de Autenticación de Mensaje**

En los sistemas simétricos, es posible autenticar el emisor así como asegurar la integridad mediante el uso de un Código de Autenticación de Mensaje (*MAC*), el cual es generado por un cierta función *C* que responde al siguiente formato:  $MAC=C(K,M)$ , donde *M* es el mensaje a encriptar y *K* es la clave secreta compartida entre Emisor y Receptor. Por lo tanto la clave *K* permite autenticar al Emisor y el Digesto obtenido al aplicar la función *C* asegura la integridad del mensaje *M*. HMAC-MD5, HMAC-SHA-1 y HMAC-SHA-256 son algunos de los algoritmos más conocidos para el cálculo de MACs.

## **4.6. Firma Digital**

En los sistemas asimétricos, la autenticación e integridad de datos se garantiza mediante el uso de lo que se conoce como firma digital. Al igual que en los métodos anteriores, se obtiene el digesto o resumen del mensaje a enviar, lo que asegura la integridad de los datos. El proceso continúa con la encriptación del digesto, por parte del emisor, para lo cual hace uso de la clave privada. El mensaje en texto plano, así como el digesto encriptado son enviados y ya en manos del receptor, el mismo procede en primer lugar a desencriptar el digesto haciendo uso de la clave pública del emisor, luego aplica el algoritmo de hash al mensaje recibido, por lo que si los valores coinciden (digesto calculado y digesto recibido), se asume que la autenticidad y la integridad están garantizadas. Algunos de los algoritmos de firma digital más usados son: RSA-MD5, RSA-SHA-1, RSA-SHA-256 y DSA con longitudes de claves de 1024 bits, 2048 bits y superiores.

## **4.7. Uso de la criptografía en DNS**

Habiendo expuesto los aspectos fundamentales de la criptografía en términos del contexto DNS, se exponen a continuación las particularidades de la implementación de tales aspectos en el mundo DNSSEC.

Los estándares de seguridad en DNS hacen uso de la criptografía de dos maneras distintas: seguridad en las transacciones, como la que se usa en las transferencia de zonas y actualizaciones dinámicas, donde se hace uso de un modelo de seguridad “punto a punto” en el cual, ambas partes involucradas en la transacción confían la una en la otra. Las partes intercambian información que permiten autenticar el origen y la integridad de los datos. Por ejemplo, *TSIG* (criptografía simétrica) y *SIG(0)* (criptografía asimétrica), [33], son métodos usados para realizar éstas validaciones.

Por otra parte, en una arquitectura basada en DNSSEC, la seguridad se presenta bajo el paradigma Cliente/Servidor, lo que permite a un receptor, validar el origen y la integridad de los datos que recibe en respuesta a una consulta. En base a lo anteriormente expuesto, es crítico para el correcto funcionamiento del sistema, asegurar que el origen de los datos es realmente de quien dicen ser, lo que normalmente se logra en presencia de una Infraestructura de Clave Pública (*PKI*), mediante la cual, las partes involucradas en la comunicación confían en una Autoridad Certificante (*CA*) lo que permite garantizar la autenticación e integridad de los datos. Los aspectos de seguridad

cubiertos en DNSSEC no se basan en una PKI, en su lugar, se hace uso de una jerarquía o “cadena de confianza”, basada en la delegación de nombres DNS.

La cadena de confianza comienza en la raíz del sistema de nombres de dominio o bien en un “Punto de Entrada de Seguridad” (*SEP*), ambos representan un punto de partida para la comprobación de la cadena cuando se quiere validar una respuesta DNS. La autenticidad de cada link de la cadena, con excepción del punto de partida, es verificada por el dominio anterior. El proceso de creación de la cadena de confianza, así como su comprobación es explicado en detalle en el capítulo correspondiente a DNSSEC.

#### **4.7.1. Autenticación e Integridad en Transferencias de Zonas**

*Transaction Signature (TSIG)*: Definido en el RFC 2845, es un mecanismo que hace uso de una clave única mediante la implementación de Códigos de Autenticación de Mensaje entre servidores maestros y esclavos. La distribución de la clave hacia los servidores esclavos debe hacerse de manera segura utilizando medios como email seguro, fax o correo postal, se recomienda para la misma, periodos de actualización de entre 30 y 60 días. En caso de presencia de más de un servidor esclavo, se deben utilizar pares de claves (maestro-esclavo) diferentes, ya que al verse comprometida alguna de las claves, solo se deshabilitarían temporalmente las transferencias entre las partes afectadas, sin involucrar al resto de las transferencias.

Registro de Recurso *SIG(0)*: Definido en el RFC 2931, este registro hace uso de criptografía de clave pública para la generación de firmas digitales que permiten autenticar a las partes involucradas, así como asegurar la integridad de los datos involucrados en cada transacción que incluyen transferencias de zona. Sin embargo a la fecha no existen herramientas disponibles para la versión BIND 9 DNS que soporten *SIG(0)* en transferencias de zonas. Sí se puede hacer uso de *SIG(0)* en implementaciones de DNS Dinámico

# 5. DNSSEC

## 5.1. Introducción

Según se describe en el RFC 4033: *DNS Security Introduction and Requirements* [11], las Extensiones de Seguridad para DNS (DNSSEC) proveen autenticación del origen e integridad de los datos intercambiados a través del protocolo DNS. Las mejoras que ofrece DNSSEC radican principalmente en el uso de una jerarquía de “firmas criptográficas” que permite proteger el flujo de información intercambiado entre Servidores Autoritativos, Servidores DNS Recursivos y Clientes DNS.

Por otro lado, en [12], el cual funciona como un repositorio central de documentación relacionada con el tema, se define a DNSSEC como: Un conjunto de extensiones para DNS, las cuales proveen: a) Autenticación del origen de los datos DNS, b) Integridad de datos, y c) Autenticación de la negación de existencia.

A continuación se describen brevemente los propósitos antes citados

### 5.1.1. Autenticación del origen de los datos

Cuando se envía una carta a alguien, ésta usualmente incluye la firma de quien la envía, este proceso le brinda al receptor, una pequeña prueba de que la carta fue realmente enviada por la persona que figura en los datos del remitente. Desde el punto de vista del entorno DNS, esta prueba tiene un gran valor, porque un Cliente DNS puede confiar en que un Servidor de Nombres Autoritativo tiene realmente autoridad sobre una cierta zona, lo que garantiza que los datos que reciban de dicha fuente fueron realmente originados por la misma.

### 5.1.2. Integridad de los datos

Continuando con el ejemplo del envío de una carta, es posible que en el proceso de transporte de la misma, alguien la intercepte con fines mal intencionados (podría reescribir, borrar o bien agregar algunas líneas), resultando tales alteraciones para el receptor, en algo desastroso. Llevado lo anterior al entorno DNS, también resulta en algo desastroso, ya que en un flujo de datos intercambiados en una arquitectura DNS, alguien podría capturar y alterar una consulta o una respuesta DNS, sin que las partes involucradas se enteraran. Por lo tanto DNSSEC debe ser capaz de detectar tales modificaciones no autorizadas, por lo que se dice que DNSSEC debe garantizar la integridad de los datos.

### 5.1.3. Autenticación de la Negación de Existencia

En una arquitectura de DNS tradicional, si se realiza una consulta por registros que no existen, el Servidor consultado responderá con un registro *SOA* de la zona adjunta, junto con un código de error indicando el error específico que ha ocurrido. Esta funcionalidad podría ser usada para un ataque del tipo “*replay attack*”, el cual repite una respuesta de “no existencia” que fuera capturada anteriormente, y de esta manera se podría falsificar la “existencia” de un *host* como “no existente”.



Por todo lo anterior, y con la finalidad de proporcionar la negación autenticada de registros que no existen para evitar el ataque mencionado, DNSSEC incorpora un nuevo tipo de registro, el registro *NSEC*, [34] (o su última versión *NSEC3*, [35]).

El diseño de DNSSEC, se inició a mediados de los 90, motivado por el creciente reconocimiento de los peligros de envenenamiento de *cache* DNS. La primera versión de estas extensiones de seguridad fueron oficialmente publicadas en Enero de 1997 en el RFC 2065, seguida de una versión mejorada que fuera publicada en el RFC 2535 (Marzo de 1999). Luego de experiencias en pruebas piloto de implementación, la versión actual fue presentada en los RFC 4033, 4034 y 4035 (Marzo 2005).

## 5.2. Conceptos generales sobre DNSSEC

A fin de lograr los propósitos antes descriptos, DNSSEC hace uso de cuatro nuevos Registros de Recursos y dos bits (CD = Comprobación Desactivada y AD = Dato Autenticado) presentes en el encabezado de un paquete DNS. Por otro lado, un Servidor *Resolver* que realiza una consulta, utiliza los mecanismos de extensión para DNS (*EDNS0*) y activa el bit DO (DNSSEC OK), presente en el Registro de Recurso Opcional (RR OPT). Por lo tanto, mediante la activación del bit anterior un *Resolver* está indicando que tiene la capacidad de procesar información relacionada con DNSSEC. Por otro lado, un Servidor de Nombres que recibe solicitudes en las cuales el bit DO no estuviera presente, no responde con Registros de Recursos relacionados con DNSSEC, con lo que se contribuye a mejorar el rendimiento de DNS, ya que evita tener que retornar información que posteriormente no será procesada.

Los Registros de Recursos para DNSSEC son:

- **DNSKEY**: Registro de Recurso habilitado para almacenar claves públicas, que posteriormente serán usadas por DNSSEC en procesos de autenticación.
- **RRSIG**: Contiene la firma para un conjunto de Registros de Recursos (RRset) con un nombre particular, clase y tipo. El registro RRSIG se genera en el proceso de firmado de una zona utilizando la clave privada y cuyo par (clave pública) es almacenada en el registro DNSKEY.
- **NSEC**: Permite validar la estructura de una zona y los Registros de Recurso que esta contiene.
- **DS**: Permite crear una cadena de confianza o de autoridad de una zona padre firmada, hacia una zona hija firmada. DS está relacionado con el Registro DNSKEY, ya que contiene un resumen (*hash* o *digesto*) de la clave (*KSK*) almacenada en éste último.

Los bits AD y CD

EL bit AD solo tiene sentido en una respuesta DNS, si el mismo almacena el valor 1, está indicando que todos los RRsets en la sección "*answer*" y toda información pertinente a Registro de Recursos de Respuestas Negativas en la sección "*authority*" son auténticas, es decir que la validación ha sido exitosa. Es importante destacar que regularmente los Servidores de Nombres Autoritativos, nunca envían respuestas con

este bit activo, dado que éstos no verifican las firmas que envían como respuestas. Lo que implica que el bit AD solo es activado por aquellos servidores que han comprobado firmas, tales como los Servidores *Resolver*.

El bit CD lo activa el Cliente que realiza la consulta para indicar al Servidor de Nombres que está dispuesto a aceptar datos que no hayan sido validados, por lo tanto el segundo simplemente devuelve una respuesta aún cuando la validación no haya sido exitosa.

Por otro lado se presenta el caso de un Servidor *Resolver* con soporte para DNSSEC pero que no realiza validaciones, puede en principio confiar en datos ya validados, si es que se encuentra en una cadena de confianza. Sin embargo, el mejor de los casos se presenta en el uso de *stub resolvers* que puedan realizar la validación criptográfica y en consecuencia, al enviar una consulta, el bit CD se activa a 1. Lo anterior provee un esquema de DNS seguro de extremo a extremo, sin la necesidad de servidores intermedios en los que se deba confiar.

### **5.2.1. Puntos de Entrada Seguros (SEPs)**

En una estructura de DNS con soporte para las extensiones de seguridad, tanto el servidor con autoridad sobre una zona (maestro o esclavo), así como el servidor *Resolver* que realiza las consultas, deben estar configurados con soporte para DNSSEC. La resolución DNSSEC comprueba si la consulta corresponde a una zona asegurada con DNSSEC. La respuesta es positiva cuando el destino se encuentra en zona segura, es decir que el archivo de zona esta criptográficamente firmado, lo que garantiza una respuesta genuina al *Resolver* que realizó la consulta. A los nodos superiores de estas estructuras se los conoce como Puntos de Entrada Seguros (*SEP*), por lo tanto, la lista de *SEPs* es el equivalente en la práctica de los proveedores de certificados CA de los navegadores web, [36].

En términos generales, DNSSEC se asegura de que tiene un solo *SEP* apuntando a la raíz de la zona DNS. Una cadena de confianza enlaza la clave de la firma con el resto de zonas que están por debajo en la jerarquía. Esto permite a los clientes de DNSSEC validar las firmas.

### **5.2.2. Compatibilidad con el esquema tradicional de DNS**

DNSSEC usa los mismos mecanismos de acceso que el DNS tradicional. Debido a que en un proceso de consultase utilizan RRs, el sistema es compatible retrospectivamente. La seguridad adicional la proporciona el cliente con capacidad para DNSSEC al validar las firmas de los RRs. Si una respuesta no está debidamente firmada, se ignora.

Se trata de un procedimiento muy seguro, ya que nunca se insta al usuario a aceptar una respuesta potencialmente comprometida. De todos modos, el usuario debe acostumbrarse a las respuestas *NXDOMAIN*, que significan que el dominio no existe.

Cuando la respuesta no proviene de una zona segura, el *Resolver* recurre a los métodos tradicionales para la resolución, devolviendo la respuesta oportuna al cliente que hace la petición. En éste último punto existen extensiones de seguridad para navegadores web,

tales como Mozilla Firefox que permiten comprobar la existencia y validez de registros DNS firmados para los dominios consultados.

En el proceso de comprobación de una respuesta DNS para un determinado conjunto de Registros de Recurso (RRset), un cliente con características DNSSEC, puede obtener la firma digital relacionada con los Registros de Recurso y comprobar esta firma usando la clave pública publicada por el administrador de la zona, comparando el valor del digesto calculado a nivel local del RRset. Por otra parte, el cliente puede validar la clave pública del administrador de la zona usando una jerarquía de firmas que conduce a un punto de confianza (*SEP*). Si todas estas comprobaciones son exitosas, entonces el cliente tiene cierta confianza en que la respuesta DNS fue completa y auténtica.

La implementación de DNSSEC implica diferentes acciones a diferentes niveles de administración. Para un administrador de zonas DNS, DNSSEC comprende el proceso de firma de todos los RRsets con una clave privada, la publicación de estas firmas para cada RRset en el archivo de la zona, y también la publicación de la clave pública del archivo de zona. Además, el administrador de la zona tiene que asegurarse de que la clave pública de la zona está firmada por el administrador de la zona padre y publicada en la zona padre. Para un administrador de una zona padre, DNSSEC también incluye tareas de validación de claves públicas de zonas hijas, mediante la utilización del registro DS (Delegación de Firma).

### 5.2.3. Cadenas de confianza

El proceso de construcción de una cadena confianza es fundamental para la rápida implementación de DNSSEC en una jerarquía DNS, ya que sin ésta característica, cada Servidor Resolver configurado con DNSSEC, debería tener un punto de anclaje seguro o *SEP* por cada dominio seguro en Internet, lo que claramente no permitiría un despliegue a escala global de tales extensiones de seguridad.

La siguiente ilustración permite observar los procesos involucrados en la creación de la cadena de confianza:

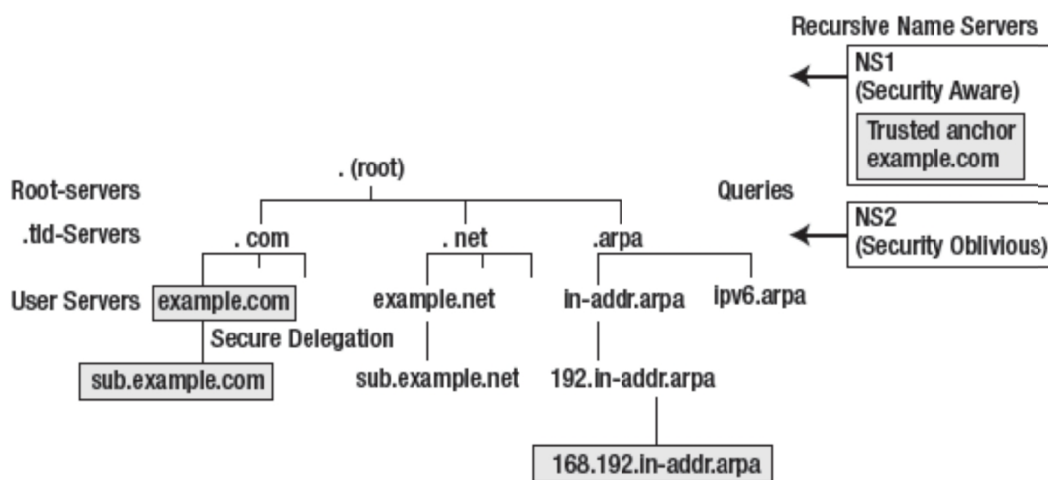


Figura 16: Creación cadena de confianza DNSSEC.

Tanto el dominio *example.com* como *sub.example.com* se encuentran asegurados, es decir que para que pueda ocurrir una delegación segura es requisito previo haber asegurado la zona hija (*sub.example.com*). El punto de anclaje seguro para *example.com* cubre las zonas seguras que son delegadas a partir de él, a través de una delegación segura creando una cadena de confianza provista por el uso del Registro de Recurso DS. Una cadena de confianza puede ser construida tanto hacia arriba como hacia abajo en una jerarquía DNS, por lo que si el dominio de nivel superior *.com* fue asegurado, el dominio *example.com* puede unirse a la cadena.

Continuando con el ejemplo de la figura, el servidor *Resolver NS1* (configurado con DNSSEC), podría ahora requerir un nuevo anclaje seguro (*SEP*) para el dominio *.com*, y este único *SEP* cubriría ahora los dominios *.com*, *example.com*, así como *sub.example.com*.

Desde Julio 2010 la zona raíz se encuentra firmada y a la fecha, son 1386 los dominios de nivel superior que fueron firmados, de los cuales, 1378 tienen puntos de anclaje seguros publicados como registros DS en la zona antes mencionada [13].

#### **5.2.4. Concepto de Clave de Zona (ZSK) y Clave de Claves (KSK)**

Según se describe en RFC 4641, las claves criptográficas usadas para el firmado de registros asociados a un dominio pueden ser de dos tipos, *ZSK* (*Zone Signing Key*) o *KSK* (*Key Signing Key*), donde la primera tiene por función la de proteger los Registros de Recursos individuales de una Zona dada, mientras que la *KSK* se encarga de proteger la *ZSK*. Operacionalmente se almacenan en un registro DNSKEY y se distinguen mediante el bit llamado SEP, [14].

##### ***Motivaciones para un uso separado de Claves***

No se requiere interacción entre una Zona Padre y una Zona Hija cuando las Claves de Zona ya han sido verificadas y se encuentran actualizadas.

La *KSK* puede configurarse con longitudes de clave mayores, lo que la convierte en una clave de mayor fortaleza. Operacionalmente tiene poco impacto en consumo de recursos, ya que solo se usa para el firmado de una pequeña porción de datos de una zona dada.

Dado que la *KSK* sólo se utiliza para firmar un conjunto de claves, ésta puede actualizarse con menos frecuencia que otros datos en la Zona y ser almacenada en una localización diferente de la *ZSK*.

En resumen, para la mayoría de los métodos de administración de claves y firmado de zonas, la clave *KSK* es usada con menor frecuencia que la clave *ZSK*. Una vez que un conjunto de claves es firmado con una clave *KSK*, todas las claves del conjunto pueden ser usadas como claves *ZSK*, si alguna de ellas fuera comprometida, ésta simplemente se elimina del conjunto de claves y el nuevo conjunto de claves debe refirmarse nuevamente con la clave *KSK*.

Con respecto a lo citado en el párrafo anterior, cabe destacar que verdaderamente se cumple según estudios comparativos presentados en reunión de la IETF (Marzo 2012), donde se destacan las siguientes observaciones, [37]:

- Todos los dominios de nivel superior (TLDs) que interactúan con los servidores raíz de la *Internet Assigned Numbers Authority (IANA)*, emplean el modelo *KSK/ZSK*.
- La afirmación de que la clave *KSK*, puede ser distinguida de la clave *ZSK*, observando el bit SEP en el Registro de Recurso DNSKEY, es verdadera.
- Sobre 80 archivos de zonas firmados que fueron observados, solo 5 usan la misma longitud de clave, tanto para la *KSK* como para la *ZSK*, el resto tiene configurada la *KSK* con una longitud de clave mayor sobre la *ZSK*.

