UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN

PROYECTO FINAL DE CARRERA

ESTUDIO SOBRE LA IMPLANTACIÓN DE SEGURIDAD EN ROUTING INTERDOMINIO

AUTORA: PAULA GONZÁLEZ MUÑOZ
TUTOR: DR. FRANCISCO VALERA PINTOR

Septiembre de 2012

"Some people grow into their dreams, instead of out of them"
"That depends on whether your dreams are reasonable," said Vorsoisson.
"No, it doesn't." Miles smiled slightly. "It depends on how hard you grow."
Lois McMaster Bujold, Komarr

Índice general

Lista de Figuras

Lista de Tablas

Capítulo 1

Motivación y objetivos

1.1. Breve repaso a la historia de Internet

El origen de Internet tal y como lo conocemos puede trazarse hasta 1961, cuando Leonard Kleinrock escribió el primer documento dónde se hablaba de conmutación de paquetes [1]. Este documento es especialmente importante ya que, durante la misma época, los científicos del programa de investigación de computadoras en DARPA[1] estaban empezando a desarrollar el concepto de las redes de ordenadores como redes a nivel global que permitiesen interacciones sociales así como el acceso a información y programas en cualquier lugar del mundo [2]. Estas investigaciones supusieron un gran avance en las redes de ordenadores puesto que fueron la pieza clave en la transición de la conmutación de circuitos a la conmutación de paquetes.

El trabajo en los aspectos arriba mencionados continuó durante la década de los sesenta con el desarrollo de ARPANET, así como la definición de los cuatro pilares en la interconexión de redes según Kahn [3]:

- La conexión a Internet de una red no requiere cambios internos de esta, ya que cada red es en diseño y propósito independiente de las demás, aunque se conecten para compartir información.
- Las comunicaciones se realizarán utilizando técnicas de "mejor esfuerzo". Si un paquete no llega al destino, se reenvía.
- Se utilizarán cajas negras[2] para conectar las redes.
- No existirá un control global a nivel operativo.

Cómo podemos imaginar, estos cuatro pilares de la interconexión de redes derivarían en el desarrollo de algoritmos y técnicas para conseguir alcanzar las funcionalidades deseadas.

De todas las consideraciones que se hicieron necesarias [4] para alcanzar con éxito estas metas, las más importantes a efectos de este documento son:

- El desarrollo de algoritmos de enrutamiento tanto a nivel de enlace como a nivel de red, así como de los sistemas de hardware necesarios para interpretarlos, es decir, el desarrollo de los protocolos TCP/IP y de los routers además de políticas de direccionamiento a nivel global.
- El desarrollo de funciones de enrutamiento que permitiesen pasar los paquetes entre las distintas redes desde una dirección de origen a una destino, independientemente de la longitud del camino.

[1] Defense Advanced Research Projects Agency
[2] Lo que posteriormente se denominaría routers y puertas de enlace

Todos estos avances consiguieron que, poco a poco, cada vez más organizaciones y, por tanto, sus redes se uniesen a Internet, lo que ocasionó un crecimiento exponencial de la misma que hizo que se tuviese que replantear cómo gestionarla. Los esfuerzos se realizaron en dos campos. En primer lugar, se definió un sistema de nombres de dominio (DNS) [5] [6] , para facilitar un mecanismo escalable de nombres de dominio jerárquicos a direcciones de red. Al mismo tiempo, el aumento de equipos conectados a la red hacía que las capacidades de los routers empezasen a verse seriamente limitadas, ya que no era suficiente con un algoritmo de enrutamiento distribuido de manera uniforme por todos los routers de Internet.

Esto hizo necesario el implantar un modelo jerárquico dónde se distinguiera entre qué ocurría dentro de una región de Internet y fuera.

En 1982 Eric Rosen y Bolt Beranek, plantearon el crecimiento estructurado de la red en su definición del protocolo EGP (Exterior Gateway Protocol) [7]. Su aportación fue especialmente importante puesto que es en este documento dónde se definen conceptos vitales para el enrutamiento interdominio tal y como hoy lo conocemos como, por ejemplo, la definición de sistema autónomo (AS) y los distintos tipos de dominios (stub o núcleo en función de su posición relativa en la red).

Este concepto se retomaría posteriormente por Paul Tsuchiya [8], quien plantearía la evolución de Internet no como una red centralizada, sino como un conjunto de divisiones administrativas (correspondiente cada una a un AS) compuesta cada una por una serie finita de nodos[3]. De esta forma, cada división administrativa se gestionaría de forma independiente y una hipotética autoridad central solo controlaría la forma que tienen dichas divisiones de informarse unas a otras sobre su contenido interno.

Este cambio de paradigma, el paso de una gestión de red centralizada a una descentralizada, añade robustez al sistema al forzar a que el funcionamiento de cada AS sea independiente de los demás, evitando que un fallo local de enrutamiento se propague. Por tanto, surge la necesidad de dividir los protocolos de enrutamiento en protocolos internos (IGP) y protocolos externos (EGP[4]). La importancia de esta división de protocolos es que aunque todos los sistemas tienen que comunicarse utilizando el mismo protocolo EGP, cada uno puede utilizar el protocolo IGP que mejor convenga a la topología y necesidades de su AS. Dichas definiciones han tenido tal aceptación que, desde entonces, se toman como un axioma fundamental del enrutamiento y se han mantenido hasta la actualidad.

Sin embargo, en la propia definición de EGP se fue consciente de las limitaciones del protocolo ya que no era capaz de generalizar para todo tipo de topologías de red[5]. Aparte de este problema, los diseñadores dejan entrever un posible problema de seguridad al no estar protegido el protocolo de ninguna manera ante inserciones maliciosas de mensajes que "falseen" la topología de la red.

Debido a las limitaciones del protocolo EGP, ha sido necesario el desarrollo de nuevos protocolos para suplir dichas carencias. El protocolo que consiguió imponerse, en el año 1995, fue BGP-4[6] [9, 10]. Se trata de un protocolo capaz de establecer la mejor ruta entre dos AS independientemente de la topología del sistema, lo cual soluciona uno de los principales problemas del protocolo EGP: el hecho de ser solo válido para topologías de tipo árbol.

1.2. BGP y los secuestros de prefijos

El protocolo BGP se basa en que los distintos sistemas autónomos anuncien los prefijos IP que están dentro de su control administrativo y sean los algoritmos de decisión de los routers los que establezcan la ruta óptima para llegar a ellos desde cualquier lugar de la red. Cómo ocurre con muchos protocolos, a la hora de diseñar su seguridad se hizo pensando en

[3] Podemos ver como estas ideas son muy similares a las postuladas por Kahn en 1972.

[4] Es importante notar que no se debe confundir el protocolo EGP con los protocolos EGP en general, conjunto del que forma parte junto a BGP.

[5] Si los AS no conforman una topología tipo árbol, en concreto un spanning tree, el protocolo no es capaz de evitar la formación de bucles a la hora de establecer rutas.

[6] En 2002 se publicaría una revisión del protocolo eliminando así ambigüedades y problemas que se habían detectado durante los primeros años de operación

un atacante externo a la red y, en general, siempre se considera de vital importancia la buena administración de los routers y su control de acceso [11]. Sin embargo, se establece una relación de confianza mutua entre todos los sistemas autónomos por lo que si uno de ellos manda anuncios erróneos, bien por errores de configuración bien por otro tipo de motivos, pueden darse situaciones en las que el funcionamiento de la red se vea comprometido (p.ej. el llamado "secuestro de prefijos").

Se denomina "secuestro de prefijos" a un fenómeno por el cuál un sistema autónomo anuncia como suyo un prefijo dentro del qué se agrupan otros que no le pertenecen. Para entender bien en que consiste este ataque[7], debemos explicar qué es un prefijo de red [12]: cuando definimos la dirección IPv4 de una máquina[8], estamos definiendo en cuatro octetos tanto a qué red pertenece como a qué máquina nos referimos dentro de dicha red. Dado que la longitud de cada campo no está fijada para permitir mayor flexibilidad de configuración por parte de las distintas subredes, se incluye un segundo campo que determina cuántos bits indican la red y cuántos la máquina. De esta forma, un prefijo IP de longitud diez indica que los diez primeros bits contienen la dirección de red, mientras que los veintidós bits siguientes contienen información del sistema destino dentro de esa red[9].

El secuestro de prefijos consiste en que tenemos un sistema autónomo A que es dueño del prefijo de red 124.207.128.0/17[10] y así lo anuncia por BGP. Sin embargo, un sistema autónomo B, ajeno a A, empieza a anunciar por algún motivo el prefijo 124.207.159.0/24. Dado el algoritmo de selección de rutas de BGP, los routers preferirán el prefijo anunciado por B antes que el prefijo anunciado por A, ya que es más preciso, haciendo que parte del tráfico que va dirigido a A (en concreto todo el que va dirigido a aquellos sistemas que tengan una dirección del tipo 124.207.159.X se vea desviado a B) aunque con una ejecución normal del algoritmo dichos paquetes no tuviesen que pasar por B.

Existe otra manera de secuestrar prefijos en BGP que consiste en que dado un prefijo de destino P, un router C anuncie que posee una ruta más corta para llegar a P, aunque esto no sea cierto. Este anuncio hará que a la hora de establecer las rutas para enviar paquetes a los sistemas que tienen el prefijo P la red prefiera mandarlos por C antes que por cualquier otra máquina.

Uno de los primeros casos documentados de secuestro de prefijos data de 1997[13], cuando un router perteneciente al AS 7007 filtró de manera accidental (probablemente debido a un fallo en el router) parte de su tabla de rutas a Internet, ocasionando lo que se denominó un agujero negro en Internet. Lo que ocurrió fue que muchas de estas rutas eran prefijos /24, más concretos que los que se anunciaban en la red BGP, indicando como parte de su ruta que pasaban por el AS 7007 y haciendo que estas se conviertan en rutas preferidas para los algoritmos de selección de rutas.

Posteriormente ha habido casos similares, siendo los dos más famosos los de Pakistán en 2008 [14] y de China en 2010 [15]. En el caso de Pakistán, un intento de impedir que se pudiese acceder a Youtube desde dentro de las fronteras de este país en 2008 ocasionó una caída total del servicio a nivel mundial desde las 18.47h del 29 de febrero de 2008 hasta las 21.01h del mismo día. Los efectos sobre la tabla de rutas de BGP pueden verse en la figura 1.1.

En el caso de China, el AS 23724 pasó de anunciar apenas cuarenta prefijos a anunciar más de treinta y siete mil. Esto provocó un apagón total o parcial de Internet en más de veinte países durante aproximadamente quince minutos, tal y como puede observarse en la figura 1.2 y en la tabla 1.1.

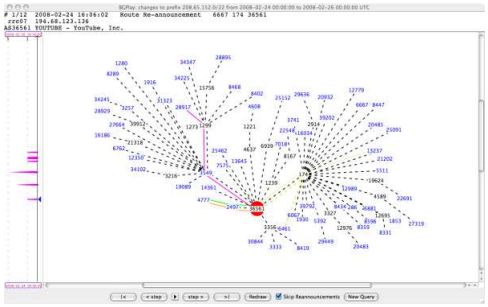
Otro caso que puede considerarse como secuestro de prefijos fue el de Egipto en 2011 [16] [17], dnde, para evitar la coordinación de protestas ciudadanas, el gobierno decidió ordenar a sus proveedores de Internet la desconexión del resto de la red. Esto se hizo generando una cantidad masiva de mensajes de eliminación de aquellos prefijos pertenecientes a Egipto, lo que provocó que, de manera efectiva, Egipto desapareciese del mapa de Internet.

[7] En el ámbito de este documento, denominaremos "ataque" a todo aquel fenómeno que implique un funcionamiento no deseado de la red de manera provocada bien por un defecto en el sistema, bien por una intrusión de agentes maliciosos externos

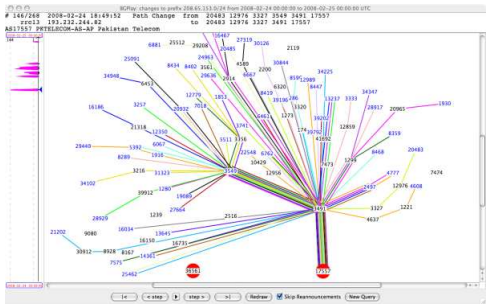
[8] Salvo que se indique lo contrario, nos referiremos siempre a direcciones IP versión 4

[9] Dicho sistema destino puede ser una máquina final o una nueva subred

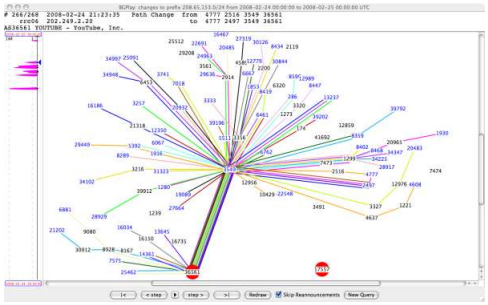
[10] Por convención, la notación utilizada en una dirección IP es W.X.Y.Z/L, donde W, X, Y y Z son los cuatro octetos de la dirección IP y L indica el número de bits correspondientes al prefijo



(a) Antes del secuestro, con las tablas de rutas apuntando al AS de Youtube



(b) Durante el secuestro, con las tablas de rutas apuntando al AS Pakistaní



(c) Después del secuestro, una vez recuperado el servicio

Figura 1.1: Rutas BGP antes, durante y después del incidente de Pakistan-Youtube. Fuente: RIPE

1.3. Iniciativas para hacer BGP más seguro

El hecho de que Internet se haya convertido en un recurso estratégico de la economía de los países, sumado a los distintos casos de fallos de funcionamiento y ataques, ha hecho que distintos grupos se hayan interesado por hacer BGP seguro ante este tipo de ataques. Desde principios de siglo ha habido múltiples iniciativas, varias de ellas dentro del IETF, aunque ha habido investigaciones ajenas al mismo. A pesar de la diversidad de métodos estudiados, y que mencionaremos en breve, todas estas soluciones coinciden con la necesidad de tener una infraestructura de clave pública (PKI) que contenga información sobre quién puede anunciar qué.

La primera de estas propuestas, tuvo una última versión antes de ser descartada y fue desarrollada en el borrador del protocolo Secure BGP (S-BGP)[18] del año 2003. Su funcionamiento consistía en que cada router verificase que todos los routers por los que ha pasado el paquete fueran miembros legítimos de la red BGP mediante la firma de los datos del paquete. Uno de los principales inconvenientes era el uso de RSA-2048, ya que la longitud de las firmas en RSA-2048 es de dos mil cuarenta y ocho bits, un ciencuenta por ciento de la longitud máxima de un mensaje BGP. Al medir el tiempo de firma y verificación[11] obtenemos los siguientes resultados:

Si comparamos los resultados de la tabla 1.2 con los datos de la tabla 4.2, podemos ver como las velocidades son claramente insuficientes para poder conseguir un funcionamiento óptimo de BGP.

Otra posible solución fue el protocolo Secure Origin BGP (soBGP) [19] cuyo último borrador data del año 2006. En este caso, el esfuerzo de la seguridad se centraba en verificar que los AS que originaban los anuncios de prefijos estaban autorizados por los dueños legales para realizar tales anuncios. Para esto cada router consultaba una base de datos local que indicaba qué AS podía anunciar qué prefijo.

[11] Las medidas se han realizado sobre la máquina de pruebas 'moscardón' cuyas características se especifican en la sección 4.2.1

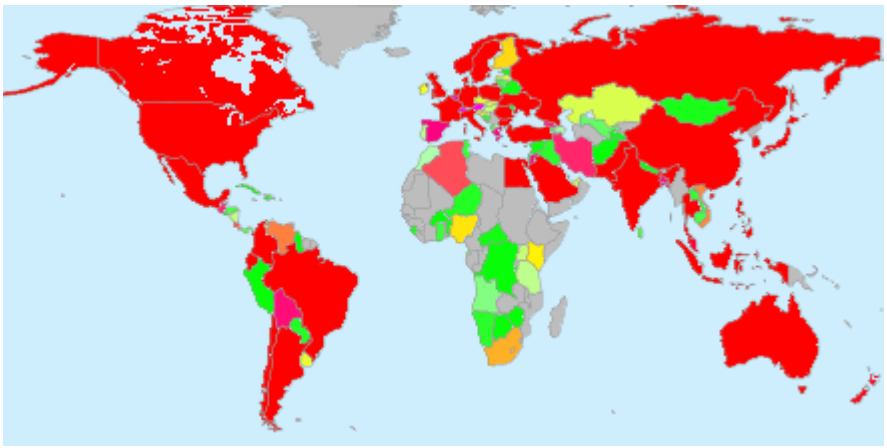


Figura 1.2: Países afectados por el secuestro de prefijos de China en 2010. Fuente: BGPMON

Una tercera línea de investigación sobre como hacer BGP más seguro fue RIVET [20] en el año 2011. Se trata de una combinación de las dos anteriores donde los routers tenían información sobre la topología de la red, de forma que solo aceptaban actualizaciones que provinieran de sus vecinos suponiendo que el origen de la actualización sea un origen válido. Esta información se obtendría, de nuevo, a través de una infraestructura de clave pública.

Ninguna de estas tres alternativas se adoptó y no ha sido hasta 2010, con el secuestro de prefijos de China, cuando se han retomado las distintas líneas de trabajo, comenzando con el desarrollo de una infraestructura de clave pública (RPKI) entre 2010 y 2012 dado que, cómo hemos visto, todas las alternativas de seguridad que se han planteado a lo largo de los años requieren una.

1.4. Objetivo del proyecto y estructura

El objetivo de este proyecto reside en estudiar y evaluar tres propuestas para dar seguridad a BGP. A lo largo de dicha evaluación trataremos de aislar las posibles debilidades de dichas propuestas y analizarlas en profundidad para comprobar si, efectivamente, se trata de posibles problemas para la implantación y ejecución de la proyecta dentro del entorno de las redes BGP o si no se trata de un inconveniente real.

A lo largo de este documento estudiaremos las soluciones que se proponen actualmente para conseguir aumentar la seguridad en BGP. Para ello primero haremos un repaso al estado del arte de la situación actual en el desarrollo de tres de las soluciones que hay planteadas para dar seguridad al protocolo.

A continuación estudiaremos los inconvenientes de dichas tecnologías, estudio que dividiremos en dos partes: una donde hablaremos de RPKI y otra segunda donde hablaremos de la validación de prefijos (BGP-PFX) y de BGP seguro (BGPSEC).

Una vez que hayamos discutido los inconvenientes de las tres tecnologías, enunciaremos las conclusiones del proyecto así como las posibles líneas de trabajo futuro sobre seguridad en routing interdominio a partir de lo expuesto en este proyecto fin de carrera. Finalmente, expondremos un presupuesto desglosando el coste de la elaboración de todo este estudio.

País	Número de prefijos
Estados Unidos	10547
China	10298
República de Corea	2857
Australia	1650
México	885
India	719
Japón	604
Brasil	592
Francia	508
Rusia	471
Canadá	425
Tailandia	372
Indonesia	369
Italia	338
Colombia	328
Reino Unido	322
Chile	302
Suecia	281
Hong Kong	276
Ecuador	272
Dinamarca	227

Tabla 1.1: Lista de los países más afectados por el secuestro de prefijos de China. Fuente: BGPMON

Tiempo de firma	Tiempo de verificación	Firmas / segundo	Verificaciones / segundo
0.002272s	0.000070s	440.2	14298.1

Tabla 1.2: Tiempos de ejecución para RSA-2048

Capítulo 2

Estado del arte

Internet es un recurso estratégico tanto desde un punto de vista geopolítico como desde un punto de vista económico. Partiendo de la premisa de que esta no se puede apagar, todos los días hay información crítica[1] circulando por Internet. Como hemos visto en la sección 1.2, en la práctica, aunque Internet no pueda apagarse, hay que tener en cuenta que, bien por errores de configuración bien por ataques reales o decisiones político-económicas, se ha conseguido el mismo efecto de pérdida de servicio atacando no a Internet como tal, sino al protocolo BGP.

A lo largo del presente capítulo, analizaremos las tres principales iniciativas en las que trabaja actualmente el IETF[2] para hacer de BGP un protocolo más robusto.

2.1. Resource Public Key Infraestructure, RPKI

Como se menciona en la sección 1.3, las distintas iniciativas de seguridad en BGP pasan por tener una infraestructura de clave pública para, de esta manera, poder distribuir a los miembros de la red BGP información sobre quienes pueden distribuir que prefijos IP. Es por ello que en 2007 empezó a trabajar en lo que culminaría con la definición de RPKI [21]. RPKI tiene una arquitectura basada en tres elementos principales:

- Una infraestructura de clave pública (PKI).
- La firma de los objetos importantes para el enrutamiento.
- Un repositorio distribuido que contiene los objetos firmados y los objetos PKI para que todos los miembros de la red BGP[3] puedan acceder a la información de seguridad.

Haciendo uso de esta estructura, cualquier entidad (empresa, organización o persona) puede asegurar a otras entidades que es el legítimo concesionario de un recurso de Internet[4].