### **Consideraciones sobre revocación de Claves**

Otro aspecto importante a tener en cuenta es el relacionado al reemplazo de las claves o lo que se conoce como “*key rollover*”, ya que se considera que las claves no tienen una fecha de expiración explícita, sino que éstas se deben reemplazar cada cierto periodo de tiempo, dependiendo del algoritmo usado y longitudes de clave, lo que permitirá evitar que un atacante pudiera deducir cualquier tipo de información a partir de las mismas. Cuando una clave de zona es revocada, la totalidad de la zona debe ser refirmada, incluyendo la firma para el RRset DNSKEY. Por otro lado, cuando una clave del tipo *KSK* es revocada, solo el Registro RRSIG del RRset DNSKEY debe ser refirmado, utilizando para esto ambas claves, la clave tipo *ZSK* y la nueva clave *KSK*. Además el administrador del nivel superior, debe ser notificado que una nueva clave *KSK* se ha generado en una de sus zonas hijas. Es decir que, el digesto de la nueva clave *KSK*, se debe agregar al Registro DS en la zona del nivel superior.

Un dato a tener en cuenta es que se debe contar con capacidad de procesamiento para generar las firmas criptográficas y digestos correspondientes. En el año 2003, cuando se iniciaron las pruebas de DNSSEC para la zona .nl, el proceso de habilitar las capacidades de DNSSEC demandó un tiempo de 90 minutos para un archivo de zona con tamaño igual a 40mb, aunque no se mencionó el hardware que fuera utilizado, [39-p11].

Además de la necesidad de contar con mayor capacidad de procesamiento, generar archivos de zonas con capacidad DNSSEC, requiere contar con mayor capacidad de almacenamiento, debido a la incorporación nuevos registros tales como: RRSIG, DS, NSEC o NSEC-3. Según observaciones realizadas en [40-p4], el tamaño de un archivo de zona, se incrementa en factor de 4 a 12, dependiendo del algoritmo y digesto elegido, longitud de clave, longitud de registro en texto plano, cantidad de Registros de Recursos y tipo de Registro de Recurso de Negación de Existencia.

Otro aspecto a tener en cuenta es el incremento en el uso de memoria, en estudios llevados adelante por *RIPE NCC*, [41-p1], se mostró un incremento en el uso de memoria de alrededor del 5%. Mientras que los mismos estudios demostraron un incremento del ancho de banda del 10%.

Por otro lado, en el *NIST* también se realizaron pruebas con respecto al rendimiento de DNSSEC, dando como resultado diferencias importantes con respecto a lo demostrado por *RIPE*. Por ejemplo, en el uso de memoria, considerando archivos de zonas de entre

500 y 30000 Registros de Recursos, el incremento en el uso de la misma, fue entre el 9% y 209%, para claves de longitud igual a 1024 bits.

Con respecto al ancho de banda, las pruebas llevadas a cabo por el *NIST*, muestran la diferencia en el uso del ancho de banda entre tráfico DNS estándar y tráfico con características DNSSEC, donde se observa un incremento del 437% del segundo con respecto al primero.

### **5.2.5. Delegación segura**

El primer paso en el proceso de implementación de DNSSEC, se inicia firmando los archivos de zona, usando la clave privada del sistema de encriptación asimétrico seleccionado. La clave pública correspondiente a la clave privada utilizada para firmar la zona, se publica usando el registro de recurso DNSKEY.

Una vez que una zona ha sido asegurada, ésta puede ahora agregarse a una cadena de confianza existente o bien ser usada en un proceso de delegación a un subdominio. En ambos casos, para llevar a cabo el proceso de delegación, se utiliza el registro de recurso de Delegación de Firma (RR DS). El Registro DS se sitúa en la Zona Padre de la zona que será delegada de manera segura y valida la siguiente clave en la cadena de confianza. El DS contiene un resumen o digesto de la clave KSK definida en el registro DNSKEY del dominio hijo. Por ejemplo: si el subdominio *sub.example.com* va a ser delegado de forma segura (unido a una cadena de confianza), un Registro DS contiene el digesto del Registro DNSKEY con el nombre de *sub.example.com* y el cual será agregado o almacenado en la zona del dominio *example.com*. Finalmente, se debe tener en cuenta que, una delegación segura ocurre solo si la zona padre e hija han sido firmadas, es decir, han sido aseguradas.

Como un ejemplo de implementación se puede mencionar el uso de la herramienta *dnssec-signzone*, con la cual se puede generar el Registro DS durante el proceso de firmado de la zona hija. El Registro DS, y en algunos casos, una copia de la clave KSK (contenida en un Registro DNSKEY) de la zona anterior, deben ser enviadas al propietario del dominio padre, para su inclusión en el archivo de zona, el cual ahora debe ser refirmado. En proceso de validación, un Servidor *Resolver* que recibe Registros de Recursos de un dominio seguro, puede hacer un seguimiento en la ruta de delegación para el dominio *sub.example.com*, a través de uno o más Registros DS de zonas firmadas, a partir del Punto de Entrada Seguro configurado en el *Resolver*.

La siguiente figura ilustra el proceso anteriormente descrito:

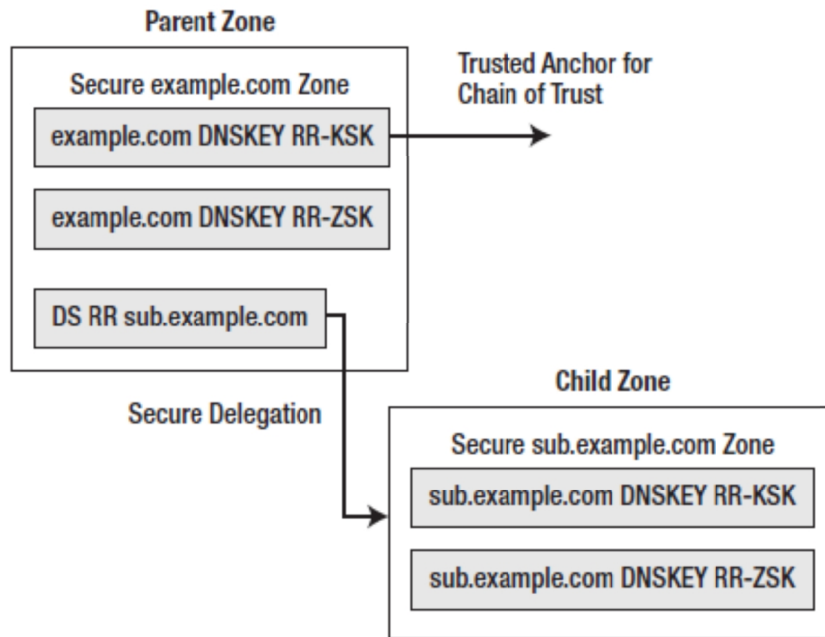


Figura 17: Proceso de delegación segura en DNSSEC.

### 5.3. Como trabaja DNSSEC

Hasta el momento, se ha descrito de manera general cual es el propósito del uso de las extensiones de seguridad para DNS y los cambios necesarios (nuevos registros de recursos, firmado de zonas, delegaciones seguras, etc.) en una arquitectura de DNS para poder poner en funcionamiento éstas extensiones. El presente apartado, tiene por objetivo exponer el flujo de datos intercambiados en un proceso de solicitud/respuesta y validaciones, necesarios para garantizar una respuesta segura.

Cuando un Servidor DNS Autoritativo recibe una consulta DNS, éste responderá agregando datos adicionales de DNSSEC. Parte de estos datos adicionales son en efecto la firma digital de los datos DNS contenidos en la respuesta. La forma de llevar a cabo este proceso es mediante el agregado del registro de recurso de firma digital (RRSIG) que permitirá a un cliente DNS autenticar la respuesta recibida. Si el servidor DNS Autoritativo no tuviera información para responder a la consulta, tal es el caso cuando el nombre de dominio no existe, entonces la respuesta incluirá un registro de recurso NSEC acompañado del registro RRSIG.

La primera tarea que debe llevar a cabo un cliente DNS con características de DNSSEC es utilizar los datos contenidos en el registro RRSIG para comprobar la validez de la respuesta DNS. Para realizar esto, el cliente DNS hace uso del algoritmo criptográfico referenciado en el registro RRSIG para generar el compendio de los datos contenidos en el RRset. La segunda tarea consiste en tomar el valor contenido en el registro RRSIG y cifrarlo con la clave pública contenida en el registro DNSKEY. Esto último resultará en descriptar el compendio contenido en el registro RRSIG. Los resultados de estas dos operaciones son comparados. Si la respuesta DNS es auténtica, entonces el compendio del RRset será igual al valor del compendio descriptado en el RRSIG.

El registro DNSKEY normalmente se proporciona como parte de la sección adicional de una respuesta DNSSEC. Si el cliente no ha validado el registro DNSKEY tras un período definido localmente, entonces el cliente también debe validar el valor del DNSKEY.

Este procedimiento de validación implica verificar el registro RRSIG asociado al registro DNSKEY, usando los mismos procedimientos descritos anteriormente. Sin embargo la validación de una clave de zona de un dominio también implica la construcción de una cadena de confianza hacia un punto de anclaje de confianza o punto de entrada seguro (*SEP*). Si esta clave de dominio no es ya un ancla de confianza, entonces el cliente debe consultar la zona padre para el registro DS de la zona hija. Una solicitud o consulta del registro DS debe retornar el valor de clave pública, el valor del registro DS, un registro RRSIG asociado al registro DS y el registro DNSKEY de la zona padre. El registro DS se valida con el registro RRSIG usando la clave pública contenida en el registro DNSKEY. Este proceso iterativo construye una cadena de confianza hacia un punto de entrada seguro. Finalizado el proceso de validaciones garantiza la validez de una respuesta DNS.

La siguiente figura ilustra la secuencia de intercambio de solicitudes/respuestas, necesarias para garantizar que la respuesta recibida al consultar por el dominio "*www.miempresa.com*" es genuina. Se asume que los Servidores Raíz soportan DNSSEC y las zonas correspondientes han sido firmadas. Del lado del Cliente se asume que el mismo posee las claves públicas de la zona raíz, obtenida de manera segura y que la misma se encuentra disponible.



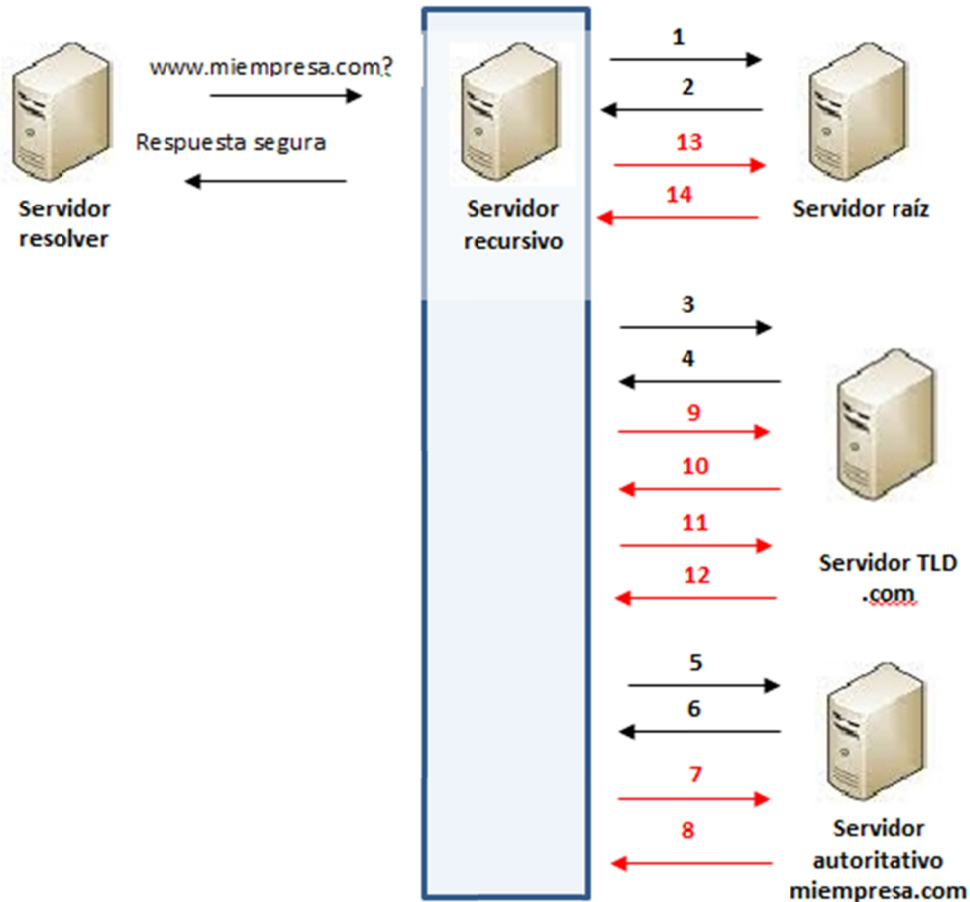


Figura 18: Intercambio de mensajes solicitudes/respuestas en DNSSEC.

1. Al recibir una consulta DNS desde un cliente, que pide resolver *www.miempresa.com*, el Servidor Recursivo, procede según el protocolo DNS estándar, para localizar el servidor de nombres para *miempresa.com*. A partir de que el mismo, es un servidor con características DNSSEC, la consulta enviada lleva activado el bit DNSSEC, indicando que desea recibir una respuesta segura, esperando que el servidor que recibe la consulta, tenga también características DNSSEC, y pueda responder como tal.
2. Siguiendo el comportamiento del protocolo DNS estándar, el Servidor Raíz devuelve como respuesta un registro NS para *nscom.com*, y su correspondiente registro de tipo A, adicionalmente se agregan a la respuesta los registros DS para la zona *.com* acompañado de la firma contenida en un registro RRSIG.
3. Análogamente al punto anterior, el Servidor Recursivo consulta al servidor de nivel superior (TLD), *nscom.com* por el nombre *www.miempresa.com*.
4. El servidor TLD *nscom*, retorna como respuesta un registro NS para el servidor con autoridad para *miempresa.com*, junto a un registro de tipo A. Son parte de la respuesta los registros DS para el mismo nombre de dominio, junto a la firma de tales registros, contenida en un registro RRSIG.

5. Ahora el Servidor Recursivo continúa el proceso, consultando ahora al Servidor Autoritativo por [www.miempresa.com](http://www.miempresa.com).
6. En este punto, se obtienen los primeros datos que permitirán validar la respuesta verificando la cadena de confianza. El Servidor Autoritativo responde con un registro de tipo A para [www.miempresa.com](http://www.miempresa.com), junto a su firma correspondiente almacenada en un registro RRSIG.
7. Para iniciar la tarea de validación de la respuesta recibida, en esta instancia el Servidor Recursivo consulta nuevamente al Servidor Autoritativo por las claves públicas (ZSK y KSK).
8. El Servidor Autoritativo devuelve los registros DNSKEY que almacenan las claves ZSK y KSK, junto a los registro RRSIG que contienen la firma para la zona *miempresa.com* y la firma del registro DNSKEY que contiene la clave ZSK, generada con la KSK anterior. En este punto hacemos un alto para observar un aspecto relevante del proceso de validación. La pregunta es; ¿cómo confiar en que el Servidor Autoritativo que acaba de responder con firmas y las correspondientes claves públicas para validarlas es genuino? La respuesta está en consultar a alguien más que tenga confianza en este Servidor Autoritativo. Esta delegación de confianza se logra a partir del envío de cierta información criptográfica (Registro DS que contiene un digesto de la clave KSK para *miempresa.com*) al responsable de la zona padre, es decir a *nsc.com*.
9. Es ahora cuando el Servidor Recursivo consulta al servidor de nivel superior *nsc.com* por el registro DS para *miempresa.com*, lo cual le permitirá continuar con el proceso de validación de la cadena de confianza.
10. El Servidor *nsc.com* responde con el registro DS consultado, más un registro RRSIG que contiene la firma del primero.
11. En esta instancia el Servidor Recursivo necesita las claves públicas de *nsc.com* que le permitan verificar la integridad de lo recibido en el punto 10, por lo que consulta nuevamente a *nsc.com* por los registros DNSKEY para la zona *com*.
12. Análogo al punto 8, el servidor *nsc.com*, retorna los registros DNSKEY correspondientes, junto a sus firmas contenidas en los RRSIG, pero en este caso para la zona *com*.
13. Siguiendo con el proceso de validación de la cadena de confianza, en esta instancia el Servidor Recursivo consulta al Servidor Raíz por el registro DS para la zona *com*.
14. El Servidor Raíz responde entonces con el registro DS y su firma correspondiente almacenada en el RRSIG.

Continuando con la descripción del ejemplo propuesto, si el proceso de validación de la cadena de confianza (pasos 7 a 14) fue exitosa, el Servidor Recursivo puede comprobar la firma para el Registro DNSKEY, firmado con la clave KSK que ya fuera autenticada. Si la comprobación es exitosa, la clave de zona se puede considerar autentica también.

Esto implica que ahora la firma para el registro tipo A puede ser verificada, de cumplirse esta última verificación, se puede asegurar que la respuesta enviada al resolver es una respuesta segura.

Del análisis anterior se observan los siguientes aspectos importantes, el primero relacionado con ocurrencia en serie de las consultas y respuestas, y el segundo, es la ocurrencia de dos etapas bien definidas, siendo la primera en un ciclo descendente correspondiente a la resolución DNS, mientras que la segunda muestra un ciclo ascendente y corresponde a la validación DNSSEC, según se muestra en la siguiente figura.



Figura 19: Proceso de consulta y validación DNSSEC

### 5.3.1. Proceso de generación y verificación de firmas DNSSEC

En base al ejemplo anterior, observemos ahora las tareas que realizan los servidores responsables del firmado de zonas, tales como los de nivel superior y principalmente los autoritativos. Mientras que para el proceso de verificación de firmas, se expondrán las tareas que realiza un Servidor Recursivo, recordando que DNSSEC hace uso de la criptografía de clave pública para la autenticación e integridad de los datos intercambiados.