El poseer este tipo de garantías sobre la propiedad de recursos tiene muchos usos potenciales como, por ejemplo, evitar que terceros no autorizados puedan publicar información sobre rutas concernientes a esos recursos, además de que es una infraestructura PKI diseñada especialmente para poder dar apoyo a iniciativas de seguridad para BGP [21]. Es interesante destacar que, a la hora de definir las capacidades y estructuras de RPKI, se ha tratado de utilizar estándares actuales del IETF siempre que ha sido posible. En particular, se utilizan los certificados X.509 [22] y sus extensiones para las direcciones IP y los identificadores de sistemas autónomos [23], además de la sintaxis para mensajes cifrados (CMS) [24].

[1] Desde el punto de vista del tiempo, el coste o la seguridad.

[2] Internet Engineering Task Force

[3] Nótese que, aunque para nuestro caso concreto hablaremos de la red BGP, esta estructura puede utilizarse para proporcionar mayor seguridad en otros protocolos.

[4] Se considera recurso de Internet a un conjunto de prefijos IP o a un conjunto de números de sistemas autónomos.

La estructura de RPKI se basa en la existencia de tres elementos bien diferenciados. En primer lugar tenemos los certificados, que podrán ser de dos tipos (certificado de Autoridad de Certificación, CA, o certificados de Usuario Final, EE) en función de para que se vayan a utilizar. Después, tenemos los objetos que se firmaran con dichos certificados como, por ejemplo, los manifiestos o las autorizaciones de origen de rutas (ROAs). Finalmente, debe existir un sistema de repositorios distribuidos para que los objetos firmados sean accesibles por los proveedores de servicios de Internet (ISPs). A continuación procederemos a describir de manera más detallada cada uno de los elementos RPKI para, finalmente, describir el funcionamiento del conjunto. Sus usos por las extensiones de seguridad para BGP los detallaremos en las secciones 2.2 y 2.3.

2.1.1. Certificados

Los certificados en RPKI garantizan que unos determinados recursos de Internet han sido asignados a un determinado concesionario. Cabe destacar que los certificados no garantizan la identidad del concesionario, al contrario de lo que ocurre en otros contextos. Solo se garantiza que esa entidad tiene asignados los recursos que firma el certificado. En otras palabras, los certificados en RPKI sirven para proporcionar *autorización*, no *autenticación* [21]. Podemos distinguir dos tipos de certificados: por un lado están los que acreditan a las autoridades de certificación y que denominaremos certificados CA, mientras que por otro tenemos los certificados que se utilizan para la firma y verificación de los recursos, llamados certificados de usuario final (EE).

Certificados CA

Estos certificados son necesarios porque todo concesionario de recursos debe poder subasignar, a su vez, recursos a sus arrendatario. En la estructura de Internet esto tiene especial significado ya que la asignación de recursos se hace en forma de árbol donde en la raíz se encuentra la IANA[5], por debajo de ella están los registros regionales de Internet (RIR)[6], en el siguiente escalón los registros nacionales de Internet (NIR), después los registros locales de Internet (LIR) y, finalmente, los proveedores de servicios de Internet (ISP). Dado que estos niveles tienen connotaciones geopolíticas y geoeconómicas, es posible que en determinadas zonas geográficas existan todos mientras que en otras se pase directamente del RIR a los distintos ISPs, tal y como puede verse en la figura 2.1.

Además, aquellas entidades que no subasignan sus recursos entre otras también necesitan los certificados CA para poder gestionar sus asignaciones internas. Un ejemplo sería una entidad propietaria de tres sistemas autónomos distintos que desea que cada uno pueda anunciar un subconjunto determinado de las IPs que tiene asignadas.

Dado que cada certificado CA implica una subasignación de recursos a un concesionario, aquellas entidades que posean recursos de varios organismos tendrán varios certificados CA donde es posible que los datos de identificación no coincidan. Esto es debido a que a la hora de expedir los certificados, como ya hemos comentado, no se proporciona autenticación. También es posible que una autoridad certificadora genere un certificado CA por cada asignación de recursos lo cual facilita la gestión de los recursos y los certificados, aunque incrementa el número de certificados a utilizar.

Finalmente, estos certificados CA son los que se utilizan para expedir los certificados EE que firman los recursos de Internet. Por tanto, si una entidad pierde la adjudicación de un conjunto de recursos, la propagación de dicho cambio a través de todo el repositorio es tan sencilla como limitarse a revocar el certificado CA que contenga la autorización sobre esos recursos.

La estructura de firmas puede verse en la figura 2.2, donde se ilustra qué certificados se utilizan para firmar qué objetos. Como podemos ver en la figura, los certificados de tipo CA se pueden utilizar en múltiples ocasiones mientras que los certificados de tipo EE, que se explican en la sección 2.1.1, se utilizan para firmar un único objeto.

[5]Internet Assigned Numbers Authority

[6]Los cinco RIR son: AfriNIC (África), APNIC (Asia y el Pacífico), ARIN (América del norte), LACNIC (Latino América y el Caribe) y RIPE (Europa).

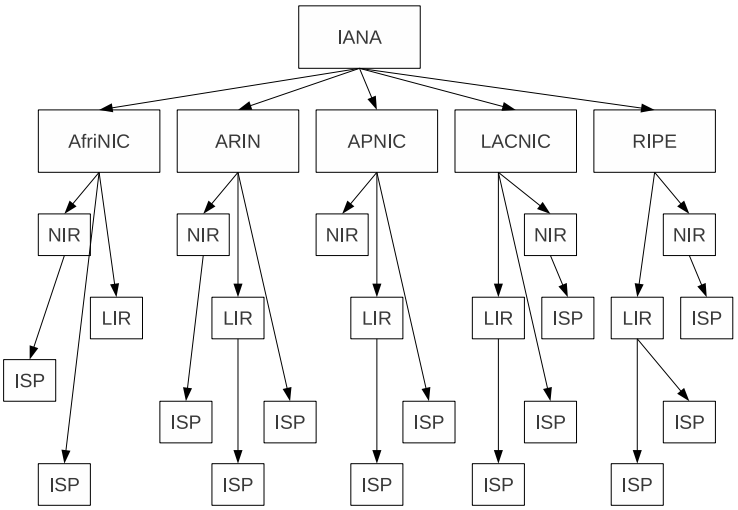


Figura 2.1: Ejemplo de la estructura de los concesionarios de recursos de Internet

Certificados EE

Estos certificados se utilizan para firmar los recursos de Internet y su formato se define en la RFC6487 [25]. Si bien las asignaciones de recursos se realizan mediante la expedición de certificados CA por parte de la autoridad de nivel superior (o la misma entidad para facilitar la administración de los mismos) son los certificados EE los que firman los recursos y los que se utilizaran para verificar que el recurso realmente pertenece a quien lo está utilizando, o que su uso por esa entidad ha sido autorizado por el concesionario.

Para los casos de las ROAs (Resource Origin Autorization) y los manifiestos (documentos que muestran el contenido del directorio al que pertenecen en RPKI), existe una correspondencia uno a uno entre certificados EE y objetos, es decir, cada certificado EE puede firmar *exactamente* un único objeto. Esto es así para que si se desea revocar todas las decisiones que se hayan tomado asumiendo que una entidad es la concesionaria de un determinado objeto, dicha operación pueda realizarse limitándose a revocar el certificado EE y, por tanto, invalidando la firma del objeto. Además, dado que cada certificado EE firmará un único objeto durante toda su vida útil, se puede eliminar su clave privada una vez se haya firmado dicho objeto, simplificando así la gestión de claves.

Hay un único caso particular de interés que es el de la firma secuencial de manifiestos. Aunque, según hemos explicado, los certificados EE se utilizarán para firmar un único objeto, es posible que una autoridad certificadora escoja utilizar la misma clave privada de un certificado EE para firmar una secuencia de manifiestos. Dado que para cada instancia de la autoridad certificadora hay un único manifiesto válido en un momento dado, se sigue cumpliendo el principio un certificado EE - un objeto firmado para un instante temporal.

Los certificados EE se transmitirán junto con el objeto firmado tal y como se define en la RFC6488 [26]. Esto simplifica también la gestión de certificados ya que, de esta forma, los certificados EE solo aparecen en el repositorio como parte del objeto firmado.

Terceros de confianza

En general, todos los sistemas que basan su confianza en certificados (siendo los sistemas PKI un caso particular) requieren de la existencia de terceros de confianza como raíces de los arboles de certificados. Dada la estructura jerárquica que se ha comentado en la sección 2.1.1, la IANA y los cinco RIRs son candidatos lógicos para actuar de terceros de confianza aunque entra dentro del ámbito de decisión de cada entidad usuaria de RPKI elegir los terceros de confianza que desee.

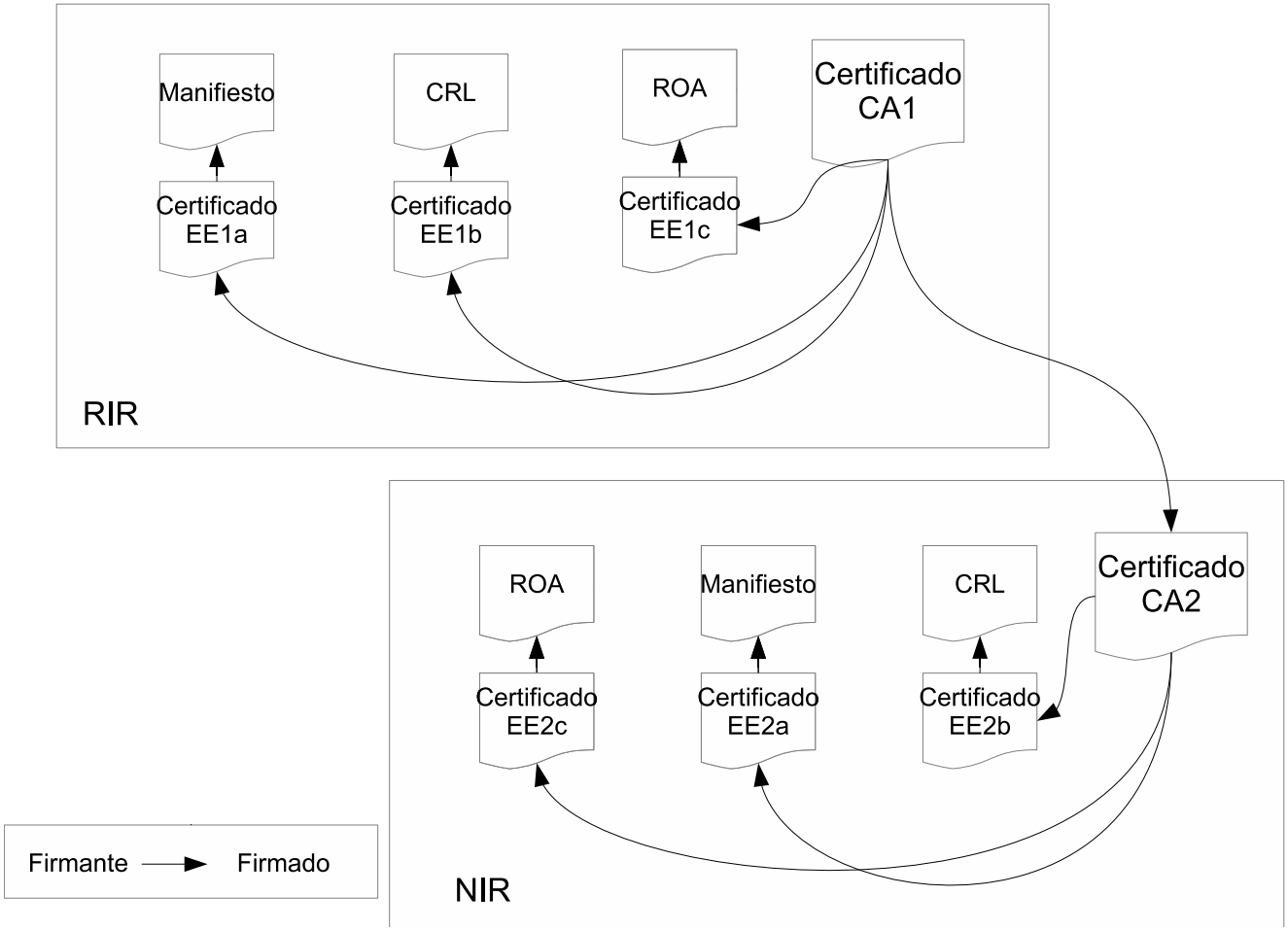


Figura 2.2: Ejemplo de estructura de directorios dentro de RPKI

Es posible que una entidad usuaria llegase a crear un tercero de confianza para poder expedir certificados dentro del espacio de recursos que tiene asignados. De hecho, aquellas entidades lo suficientemente grandes como para utilizar BGP de manera interna, y que tengan un espacio privado de direcciones IP [27] deberían crear un tercero de confianza acorde con los requisitos de la RFC3779 [23] para poder expedir todos los certificados CA necesarios para gestionar su espacio privado de direcciones IP.

Sin embargo, una entidad que decidiese depender únicamente de los terceros de confianza que crease y mantuviese por sí misma se arriesgan a ser vulnerable a errores de asignación, aunque sería una vulnerabilidad local.

Como ocurre con todos los sistemas PKI, un tercero de confianza (TA) no es más que un certificado CA auto-firmado. Sin embargo, en RPKI se define un formato para almacenar la información necesaria para poder verificar al TA de una manera sencilla [28]. A este formato se le denomina Trust Anchor Locator (TAL). Una ventaja de definir el TAL es el hecho de que, de esta forma, la información del TA puede variar con el tiempo sin necesidad de volver a distribuirla por todo el sistema. A pesar de esto, el TAL es totalmente análogo a la estructura TrustAnchorInfo definida en la RFC5914 [29]. Finalmente, la estructura de un TAL es:

1. Una URI rsync [30]
2. Una ruptura de linea <CLRF> o <LF>
3. Un campo subjectPublicKeyInfo [22] en formato DER codificado en Base64.

Respecto a esta estructura, es la URI rsync la que requiere más comentarios al respecto. El primero es que dicha URI debe apuntar a un objeto, no a un directorio o cualquier otro tipo de colecciones de objetos. Además, dicho objeto debe ser un certificado CA auto-firmado, que es como se define un tercero de confianza, acorde con la RFC6487 [25]. Este certificado será el tercero de confianza en el camino de descubrimiento [31] y de validación [22].

Además, hay que tener en cuenta que, dado que la definición del TAL permite que haya modificaciones en el TA que son transparentes para el resto del sistema, esto tiene varias consecuencias:

- La URI que apunta al TA debe ser una URI estable.
- La clave pública del TA debe ser estable y no debe modificarse por ningún motivo que no sea un cambio de clave (p. ej., modificaciones en los prefijos que tienen su origen de verificación en el TA o la renovación del TA antes de su fecha de expiración).
- Dada la naturaleza estable de campos como la clave pública, es muy recomendable que el TA trabaje de manera off-line. Para ello, puede expedir un certificado CA que sea el que trabaje en línea para expedir o verificar todos los certificados que dependan de el TA.

Finalmente, el TA es el único certificado que no tiene un CRL (Certificate Revocation List) ni aparece en el correspondiente manifiesto. Esto se debe a que es auto-firmado. Por ello, si se desea revocar, la única manera de hacerlo es borrando el objeto firmado que aparece en la URI que figura en el TAL. Este es otro motivo por el que es especialmente importante almacenarlo de manera off-line, ya que su modificación por parte de un atacante tiene consecuencias en todo el sistema RPKI.

2.1.2. Objetos Firmados

Dentro de RPKI existen dos tipos de objetos que se firmarán con las claves privadas de los certificados EE. Estos son las autorizaciones de origen de rutas (ROAs) y los manifiestos. Cada uno de estos objetos es muy distinto al otro ya que en el caso de las ROAs, estas contienen información directa sobre los recursos de Internet, mientras que los manifiestos contienen información sobre los objetos firmados por una autoridad de certificación.

Manifiestos

Como ya hemos dicho, los manifiestos no son más que una lista firmada de todos los objetos firmados por una autoridad de certificación excluyendo el propio manifiesto [32]. El manifiesto asociado con el punto de publicación de una autoridad de certificación contiene:

- El conjunto de todos los certificados que no hayan sido revocados ni estén caducados firmados por la autoridad de certificación[7].
- La lista de revocación de certificados (CRL) más reciente que haya sido expedida por la autoridad de certificación.
- Todos los objetos publicados que puedan verificarse utilizando certificados EE expedidos por la autoridad de certificación.

Dado que cabe la posibilidad de que haya más de una instancia de una autoridad de certificación conviviendo en un mismo punto de publicación, es posible encontrarse varios manifiestos cada uno con información de una instancia distinta.

La estructura de un manifiesto RPKI será la que sigue [32]:

- Primero poseerá un campo tipo eContentType definido como id-ct-rpkiManifest y con un valor númerico de 1.2.840.113549.1.9.16.1.26.

  id-smime OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs9(9) 16 }
  id-ct OBJECT IDENTIFIER ::= { id-smime 1 }
  id-ct-rpkiManifest OBJECT IDENTIFIER ::= { id-ct 26 }

- El contenido de un manifiesto es de tipo ASN.1 y estará codificado según se define en la recomendación ITU-T[8] X.690 [33]. Dicho contenido viene definido de la siguiente forma:

  Manifest ::= SEQUENCE {
  version [0] INTEGER DEFAULT 0,
  manifestNumber INTEGER (0..MAX),
  thisUpdate GeneralizedTime,
  nextUpdate GeneralizedTime,
  fileHashAlg OBJECT IDENTIFIER,
  fileList SEQUENCE SIZE (0..MAX) OF FileAndHash
  }
  FileAndHash ::= SEQUENCE {
  file IA5String,
  hash BIT STRING
  }

Es a través de estos campos que cualquiera de las partes puede establecer el estado del manifiesto, si es válido o inválido. Para ello, tan solo hay que realizar las siguientes comprobaciones además de las especificadas en la RFC6488 [26]:

1. El valor del campo eContentType en EncapsulatedContentInfo es id-ad-rpkiManifest (OID 1.2.840.113549.1.9.16.1.26).
2. La versión del manifiesto es 0.
3. El valor de thisUpdate precede al de nextUpdate.

[7] Dado que cada objeto firmado está contenido junto a su certificado EE como parte de un mensaje CMS, estos certificados no se publicarán por separado en el punto de publicación.

[8] International Telecommunication Union

El objetivo final, tanto de la estructura de los manifiestos como de las comprobaciones sobre su validez, no es otro que el de que los miembros de la red BGP puedan decidir que objetos utilizar para, por ejemplo, crear las tablas de enrutamiento. Sin embargo, como ocurre también con BGP-PFX, independientemente del resultado de las comprobaciones, el uso de unos objetos u otros depende de la política local.

Autorizaciones de Origen de Rutas

La información sobre la asignación de IPs que proporciona una infraestructura de clave pública no es suficiente por sí misma para guiar las decisiones de enrutamiento. De hecho, BGP funciona bajo la suposición de que los sistemas autónomos que originan los anuncios de rutas han sido autorizados por el concesionario de dichos prefijos para hacer el anuncio. El propósito de las autorizaciones de origen de rutas (ROAs) es, por tanto, el hacer explícita la existencia de dicha autorización.

La estructura de las ROAs se define en la RFC6482 [34], aunque la confirmación de que el sistema autónomo está autorizado para anunciar el prefijo correspondiente se hace explícita al contener el objeto el certificado EE correspondiente. De la estructura de las ROAs podemos ver como, potencialmente, una ROA puede contener todos los prefijos IP que un concesionario dado autorice a un sistema autónomo a anunciar. Esto es especialmente importante ya que, de este modo, se compacta la información. En cambio, tiene como desventaja que, en caso de necesidad, no se puede revocar un único prefijo, sino que hay que eliminar todos los que contenga la ROA y generar una nueva.

Además, es importante notar que antes de validar un anuncio de rutas, la parte involucrada en la validación debe validar la ROA según el procedimiento que se indica en la RFC6488 [26], además de verificar que los prefijos IP de la ROA coinciden con aquellos contenidos en el certificado EE en el campo de delegación de direcciones. En otras palabras, no solo hay que comprobar que la ROA ha sido firmada por un certificado válido y no ha caducado, también hay que comprobar que el contenido de la autorización del certificado es coherente con el contenido de la ROA.

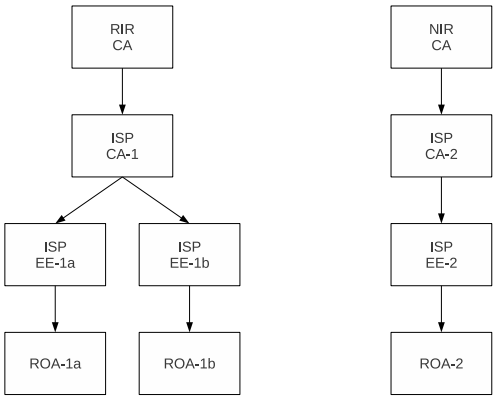


Figura 2.3: Ejemplo de asignación de recursos y firma de ROAs

2.1.3. Repositorio distribuido

Para el correcto funcionamiento del sistema RPKI, cada ISP debe poder acceder a todo el conjunto existente de ROAs y a toda la información necesaria para validarlas, esto es, certificados y CRLs. Es por ello que la principal función del repositorio RPKI es hacer accesible esta información a todos los ISP para que puedan consultarla con la frecuencia que sea necesaria. Además, el repositorio es un punto de seguridad en el sistema, ya que se garantiza que solo aquellos con la autorización correspondiente pueden modificar su contenido. Debe notarse que esto último es de vital importancia, ya que todo el sistema de

verificación en las autorizaciones en el uso de los distintos recursos depende de la integridad de la información contenida en el repositorio. La estructura de este es el de un único repositorio compuesto por múltiples bases de datos, como mínimo una por cada RIR, que contenga toda la información (certificados CA y EE, manifiestos, CRLs y ROAs) que esté asociada con ese registro, es decir, que si se siguiese el árbol de certificados de cada uno de esos elementos, su raíz estuviese en el certificado CA del registro. A cada una de estas bases de datos que contienen solo parte de la información del repositorio se les denomina puntos de publicación (PP). Por tanto, si un usuario desea descargarse a una copia local todo el repositorio, deberá conectarse a los puntos de publicación existentes y descargar toda la información contenida en cada uno de ellos.