En un servidor DNS Autoritativo, a los conjuntos de registros de recursos de una zona, se les aplica una función de *hashing*, seguidamente, el compendio obtenido, es encriptado con su correspondiente clave privada, obteniendo así la firma digital. La siguiente figura ilustra en términos generales este proceso:

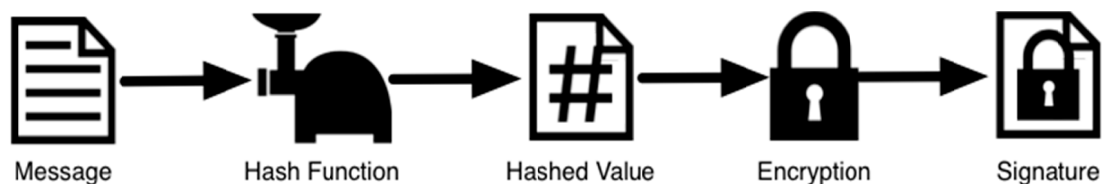


Figura 20: Proceso de generación de firma DNSSEC

Cuando un Servidor Recursivo, consulta por un registro de recurso, éste recibe, la respuesta en texto plano, así como su firma digital. El Servidor Recursivo, conoce la función de *hashing* utilizada, ya que la misma forma parte de la respuesta recibida. Es así que, a partir del mensaje en texto plano, le aplica dicha función de *hash*, obteniendo así un cierto valor “x”. El recursivo obtiene también, la clave pública, (presente en un registro DNSKEY), necesaria para descryptar el valor de *hashing* que recibió como

respuesta por parte del Servidor Autoritativo, por lo que luego de proceder a descryptar la respuesta, obtiene un cierto valor, al que llama “y”. Si los valores “x” e “y” son idénticos, y se verifica además que cumplen con llevar marcas de tiempo válidas, la respuesta pudo ser verificada, lo que significa que la misma proviene de un Servidor Autoritativo autenticado y que los datos contenidos en dicha respuesta no han sufrido alteraciones en el camino, por lo tanto, los mismos están íntegros. La siguiente figura ilustra el proceso antes descrito:

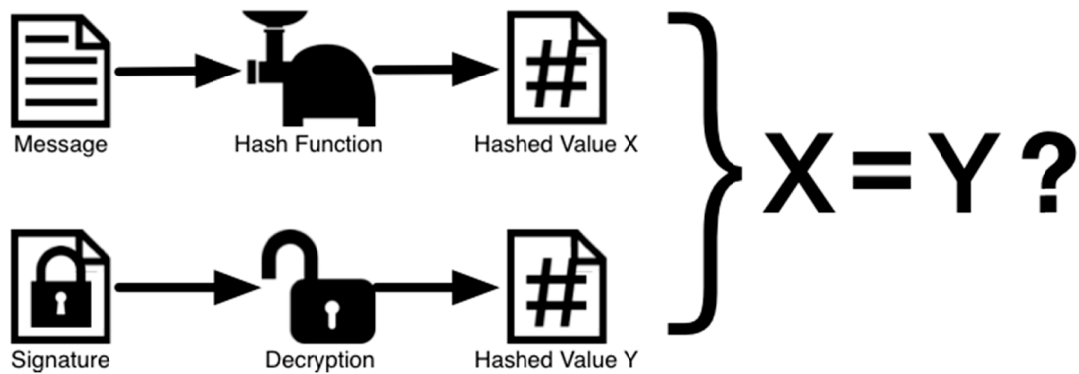


Figura 21: Proceso de verificación de firma DNSSEC

#### 5.4. Contexto a nivel mundial

Uno de los objetivos propuestos en el presente trabajo es el de acompañar el despliegue de DNSSEC a nivel mundial, por lo cual, se considera importante exponer cual es la situación a la fecha en cuanto a la implementación en los distintos niveles de la jerarquía DNS, como ha evolucionado desde la fecha en que se anunció el firmado de la zona raíz y cuáles son los procesos hasta alcanzar la implementación final para un nivel de dicha jerarquía.

Con respecto a los dominios de nivel superior, el primer aspecto que se debe considerar son las etapas que se deben cumplimentar hasta alcanzar el despliegue final de tales características de seguridad. Según se describe en *Internet Society – Deploy360 Programme*,[48], las cinco etapas son:

- Experimental: La entidad registrante responsable para un dominio de nivel superior se encuentra realizando pruebas de manera interna o trabajando sobre un programa piloto. Tal entidad anuncia el inicio de esta etapa, a través de participación en listas de correo de la temática, presentaciones en conferencias o eventos, participación en workshops de entrenamiento, publicaciones de artículos.
- Anuncio público: La entidad responsable, hace una declaración pública, comprometiéndose a desplegar DNSSEC y firmar el TLD. Esta novedad puede hacerse pública mediante publicación como noticia, anuncio en conferencias, o simplemente enviando un email reportando esta novedad a la *Internet Society*.

- **Parcial:** En esta etapa, el dominio de nivel superior hace público sus registros de recursos firmados con DNSSEC, pero aún el registro de delegación de firma, (DS) no se publicó en la zona raíz, es decir que aún no se ha unido a la cadena de confianza global.
- **DS en zona raíz:** Cuando la zona raíz publica el registro DS para un dominio de nivel superior, entonces éste ya se encuentra unido a una cadena de confianza global, en esta instancia los dominios de segundo nivel, por debajo del TLD pueden realizar validaciones DNSSEC y verificar firmas hasta la zona raíz.
- **Operacional:** La etapa final para el despliegue de DNSSEC se concreta cuando la entidad registrante para el TLD se encuentra en condiciones de aceptar delegación segura de dominios de nivel inferior. No existe una manera directa de conocer que un TLD se encuentra en esta etapa, es por esto que la entidad registrante es responsable de realizar el anuncio a través su sitio web o enviando un email a la *Internet Society* informando tal situación.

Las siguientes figuras ilustran las etapas anteriormente descritas, según distribución mundial para el caso de dominios de nivel superior por código de país, [49]:

#### ccTLD DNSSEC Status on 2016-12-05

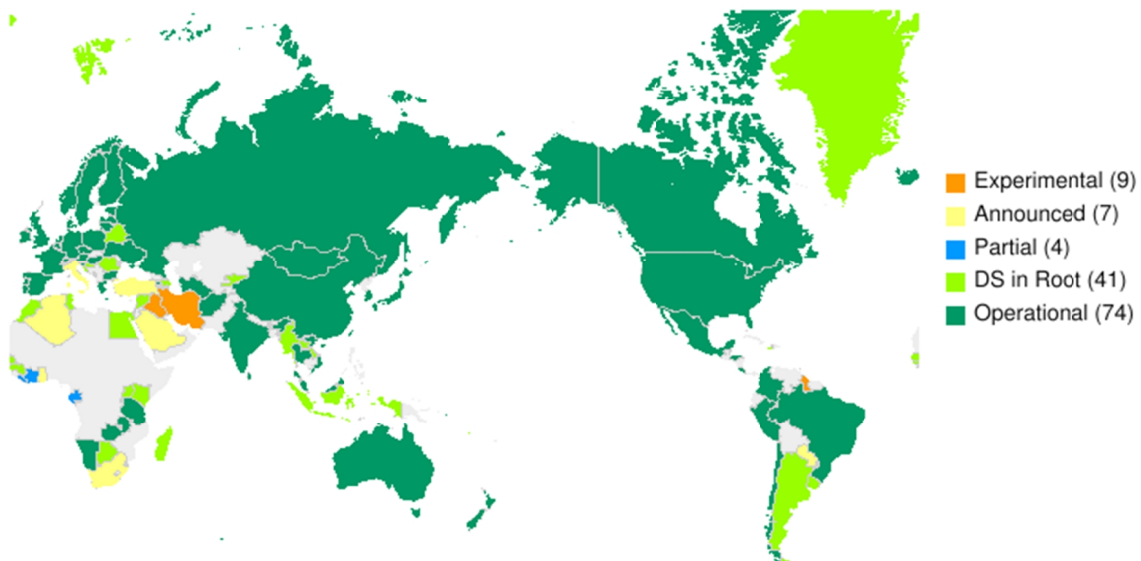


Figura 22: Mapa mundial despliegue DNSSEC ccTLD

### AF ccTLD DNSSEC Status on 2016-12-05

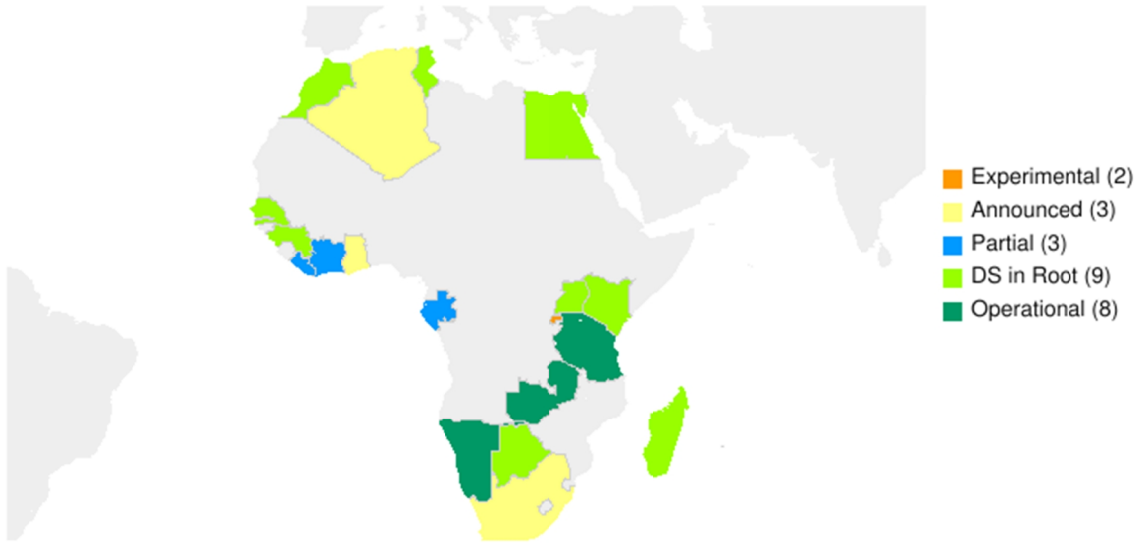


Figura 23: Mapa África despliegue DNSSEC ccTLD

### NA ccTLD DNSSEC Status on 2016-12-05

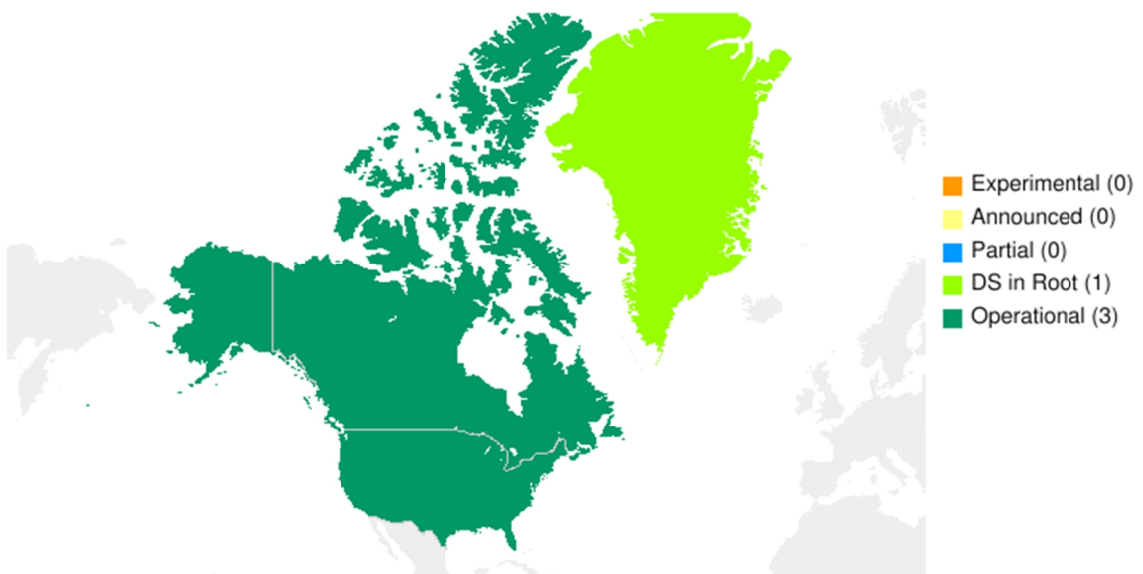


Figura 24: Mapa América del Norte despliegue DNSSEC ccTLD

### LAC ccTLD DNSSEC Status on 2016-12-05

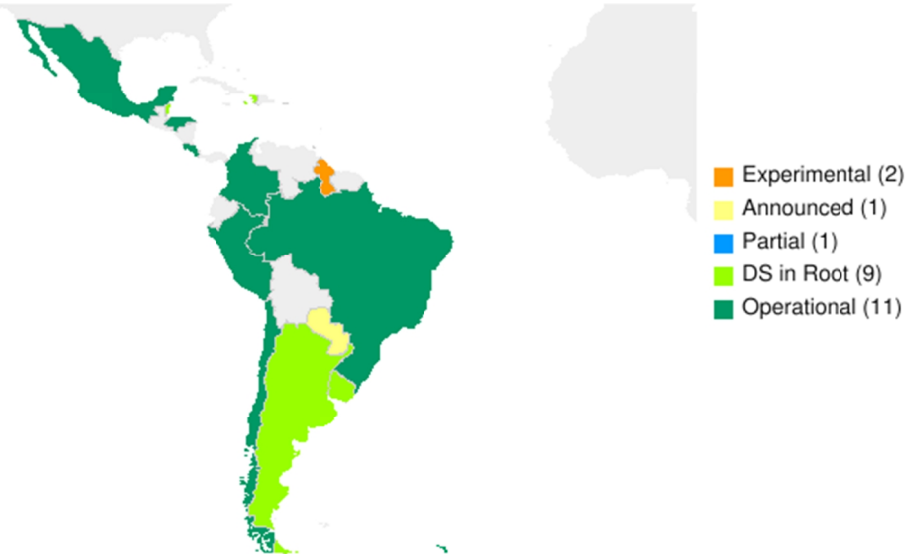


Figura 25: Mapa Latinoamérica y Caribe despliegue DNSSEC ccTLD

### AP ccTLD DNSSEC Status on 2016-12-05



Figura 26: Mapa Asia y Oceanía despliegue DNSSEC ccTLD

## EUR ccTLD DNSSEC Status on 2016-12-05

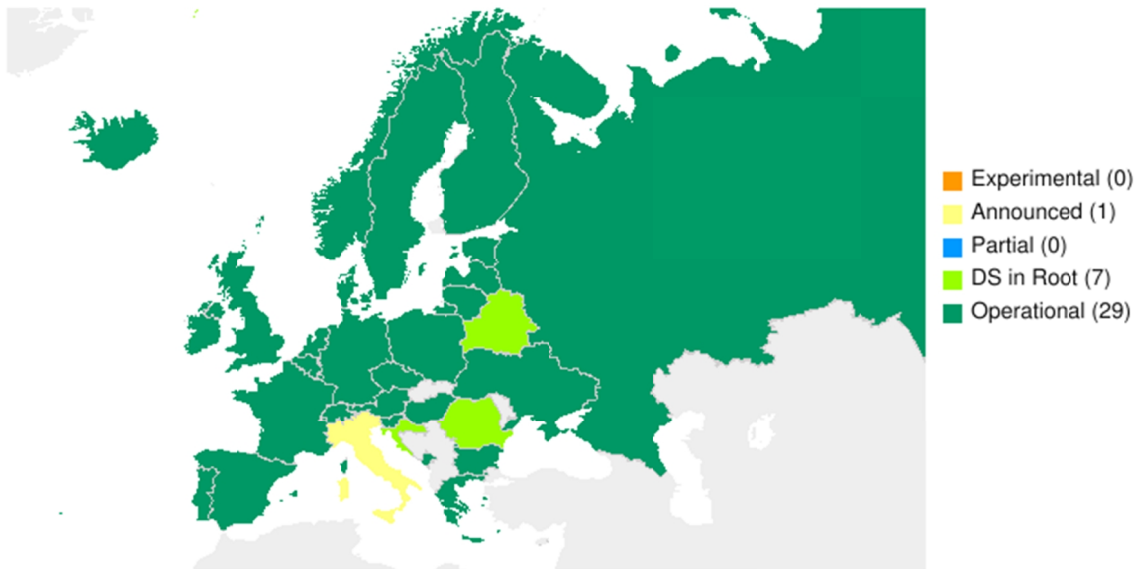


Figura 27: Mapa Europa despliegue DNSSEC ccTLD

Continuando con la serie de reportes que muestran el avance del despliegue DNSSEC, se considera relevante el reporte que presenta el *Dr. Richard Lamb*, (*Sr. Program Manager DNSSEC – ICANN*),[50], donde brinda información detallada de los TLDs que adoptan DNSSEC. Un dato importante que se destaca de este reporte es el bajo porcentaje de dominios de segundo nivel que adoptan DNSSEC, donde tan solo un 1% lo han implementado. Las razones de esta baja adopción pueden ser varias y requiere un análisis detallado que se presenta en la siguiente sección. La Figura 28 muestra un análisis comparativo para los porcentajes de adopción.

### *DNSSEC Deployment Report*

Sun Jan 29 23:11:56 UTC 2017

Total TLDs: 1528 / Signed TLDs in root: 1373 / Recently added: ws. (01/25/2017)

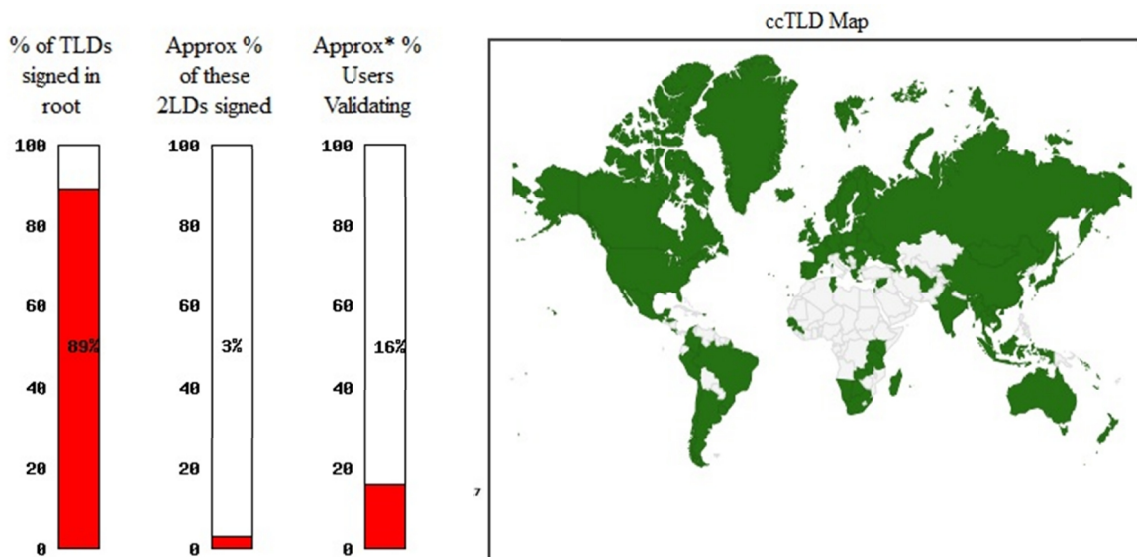


Figura 28: Reporte porcentajes de adopción



En lo que respecta al dominio *.ar*, en Junio del año 2015, se informa desde NIC Argentina, que se realizó el firmado y anuncio para la zona *.ar*, completando así la etapa cuatro para el despliegue final de DNSSEC, restando aún la posibilidad de la aceptación de delegación segura, para alcanzar la quinta y última etapa [51].

### **5.5. Evolución del despliegue a nivel mundial**

Para que DNSSEC funcione según lo previsto, el despliegue debe abarcar todos los niveles de la arquitectura DNS, es por esto que se considera relevante presentar cuales son los aspectos que retrasan de alguna manera la completa adopción.

Algunas de las causas de una baja adopción por parte de los servidores Autoritativos, así como dominios de nivel superior genéricos, se presentan en “*A measurement study of DNSSEC misconfigurations*” [52]. Estas son:

- La implementación de DNSSEC aumenta la complejidad de la gestión de la infraestructura de DNS.
- Una mala configuración podría hacer que los usuarios de Internet no puedan acceder a la red protegida.

A modo de resumen se describen los resultados expuestos en el trabajo citado, donde se realizaron pruebas sobre los dominios *.bg*, *.br*, *.co*, *.com*, *.nl* y *.se*. Se utilizaron dos escenarios de prueba, el primero basado en un enfoque descendente, donde primeramente se verifica la zona raíz, seguido por la zona de nivel superior y su correspondiente registro DS en la zona padre, continuando así el proceso hasta llegar a la zona autoritativa. El segundo enfoque es en el orden inverso, es decir que primero se verifica la zona autoritativa, continuando con la zona de nivel superior hasta llegar a la zona raíz.

Basados en el primer escenario de pruebas, (enfoque descendente), la primera observación que se realiza está relacionada con la configuración del registro DNSKEY, donde las dos terceras partes presentan errores de configuración, tales como pérdida del registro DNSKEY, claves KSK incorrectamente firmadas por sus correspondientes claves ZSK y expiración de clave. Curiosamente el tercio restante se debe a configuraciones erróneas generales de DNS, probablemente debido a incompatibilidades con las consultas DNSSEC. La siguiente figura muestra en detalle los porcentajes de las configuraciones erróneas antes descriptas.