Desde un punto de vista de almacenamiento de la información, el repositorio es un árbol de directorios donde cada directorio corresponde a un certificado y contiene todos aquellos elementos que son verificables utilizando este certificado. Para comprobar las relaciones entre los certificados, hay que notar que la extensión SIA (Subject Information Access) [22] de un determinado certificado contiene una URI (Uniform Resource Identifier) que apunta al directorio correspondiente a los objetos verificables por dicho certificado. De la misma manera, la extensión AIA (Authority Information Access) de un certificado contiene una URI que apunta a la localización en la que está el certificado bajo el que ha sido expedido. Puede verse dicha estructura de manera más clara en la figura 2.4.

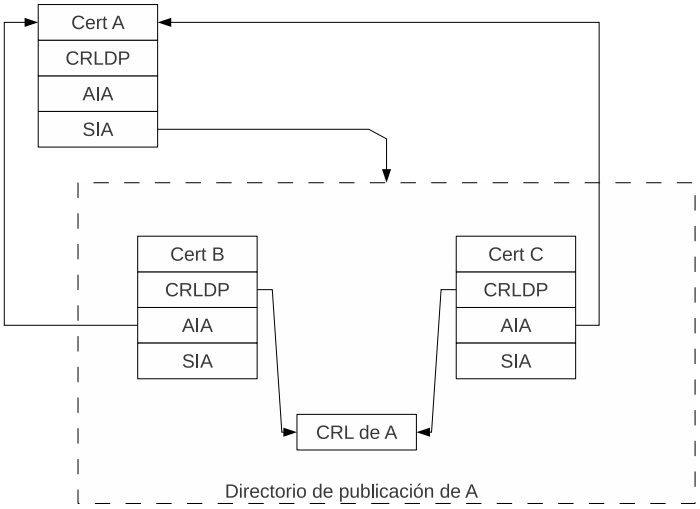


Figura 2.4: Uso de las extensiones SIA y AIA en RPKI.

Hay que tener en cuenta que esto implicaría que cuando hubiese que volver a expedir un certificado, habría que volver a generar todos los objetos dependientes de dicho certificado, es decir, todos aquellos contenidos en el directorio correspondiente a dicho certificado. Para evitarlo, se permite que se vuelvan a expedir certificados con la misma clave pública. Una consecuencia directa de esto es el hecho de poder utilizar un sistema persistente de URIs para la composición del árbol de directorios del repositorio.

Otro punto importante en el repositorio es que debe ser accesible. Para ello, es necesario que los protocolos que los operadores del repositorio elijan para acceder al mismo deben permitir las siguientes operaciones:

1. Descarga: los protocolos escogidos deben soportar tanto la descarga en bloque de todo el contenido del repositorio como la descarga de objetos individuales del mismo y otros tipos de descargas.

2. Subida/modificación/borrado: los protocolos escogidos también deben proporcionar mecanismos para que los concesionarios de los recursos de Internet puedan acceder al

repositorio para subir o borrar objetos así como modificarlos. Todas estas operaciones tienen que requerir la autenticación de quien las realiza para poder hacerlas y, de esta forma, proteger la integridad de la información contenida. En general, cualquier operación que implique una modificación del repositorio necesitará que el usuario que la realiza se autentique para poder efectuarla.

Dados los requisitos anteriores, y para fomentar el que todos los usuarios que lo necesiten puedan acceder al repositorio, este debe permitir, como mínimo, el acceso mediante el protocolo rsync [30], [35]. Esto no implica, no obstante, que no se puedan utilizar otros protocolos, como ya se ha dicho, siempre y cuando se comunique su uso a todas las autoridades de certificación que publiquen sus datos en el punto de publicación que utilicen dicho punto de publicación.

2.1.4. Cachés Locales

Para poder realizar las comprobaciones necesarias, los sistemas que dependen de RPKI deben obtener, en primer lugar, una copia local de todos los certificados EE válidos. Para ello el procedimiento a seguir es:

1. Obtener una copia local de todos los certificados, CRLs y manifiestos expedidos utilizando la arquitectura RPKI.

2. Por cada manifiesto, comprobar la firma del mismo con el certificado CA correspondiente[9], así como que las marcas temporales del mismo son correctas.

3. Por cada manifiesto, comprobar la lista de certificados y CRLs que contiene, así como sus hash[10].

4. Validar cada certificado EE construyendo y verificando el camino de certificación hasta el conjunto de TAs configurado localmente.

Dado que estas operaciones se realizarán con determinada frecuencia, es conveniente el permitir actualizaciones incrementales, es decir, actualizar solo aquellos objetos que han sido modificados desde la última consulta de la caché al repositorio.

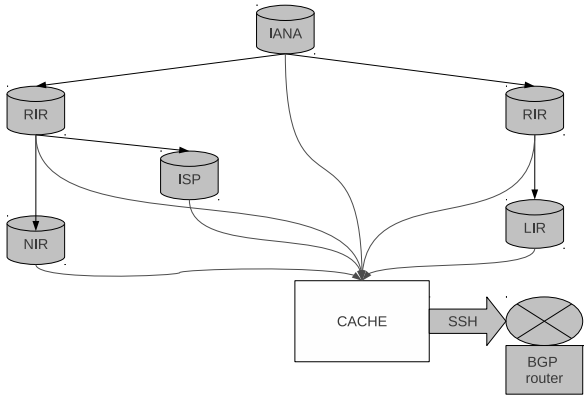


Figura 2.5: Adquisición de datos por parte de una caché local

[9]Es decir, verificar el certificado EE que firma el manifiesto con el certificado CA que lo ha expedido
[10]funciones resumen [36] .

2.1.5. Distribución de la información RPKI

Una vez que la información ha sido copiada en las cachés locales, estas permitirán acceso a los routers que tengan asignados para que puedan adquirir la información sobre la situación de la red (quién posee qué recursos). Para dichas actualizaciones se está realizando actualmente un protocolo de actualización incremental [37] que se implementaría mediante el uso de túneles SSH.

El funcionamiento de dicho protocolo se basa en que los routers estén conectados a varias cachés, una primaria y una o más secundarias, con las que se sincronizan. Cuando las cachés se actualizan al descargar nueva información del repositorio, envían una notificación a los routers que estén conectados a ellas, lo que indica que es un buen momento para conectarse, aunque no implica una obligación de hacerlo. Entonces los routers pueden conectarse para actualizarse o esperar a algún evento determinado en la configuración[11]. Independientemente de que llegue o no una notificación de actualización, los routers se comunican con la caché por lo menos una vez cada hora para comprobar las actualizaciones, y mantener abiertos los túneles de comunicación.

Sea cuando fuere, si al llegar una notificación, las cuales se envían de manera periódica (salvo que haya algún cambio en cuyo caso se envían inmediatamente), el número de la última actualización que se ha recibido en el router se compara con el que va a mandar la caché. Si ambos números son distintos, se envían todas las que sean necesarias hasta que la información en ambos puntos sea la misma.

En caso de conectarse por primera vez el router a una caché o si hay una caída en la conexión, la caché enviaría toda la información al router sobre el estado actual en toda la red hasta completar la transmisión. Para evitar pérdidas de información es por lo que se recomienda que el router esté conectado, y sincronizado, con varias cachés, aunque utilice preferentemente una de ellas.

2.2. BGP Prefix Origin Validation, BGP-PFX

El algoritmo de validación de origen (BGP-PFX) [38] está fuertemente inspirado por la iniciativa Secure Origin BGP (soBGP) [19]. Sin embargo, al contrario de lo que ocurría con la definición de soBGP, la validación de origen se ve como una parte de un conjunto de medidas para ayudar a hacer BGP un protocolo seguro, en vez de un método único y suficiente en sí mismo de alcanzar dicha meta.

Partiendo de la información proporcionada por RPKI, el algoritmo proporciona información al router BGP sobre si el sistema autónomo de origen es válido, inválido o no se tiene información del mismo, tal y como se ilustra en la figura 2.6. Sin embargo, el resultado del algoritmo no es vinculante, es decir, la influencia del algoritmo se determina en la configuración de cada router, donde se define qué hacer con las rutas en función del resultado de la validación como una política local más a tener en cuenta a la hora de tomar la decisión de encaminamiento.

De este modo podemos tener ISPs que decidan configurar sus routers de forma que utilicen preferentemente las rutas válidas sobre las desconocidas y descarten las inválidas, otros que no descarten las inválidas, pero las utilicen cuándo no les quede otra alternativa u otros que decidan tratar todos los resultados del algoritmo por igual.

2.3. Border Gateway Protocol Security, BGPSEC

BGPSEC es la expansión de seguridad para BGP propiamente dicha. Su funcionamiento se basa en comprobar que el origen del anuncio BGP es correcto mediante el uso del algoritmo BGP-PFX [39] tal y como se ha explicado en la sección 2.2, unida a la verificación de que el mensaje UPDATE no ha sido modificado. Como en el caso de BGP-PFX, lo que el router haga según el resultado de la verificación del mensaje es una cuestión de política interna del ISP. Además es importante notar que, por razones de diseño [40], el campo NRLI contendrá un único prefijo IP.

[11] Por ejemplo, un temporizador

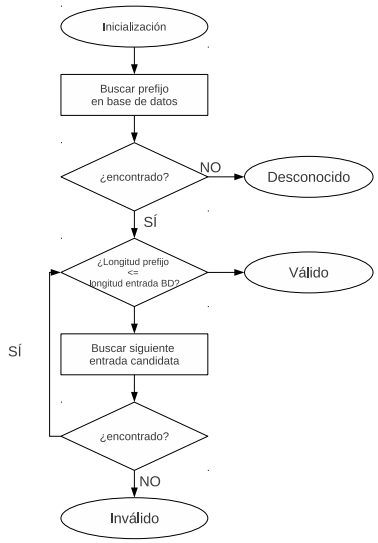


Figura 2.6: Diagrama de flujo de BGP-PFX

Para ello, BGPSEC sustituye el atributo AS_Path del mensaje BGP estándar, por un nuevo atributo llamado BGPSEC_Path_Signatures. Dicho atributo se compone de tres campos bien diferenciados y todos ellos de longitud variable, tal y como puede verse en la figura 2.8.