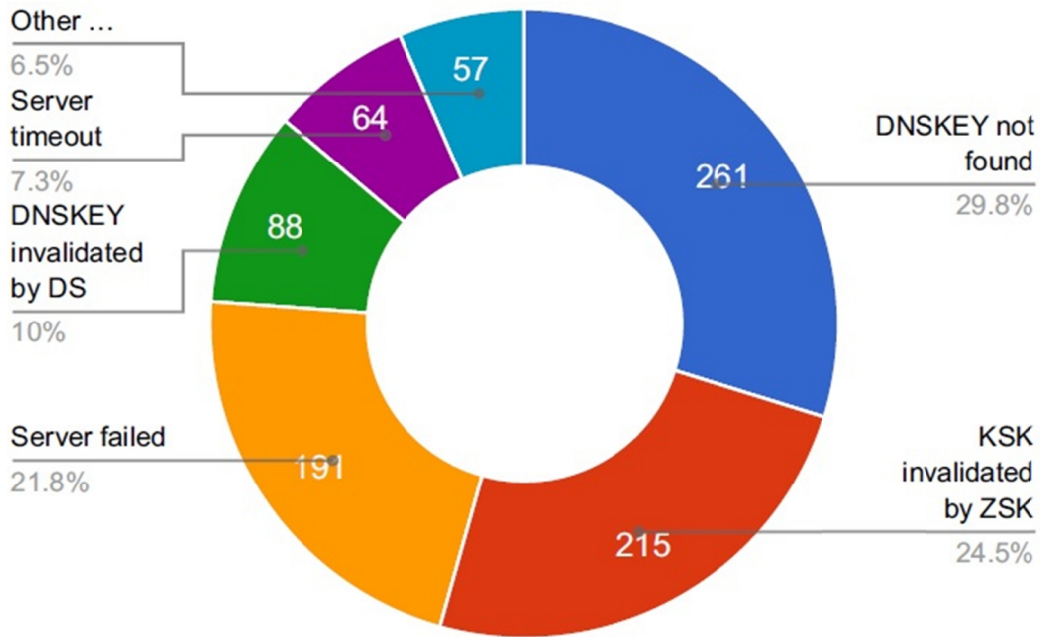


Figura 29: Configuraciones erróneas enfoque descendente

Paso seguido, se analizó el impacto de éstas configuraciones erróneas en la disponibilidad del servicio, al momento de realizar una consulta/respuesta en lo que concierne específicamente a DNSSEC. Se observó que la pérdida del Registro DNSKEY, así como una clave KSK invalidada por su correspondiente clave ZSK, provocó prácticamente un 100% de inaccesibilidad. Por otro lado, un Registro DNSKEY invalidado por su correspondiente Registro DS presente en la zona padre, alcanzó un 3,41% de afectación a la disponibilidad del servicio, sin embargo, este tipo de errores se consideran de gravedad ya que comprometen la comprobación de la cadena de confianza.

Continuando con el análisis y en base al escenario con enfoque ascendente, los resultados muestran que la mayoría de las configuraciones erróneas se corresponden a dominios que no tienen registros DS válidos publicados en sus respectivas zonas de nivel superior. Esto se debe a que en este enfoque, primero se verifican los registros RRSIG y DNSKEY de manera local, antes de solicitar el registro DS a la zona padre, lo que implica que la configuración de la zona es correcta, pero la cadena de confianza se rompe en ausencia del registro DS. Las configuraciones erróneas relacionadas a este tipo de error representan un 68% de los casos. La siguiente figura ilustra los detalles de los resultados obtenidos.

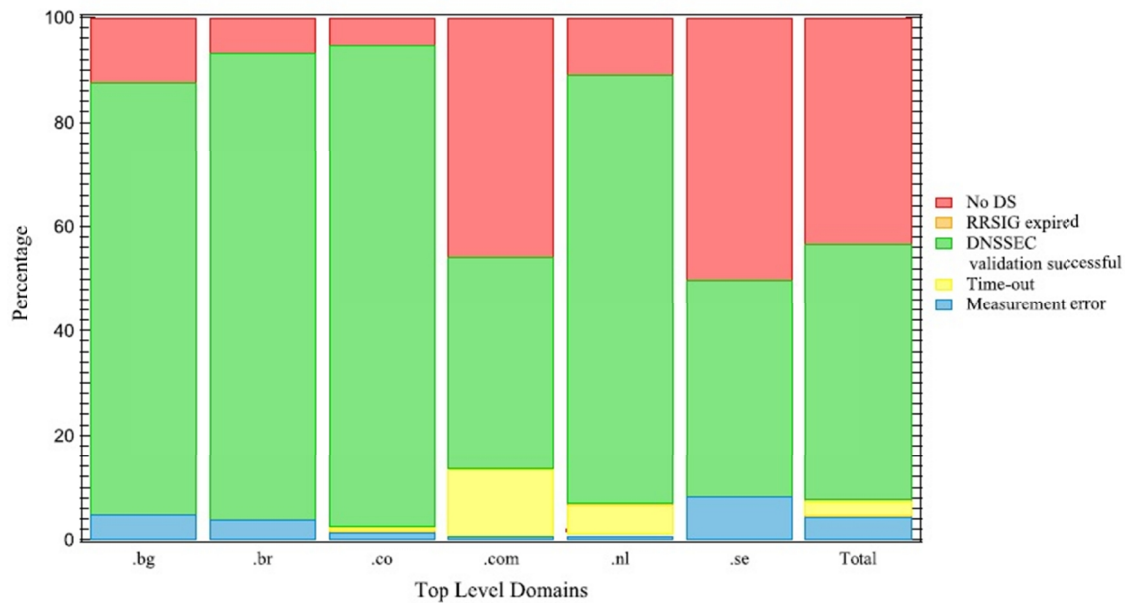


Figura 30: Configuraciones erróneas enfoque ascendente

A modo de conclusión, y en base a los análisis de configuraciones erróneas para los dominios bajo estudio, (.bg, .br, .co y .se), los resultados obtenidos muestran que la implementación de DNSSEC no es una tarea trivial. La mayoría de los casos de configuraciones erróneas están relacionadas con el registro DNSKEY, donde en el 99% de los casos, una pérdida del registro DNSKEY o un error en las etapas de firmado provocan la no disponibilidad del servicio de resolución, generando que el dominio sea inalcanzable. Por lo tanto, las organizaciones que implementan DNSSEC, deben verificar frecuentemente, una correcta configuración de los parámetros DNSSEC e implementar mecanismos que garanticen verdaderamente los recursos.

En lo que respecta a las tareas de implementación de DNSSEC propuestas en este trabajo de tesis, desarrolladas en las secciones “Comprobaciones sobre configuración DNSSEC”, “Observaciones sobre el proceso de configuración DNSSEC” y “Análisis de falla de proceso de validación”, las conclusiones antes descriptas confirman y validan las observaciones y sugerencias relacionadas a los procesos de generación de claves, ubicación de archivos, firmado de zonas, expiración de firmas, y otros procesos relacionados al despliegue final de DNSSEC.

## 5.6. Topología DNSSEC

### 5.6.1. Introducción

Expuesto el funcionamiento de DNSSEC de manera general, se presenta a continuación detalles de cuáles son las características con las que deben contar cada una de las entidades (servidores autoritativos, *resolvers*, clientes DNS) y funciones que cumplen en un proceso de validación de respuesta DNS.

### 5.6.2. Servidor de Nombres Autoritativo

Un Servidor de Nombres Autoritativo con soporte para DNSSEC, implícitamente debe soportar también EDNS por las siguientes razones. Debe ser capaz de recibir e interpretar paquetes de datos DNS, donde el bit DO se encuentra activo, lo que indica que el Cliente que realizó la consulta, está dispuesto a recibir respuestas DNSSEC. Del mismo modo, el hecho de que se encuentre configurado con EDNS, le permitirá soportar paquetes DNS de mayor longitud, utilizando UDP como protocolo de transporte, es decir que aún cuando el paquete tenga una longitud de 65535 bytes, éste se puede enviar en un único paquete UDP, debido a que las respuestas DNSSEC tienen una mayor longitud que una respuesta DNS tradicional, como consecuencia del agregado de firmas y claves relacionadas.

Por otro lado, si el Servidor es Autoritativo, éste debe ser capaz de generar firmas para todos los datos sobre los que tiene autoridad. Las firmas son generadas de manera offline, ya que realizar esta tarea de manera online para cada consulta que se reciba, resultaría poco factible debido a las limitaciones de velocidad y todo lo que concierne a la estabilidad y disponibilidad.

Asimismo, un Servidor Autoritativo, debe poder enviar los nuevos Registros de Recursos que incorpora DNSSEC (DNSKEY, RRSIG, DS y NSEC y/o NSEC3), con la particularidad de que todos estos Registros fueron generados de manera offline.

Por último, un Servidor de las características mencionadas anteriormente, nunca enviará como respuesta paquetes DNSSEC con el bit AD activo, ya que solo devuelve como respuesta Registros acompañados de su firma correspondiente, es decir que no realiza ningún tipo de validación.

### 5.6.3. Servidores de Nombres Cache

Como se ha descrito anteriormente, los Servidores de Nombres Cache cumplen dos roles diferentes, por un lado, reciben consultas de los Clientes y eventualmente, devuelven una respuesta, utilizando un mismo canal de comunicación, de este modo se puede considerar que cumple el rol de Servidor. Mientras que del otro lado es el encargado de resolver las consultas recibidas, mediante el envío de consultas a los Servidores de Nombres Autoritativos, comportándose como un Cliente. En el RFC 4035 [38-p17], se hace una distinción entre los dos roles, donde del lado del Servidor se lo llama Servidor de Nombres Recursivo, mientras que del lado del Cliente se lo llama Servidor *Resolver*.

Un Servidor de Nombres Cache con capacidades de DNSSEC, intentará enviar sus consultas, indicando que tiene la capacidad para tratar las respuestas DNSSEC que reciba, lo que significa que puede interpretar los nuevos Registros de Recursos de DNSSEC.

Regularmente todos los Servidores de Nombres Caché validaran las respuestas DNSSEC por sí mismos, lo que implica que cuando se reciba una respuesta con éstas características, se comprobaran las firmas.

Cuando un Cliente con soporte para DNSSEC envía una consulta, el Servidor de Nombres Cache activará el bit AD solo si considera que todos los Conjuntos de Registros de Recursos en las secciones “*Answer*” y “*Authority*” de la respuesta, son auténticas, o bien si considera que todos los Conjuntos de Registros de Recursos en la sección “*Answer*” y cualquier Registro de Recurso de Respuesta Negativa en la sección “*Authority*” son auténticas.

Si el Cliente activa el bit CD, el Servidor de Nombres Cache, no realizará ninguna validación, simplemente se comportará como un proxy, reenviando todos los datos, tal cual los ha recibido, incluido cualquier Registro DNSSEC. Estos datos, incluso podrían provenir de la memoria caché, e incluso tratarse de datos manipulados conteniendo firmas inválidas. En este caso toda la responsabilidad reside en el Cliente. En conclusión, la activación del bit CD en una consulta, simplemente está afirmando que es el Cliente el que realizará la verificación de las firmas. Y lo que es más importante aún, el bit CD no tiene ninguna influencia sobre el bit AD, un Servidor de Nombres Caché podría establecer el bit AD en una respuesta, aún cuando el bit CD haya sido activado en una consulta, indicando que los datos son auténticos.

#### **5.6.4. Stub Resolver**

*Stub Resolver* es la implementación de DNS en un dispositivo host cliente, es un conjunto de librerías encargadas de dialogar con los programas (navegador web, clientes de correo, cliente FTP, etc.) cuando éstos intentan resolver un nombre de dominio. Este conjunto de librerías por lo general se implementan como parte del Sistema Operativo Cliente y no involucran mucho código, dado que la mayor funcionalidad se encuentra en los Servidores de Nombres Cache.

Si un *Stub Resolver* no tiene soporte para DNSSEC, el Servidor de Nombres Cache, no recibirá consultas con el bit DO activado, por lo tanto este responderá de la manera regular, es decir sin adjuntar ningún tipo de firmas en la respuesta.

Por el contrario, si se encontrara configurado con soporte DNSSEC, se presentan dos escenarios posibles, en el primer caso, es el *Stub Resolver* quien hace la verificación por cuenta propia, en el segundo caso, se basa en el Servidor Cache, para que éste se encargue de tal verificación. En ambos casos, el *Stub Resolver* envía la consulta con el bit DO activo.

Realizar el proceso de verificación por cuenta propia, tiene sentido cuando los Clientes implementan sus propias políticas con respecto a DNSSEC, en lugar de seguir las políticas con las que opera el Servidor Cache, o bien si el *Stub Resolver* no tiene confianza en el Servidor Cache. Para indicar al Servidor de Resolución, que se quiere recuperar todos los registros, el *Stub Resolver*, deberá activar el bit CD en la consulta, el Servidor Cache, responderá con los datos sin procesar, incluyendo las firmas correspondientes. Sin embargo, cabe destacar, que los datos que residen en cache, tendrán una doble verificación, una realizada por el Servidor Cache, antes de almacenar los mismos y la otra la realizará el *Stub Resolver*.

Si un *Stub Resolver* tiene soporte para DNSSEC, pero deja la verificación de firmas en manos del Servidor Cache, debe tener plena confianza en este último. En el RFC 4035,

se expresa que, las respuestas recibidas por un *Resolver*, dependen en gran medida de la política local del Servidor Recursivo, [38-p24].

Actualmente, los Clientes finales, no consideran las opciones de DNSSEC, por lo general solo se limitan a la resolución de nombres de dominio. Sin embargo, para algunos servicios, las respuestas DNS autenticadas pueden ser necesarias. Ejemplos de estos servicios pueden ser, “*daemon mail*”, donde es necesario asegurarse de enviar emails al Servidor correcto, y no al de un atacante; navegadores web, para los casos de realizar transacciones bancarias, donde es necesario asegurar que se está dirigiendo a la página correcta y no a una copia de ésta montada por un atacante. En caso de necesitar habilitar tales opciones de DNSSEC, es el administrador del sistema, la persona encargada de configurar estas extensiones.

## **5.7. Especificaciones**

Presentados de manera general los nuevos registros de recursos presentes en DNSSEC, así como el rol que desempeñan cada uno de ellos en un intercambio de flujo DNS seguro, se presenta a continuación una descripción detallada de cada Registro, analizando estructura y campos presentes. Se tomó como referencia lo descrito en el RFC 4034.

### **5.7.1. Registros de Recurso de clave pública(DNSKEY)**

DNSSEC hace uso de criptografía de clave pública para “firmar” y autenticar conjuntos de registros de recursos DNS (RRsets). Es por esto que se ha introducido el registro de recurso DNSKEY, el cual está habilitado para almacenar claves públicas, que posteriormente serán usadas por DNSSEC en procesos de autenticación.

Según se describe en [34-p4], el registro DNSKEY “no pretende ser un registro para el almacenamiento de claves públicas arbitrarias, que no estén directamente relacionadas con la infraestructura DNS.

Una zona firma sus RRsets autoritativos usando una clave privada y almacena su correspondiente clave pública en un registro DNSKEY. Un *Resolver* puede luego utilizar la clave pública para validar las firmas que cubren la zona de los RRsets, y así autenticarlos.

#### **Formato de la porción RDATA**

La porción RDATA para un Registro de Recurso DNSKEY consiste en un campo *Flags* de dos octetos de longitud, un campo *Protocol* de un octeto de longitud, un campo *Algorithm* de un octeto de longitud y por último un campo de longitud variable para la clave pública.

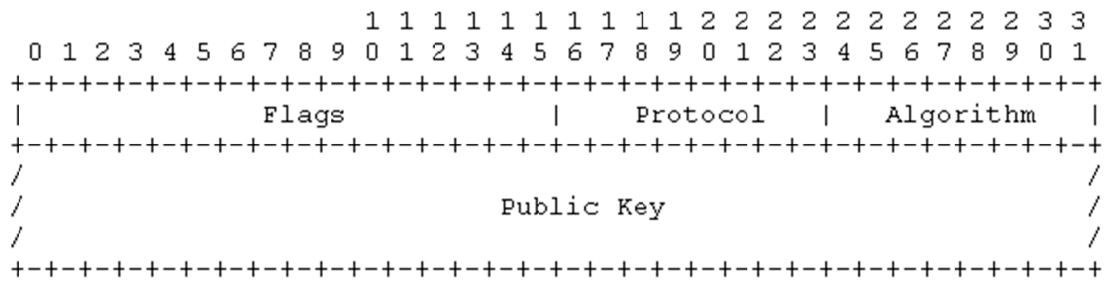


Figura 31: Formato porción RDATA en Registro DNSKEY.

### Campo *Flags*

- Del bit 0 al bit 6, actualmente no tienen uso, por lo tanto almacenan el valor 0.
- A través del bit 7, se interpreta la clave para la zona. Si el bit tiene el valor 1, el registro DNSKEY mantiene la clave para la zona (*ZSK*). Si el bit tiene el valor 0 entonces el registro mantiene algún otro tipo de clave pública DNS y no debe ser usada para verificar los registros RRSIGs.
- El uso de este bit adquiere relevancia según lo descrito en [42-p4] y aplica solo a claves del tipo KSK. Si el bit 8 tiene el valor 1, está indicando que el contenido del campo *key-data* ha sido revocado y en consecuencia, éste no debe ser usado para autenticar firmas (Registros RRSIG). Si esta seteado a 0, la clave puede ser usada sujeto a los plazos de aceptación.
- Del bit 9 al 14, se encuentran reservados, por lo tanto deben valer 0.
- Continuando con la interpretación de los bits banderas, el bit 15 es usado para indicar un Punto de Entrada Seguro,[43-p4]. Si el bit tiene el valor 1, el registro DNSKEY mantiene una clave destinada para uso como un Punto de Entrada Seguro. En este caso el bit 7 debe contener el valor 1. Lo que indica, solo para propósitos administrativos (indica el propósito de la clave), que la clave se trata de una clave KSK.

Por lo tanto, las únicas combinaciones válidas para el campo *Flags* son: 0000, 0100 y 0101, cuyos valores en decimal correspondientes son: 0, 256 y 257 respectivamente.

### Campo *Protocol*

El campo *Protocol* debe tener el valor 3 y el registro DNSKEY debe ser tratado como invalido durante el proceso de verificación de firma, si se encontrara un valor diferente.

### Campo *Algorithm*

El campo *Algorithm* identifica el algoritmo criptográfico de clave pública y determina el formato del campo Clave Pública. La figura siguiente muestra un listado de los algoritmos posibles para DNSSEC según la IANA [44].

Number	Description	Mnemonic	Zone Signing	Trans. Sec.	Reference
0	Reserved				[RFC4034]
1	RSA/MD5 (deprecated, see 5)	RSAMD5	N	Y	[RFC4034][proposed standard][RFC2537][proposed standard]
2	Diffie-Hellman	DH	N	Y	[RFC2539][proposed standard]
3	DSA/SHA1	DSA	Y	Y	[RFC3755][proposed standard][RFC2536][proposed standard][Federal Information Processing Standards Publication (FIPS PUB) 186, Digital Signature Standard, 18 May 1994.][Federal Information Processing Standards Publication (FIPS PUB) 180-1, Secure Hash Standard, 17 April 1995. (Supersedes FIPS PUB 180 dated 11 May 1993.)]
4	Reserved for Elliptic Curve	ECC			
5	RSA/SHA-1	RSASHA1	Y	Y	[RFC3755][proposed standard][RFC3110][proposed standard]
6	DSA-NSEC3-SHA1	DSA-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
7	RSASHA1-NSEC3-SHA1	RSASHA1-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
8	RSA/SHA-256	RSASHA256	Y	*	[RFC5702][proposed standard]
9	Unassigned				
10	RSA/SHA-512	RSASHA512	Y	*	[RFC5702][proposed standard]
11	Unassigned				
12	GOST R 34.10-2001	ECC-GOST	Y	*	[RFC5933][standards track]
13	ECDSA Curve P-256 with SHA-256	ECDsap256SHA256	Y	*	[RFC6605][standards track]
14	ECDSA Curve P-384 with SHA-384	ECDsap384SHA384	Y	*	[RFC6605][standards track]
15-122	Unassigned				
123-251	Reserved				[RFC6014][standards track]
252	Reserved for Indirect Keys	INDIRECT	N	N	[RFC4034][proposed standard]
253	Private algorithms - domain name	PRIVATEDNS	Y	Y	[RFC3755][proposed standard][RFC2635][proposed standard]
254	Private algorithms - OID	PRIVATEOID	Y	Y	[RFC3755][proposed standard][RFC2635][proposed standard]
255	Reserved				[RFC4034][proposed standard]

Tabla 9: Listado algoritmos para DNSSEC.