2.3.1. Secure_Path

El campo Secure_Path es totalmente equivalente al atributo AS_Path en un mensaje BGP. Su formato se define con un primer campo de dos octetos que contiene la longitud, en octetos, de la ruta de sistemas autónomos que ha seguido el mensaje UPDATE y un segundo campo llamado Secure_Path_Segment que registra cada sistema autónomo *distinto* por el que ha pasado el mensaje desde el NRLI, tal y como se indica en la figura 2.9.

Además, como podemos observar en la figura 2.9, cada Secure_Path_Segment está compuesto de seis octetos donde los cuatro primeros muestran el número de sistema autónomo. El siguiente (pCount) especifica el número de pasos por el sistema autónomo que cubre la firma, lo que permite a un router BGPSEC añadir varias copias de su AS sin necesidad de firmar varias veces. Finalmente, el último octeto (Flags) se reserva para uso futuro.

El hecho de que la longitud se exprese en octetos hace que la longitud mostrada en el campo Length sea seis veces mayor que el número de Secure_Path_Segment.

2.3.2. Additional_Info

El nuevo campo Additional_info (figura 2.10) tiene un primer octeto que determina el tipo de la información, otro segundo octeto que contiene la longitud y, finalmente, una información de longitud variable. En la especificación actual de BGPSEC las componentes de tipo y longitud tienen ambas valor cero y, por tanto, no hay información adicional a transmitir en el mensaje UPDATE. No obstante, se reserva este campo para uso futuro. Se prevé, sin embargo, que la sintaxis dependa del tipo de información que contenga.

2.3.3. Signature_Block

Finalmente, el campo Signature_Blocks contiene uno o dos bloques de firmas cada uno utilizando un algoritmo distinto. Esto se define así para que pueda ser posible la actualización a nuevos algoritmos de firma en un futuro sin que sea necesario modificar todos

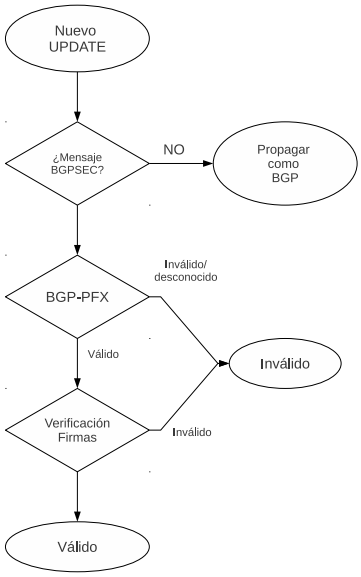


Figura 2.7: Procedimiento de verificación de mensajes UPDATE

los equipos que soporten BGPSEC a la vez, sino que se pueda hacer de forma incremental. En principio, para la versión '0' del protocolo, el algoritmo de firma que se utilizará será ECDSA-256[12] utilizando SHA256 como algoritmo de hash.

Como puede verse en la figura 2.11, el primer octeto indica tanto el algoritmo que se ha utilizado para firmar como el que se ha utilizado para realizar la función de hash. Los dos siguientes octetos indican la longitud del segmento de firmas y, finalmente, se encuentra el Signature_Segment, es decir, las firmas propiamente dichas.

En un bloque Secure_Path hay exactamente el mismo número de Signature_Segments que de Secure_Path_Segments. El campo Signature_Key_Identifier contiene el Signature_Key_Identifier del certificado EE correspondiente, ya que es necesario para la verificación de la firma y ocupa los veinte primeros octetos del bloque. El siguiente octeto es el campo Signature_Length contiene la longitud en octetos de la firma. Finalmente, la firma protege el campo NRLI y el Secure_Path, tal y como se muestra en la figura 2.12.

[12]Cifrado basado en curvas elípticas

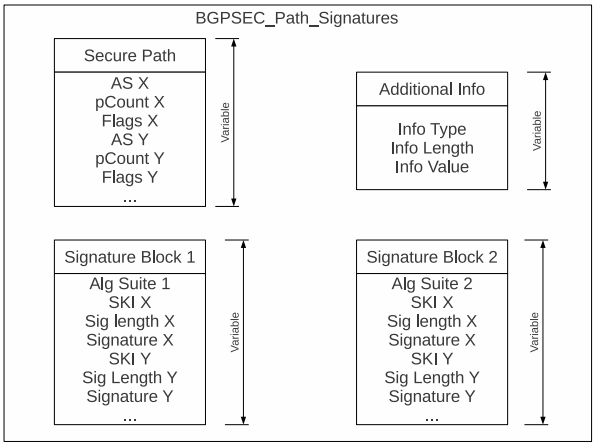


Figura 2.8: Estructura del atributo BGPSEC_Path_Signatures

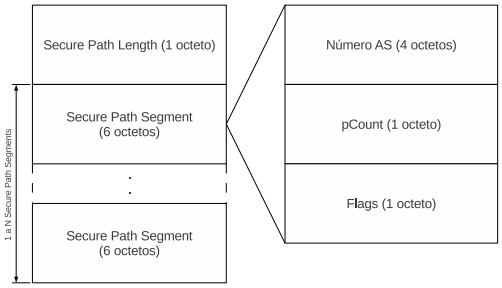


Figura 2.9: Campo Secure_Path

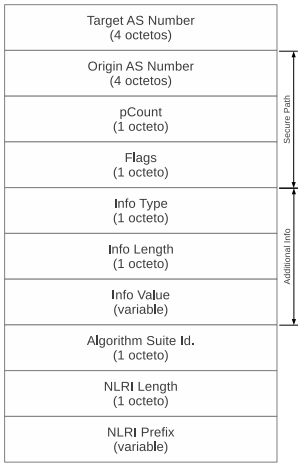


Figura 2.12: Campos protegidos por la firma en un UPDATE BGPSEC

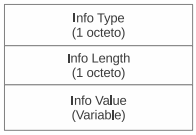


Figura 2.10: Campo Additional_Info

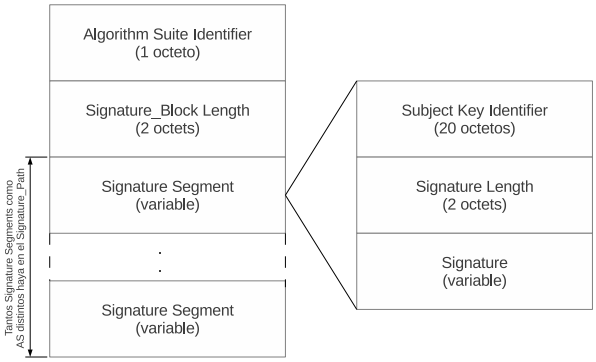


Figura 2.11: Campo Signature_Block

Como consideraciones finales sobre BGPSEC, debemos notar que ya que los que firman los mensajes son los *routers* por los que el mensaje UPDATE ha pasado, para su correcto funcionamiento es necesario el acceso a dos infraestructuras PKI: una de ellas es RPKI, para poder utilizar BGP-PFX sobre los mensajes recibidos, y otra contendría los certificados de todos los routers BGPSEC de la red.

Capítulo 3

Problemática RPKI

Como hemos podido ver a lo largo de la sección 2.1, RPKI es una estructura de repositorio distribuido a lo largo de distintas bases de datos, o puntos de publicación. Por tanto, hay que tener en cuenta los problemas inherentes a dicho tipo de estructuras, además de aquellos que vienen dados por el volumen de la información que se esta manejando.

Es por ello que podemos dividir los posibles problemas de RPKI en:

- Aquellos derivados del volumen de datos con el que se trabaja.
- Aquellos considerados problemas derivados de la seguridad en el diseño de la base de datos RPKI.

Es por ello que a lo largo de este capítulo hemos realizado un análisis cuantitativo de los principales problemas de RPKI. Aunque vayamos a comentar los distintos casos por separado, debemos tener en cuenta que muchos ataques tratarán de aprovecharse de las vulnerabilidades derivadas del gran volumen de datos a tratar.

3.1. Problemas derivados del volumen de datos

Si observamos la tabla 3.1, podemos observar cómo el volumen de datos con el que se trabaja a nivel global es de cientos de megas. Para los estándares actuales dicho volumen de datos no es muy elevado, pero debemos considerar que habrá muchos equipos conectándose simultáneamente a los distintos puntos de publicación. De hecho, en la tabla 3.2 podemos ver cómo para los casos en que haya una, dos o tres cachés por AS la capacidad de atender peticiones de los puntos de publicación debe ser del orden de gigabytes (en el caso de AfriNIC) a terabytes (en el caso de ARIN o RIPE).

Para poder obtener datos más precisos sobre el volumen de tráfico, es necesario conocer el número medio de routers por AS y el número medio de routers a los que puede servir una caché de manera simultánea, información que no está disponible.

3.1.1. Escalabilidad

Como puede verse en la figura 3.1, el número de prefijos activos en las tablas BGP ha aumentado entre 2011 y 2012 en una cantidad semejante a la que lo hizo en el período de 2008 a 2011. Por tanto, es muy importante comprobar que la infraestructura sea capaz de manejar dichos aumentos en el número de recursos sin necesidad de modificaciones sustanciales.

Uno de los peligros en el aumento del número de recursos radica en que la infraestructura no sea capaz de servir toda la información necesaria a las cachés para que estas se actualicen en un tiempo aceptable, evitando así que haya momentos en que el sistema sea vulnerable y que, en caso de que un volumen grande de equipos hayan hecho un reinicio y necesiten

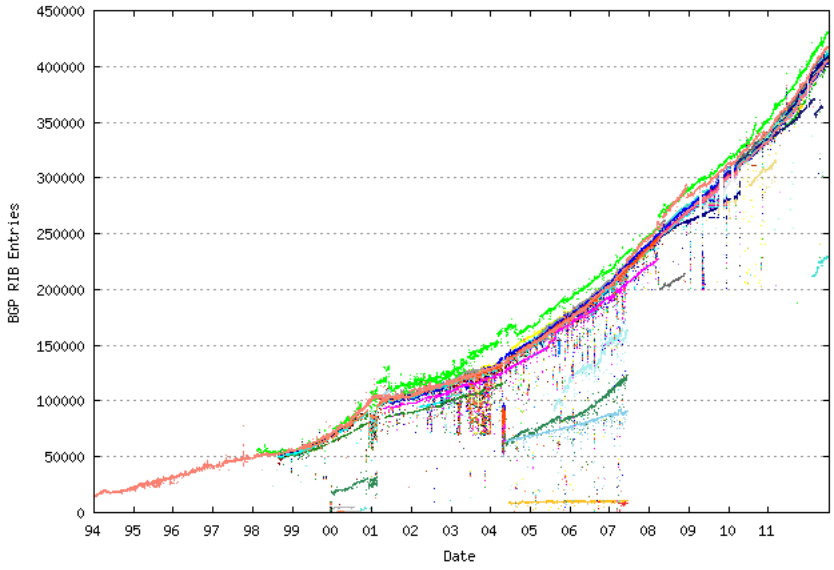


Figura 3.1: Crecimiento de la tabla BGP desde 1994 hasta ahora. Fuente: potaroo.net

descargar de nuevo toda la información contenida en la base de datos RPKI, esta no sufra una degradación del servicio[1].