Los valores 6 = DSA-NSEC3-SHA1 y 7 = RSASHA1-NSEC3-SHA1, son alias para los valores 3 y 5 respectivamente. Sin embargo, los valores 6 y 7, deben ser usados si la zona firmada usa NSEC3, para evitar problemas de incompatibilidad en *Resolvers* configurados sin soporte para el protocolo NSEC3. Por el contrario, si el protocolo anterior no se usa dentro de una zona, entonces los valores 3 y 5 deben ser usados

### Campo *Public Key*

El campo *Public Key* contiene información de la clave, el formato depende del algoritmo de la clave que se almacena y se describe en documentos separados.

### Formato de presentación del Registro *DNSKEY*

El formato de presentación de la porción *RDATA* es el siguiente:

- El campo *Flags* debe ser representado como un entero decimal sin signo. Posibles valores son 0, 256 y 257.
- El campo *Protocol* debe ser representado como un entero decimal sin signo con un valor igual a 3.
- Los valores posibles para el campo *Algorithm* se muestran en la tabla anterior.
- El campo *Public Key* debe ser representado usando una codificación Base 64.

### Ejemplo de un Registro *DNSKEY*

El siguiente registro *DNSKEY* almacena la clave para la zona del dominio *example.com*



```
example.com. 86400 IN DNSKEY 256 3 5 ( AQPskmynfzW4kyBv015MUG2DeIQ3
Cbl+BBZH4b/OPY1kxkmvHjcZc8no
kfzj31GajIQKY+5CptLr3buXA10h
WqTkF7H6RfoRqXQeogmMHfpftf6Z
Mv1LyBUgia7za6ZEzOJBOztyvhjL
742iU/TpPSEDhm2SNKLi jfUppn1U
aNvv4w== )
```

Figura 32: Ejemplo Registro DNSKEY.

Los primeros cuatro campos especifican: propietario del Registro de Recurso (nombre de zona), TTL, Clase y tipo de registro (DNSKEY). EL valor 256 es bit 7 activado (bit de clave de zona). Campo *Protocol*, siempre vale 3. El valor 5 en el campo *Algorithm* significa que se trata de RSA/SHA1. El texto restante es la codificación en Base 64 de la clave pública.

### 5.7.2. Registro de firma digital (RRSIG)

Un registro RRSIG contiene la firma para un conjunto de Registros de Recursos (RRset) con un nombre particular, clase y tipo. El registro RRSIG especifica un intervalo de validez de la firma y utiliza el Algoritmo, Nombre del propietario (zona firmante) y etiqueta de la clave (*Key Tag*) para identificar el registro que contenga una clave pública, necesarios para el proceso de verificar la firma. El registro RRSIG se genera en el proceso de firmado de la zona, por ejemplo, mediante la utilidad *dnssec-signzone*, utilizando la clave privada y cuyo par (clave pública) es almacenada en el registro DNSKEY. El Registro RRSIG se define en [34-p6].

#### Formato de la porción RDATA

La porción RDATA para un RRSIG consta de los siguientes campos:

Nombre campo	Longitud
<i>TypeCovered</i>	2 octetos
<i>Algorithm</i>	1 octeto
<i>Labels</i>	1 octeto
<i>Original TTL</i>	4 octetos
<i>SignatureExpiration</i>	4 octetos
<i>SignatureInception</i>	4 octetos
<i>Key Tag</i>	2 octetos
<i>Signer'sName</i>	Variable
<i>Signature</i>	Variable

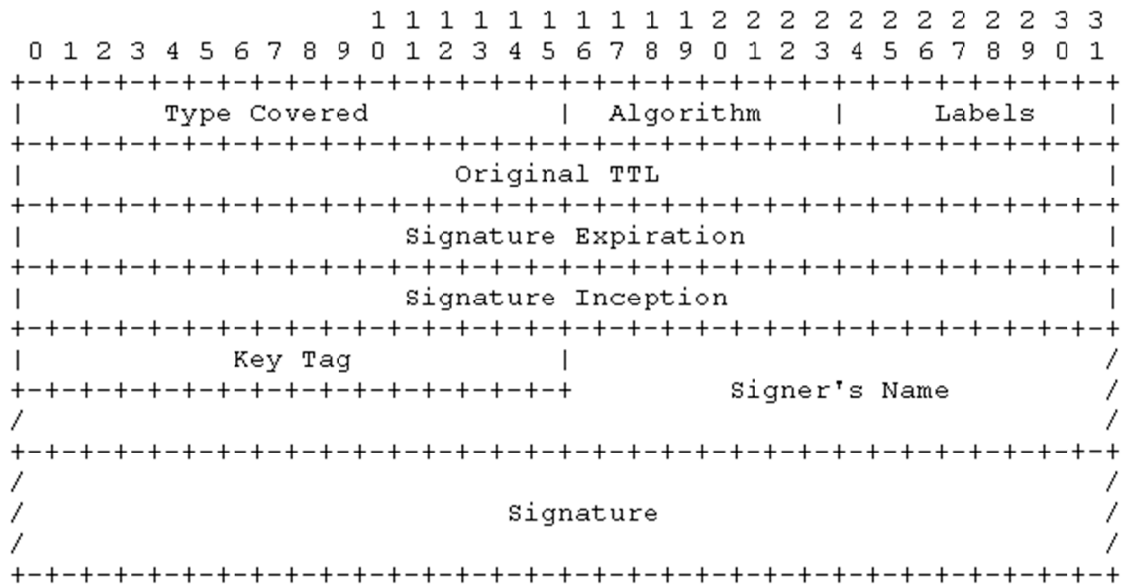


Figura 33: Formato porción RDATA en Registro RRSIG.

### Campo *Type Covered*

El campo *Type Covered* identifica el tipo de RRset que es cubierto por el registro RRSIG.

### Campo *Algorithm*

El campo *Algorithm* identifica el algoritmo criptográfico de clave pública y determina el formato del campo Clave Pública. Solo DSA/ SHA-1 y RSA/ SHA-1 (valores 3 y 5 respectivamente) están definidos para uso con RRSIG, acorde a lo especificado en el RFC 4034, sin embargo en [45], se contempla el uso de los algoritmos SHA-256 y SHA-512 también llamados SHA-2 (valores 8 y 10 respectivamente); por otro lado en [46], se describe el uso de los algoritmos GOST R 34.10-2001 y GOST R 34.11-94.

### Campo *Labels*

El campo *Labels* se utiliza como un contador del número de etiquetas en el nombre original de registro RRSIG. El valor de este campo es usado al momento de comprobar la firma, ya que se necesita el nombre original usado para la creación de tal firma.

El valor presente en el campo *Labels*, debe ser menor o igual al número de etiquetas en el nombre del RRSIG. Por ejemplo:

- “*www.ejemplo.com*” tiene el valor 3 en el campo Labels.
- “*\*.ejemplo.com*” tiene el valor 2 en el campo Labels.
- “.” Tiene el valor 0 para el campo anterior.

### Campo *Original TTL*

En este campo se define el TTL del RRset cubierto. El motivo de almacenar este valor de forma repetida en la firma de debe a que en el proceso de firmado, se utiliza el campo TTL original del RRset (se firma el RRset completo original), pero este valor se

decrementa en el transcurso de almacenamiento en caché, por lo que de no tener el valor original la firma no sería correcta.

### Campos *Signature Expiration* y *Signature Inception*

Para que una firma sea válida, ésta debe existir entre las fechas de inicio y de expiración. Si el momento actual es anterior al de inicio o posterior al de expiración, la firma no será válida. Observemos que esta característica obliga a que firmante y *resolver* tengan sus relojes sincronizados, esto es, si el reloj de un *resolver* marca un día incorrecto, podrían dejar de validarse registros y aparentar una pérdida de conexión con Internet. La zona horaria utilizada es UTC. Ambos tienen el formato YYYYMMDDHHMMSS.

### Campo *Key Tag*

El campo *Key Tag* tiene el propósito de facilitar la búsqueda de la clave de firma. Cada DNSKEY tiene una etiqueta (*tag*) que se calcula a partir de su RDATA. Si una zona tiene más de un DNSKEY, esta etiqueta puede ayudar a encontrar el correcto de manera más rápida. Para el caso de que existan más de un DNSKEY con la misma etiqueta, se procederá a intentar con cada una de las claves.

### Campo *Signer'sName*

En este campo se especifica el nombre del propietario del registro DNSKEY, el valor presente en este campo más los valores de los campos *Algorithm* y *Key Tag*, son utilizados para localizar el DNSKEY que contiene la clave pública para verificar la firma, (realizada con su correspondiente par privada).

### Campo *Signature*

El campo *Signature* contiene la firma criptográfica que cubre la porción RDATA del registro RRSIG (excluido el campo *Signature*). La firma se almacena utilizando una codificación Base64. En [34 p9 y 10], se especifica que la firma se genera a partir de la siguiente cadena de caracteres: RRSIG\_ RDATA | RR(1) | RR(2) | ..., donde RRSIG\_ RDATA es la cadena de todos los campos descriptos anteriormente, excepto el campo *Signature*. Los Registros de Recursos RR(i) son los elementos del Conjunto de Registros de Recursos (RRset) y cada Registro de Recurso compuesto por sus campos correspondientes (Nombre, Tipo, Clase, TTL, etc.).

### **Ejemplo de un Registro RRSIG**

El siguiente Registro RRSIG almacena la firma para un RRset tipo A del dominio *host.example.com*:

```
host.example.com. 86400 IN RRSIG A 5 3 86400 20030322173103 (
    20030220173103 2642 example.com.
    oJB1W6WNGv+ldvQ3WDG0MQkg5IEhjRip8WTr
    PYGv07h108dUKGMeDPKijVCHX3DDKdfb+v6o
    B9wfuh3DTJXUAFI/M0zm0/zz8bW0Rzn1803t
    GNazPwQKkRN20XPXV6nwwfoXmJQbsLnrLfkG
    J5D6fwFm8nN+6pBzeDQfsS3Ap3o= )
```

Figura 34: Ejemplo Registro RRSIG.

Los primeros cuatro campos especifican: propietario del Registro de Recurso, TTL, Clase y y Tipo de Registro de Recurso (RRSIG). La “A” representa el tipo de RRset cubierto. El valor 5 identifica el algoritmo usado para crear la firma (RSA/SHA1). El valor 3 representa el número de etiquetas en el nombre (original) del propietario. El valor 86400 es el TTL original del RRset cubierto. 20030322173103 y 20030220173103 son las fechas de expiración e inicio, respectivamente. 2642 es el *Key Tag* (etiqueta de la clave escogida para la firma) y *example.com* es el nombre del Firmante (zona con DNSKEY). El texto restante es una codificación en Base64 de la firma.

### 5.7.3. Negación de la existencia autenticada(NSEC)

Uno de los objetivos de DNSSEC es el de ser capaz de autenticar respuestas negativas o de “no existencia”, en otras palabras, poder verificar si una respuesta negativa (*NXDOMAIN*), fue recibida del host consultado. Es por esto que en el esquema inicial de DNSSEC se ha creado el Registro NSEC. En términos generales, NSEC permite validar la estructura de una zona y los Registros de Recurso que ésta contiene.

En una primera forma, y en respuesta a una consulta, NSEC permite verificar la no existencia de un nombre de host. Cada nombre tiene su correspondiente Registro NSEC que apunta al siguiente nombre de host válido en la zona. NSEC provee una cadena de nombres de hosts válidos, y por implicancia, cualquier cosa que no estuviera en la cadena, entonces no existe. En una segunda forma, contiene una lista de tipos de Registros de Recursos con el mismo nombre que el Registro NSEC, por lo tanto, cualquier tipo de Registro de Recurso que no estuviera en la lista, no existe. NSEC se genera de manera automática mediante la utilidad *dnssec-signzone*.

Para ilustrar el funcionamiento del Registro NSEC, consideremos el siguiente ejemplo a partir de una pequeña zona DNS con tres nombres (*example.org*, *a.example.org* y *d.example.org*) y tres tipos de Registros (SOA, A y TXT). Para resumir el ejemplo, la Clase no se visualiza. Resultando la siguiente zona:

```
example.org.      SOA ( ... )
a.example.org.   A 127.0.0.1
                  TXT "a record"
d.example.org.   A 127.0.0.1
                  TXT "d record"
```

Figura 35: Ejemplo Zona DNS (Funcionamiento NSEC).

A partir de la zona anterior, la cual aún no está firmada, como primer paso, se necesita que la misma se encuentre ordenada canónicamente [34-p18]. El proceso continúa con la creación de los Registros NSEC, se agrega uno por cada nombre, donde cada Registro agregado “cubre” un cierto intervalo, el último Registro NSEC apunta al primer nombre. Dando por resultado, según la zona anterior:

1. El primer registro NSEC cubre el intervalo entre *example.org* y *a.example.org*.
2. El segundo NSEC cubre desde: *a.example.org* a *d.example.org*.
3. El tercer NSEC apunta a *example.org*, y cubre desde *d.example.org* a *example.org*.

Con los intervalos definidos y puestos éstos en Registros de Recursos, la zona puede ser firmada, resultando lo siguiente:

```

example.org.      SOA ( ... )
                  DNSKEY ( ... )
                  NSEC a.example.org. SOA NSEC DNSKEY RRSIG
                    RRSIG(SOA) ( ... )
                    RRSIG(DNSKEY) ( ... )
                    RRSIG(NSEC) ( ... )
a.example.org.   A 127.0.0.1
                  TXT "a record"
                  NSEC d.example.org. A TXT NSEC RRSIG
                    RRSIG(A) ( ... )
                    RRSIG(TXT) ( ... )
                    RRSIG(NSEC) ( ... )
d.example.org.   A 127.0.0.1
                  TXT "d record"
                  NSEC example.org. A TXT NSEC RRSIG
                    RRSIG(A) ( ... )
                    RRSIG(TXT) ( ... )
                    RRSIG(NSEC) ( ... )

```

Figura 36: Ejemplo Zona DNS firmada (Funcionamiento NSEC).

Continuando con el ejemplo, si ahora un Servidor *Resolver* con características DNSSEC consulta por el dominio *b.example.org*, se retorna un paquete *NXDOMAIN*, el cual no puede considerarse confiable, ya que los datos de la cabecera podrían haber sido modificados. Para poder detectar con seguridad que *b* no existe, debe existir también un registro NSEC que cubre el espacio de nombres donde reside *b*:

```
a.example.org.   NSEC d.example.org.
```

Es decir, *b* debería preceder a *a*, pero el siguiente nombre propietario es *d.example.org*, en conclusión, *b* no existe. Haciendo el cálculo anterior, un *Resolver* puede concluir que el nombre *b* no existe. Por lo que, si la firma del registro NSEC es válida, se prueba que *b* no existe, es lo que se denomina: negación de existencia autenticada.

### Formato de la porción RDATA

```

          1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
/                               Next Domain Name                               /
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
/                               Type Bit Maps                               /
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figura 37: Formato porción RDATA en Registro NSEC.

### Campo *Next Domain Name*

Como se mencionó anteriormente, el Registro NSEC es usado para formar una cadena de nombres correspondientes a RRsets dentro de una zona, en consecuencia un RRset que no estuviera presente en la cadena, demuestra que no existe. El campo *Next Domain Name*, mantiene la próxima entrada en la cadena de nombres de dominio, ordenadas canónicamente. Si el Registro de Recurso para el cual se está agregando un NSEC fuera el último, en el campo *Next Domain Name* se define un puntero hacia el Registro SOA, creando así un ciclo.

### Campo *Type Bit Maps*

Mantiene un mapeo de bits de los tipos de Registro de Recursos definidos para un nombre de dominio. Para codificarla presencia de un tipo, la totalidad del espacio de tipos de Registros de Recursos se divide en bloques de tamaño igual a 256 bits, numerados de 0 a 255. Por cada número de bloque, la presencia de hasta 256 tipos de Registros pueden ser codificados usando una máscara de bits. Dado un número de bloque N y un bit de posición P, el correspondiente número de tipo de Registro es  $(N*256 + P)$ . Por ejemplo, para el bloque 1 y bit de posición 2, corresponde el tipo de Registro 258, (tipo actualmente no definido). El campo se codifica de la siguiente manera:

$$\textit>Type Bit Maps} = (\textit>window block number} | \textit{bitmaplength} | \textit{bitmap})^*$$

Donde | es un operador de concatenación y \* representa el operador *Kleene*. Cada instancia de número de bloque, contiene un valor dentro del rango (0-255) y la longitud del mapa de bits, contiene dicha longitud expresada en bytes (valor máximo de 32). El número de bloque y longitud del mapa de bits tienen una longitud de un byte, el mapa de bits puede tener un máximo de longitud de 32 bytes (256 bits, uno por cada posible tipo de Registro de Recurso en el bloque). Los bloques en los que ningún tipo de RR está presente no son incluidos. Por ejemplo: Si solo están presentes los tipos de Registros 1 (A) y 15 (MX), la codificación para el campo sería la siguiente: 0x00024001 = (0x00 | 0x02 | 0x4001).

Presentado un ejemplo de funcionamiento del Registro NSEC y según se describe en [35-p4], NSEC presenta dos problemas en cuanto a su diseño, el primero se conoce como “enumeración de zona”, donde cualquiera es capaz de enumerar los registros autoritativos en una zona, recorriendo la cadena NSEC, lo que permitiría a un atacante obtener todos los nombres y tipos de Registros de Recursos para una zona dada, es decir, reconstruir la totalidad de la zona.

El segundo problema se presenta para aquellas zonas de gran tamaño, donde cada nombre presente en la zona recibe un Registro NSEC más un Registro RRSIG, lo que conlleva a un gran aumento en el tamaño de la zona, una vez firmada.

En consecuencia, se introducen dos nuevos Registros de Recursos destinados a sustituir al registro NSEC, los cuales se encuentran definidos en [35]. El primero de ellos es el Registro NSEC3, el cual contiene un resumen criptográfico del siguiente RRset para una zona, en un orden canónico.

### 5.7.4. Registro NSEC3

En NSEC3, para cada nombre se obtiene su digesto, incluido el nombre propietario, por lo general se usa como algoritmo SHA-1 y para incrementar el nivel de seguridad, la función de resumen se puede aplicar varias veces, tomando como entrada el digesto anterior.

La siguiente figura muestra el formato del Registro NSEC3, la porción RDATA, almacena el hash o resumen del siguiente RRset en un orden canónico. La función de hash ha sido aplicada el número de veces indicado en el campo *Iterations*. El contenido del campo *Salt* (de longitud variable), es agregado al nombre, antes de aplicar la función de hash, para prevenir ataques de diccionario. El campo Type Bit Maps tiene la misma estructura definida en el Registro NSEC.

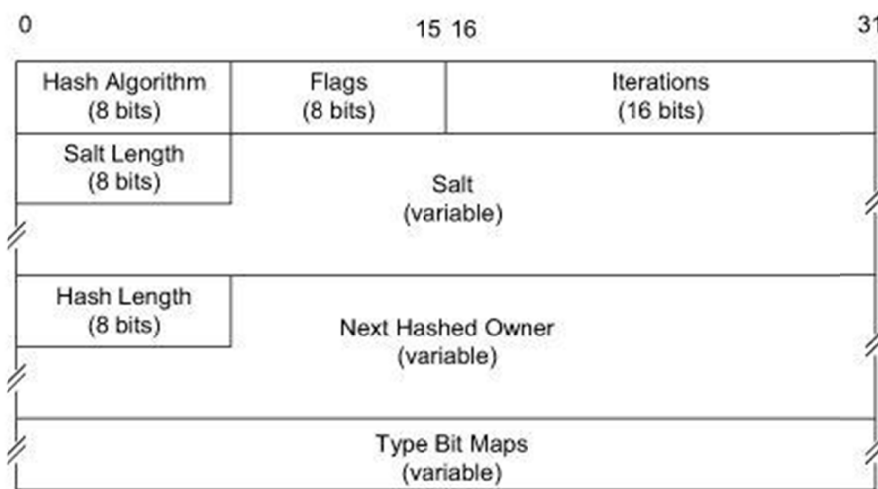


Figura 38: Formato porción RDATA en Registro NSEC3.

El campo Hash Algorithm, identifica la función de hash aplicada al siguiente nombre de dominio, el resumen obtenido es almacenado en el campo Next Hashed Owner. A la fecha solo el algoritmo SHA-1 (valor 1) se encuentra definido. El bit de orden inferior del campo Flags contiene el bit de bandera opt-out, si el mismo se encuentra en 1, estaría indicando que el Registro NSEC puede cubrir delegaciones sin firmas, como es el caso de una delegación (NS RRset) que hace referencia a una zona hija que no requiere o no desea ser firmada. El campo Salt Length indica la longitud del campo Salt expresada en bytes.

Continuando con el ejemplo para la zona *example.org*, se presenta a continuación el Registro NSEC3, el cual contiene un digesto calculado dos veces y usa el valor de *Salt* "DEAD":