Número de cachés

Ya hemos mencionado en este capítulo que el número de cachés que haya en cada AS influye en el volumen de tráfico que debe soportar RPKI, además del hecho de que a mayor número de cachés, mayor número de conexiones simultáneas que deben soportar los distintos puntos de publicación, con lo cual el sistema debe ser más robusto.

Desde otro punto de vista, a mayor número de cachés en un mismo AS, más sencillo es para este el tener la información distribuida por todos sus routers de manera rápida y eficaz sin necesidad de tener una infraestructura con una capacidad especialmente grande, ya que cada grupo de cachés sirve a un grupo de routers determinado.

Una posible solución para este problema consiste en que solo un pequeño porcentaje de las cachés del AS se actualicen de los puntos de publicación, y sean estas las que el resto de cachés del AS utilicen para descargarse la información [37].

Otro punto a tener en cuenta es la necesidad de encontrar un equilibrio entre la necesidad de cubrir la demanda interna dentro de un AS con capacidad de servir la información obtenida de RPKI[2] con el hecho de que un número de cachés excesivamente elevado puede tener efectos adversos sobre RPKI al crear una mayor congestión de tráfico [37].

Número de routers por caché

De la misma forma que en la sección 3.1.1 se ha hablado de la necesidad de equilibrio en el número de cachés para no saturar el sistema, existe un problema similar con los routers dentro del AS. Todos los routers que necesiten utilizar la información contenida en la arquitectura RPKI para su funcionamiento (p. ej., routers que utilicen BGP-PFX y BGPSEC) deben poder descargar dicha información de una caché primaria así como poder conectarse a una o más cachés secundarias para poder recuperarse en caso de que la conexión con la primaria se interrumpiese por algún motivo.

[1] En el peor de los casos, el efecto sería el mismo que el de ser el objetivo de un ataque de denegación de servicio distribuido (DDoS) [41].

[2] Que es posible el que un AS pequeño contrate este servicio a un tercero de su confianza con infraestructuras suficientes.

Por tanto, el número de routers que pueden conectarse a una caché de manera simultánea puede resultar problemático por varios motivos:

- Si la relación routers por caché es muy reducida, en el límite un router por caché:
  - Serán necesarias muchas cachés para poder dar servicio a todos los routers.
  - Cuando aumente el número de routers en el AS, será necesario aumentar el número de cachés.
- Si la relación routers por caché es alta:
  - Es posible que la caché no pueda dar servicio de manera simultanea a todos los routers.
  - Las actualizaciones pueden tardar en propagarse.

De nuevo es necesario encontrar un equilibrio entre velocidad y los recursos reales del AS para, de esta forma, hacer el diseño de forma óptima.

3.1.2. Tiempo de recuperación en caso de caída del sistema

Otro parámetro que es importante (dado que lo que pretende RPKI es aumentar la seguridad en BGP para evitar errores debidos a ataques o a fallos de configuración) es el tiempo que tardaría en recuperarse después de una caída del sistema, es decir, si la base de datos distribuida sufriese una degradación total o parcial del servicio ¿cuánto tardaría el sistema en volver a ser seguro?

El primer dato a tener en cuenta es que durante el tiempo en que el servicio sufra la degradación en su funcionamiento la información que tengan las cachés locales almacenadas puede quedar anticuada (p. ej., puede haber revocaciones de certificados que no hayan llegado a transmitirse a las cachés). Además, cuándo RPKI vuelva a estar en linea todas las cachés se conectarán a los puntos de publicación para sincronizarse y, en caso de que haya habido modificaciones, empezarán a actualizarse, lo que puede ocasionar una nueva pérdida del servicio.

Es por ello que la estructura, antes de entrar en su fase de producción, debería superar pruebas de estrés global así como tener establecidas procedimientos y protocolos de actuación para casos como este.

3.1.3. Tiempo de propagación de cambios en el estado de la red

Finalmente, el último gran problema derivado del tamaño de RPKI reside en el tiempo de propagación de modificaciones. Como se ha podido ver en la sección 1.2, perdidas de servicio de unos pocos minutos son inadmisibles dada la naturaleza global de Internet. Es por ello que resulta necesario el conocer cuanto tarda la estructura en propagar los cambios para, de ser necesario, establecer los protocolos oportunos para minimizar ese tiempo hasta un nivel aceptable.

3.2. Problemas relativos a la seguridad

A pesar de que la seguridad interna de RPKI se sale del ámbito de este documento, es interesante señalar dos puntos importantes que pueden suponer una vulnerabilidad para la red BGP en caso de que dichos ataques tuviesen lugar durante la fase de producción del sistema.

3.2.1. Control de acceso

En primer lugar, es muy importante que el control de acceso dentro de los puntos de publicación esté muy bien diseñado para que todos los usuarios puedan leer de todos los directorios y, por tanto, obtener toda la información que necesitan respecto a certificados y objetos validados para poder utilizarla a nivel local.

Por otra parte, es igualmente importante que solo el dueño de cada directorio pueda escribir en él. Este concepto se traduce en que aunque se utilice una arquitectura jerárquica en la construcción del árbol de certificados, y por tanto del de directorios, el hecho de que un usuario tenga permiso para escribir en un directorio no implica que tenga permiso para escribir en sus subdirectorios. En la figura 3.2 podemos observar cómo en el caso de dos entidades distintas, donde la entidad 1 certifica a la entidad 2 (aunque la información de la entidad 2 sea un subdirectorio) no tiene sentido que pueda manipularse desde la entidad 1.

Estas consideraciones son especialmente interesantes ya que, de este modo, un posible atacante solo podría tener acceso a partes aisladas de la base de datos, de forma que la detección del ataque y posterior recuperación del mismo sería mucho más sencilla.

3.2.2. Creación de un número excesivo de objetos RPKI

Otro aspecto a tener en cuenta con la seguridad en RPKI es el supuesto de que un usuario empezase a crear un número muy elevado de objetos RPKI (certificados, manifiestos, CRLS o ROAs) habría varias consecuencias adversas para el sistema:

- Las cachés lo verían como nuevas actualizaciones de forma que intentarían conectarse todas a descargar un volumen muy grande de datos lo que podría ocasionar los mismos efectos que un posible DDoS.
- El usuario podría consumir todos los recursos del sistema.
- La recuperación del error sería costosa ya que habría que verificar que objetos son válidos y cuales no.

Por ello, además del control de acceso anteriormente mencionado (para evitar que sea un tercero que quisiera perjudicar el funcionamiento de RPKI) es muy recomendable que se establezca una política de cuotas de disco y de nodos-i[3].

[3] O equivalente en función del sistema operativo en que este desplegado RPKI.

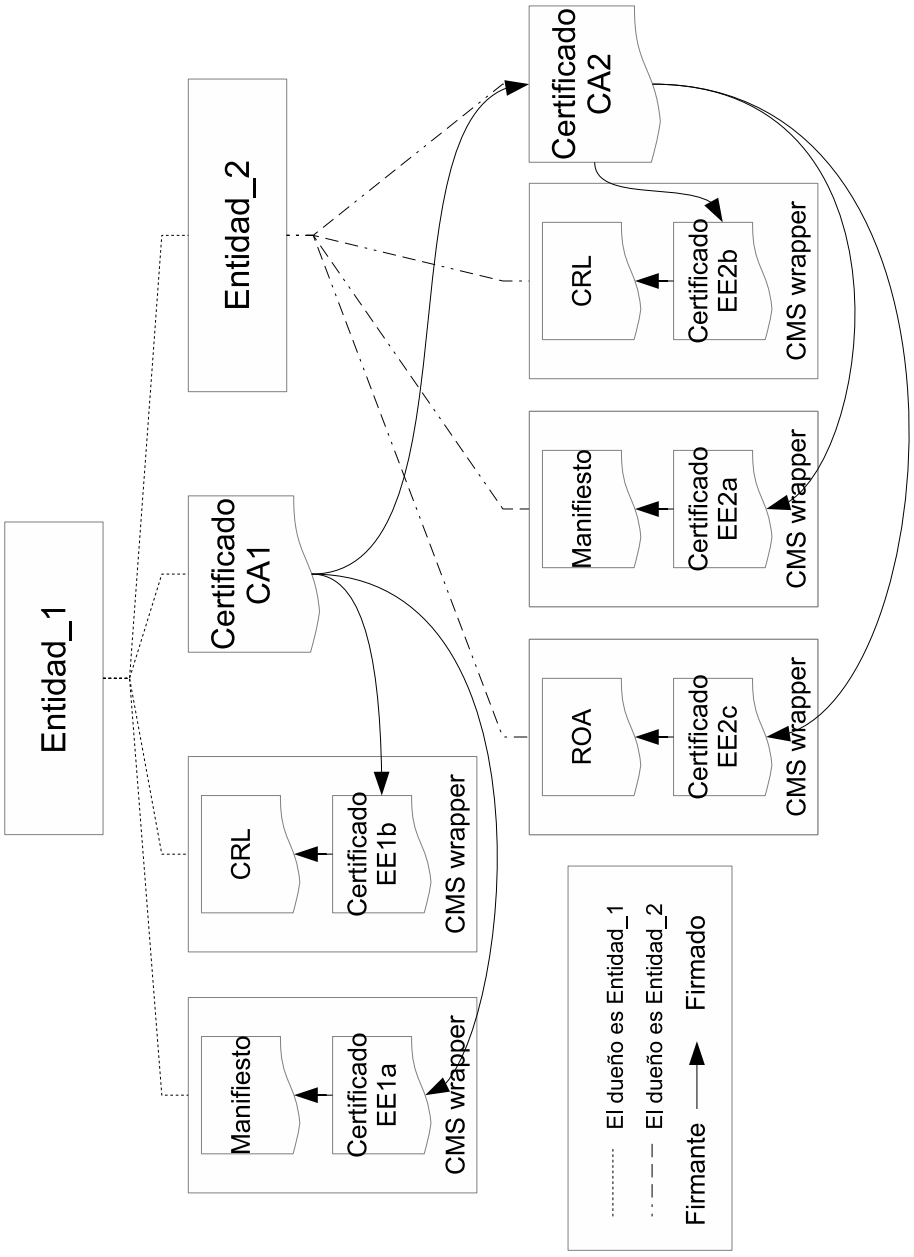


Figura 3.2: Ejemplo de estructura de directorios dentro de RPKI

	WORLDWIDE	APNIC	ARIN	RIPE	LACNIC	AfriNIC
AS number announced	37.427	4.426	14.348	15.553	1.441	448
prefixes per Asn	10	19	8	5	22	14
CA certificates	37.427	4.426	14.348	15.553	1.441	448
EE certificates	374.270	88.520	114.784	77.765	31.702	6.272
certificate size (KB)	1	1	1	1	1	1
Space required(only certificates) [KB]	411.697	92.946	129.132	93.318	33.143	6.720