```
15BG9L6359F5CH23E34DDUA6N1RIHL9H.example.org. (
  NSEC3 1 0 2 DEAD 04SKNAPCA5AL7Q0S3KM2L9TL3P50KQ4C
  SOA RRSIG DNSKEY NSEC3PARAM )
```

En la primera línea se observa el digesto para el nombre propietario: *15BG9L6359F5CH23E34DDUA6N1RIHL9H.example.org*, éste es el digesto para

*example.org*. Obsérvese que aún cuando se ha calculado el resumen de *example.org*, el nombre de la zona es agregado para que se visualice como un nombre de dominio nuevo.

El siguiente nombre, *a.example.org*, (línea 2), se le calcula su digesto, resultando, *04SKNAPCA5AL7QOS3KM2L9TL3P5OKQ4C*, en este caso *.example.org*, no es agregado al siguiente nombre, ya que este nombre cae siempre en la zona actual.

La siguiente sección está conformada por los valores: “*1 0 2 DEAD*”

- Algoritmo de *hash* = 1 (SHA-1 por defecto);
- OptOut = 0 (desactivado);
- Cantidad de iteraciones para *hash* = 2;
- Valor de *Salt* = “DEAD”.

El registro finaliza con el tipo *bit map*, cuya función de explica a continuación.

El segundo Registro especificado en el RFC 5155, es el Registro NSEC3PARAM, el cual tiene el mismo formato del Registro anterior, excepto por los campos *Hash Length*, *Next Hashed Owner* y *Type Bit Maps*, los cuales no están presentes. NSEC3PARAM es usado por los Servidores de Nombres Autoritativos configurados con soporte NSEC3 para proporcionar respuestas negativas. El Registro NSEC3PARAM proporciona los parámetros necesarios para el cálculo del hash de los nombres de dominio.

Para obtener el valor de hash a almacenar en el campo *Next Hashed Owner*, se lleva a cabo el siguiente cálculo:

$$\begin{aligned} \text{IH}(0) &= \text{H}(\text{owner name} \mid \text{Salt}) \\ \text{IH}(k) &= \text{H}(\text{IH}(k - 1) \mid \text{Salt}) \text{ if } k > 0 \\ \text{Next Hashed Owner} &= \text{H}(\text{IH}(\text{Iterations}) \mid \text{Salt}) \end{aligned}$$

Donde H es la función de hash especificada en el campo *Hash Algorithm*. El valor para *owner name* se especifica en la forma canónica. Los valores *Iterations* y *Salt*, se toman de los campos correspondientes en el Registro NSEC3.

Para evitar confusión los tipos de Registros NSEC y NSEC3, en el RFC 5155 se especifica el uso de los números de algoritmos de seguridad especiales 6 y 7 como alias para los identificadores 3 (DSA) y 5 (SHA-1) en zonas configuradas con soporte para Registros NSEC3. (*Ver campo Algorithms en Registro DNSKEY descripto anteriormente*).

Otra característica a destacar es con respecto al resultado de aplicar la función de resumen a los nombres, donde luego de aplicar dicha función, se obtiene un nombre de una sola etiqueta. Por ejemplo: nombres como *l.h.example.org*, resultan en nombres como: *117GERCPRCJGG8J04EVINDRK8DIJT14K.example.org*. Es decir que al obtener el digesto de un nombre se pierde la “profundidad” de una zona. La función de *hash* introduce un espacio de nombres planos.

### 5.7.5. Registro DS (Delegación de Firma)





Contiene el número de algoritmo del Registro DNSKEY al que hace referencia el Registro DS. El número de algoritmo usado por el Registro DS es idéntico al usado por el Registro RRSIG.

### Campo Digest Type

El DS se refiere a un DNSKEY mediante la inclusión de un resumen (digesto) del Registro DNSKEY. El campo *Digest Type* identifica el algoritmo usado para construir el digesto. Los valores posibles se muestran en la siguiente lista, [47]:

Value	Description	Status	Reference
0	Reserved	-	<a href="#">[RFC3658]</a>
1	SHA-1	MANDATORY	<a href="#">[RFC3658]</a>
2	SHA-256	MANDATORY	<a href="#">[RFC4509]</a>
3	GOST R 34.11-94	OPTIONAL	<a href="#">[RFC5933]</a>
4	SHA-384	OPTIONAL	<a href="#">[RFC6605]</a>
5-255	Unassigned	-	

Tabla 10: Listado algoritmos para digesto Registro DS.

### Campo Digest

El campo Digest contiene el resumen del Registro DNSKEY al que se hace referencia. La siguiente expresión muestra como se calcula el digesto:

$$\text{digest} = \text{digest\_algorithm}(\text{DNSKEY ownername} | \text{DNSKEY RDATA})$$

Donde | es el operador de concatenación y el valor DNSKEY RDATA se calcula a partir del Registro DNSKEY referenciado, según la siguiente expresión:

$$\text{DNSKEY RDATA} = \text{Flags} | \text{Protocol} | \text{Algorithm} | \text{Public Key}$$

Para el caso del algoritmo SHA-1, el digesto tiene una longitud de 20 bytes y en el caso de SHA-256 el mismo tiene una longitud de 256 bytes. El Registro DS se utiliza para proporcionar un enlace descendente en la cadena de la autenticación entre los límites de la zona, por lo que el Registro DNSKEY debe almacenar la clave de zona (bit 7 del campo *Flags* en el Registro DNSKEY debe almacenar el valor 1)

### Ejemplo de un Registro DS

El siguiente ejemplo muestra un Registro DNSKEY y su correspondiente Registro DS

```

dskey.example.com. 86400 IN DNSKEY 256 3 5 ( AQOeiiR0GOMYkDshW0SKz9Xz
fwJr1AYtsmx3TGkJaNXVbfi/
2pHm822aJ5iI9BMzNXxeYcmZ
DRD99WYwYqUSdjMmmAphXdvx
egXd/M5+X7OrzKBaMbCVdFLU
Uh6DhweJBjEVv5f2wwjM9Xzc
nOf+EPbtG9DMbMADjFDc2w/r
ljwvFw==
) ; key id = 60485

dskey.example.com. 86400 IN DS 60485 5 1 ( 2BB183AF5F22588179A53B0A
98631FAD1A292118 )

```

Figura 40: Ejemplo Registro DS.

Los primeros cuatro campos especifican el Nombre, TTL, Clase y Tipo de Registro (DS). El valor 60485 es la etiqueta de clave para el correspondiente Registro DNSKEY “*dskey.example.com.*”, el valor 5 denota el algoritmo usado por el DNSKEY. El valor 1 indica el tipo de algoritmo usado para la construcción del Digesto, el texto restante (RDATA) es el digesto en formato hexadecimal.

## 5.8. Un método de validación alternativo (DLV)

En situaciones en las que no sea posible delegar la confianza en un nivel superior de la jerarquía DNS, ya sea porque el nivel superior o niveles intermedios no estuvieran aún firmados, existe la posibilidad de implementar un proceso denominado “*Domain Lookaside Validation*” [15] [16], que proporciona un método alternativo mediante el cual se puede crear y verificar una cadena de confianza mediante el uso del Registro de Recurso DLV, que funciona de manera similar al Registro DS, al igual que el DS, éste puede ser generado mediante la herramienta de soft “*dnssec-signzone*”. El Registro DLV, es colocado en una zona especial, la cual está firmada y se denomina *lookaside zone*, lo que elimina la necesidad de disponer de una Zona Padre que se encuentre firmada.

El servicio de DLV provee un punto de entrada adicional (además de la zona raíz), del cual se puede obtener información de validación DNSSEC. Sin DLV, en ausencia de una ruta totalmente firmada desde la raíz a una zona en particular, los administradores que deseen habilitar DNSSEC en Servidores *Resolvers* tendrían que configurar y mantener múltiples claves de confianza, lo que claramente se convierte en una tarea inmanejable.

Organizaciones como *Internet Systems Consortium*, proveen un repositorio de puntos de entrada seguros (SEPs), donde a través de una interface web, se puede gestionar el alta, actualización y baja de tales SEPs.

Actualmente, el sitio de la *Internet Systems Consortium*, ha decidido dar de baja el servicio de DLV, que estuvo disponible desde al año 2006 hasta el año 2016, justificando la baja, debido al progreso que se ha alcanzado en el despliegue de forma nativa de DNSSEC. Todos los registros administrados, serán dados de baja en el corriente año. Sin embargo, se consideró importante describir el funcionamiento de esta alternativa, ya que formó parte y realizó un gran aporte al despliegue de DNSSEC.

Considérese el siguiente ejemplo que muestra el proceso de validación de datos DNS mediante el uso de DLV:

- El dominio *lookaside* presente se denomina *dlv.ejemplo.net*.
- El servidor de nombres recursivo NS1, intenta verificar la cadena de confianza para la zona firmada *ejemplo.com*.
- En una secuencia normal, cuando el servidor resolver, intenta verificar la cadena de confianza para *ejemplo.com*, verificará sus puntos de entrada seguros, configurados en su cláusula *trusted-keys*, si no encontrara alguno, emitirá una consulta para encontrar el registro DS en la zona padre *.com*, si no se localiza el mismo, la zona será marcada como insegura.
- En presencia de DLV, se agrega un paso adicional, permitiendo al servidor resolver generar una consulta por la zona *lookaside*, para lo cual debe tener configurado un punto de entrada seguro, para el registro DLV de la zona que está siendo verificada.
- Cuando el servidor resolver, detecta que la característica *lookaside* se encuentra activa, generará una consulta DLV con el nombre de dominio *ejemplo.com.dlv.ejemplo.net*.
- Si la consulta es exitosa y asumiendo que el punto de entrada seguro para *dlv.ejemplo.net* se encuentra presente en la cláusula *trusted-keys*, entonces la zona *ejemplo.com* es verificada como segura.

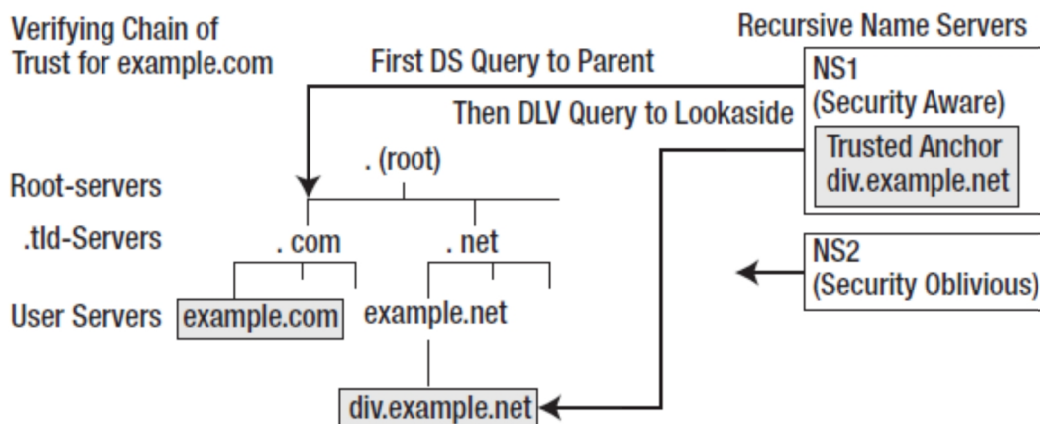


Figura 41: Proceso de validación con DLV.

La alternativa DLV fue diseñada originalmente como un método para reducir la cantidad de puntos de entrada seguros requeridos por un servidor resolver, antes de que la zona raíz fuera firmada. La misma, fue firmada en Julio del 2010, lo que eliminó la necesidad de los servicios proporcionados por DLV. Sin embargo DLV se sigue considerando como una opción útil para los siguientes casos:

- Pruebas de implementación de DNSSEC: Cuando se requiere implementar DNSSEC en entornos donde el número de zonas administradas es grande, la tarea de pasar de un ambiente de pruebas a un ambiente de desarrollo representa un salto muy grande, ya que además del proceso de firmado de cada una de las zonas, cada servidor resolver debe ser configurado con un punto de entrada seguro para cada zona a probar y la configuración final solo tiene efecto cuando se puede comprobar la misma en toda la cadena de confianza. En estos casos,

DLV, presenta una alternativa más directa, ya que se necesita configurar en un único caso (clausula *trusted-keys*) en cada servidor resolver que será usado para comprobar el proceso de validación. De esta forma solo se realizarán pruebas sobre un número reducido de nombres de dominio, dejando el resto bajo los procedimientos normales de resolución.

- Redes privadas: Las redes privadas no son inmunes a ataques internos, es aquí donde DLV proporciona un método para asegurar tales redes, utilizando un único punto de entrada seguro para configurar los Servidores *Resolvers*.

## 6. Análisis de funcionamiento de DNSSEC

En este capítulo se describirán los aspectos de la implementación de DNSSEC, con el objetivo de obtener datos comparativos en cuanto a *overhead* y tiempo de respuesta, en relación a una arquitectura DNS tradicional. Esta evaluación permitirá sentar las bases para estimar una aproximación del impacto en la implementación de estas extensiones de seguridad en Servidores Autoritativos, así como en Servidores Recursivos.

### 6.1. Descripción del ambiente de pruebas utilizado

Con el propósito de analizar el flujo de datos intercambiado en procesos de consulta/respuesta, se creó una jerarquía de servidores DNS, utilizando la herramienta de virtualización VMWare en su versión player. Se configuraron dos instancias, la primera correspondiente a una arquitectura DNS tradicional y la segunda, configura con DNSSEC. La siguiente figura ilustra para ambos casos, servidores presentes, funcionalidades de los mismos y datos de configuración de red utilizados.

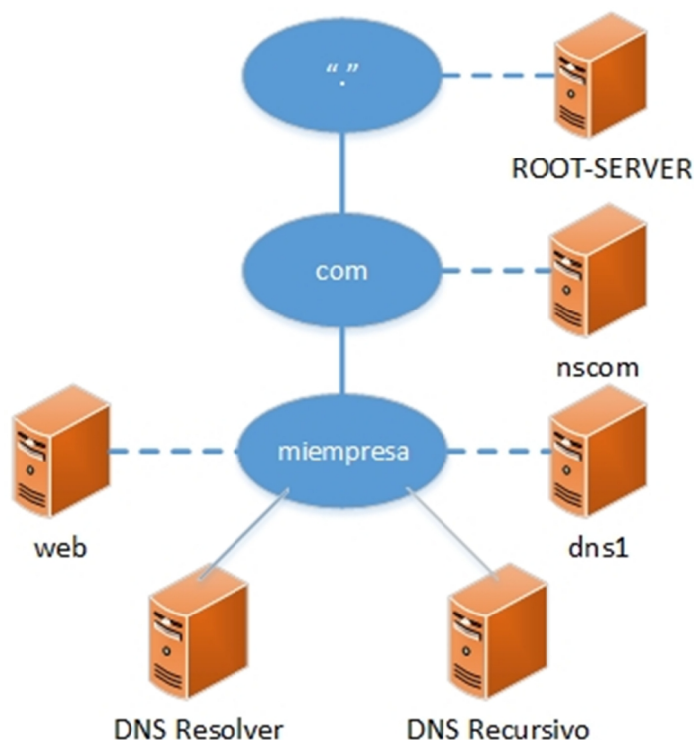


Figura 42: Ambiente de pruebas utilizado

Nombre servidor	Dirección IP	Observaciones
ROOT-SERVER	10.0.0.254	Servidor DNS Raíz
nscom	10.0.0.253	Servidor TLD .com
dns1	10.0.0.2	Servidor Autoritativo miempresa.com

web	10.0.0.100	Servidor web miempresa.com
recursivo	10.0.0.11	Servidor Recursivo Cache
resolver	10.0.0.12	Cliente DNS

Tabla 11: Configuración de red ambiente de pruebas

Todos los servidores están basados en sistema operativo Debian Linux con software DNS BIND versión 9.8.4

### 6.1.1. Descripción del proceso de configuración DNSSEC

Tomando como base el escenario presentado, se implementó DNSSEC en todos los servidores involucrados en la jerarquía DNS, es decir que, se configuró el Servidor Recursivo para validación DNSSEC, se firmaron zona *raíz*, zona *.com* y zona *miempresa.com*. El proceso se completó creando la cadena de confianza a partir de la publicación del registro DS desde zona hija a zona padre.

La implementación de DNSSEC para el escenario propuesto, se resume en los siguientes pasos:

- a) Generación de claves de firmado pública/privada ZSK y KSK.
- b) Publicación de claves pública ZSK y KSK en archivo de zona.
- c) Firmado de archivo de zona con clave privada ZSK.
- d) Publicación de archive DS en zona padre. Esto se realizó de zona *.com* a zona *raíz* y de zona *miempresa.com* a zona *.com*
- e) Refirmado de zona padre.
- f) Configuración de clave pública de validación inicial (KSK), para todos los servidores de la jerarquía, incluidos Recursivo y Resolver. Esta clave se obtuvo de lo generado en zona *raíz* (ROOT-SERVER).

Inicialmente las primeras pruebas de configuración se realizaron con los software DNSSEC Tools, [55] y OpenDNSSEC, [56], los cuales permiten automatizar las tareas de generación de claves y firmado, entre otros procesos, mediante un conjunto de *scripts* y librerías. Finalmente se optó por la utilización de los comandos nativos provistos por BIND a fin de analizar y describir cada uno de los pasos necesarios para el despliegue de DNSSEC.

A modo de ejemplo se describe a continuación la secuencia de comandos y resultados obtenidos para los incisos a), b) y c) para la zona raíz:

- 1) Creación de clave ZSK con el sgte comando:
  - a. `dnssec-keygen -r /dev/urandom -a rsasha256 -b 2048 -n zone .`
  - b. Para la zona raíz se obtuvo: `K.+008+07158.key`  
`K.+008+07158.private`
- 2) Creación de clave KSK con el comando:
  - a. `dnssec-keygen -r /dev/urandom -a rsasha256 -b 2048 -f KSK -n zone .`
  - b. Para la zona raíz se obtuvo: `K.+008+30385.key`  
`K.+008+30385.private`
- 3) Se agregó las claves públicas al archivo `db.root` con la sgte cláusula:
  - a. `$INCLUDE K.+008+30385.key ; KSK`
  - b. `$INCLUDE K.+008+07158.key ; ZSK`

4) Se firmó el archivo `db.root` con el comando:

```
dnssec-signzone -o . -t -k K.+008+30385 db.root K.+008+07158
```

Se obtuvo como resultado:

*Verifying the zone using the following algorithms: RSASHA256.*

*Zone signing complete:*

*Algorithm: RSASHA256: KSKs: 1 active, 0 stand-by, 0 revoked*

*ZSKs: 1 active, 0 stand-by, 0 revoked*

*db.root.signed*

*Signatures generated: 8*

*Signatures retained: 0*

*Signatures dropped: 0*

*Signatures successfully verified: 0*

*Signatures unsuccessfully verified: 0*

*Signing time in seconds: 0.057*

*Signatures per second: 138.547*

*Runtime in seconds: 0.095*

Continuando con la secuencia de pasos, se describen los incisos d) y e), los cuales corresponden a la delegación de confianza de la zona `.com` a zona raíz.

1) En el proceso de firmado de la zona `.com`, se obtuvo el archivo `dsset-com`. El mismo debe ser copiado al directorio `/etc/bind` del servidor `root`

2) Se agregó la cláusula `$INCLUDE dsset-com. ; DS RR` en el archivo `db.root`

3) Se refirmó la zona raíz con el comando:

```
a. dnssec-signzone -o . -t -k K.+008+30385 db.root K.+008+07158
```

Con respecto al inciso f), la configuración de clave pública de validación inicial (KSK), tiene especial importancia ya que la misma permite iniciar el proceso de validación de la cadena de confianza.

### 6.1.2. Comprobaciones sobre configuración DNSSEC

A modo de ejemplo, se describe a continuación, los primeros pasos que se llevaron a cabo a fin de comprobar las configuraciones realizadas en el punto anterior. Las comprobaciones de configuración se realizaron utilizando sentencias y comandos propios de BIND, ya que proporcionan un método directo, simple y certero, y en caso de presencia de errores, se sugiere la activación y verificación de archivos de `logs`.

Los dos primeros comandos ejecutados fueron:

- `named-checkconf`, el cual permite la verificación de la sintaxis de los archivos de configuración, `named.conf.local` y `named.conf.options`. Si la comprobación es exitosa, no se devuelve ningún mensaje.
- `Named-checkzone`, permite la comprobación del archivo de zona, para el dominio bajo administración. Si pasa la validación, se recibe un mensaje indicando que la misma fue cargada, junto a la leyenda "OK".



La siguiente instancia consistió en verificar la comunicación entre el servidor Recursivo y el servidor Autoritativo, es decir si la respuesta recibida por parte del segundo, “pasó” el proceso de validación, lo que significaría que se puede confiar en la autenticidad e integridad de dicha respuesta. Este proceso se llevó a cabo mediante el uso del comando *dig*, junto a la interpretación de los *flags* retornados, según se presenta en el siguiente ejemplo:

- *dig @10.0.0.12 www.miempresa.com A +dnssec +multiline*.

Respuesta obtenida:

```
root@resolver:~# dig @10.0.0.12 www.miempresa.com A +dnssec +multiline

; <<> DiG 9.8.4-rpz2+rl005.12-P1 <<> @10.0.0.12 www.miempresa.com A +dnssec +m
ultiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3920
;; flags: qr rd ra ad: QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do: udp: 4096
;; QUESTION SECTION:
;www.miempresa.com.      IN A

;; ANSWER SECTION:
www.miempresa.com.      604800 IN A 10.0.0.100
www.miempresa.com.      604800 IN RRSIG A 8 3 604800 20161219000250 (
20161119000250 58908 miempresa.com.
XyYvtNqMA7ohdmxKGjtn57dxS6b/9ZkexJWUoNXSp9Qj
xSBclGjF71yIa8sPL/hTj/Y8eR6XKe1zEQTKD/EBthhD
Zt1DK4L2gnx10se2/8pL8RgtZ7AMgcde0QVb2SdeAb14
eiBb/8n8SdY4DZbKpPe6EfoVGGN0edqWQ1VPtiPXoEOT
ou2DGzBwWTNoIhbJDMX8HQZGKycIO/7UjAgyuPxPxBKa
gpFwCU9z8Hd2Zjp/3spi9UhsTap3vMpNNWGqyinQtpyN
+GZMFdrq7+62BIQx8ujrqqFuJh3hYcMHvmihowUu2bQy
09mwjsXRcCjjhLmmGH7PrsAmwvptKbxe8g== )

;; Query time: 31 msec
;; SERVER: 10.0.0.12#53(10.0.0.12)
;; WHEN: Tue Nov 22 17:02:43 2016
```

Interpretación de la respuesta obtenida:

1. El mensaje de estado es auto-descriptivo, sin presencia de Error
2. Interpretación de los flags:
  - a. qr: indica que se trata de una respuesta.
  - b. rd: Se solicitó recursión.
  - c. ra: Recursividad según se indica en la consulta.
  - d. ad: Indica que la respuesta recibida ha pasado el proceso de validación, es decir que se garantiza la autenticidad y la integridad de la misma.
3. Indica que se muestra registro RRSIG conteniendo firma.
4. Contenido de la firma para el registro consultado.

El siguiente ejemplo tiene por objeto analizar el comportamiento de DNSSEC ante la consulta por un nombre de dominio inexistente, y comprobar cuál es el valor retornado

en el registro NSEC. A modo de ejemplo, se realizó una consulta ídem a la anterior, pero en este caso, por el nombre *dns2.com*.

- *dig @10.0.0.12 dns2.com A +dnssec +multiline.*

Respuesta obtenida:

```
; <<>>DiG 9.8.4-rpz2+r1005.12-P1 <<>> @10.0.0.12 dns2.com A +dnssec +multiline
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 28642
;; flags: qrdr ad; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; AUTHORITY SECTION:
com.                0 IN RRSIG NSEC 8 1 0 20161219000136 (
                    20161119000136 37890 com.
                    XkUctvLht9iJXhBZmXHbr65RS5fXd2eka2+/IlbOqU/f
                    /R4HxwK91u2VyZzYLaqclGxVPMIDYS7Esr+UuBmgnbRt
                    4Kyw2LIb/Q/KMvvi+zha9JPc64qhmPv8kQ8UpW5PK3L8
                    s8+bUYNV7FoQVd1qCC/aaXYdgOWV9/gEWLgBFULmqDBW
                    /BeRbRqAPkdeC05wk7j3bfflXxpJy/aCDxsKt9cVUOZ7
                    Qk63bKIDoe8iZ4zr+L5Qndw7Uk8Usm30GQKeE7iOhtN4
                    UsZa5+VWzOb2l+5Bjrkjhn52cLZ7Tyzs1kwRxH5Eh36
                    CrBA3wxrPSjN99nlPbrjngdsK9GZluymWw== )
com.                0 IN NSEC miempresa.com. NS SOA RRSIG NSEC DNSKEY
```

Interpretación de la respuesta obtenida:

1. Campo *status=NXDOMAIN* presente en el encabezado de la respuesta indicando que el nombre no existe.
2. Presencia de registro NSEC que apunta al siguiente nombre de dominio válido en la estructura de la zona, lo que indica que entre *dns1.com* y *miempresa.com*, no existen otros nombres de dominio.

## **6.2. Observaciones sobre el proceso de configuración DNSSEC**

En base a la secuencia de pasos listada en el proceso de configuración, se considera oportuno describir las observaciones que surgieron a partir de los resultados obtenidos, con la finalidad de aportar sugerencias a implementaciones futuras.

- El proceso de generación de claves, publicación de las mismas y firmado de zonas, es un proceso que no demanda tiempo de ejecución ni de procesamiento, si se trata de zonas almacenadas en servidores autoritativos; sin embargo para casos de zonas en niveles superiores como ccTLDs, se recomienda el uso de hardware HSM [57], tal como se describe en “*Guidelines for Deploying DNSSEC*” [58]. En otro orden, se requiere prestar especial atención a la ubicación de los archivos de claves y la edición de los archivos de zona, ya que un error de sintaxis o de inconsistencia en nombres provocaría que no se lograra el firmado de la misma.

- La creación de la “cadena de confianza” mediante la publicación del registro DS en zona padre, es un proceso que en situaciones reales, requiere la distribución por medios seguros, tales como vía email o soportes de almacenamiento.
- Con respecto a la configuración de clave pública de validación inicial (KSK root), requiere especial atención al momento del agregado de la misma a los archivos de configuración de los servidores que posteriormente realizaran el proceso de validación de firmas, ya que de existir alguna inconsistencia, provocaría una falla en toda la verificación de la cadena de confianza.
- Es necesario la configuración de registros de eventos mediante archivos logs, ya que brindan información importante al momento de realizar comprobación de errores tanto en configuración generales como en procesos de validación de firmas.
- Un aspecto importante a tener en cuenta, es con respecto al tiempo de validez de las zonas firmadas, el cual, por defecto caduca a los treinta días del firmado de las mismas. En caso de expirar dichas firmas, el proceso de validación devuelve un error general y por lo tanto no es posible la resolución de nombres. El proceso de refirmado se realiza simplemente mediante el comando correspondiente.

### **6.3. Análisis comparativo resolución de nombres DNS tradicional vs DNSSEC**

La primera propuesta de análisis tiene por objeto analizar el comportamiento de un servidor recursivo tradicional frente a otro configurado con DNSSEC, cuantificando “el costo” en términos de cantidad de consultas, tráfico enviado y tiempo de respuestas. Para la obtención de los datos a analizar se capturó el tráfico para una consulta ([www.miempresa.com](http://www.miempresa.com)), para el caso sin almacenamiento en cache y una segunda consulta, donde el servidor Resolver contiene la respuesta ya almacenada en cache. La siguiente tabla muestra los resultados comparativos para estos casos.

	Consultas	Tiempo	Bytes enviados	Bytes recibidos
DNS sin firma	4	0,0108 seg	341	514
DNS sin firma con rta en cache	1	0,0039 seg	77	128
DNSSEC	9	0,0297 seg	727	6705
DNSSEC con rta en cache	1	0,0024 seg	77	128

**Tabla 12: Resultados comparativos DNS vs DNSSEC**

Las primeras observaciones que surgen a partir de la tabla anterior son; la cantidad de tiempo total para resolver una consulta basada en DNSSEC es tres veces mayor frente a una consulta DNS tradicional, mientras que para el tráfico enviado el incremento es de 2,5 veces más y otras 12 veces más de carga de tráfico de salida, en comparación con el escenario DNS tradicional. Sin embargo, una vez que la respuesta ha sido almacenada

en cache, observamos que éste absorbe completamente las cargas de tiempo y tráfico, convirtiéndose en una herramienta importante a la hora de mejorar la performance en la resolución de nombres.

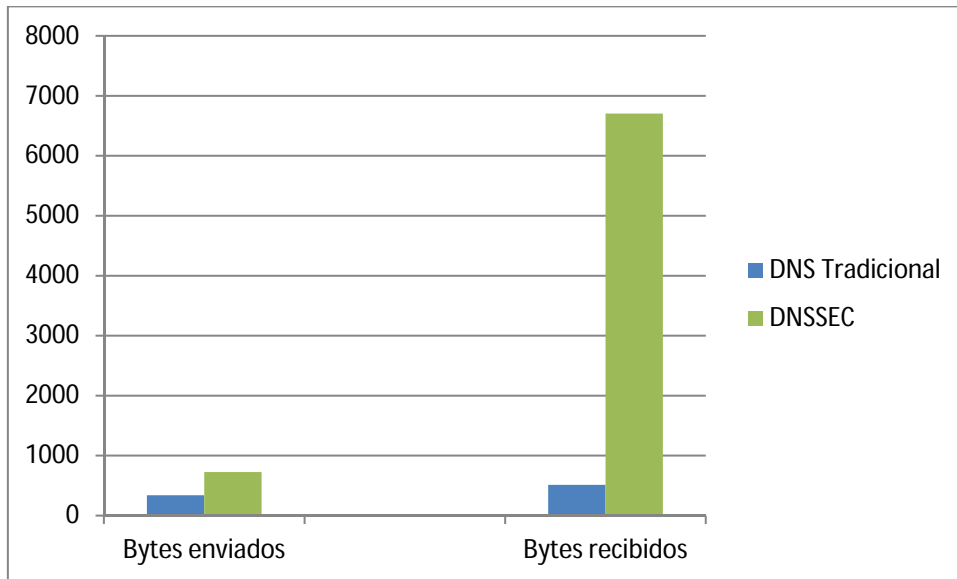


Figura 43: Bytes enviados y recibidos. DNS vs DNSSEC

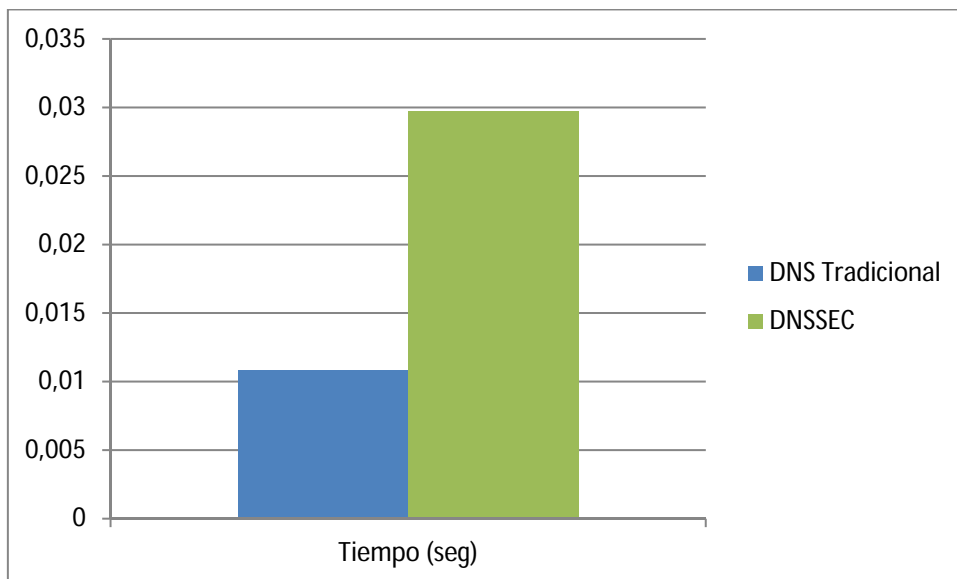


Figura 44: Tiempos de respuestas. DNS vs DNSSEC

A modo de conclusión, desde la perspectiva de un cliente DNSSEC, se observa que el incremento en la carga de tráfico se origina por la presencia de firmas almacenadas en los nuevos registros de recursos. Por otro lado, el incremento en los tiempos de respuestas, se origina por la ocurrencia en serie de los mensajes intercambiados en el proceso de validación de la cadena de confianza, necesario para garantizar la autenticidad e integridad de las respuestas obtenidas. Sin embargo, estos costos son efectivamente mitigados por la función de almacenamiento en cache.

Considerando ahora cual es el impacto de DNSSEC en el lado de un servidor Autoritativo, tomando los datos anteriores, se observa que, éste debe responder a dos consultas adicionales, la primera corresponde a la clave de firmado de zona (ZSK), almacenada en un registro DNSKEY, y la segunda corresponde al digesto de la clave de firmado de clave de zona (KSK) y la ZSK que firmó dicho digesto, ya que en este caso es autoritativo para la zona firmada y la zona que le fuera delegada. Como resultado, el servidor Autoritativo, recibe 3 consultas, con un total de 246 bytes, y devuelve 2165 bytes como respuestas. En conclusión, el costo para un servidor con características DNSSEC, tiene un incremento de carga de tráfico recibido en un factor de 3 y un incremento de carga de tráfico en respuestas de un factor de 12. La siguiente figura muestra éstas tres consultas mencionadas anteriormente, junto a los valores que representan en bytes.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.001982	10.0.0.12	10.0.0.11	DNS	77	Standard query 0xdb0a A www.miempresa.com
4	0.002895	10.0.0.12	10.0.0.11	DNS	77	Standard query 0xbd4d AAAA www.miempresa.com
7	0.005963	10.0.0.11	10.0.0.254	DNS	88	Standard query 0x9da3 A www.miempresa.com OPT
8	0.005973	10.0.0.11	10.0.0.254	DNS	70	Standard query 0x1c10 NS <Root> OPT
9	0.005975	10.0.0.11	10.0.0.254	DNS	88	Standard query 0xa6f9 AAAA www.miempresa.com OPT
10	0.081754	10.0.0.254	10.0.0.11	DNS	495	Standard query response 0x9da3 A www.miempresa.com NS nscom.com DS DS RRSIG A 10.0.0.253 OPT
11	0.082766	10.0.0.254	10.0.0.11	DNS	683	Standard query response 0x1c10 NS <Root> NS ROOT-SERVER RRSIG A 10.0.0.254 RRSIG OPT
14	0.085509	10.0.0.11	10.0.0.253	DNS	88	Standard query 0xd089 A www.miempresa.com OPT
15	0.086335	10.0.0.253	10.0.0.11	DNS	498	Standard query response 0xd089 A www.miempresa.com NS dns1.miempresa.com DS DS RRSIG A 10.0.0.2 OPT
16	0.087441	10.0.0.254	10.0.0.11	DNS	495	Standard query response 0xa6f9 AAAA www.miempresa.com NS nscom.com DS DS RRSIG A 10.0.0.253 OPT
19	0.089194	10.0.0.11	10.0.0.2	DNS	88	Standard query 0x023a A www.miempresa.com OPT
20	0.089204	10.0.0.11	10.0.0.2	DNS	88	Standard query 0x54b2 AAAA www.miempresa.com OPT
21	0.091098	10.0.0.2	10.0.0.11	DNS	1042	Standard query response 0x023a A www.miempresa.com A 10.0.0.100 RRSIG NS dns1.miempresa.com RRSIG A 10.0.0.2 RRSI...
22	0.091107	10.0.0.2	10.0.0.11	DNS	772	Standard query response 0x54b2 AAAA www.miempresa.com SOA dns1.miempresa.com RRSIG NSEC miempresa.com RRSIG OPT
23	0.091874	10.0.0.11	10.0.0.2	DNS	84	Standard query 0x4a63 DNSKEY miempresa.com OPT
24	0.092108	10.0.0.2	10.0.0.11	DNS	1238	Standard query response 0x4a63 DNSKEY miempresa.com DNSKEY DNSKEY RRSIG RRSIG OPT
25	0.093949	10.0.0.11	10.0.0.253	DNS	84	Standard query 0x57fa DS miempresa.com OPT
26	0.094159	10.0.0.253	10.0.0.11	DNS	459	Standard query response 0x57fa DS miempresa.com DS DS RRSIG OPT
27	0.095230	10.0.0.11	10.0.0.253	DNS	74	Standard query 0xbfc8 DNSKEY com OPT
28	0.095238	10.0.0.253	10.0.0.11	DNS	1208	Standard query response 0xbfc8 DNSKEY com DNSKEY DNSKEY RRSIG RRSIG OPT
29	0.097014	10.0.0.11	10.0.0.254	DNS	74	Standard query 0x556b DS com OPT

Figura 45: Captura de tráfico. Resolución DNSSEC

Señalado en rojo pueden verse las tres consultas que suman un total de 246 bytes, mientras que resaltado en azul, las correspondientes respuestas, con un total de 2165 bytes.

#### **6.4. Análisis de falla de proceso de validación**

En situaciones en donde el proceso de validación DNSSEC falla, las razones por la que esta situación puede presentarse son varias; caducidad de firma, la relación entre claves de zona padre y zona hijo puede fallar, corrupción en archivo de zona. Por otra parte y desde el punto de vista de un servidor Recursivo, la validación puede fallar, debido a que éste no soporta el algoritmo de firmado o ya sea que la clave inicial de validación puede haber expirado.

Desde la perspectiva de un cliente DNSSEC, no es sencillo identificar la razón por la cual el proceso de validación ha fallado, ya que cuando un servidor Recursivo encuentra un problema en la validación, responde a la consulta que recibió, con un estado que indica “*server fail*”

# 7. La alternativa DNSCurve

## 7.1. Introducción

Al momento de plantear los objetivos para el desarrollo del trabajo de tesis, se propuso comparar DNSSEC frente a DNSCurve, ya que ambas se presentaban como las principales opciones para dotar de seguridad al Sistema de Nombres de Dominio. Sin embargo, el anuncio, despliegue y adopción a nivel mundial de DNSSEC, hizo que se optara por exponer e implementar la primera opción.

Aun así, se consideró relevante presentar los aspectos teóricos de DNSCurve, justificar por qué no puede compararse con DNSSEC y una breve descripción de la situación actual de dicha alternativa.

## 7.2. Conceptos sobre DNSCurve

DNSCurve es un protocolo de DNS seguro diseñado por Daniel Bernstein [53], que hace uso de la criptografía de curva elíptica de alta seguridad y velocidad para mejorar drásticamente todas las dimensiones de la seguridad del DNS. DNSCurve trata de mitigar las vulnerabilidades conocidas del DNS estándar, mediante el uso de:

### 7.2.1. Confidencialidad

Protege la confidencialidad mediante el cifrado de las consultas y respuestas DNS, por lo que si alguien interceptara el tráfico de red, no se podría deducir la información en la comunicación entre un cliente y un servidor DNS. Cabe destacar que el cifrado sólo se realiza sobre los datos, no sobre las cabeceras como TCP o UDP.

### 7.2.2. Integridad

DNSCurve autentica criptográficamente todas las respuestas DNS, eliminando la posibilidad de la creación de mensajes falsificados.

### 7.2.3. Disponibilidad

Asociado con la anterior propiedad, DNSCurve, reconoce y descarta rápidamente todos los mensajes falsificados.

### 7.2.4. Primitivas Criptográficas

Para llevar a cabo las tareas criptográficas a una alta velocidad, DNSCurve utiliza Curve25519XSalsa20Poly1305, una combinación de las primitivas Curve25519, que implementa curvas elípticas y una variante del intercambio de claves de Diffie-Hellman, Salsa20, cifrado de flujo, y Poly1305, código de autenticación de mensajes; provistas por la librería de software NaCl (Network and Cryptographylibrary) [54].



NaCl provee una función de alto nivel llamada `crypto_box` que hace todo en un solo paso, convirtiendo el paquete en una caja cifrada, que lo protege contra espionaje y sabotaje.

### **7.2.5. DNSCurve frente a DNSSEC**

A primera vista, se observa claramente que la alternativa DNSCurve centra la seguridad en base a la encriptación, mientras que DNSSEC se basa en la autenticación. Ambas alternativas abordan la problemática de manera diferente. DNSCurve, hace uso de criptografía de clave pública “punto a punto” para asegurar el intercambio de los paquetes de datos DNS, las claves están asociadas a los servidores de nombres, con clave privada disponible de manera online en tales servidores, mientras que DNSSEC, hace uso de criptografía de clave pública para firmar Registros de Recursos, firmas precalculadas con claves asociadas a zonas, y así poder asegurar la exactitud en las respuestas de éstos registros.

DNSCurve, se presentó como un desarrollo personal, mientras que DNSSEC fue diseñado por los grupos de trabajo y organizaciones responsables de Internet, para garantizar la seguridad en de manera que sea compatible con todos los objetivos de DNS, es por esto que se considera que ambas alternativas no pueden ser comparadas. No obstante, DNSCurve parece ser una alternativa atractiva para garantizar la seguridad en la comunicación con un servidor DNS Resolver en un ámbito local, ya que a la fecha, es incompatible con la implementación de DNSSEC en los servidores raíz.

## 8. Conclusiones

Desde el año 2010, fecha en que la ICANN publica el despliegue de DNSSEC en los servidores Raíz, a la actualidad el mismo se ha convertido en la alternativa elegida por la comunidad global responsable de la administración de los dominios de nivel superior. Prácticamente el 100% de éstos dominios se encuentran firmados, con delegación en la zona raíz y aplican en la práctica las especificaciones descritas en los estándares.

A la fecha, es muy bajo el porcentaje de dominios de segundo nivel que adoptan DNSSEC, por lo que se espera que los resultados obtenidos y casos de pruebas descritos en este trabajo, contribuyan a manera de guía de uso para futuras implementaciones.

En Junio del año 2015, NIC Argentina informa que se realizó el firmado y anuncio para la zona .ar, sin embargo, resta aún alcanzar la última etapa del despliegue DNSSEC, lo que permitirá realizar una delegación segura desde dominios de niveles inferiores, con la finalidad de comprobar la cadena de confianza en todos los niveles de la jerarquía DNS.

De las observaciones que surgen del análisis comparativo en el proceso de resolución de nombres sobre DNS tradicional frente a un sistema con características DNSSEC, para el caso de un Servidor DNS Recursivo, se evidencia que la cantidad de tiempo total para resolver una consulta basada en DNSSEC es tres veces mayor frente a una consulta DNS tradicional, mientras que para el tráfico enviado el incremento es de 2,5 veces más y otras 12 veces más de carga de tráfico recibido. Sin embargo, una vez que la respuesta ha sido almacenada en cache, y ante futuras consultas para un mismo nombre de dominio, se observa que se absorben completamente las cargas de tiempo y tráfico, convirtiéndose éste almacenamiento en cache, en una herramienta importante a la hora de mejorar la performance en la resolución de nombres.

Desde la perspectiva de un cliente DNSSEC, (Servidor DNS Recursivo), se evidencia que el incremento en la carga de tráfico se origina por la presencia de firmas almacenadas en los nuevos registros de recursos. Por otro lado, el incremento en los tiempos de respuestas, se origina por la ocurrencia en serie de los mensajes intercambiados en el proceso de validación de la cadena de confianza, necesario para garantizar la autenticidad e integridad de las respuestas obtenidas.

En lo que respecta a la alternativa DNSCurve, la misma resulta de un desarrollo personal, alejado de los avances y anuncios a nivel mundial con respecto a DNSSEC, tal es así que es incompatible con la implementación de DNSSEC en las zonas raíz y de nivel superior. Sin embargo, dada la naturaleza jerárquica del Sistema de Nombres de Dominio, DNSCurve se puede integrar a niveles inferiores, en el proceso de comunicación con Servidores Resolver, ya que en aspectos de seguridad, se debe pensar en un conjunto de mecanismos y reglas, con el propósito de asegurar el flujo de información DNS

Considerando las observaciones descriptas en el proceso de configuración de DNSSEC, se puede afirmar que éstas características de seguridad se presentan en la práctica, como un conjunto de procedimientos complejos, que exige estar familiarizado con el

protocolo. La implementación de éstas extensiones, requiere tareas como la generación de claves para el firmado de zonas, firmado y refirmado de zonas, delegación de confianza, configuración de registros de eventos para la gestión de errores y control sobre tiempo de validez de claves de firmado. Es así que el proceso de migración a un ambiente con DNS Seguro, no es una tarea fácil y mucho menos rápida.

# 9. Apéndice: Contribuciones

## 9.1. Publicaciones con referato

- “*Análisis de impacto en la implementación de DNSSEC en un servidor DNS Recursivo*”. (Ernesto Sánchez, Daniel Arias Figueroa, Sergio Rocabado, Javier Díaz). XXII Congreso Argentino de Ciencias de la Computación, CACIC 2016 – San Luis Argentina. ISBN En construcción. <http://libros.unlp.edu.ar/index.php/unlp/catalog/book/620#accordion-1>. Pág. 854 - 862
- “*Una experiencia en la implementación de Extensiones de Seguridad para el Sistema de Nombres de Dominio*”. (Ernesto Sánchez, Daniel Arias Figueroa, Sergio Rocabado, Álvaro Gamarra, Javier Díaz). IV Congreso Nacional de Ingeniería en Informática/Sistemas de Información, CONAIISI 2016 – Salta Argentina ISSN 2347-0372. <https://drive.google.com/file/d/0Byd3HXEW6NCBNGJMY0JDOUxpdms/view>
- “*Un estudio comparativo en extensiones de seguridad para el Sistema de Nombres de Dominio*”. XVII Workshop de Investigadores en Ciencias de la Computación (WICC 2015). ISBN 978-987-633-134-0. Universidad Nacional de Salta. Autores: Ernesto Sánchez, Daniel Arias Figueroa, Sergio Rocabado, Verónica Agüero, Gustavo Molina. Abril 2015.
- “*DNS Curve, una alternativa para la seguridad en el Sistema de Nombres de Dominio*”. 2º Congreso Nacional de Ingeniería Informática/Sistemas de Información (CoNaIISI 2014). Universidad Nacional de San Luis. ISSN: 2346-9927. Autores: Ernesto Sánchez, Daniel Arias Figueroa, Gustavo Molina, Verónica Agüero. Noviembre 2014. <http://www.conaiisi.unsl.edu.ar/ProceedingsCoNaIISI2014.pdf>.
- “*Transaction Signature (TSIG). Una alternativa de seguridad para transferencias de zona DNS*”. XV Workshop de Investigadores en Ciencias de la Computación (WICC 2013). ISBN 978-987-28179-6-1. Paraná, Entre Ríos. Autores: Ernesto Sánchez, Juan Di Mauro, Jorge Silvera, Daniel Arias Figueroa. Abril 2013. [http://sedici.unlp.edu.ar/bitstream/handle/10915/27073/Documento\\_completo\\_.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/27073/Documento_completo_.pdf?sequence=1)
- “*Un caso de estudio en Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNSSEC). Resultados Parciales*”. XVIII Congreso Argentino de Ciencias de la Computación (CACIC). ISBN 978-987-1648-34-4. Bahía Blanca Buenos Aires. Autores: Ernesto Sánchez, Daniel Arias Figueroa, Sergio Rocabado, Javier Díaz. Octubre 2012. [http://redunci.info.unlp.edu.ar/files/indice\\_Cacic\\_2012.pdf](http://redunci.info.unlp.edu.ar/files/indice_Cacic_2012.pdf).

## **9.2. Actividades de transferencia**

- Curso de Postgrado “Extensiones de Seguridad para el Sistema de Nombres de Dominio”. Facultad de Ciencias Exactas – Universidad Nacional de Salta. ResCD-EXA n° 460/2013. Con una duración de 60hs. Setiembre 2013.

## **9.3. Proyectos relacionados a la investigación**

- Trabajo Final Integrador: “Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNSSEC)” aprobado el día 13 de mayo de 2013 con calificación EXCELENTE, de la carrera de ESPECIALIZACION EN REDES Y SEGURIDAD. Facultad de informática - Universidad Nacional de la Plata
- Proyecto CIUNSa, (Consejo de Investigación Universidad Nacional de Salta), Tipo C, N° 2123/0. “EXTENSIONES DE SEGURIDAD PARA EL SISTEMA DE NOMBRES DE DOMINIO”. Fecha de inicio: 01/01/2012, fecha de finalización: 31/12/2013.

# 10. Referencias

- [1] FALL Kevin R. - STEVENS W. Richard, TCP/IP Illustrated, Volume 1. The Protocols, 2da Edición, 2012, ISBN-13: 978-0-321-33631-6.
- [2] Internet Corporation for Assigned Names and Numbers, Internationalized Domain Names (IDNs), disponible en: <<http://www.icann.org/en/resources/idn>>, fecha de consulta: Junio2012.
- [3] IANA RootZoneDatabase, disponible en: <<http://www.iana.org/domains/root/db>>, fecha de consulta: Julio 2012.
- [4] Cooperative Research and Development Agreement between ICANN and US Department of Commerce, disponible en: <<http://www.icann.org/en/about/agreements/crada>>, fecha de consulta: Julio 2012.
- [5] EASTLAKE 3rd, D. RFC 6195: Domain Name System (DNS) IANA Considerations. Marzo 2011.
- [6] BELLIS, L. RFC 5966: DNS Transport over TCP - Implementation Requirements. Agosto 2010.
- [7] KUMAR, A, POSTEL, J, NEUMAN, C, DANZIG, P, MILLER, S. RFC 5966: Common DNS Implementation Errors and Suggested Fixes. Octubre1993.
- [8] EASTLAKE 3rd, D. RFC 2930: Secret Key Establishment for DNS (TKEY RR). Setiembre2000.
- [9] VIXIE P, GUDMUNDSSON O, EASTLAKE 3rd, D, WELLINGTON, B. RFC 2845: Secret Key Transaction Authentication for DNS (TSIG). Mayo2000.
- [10] VIXIE P, THOMSON, S, REKHTER, Y, BOUND, J. RFC 2136: Dynamic Updates in the Domain Name System (DNS UPDATE). Abril1997.
- [11] ARENDS, R, AUSTEIN, R, LARSON, M, MASSEY, D, ROSE, S. RFC 4033: DNS Security Introduction and Requirements. Marzo 2005.
- [12] DNSSEC.net, disponible en: <<http://www.dnssec.net>>, fecha de consulta: Agosto 2012.
- [13] TLD DNSSEC Report, Internet Corporation for Assigned Names and Numbers (ICANN). <[http://stats.research.icann.org/dns/tld\\_report/](http://stats.research.icann.org/dns/tld_report/)>. Fecha de consulta: Julio 2012.
- [14] KOLKMAN, O, GIEBEN, R. RFC 4641: DNSSEC Operational Practices, Setiembre 2006.
- [15] ANDREWS, M, WEILER, S. RFC 4431: The DNSSEC Lookaside Validation (DLV) DNS Resource Record, Febrero 2006.

- [16] WEILER, S. RFC 5074: DNSSEC Lookaside Validation (DLV), Noviembre 2007.
- [17] HUSTON, G. RFC 3172: Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa"). Setiembre 2001.
- [18] Root Servers Org, disponible en: <<http://root-servers.org/>>, fecha de consulta: Agosto 2012.
- [19] MOCKAPETRIS, P. RFC 1035: Domain Names - Implementation and Specification. Noviembre 1987.
- [20] HUSTON, G. RFC 3172: Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa"). Setiembre 2001.
- [21] Technet Microsoft, disponible en: <<http://technet.microsoft.com/es-es/library/cc781340%28WS.10%29.aspx>>, fecha de consulta: Agosto 2011.
- [22] MOCKAPETRIS, P. RFC 1034: Domain Names - Concepts and Facilities. Noviembre 1987.
- [23] OHTA, M. RFC 1995: Incremental Zone Transfer in DNS. Agosto 1996.
- [24] VIXIE, P. RFC 1996: A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY). Agosto 1996.
- [25] AUSTEIN, R. RFC 5001: DNS Name Server Identifier (NSID) Option. Agosto 2007.
- [26] VIXIE, P. RFC 2671: Extension Mechanisms for DNS (EDNS0). Agosto 1999.
- [27] Domain Name System (DNS) Parameters, disponible en: <<http://www.iana.org/assignments/dns-parameters>>. Fecha de consulta: Agosto 2012.
- [28] Agecic, Ciclo de charlas 2010 – CERT Uruguay, “Seguridad en DNS y DNSSEC”. Disponible en: [http://www.cert.uy/historico/pdf/DNSSEC\\_-\\_parte1\\_-\\_CERTificate.pdf](http://www.cert.uy/historico/pdf/DNSSEC_-_parte1_-_CERTificate.pdf). Fecha de consulta: Agosto 2012.
- [29] AITCHISON Ron, Pro DNS and BIND 10, 2da Edición, Apress, 2011, ISBN 978-1-4302-3049-6.
- [30] Steve Friedl's Unixwiz.net Tech Tips: An Illustrated Guide to the Kaminsky DNS Vulnerability. Disponible en: <http://www.unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html>. Fecha de consulta: Agosto 2012.
- [31] FERGUSON, P, SENIE, D. RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. Mayo 2000.
- [32] Internet Systems Consortium - Reportes de Seguridad. Disponible en: <https://www.isc.org/advisories>. Fecha de consulta: Agosto 2012.

- [33] EASTLAKE 3rd, D. RFC 2931: DNS Request and Transaction Signatures ( SIG(0)s ). Setiembre 2000.
- [34] ARENDS, R, AUSTEIN, R, LARSON, M, MASSEY, D, ROSE, S. RFC 4034: Resource Records for the DNS Security Extensions. Marzo 2005.
- [35] LAURIE, B, SISSON, G, ARENDS, R, BLACKA, D. RFC: 5155: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. Marzo 2008.
- [36] Amberg, Eric: Trusted name resolution with DNSSEC. In: Linux Magazine Issue 90, pp. 65. (2008).
- [37] Lewis, Edward: Comparing TLD DNSSEC Practices with RFCs. At the IEPG on the day before the 83rd IETF. Disponible en: <http://www.potaroo.net/iepg/2012-03-ietf83/index.html>. Marzo 2012.
- [38] ARENDS, R, AUSTEIN, R, LARSON, M, MASSEY, D, ROSE, S. RFC 4035: Protocol Modifications for the DNS Security Extensions. Marzo 2005.
- [39] Gieben, R. DNSSEC in NL – Final Report. Disponible en: <http://www.nlnetlabs.nl/downloads/publications/dnssec/dnssecnl/>. Fecha de consulta: Setiembre 2012.
- [40] Ager, B, Dreger, H, Feldmann, A. Exploring the Overhead of DNSSEC. Disponible en: <http://www.dnsops.gov/dnssec-perform.html>. Fecha de consulta: Setiembre 2012.
- [41] Kolkman, O. Measuring the resource requirements of DNSSEC. Disponible en: <http://www.ripe.net/ripe/docs/ripe-352>. Fecha de consulta: Setiembre 2012.
- [42] STJOHNS, M. RFC 5011: Automated Updates of DNS Security (DNSSEC) Trust Anchors. Setiembre 2007.
- [43] KOLKMAN, O, SCHLYTER, J, LEWIS, E. RFC 3757: Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag. Abril 2004.
- [44] IANA - Domain Name System Security (DNSSEC) Algorithm Numbers. Disponible en: <http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml>. Fecha de consulta: Setiembre 2012.
- [45] JANSEN, J. RFC 5702: Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC. Octubre 2009.
- [46] Dolmatov, V, Chuprina, A, Ustinov, I. RFC 5933: Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC. Julio 2010.
- [47] IANA - Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms. Disponible en: <http://www.iana.org/assignments/ds-rr-types/ds-rr-types.xml>. Fecha de consulta: Setiembre 2012.



- [48] Internet Society – Deploy360 Programme. Disponible en: <http://www.internetsociety.org/deploy360/dnssec/maps/5-stages/>. Fecha de consulta: Diciembre 2016.
- [49] Internet Society – DNSSEC DeploymentMaps. Disponible en: <https://elists.isoc.org/pipermail/dnssec-maps/>. Fecha de consulta: Diciembre 2016.
- [50] Dr. Richard Lamb, Sr. Program Manager DNSSEC – ICANN. <https://www.icann.org/profiles/richard-lamb>.
- [51] NIC ar Primera Ceremonia de DNSSEC para la zona .ar. <https://nic.ar/Enterate/Noticias/primera-ceremonia-de-dnssec-para-la-zona-ar>.
- [52] Niels L. M. van Adrichem. A measurement study of DNSSEC misconfigurations. Security Informatics – SpringerOpen Journal. Octubre 2015.
- [53] Daniel J. Bernstein. [https://es.wikipedia.org/wiki/Daniel\\_J.\\_Bernstein](https://es.wikipedia.org/wiki/Daniel_J._Bernstein)
- [54] CACE (Computer Aided Cryptography Engineering). NaCl: Networking and Cryptography library. <https://nacl.cr.yp.to/index.html>
- [55] U.S. Department of Homeland Security/Science & Technology (S&T). DNSSEC Tools Project. <https://www.dnssec-tools.org/>
- [56] Open DNSSEC Project. <https://www.opendnssec.org/>
- [57] Hardware Security Modules. <https://wiki.opendnssec.org/display/DOCREF/HSM>
- [58] Antti Ristimäki (CSC/Funet), Tuukka Vainio (University of Turku), Kaisa Haapala (CSC/Funet). “Guidelines for Deploying DNSSEC”. [http://services.geant.net/cbp/Knowledge\\_Base/Campus\\_Networking/Documents/CBP-07\\_guidelines-for-deploying-DNSSEC.pdf](http://services.geant.net/cbp/Knowledge_Base/Campus_Networking/Documents/CBP-07_guidelines-for-deploying-DNSSEC.pdf).

# 11. Abreviaturas

- IANA: Internet Assigned Numbers Authority.
- ICANN: Internet Corporation for Assigned Names and Numbers.
- CRADA: Cooperative Research and Development Agreements.
- RSSAC: Root Server System Advisory Committee.
- IETF: Internet Engineering Task Force.
- IAB: Internet Architecture Board.
- ARIU: Asociación Redes de Interconexión Universitaria (Autoridad para el registro de nombres de dominio a entidades educativas de la República Argentina).
- NIC: Network Information Center.
- BIND: Berkeley Internet Name Domain.
- FQDN: Fully Qualified Domain Name.
- SSL: Secure Sockets Layer.
- TLS: Transport Layer Security.
- AES: Advanced Encryption Standard.
- DES: Data Encryption Standard.
- IDEA: International Data Encryption Algorithm.
- RC4: Rivest Cipher 4.
- RSA: Rivest, Shamir y Adelman.
- PKI: Public-key Infrastructure.
- MAC: Message Authentication Code.
- CA: Certificate Authority.
- SEP: Secure Entry Point.
- KSK: Key Signing Keys.
- RIPE NCC: Regional Internet Registries Network Coordination Centre.
- NIST: National Institute of Standards and Technology.