Tabla 3.1: Volumen de datos esperado en el sistema RPKI de forma global y desglosado por RIRs

Number of caches	WORLDWIDE	APNIC	ARIN	RIPE	LACNIC	AfriNIC
1 per Asn	37.427	4.426	14.348	15.553	1.441	448
2 per Asn	74.854	8.852	28.696	31.106	2.882	896
3 per Asn	112.281	13.278	43.044	46.659	4.323	1.344
required channel capacity (MAXIMUM)						
1 cache per Asn(KB)	15.408.583.619	411.378.996	1.852.785.936	1.451.374.854	47.759.063	3.010.560
2 cache per Asn	30.817.167.238	822.757.992	3.705.571.872	2.902.749.708	95.518.126	6.021.120
3 cache per Asn	46.225.750.857	1.234.136.988	5.558.357.808	4.354.124.562	143.277.189	9.031.680

Tabla 3.2: Número de cachés y tráfico esperado para tres casos (una, dos o tres cachés por AS)

Capítulo 4

Problemática BGP-PFX y BGPSEC

A lo largo de este capítulo vamos a estudiar los principales problemas tanto para la implantación como para la explotación de BGP-PFX y BGPSEC. Debemos tener en cuenta que. aunque se estudien por separado, ambos sistemas comparten la problemática base de RPKI al utilizar esta tecnología como base fundamental para su ejecución. Es por ello que, incluso si no se detectase ningún tipo de dificultad en ambos protocolos, habría que tener en cuenta todo lo discutido en el capítulo 3.

4.1. Consideraciones previas

Antes de proceder a estudiar los posibles problemas en BGP-PFX y BGPSEC debemos hacer un breve estudio de los requisitos en BGP para que se considere que los routers funcionan de forma adecuada. Para ello, el tráfico no debe sufrir una degradación de servicio, esto es, se deben minimizar los descartes. Por tanto, debemos estudiar los parámetros del tráfico para que, cuando comparemos dichos parámetros con los datos calculados o simulados para ambos protocolos, poder concluir si el rendimiento se verá afectado y en qué medida.

En primer lugar, debemos tener en cuenta las características de tráfico en BGP. Para ello es necesario conocer el número medio de mensajes de tipo UPDATE por segundo, así como cuántos mensajes de este tipo se reciben en los momentos de carga máxima, ya que el sistema debe funcionar correctamente en ambos casos.

Sin embargo, también debemos tener en cuenta el hecho de que, por motivos de diseño [40], no se permite mandar varios anuncios distintos en un mismo mensaje. Esto hace que el tráfico BGP aumente, como puede verse en la tabla 4.2.

Por tanto, para poder considerar que no existe una degradación de las prestaciones ofrecidas por el protocolo, debemos cotejar el peor caso posible de los que muestra la tabla 4.2 con los datos medidos o calculados para los protocolos y verificar si estos cumplen dichas prestaciones.

4.2. Problemática BGP-PFX

Como ya hemos explicado en la sección 2.2, el algoritmo de validación de prefijos se limita, únicamente, a comprobar que el AS que origina el mensaje BGP con el anuncio correspondiente esta autorizado a hacerlo, siguiendo el proceso descrito en la figura 2.6. Debemos destacar que dicha comprobación se realiza únicamente sobre los prefijos contenidos en el campo NRLI del mensaje BGP y no sobre aquellos que aparecen al campo WITHDRAW.

Para poder evaluar la eficiencia de este algoritmo, debemos contestar a tres cuestiones muy importantes:

	3-July-2012 00:00 - 9-July-2012 23:59	10-July-2012 00:00 - 16-July-2012 23:59
UPDATES por segundo (medio)	1,31	0,37
UPDATES por segundo (pico)	1842	270
tiempo entre updates (medio) (s)	0,763358779	2,702702703
tiempo entre updates (mínimo) (s)	0,000542888	0,003703704
	17-July-2012 00:00 - 23-July-2012 23:59	24-July-2012 00:00 - 30-July-2012 23:59
UPDATES por segundo (medio)	0,92	1,1
UPDATES por segundo (pico)	2641	3128
tiempo entre updates (medio) (s)	1,086956522	0,909090909
tiempo entre updates (mínimo) (s)	0,000378644	0,000319693

Tabla 4.1: Número de UPDATES por segundo medio y de pico. Fuente: potaroo.net

	3-July-2012 00:00 - 9-July-2012 23:59	10-July-2012 00:00 - 16-July-2012 23:59
Número medio de prefijos por UPDATE	2,51	3,47
UPDATES por segundo (medio)	3,2881	1,2839
UPDATES por segundo (pico)	4623,42	936,9
tiempo entre updates (medio) (s)	0,304127003	0,77887686
tiempo entre updates (mínimo) (s)	0,00021629	0,00106735
	17-July-2012 00:00 - 23-July-2012 23:59	24-July-2012 00:00 - 30-July-2012 23:59
Número medio de prefijos por UPDATE	2,88	2,25
UPDATES por segundo (medio)	2,6496	2,475
UPDATES por segundo (pico)	7606,08	7038
tiempo entre updates (medio) (s)	0,377415459	0,404040404
tiempo entre updates (mínimo) (s)	0,000131474	0,000142086

Tabla 4.2: Número de UPDATES por segundo medio y de pico con el incremento de tráfico. Fuente: potaroo.net

- ¿El tiempo que tarda el algoritmo en validar los prefijos recibidos cumple con los requisitos temporales expresados en la tabla 4.2?
- ¿Qué porcentaje de los mensajes anunciados por BGP resultarían inválidos con este algoritmo y que consecuencias tendría para las comunicaciones?
- ¿Cómo de fácil resulta atacar al algoritmo con inserciones, modificaciones u otros ataques MITM?

Para evaluar este algoritmo hemos utilizado una lista ordenada para simular la estructura de datos donde se almacena la información obtenida a través de RPKI. La búsqueda se realizará mediante un algoritmo de búsqueda binaria para encontrar el primer resultado de la lista que cumpla los requisitos del algoritmo, es decir, contenga[1] el prefijo anunciado en el mensaje. Una vez que se encuentra dicho prefijo, en caso de que no de un resultado válido al realizar las comprobaciones propias de BGP-PFX tal y como podemos ver en la figura 2.6, al estar los prefijos almacenados en una lista ordenada solo es necesario recorrer dicha lista hasta que no haya más resultados que contengan al prefijo anunciado. Esto hace que una vez que se ha alcanzado la primera posición las búsquedas sucesivas sean muy rápidas.

Dichas decisiones de diseño se deben a que, aunque en general, routers software como XORP[42] utilizan tries para almacenar su tabla de rutas, de esta manera podemos proporcionar una cota superior a los tiempos ya que la complejidad computacional y el tiempo de búsqueda en nuestro caso sería el del caso de un trie que tuviese información en todas las hojas, es decir, el trie más grande posible para nuestro conjunto de datos.

La información que contiene dicha estructura por cada prefijo es el prefijo, su longitud mínima, su longitud máxima y el AS que esta autorizado como origen tal y como puede verse en la tabla 4.3. Dado que no tenemos disponible la información sobre que AS están autorizados a anunciar que prefijos o, alternativamente, que AS son los dueños de que prefijos, el método que hemos utilizado consiste en analizar la información sobre los prefijos anunciados y los As origen que puede encontrarse en http://thyme.rand.apnic.net/current/. Una vez descargada la información para cada RIR hemos procedido a agregar los prefijos para obtener las longitudes máxima y mínima de los anuncios realizados por cada AS. Posteriormente hemos procedido a ordenar el resultado de la agregación para obtener la estructura de datos que hemos comentado anteriormente.

prefijo	*mascara_subred*	longitud_máxima	AS
10.0.0.0	(8)	/24	12345

Tabla 4.3: Ejemplo de formato de los prefijos.

Posteriormente hemos utilizado una captura con BGPDUMP proporcionada por RIPE (http://www.ripe.net/data-tools/stats/ris/ris-raw-data) del día veintiséis de Julio de 2012 a las 18.15h. La elección de esta traza en concreto se debe a que el tráfico en ese intervalo fue de más de treinta y seis mil mensajes en total, con lo que después de eliminar los mensajes de tipo withdraw y los que contienen anuncios IPv6, seguimos teniendo más de diecinueve mil muestras para poder medir los tiempos de búsqueda del algoritmo.

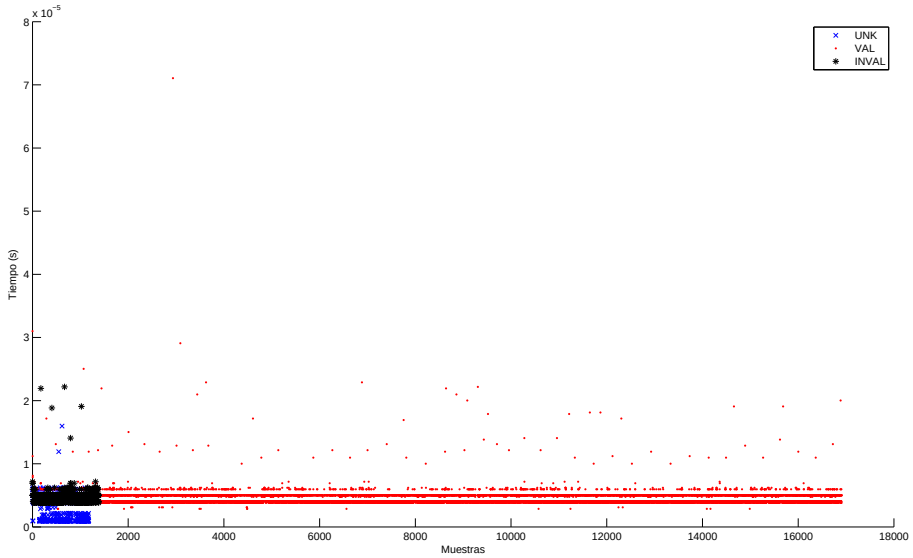
4.2.1. Tiempo de ejecución del algoritmo

Para poder obtener resultados significativos, hemos utilizado equipos con una potencia de cálculo similares a las de los routers del apéndice A. Hemos utilizado dos equipos, un primer equipo denominado 'lapa', con un procesador Intel(R) Pentium(R) 4 CPU 1.70GHz y 774380 kB de memoria RAM y otro equipo, con denominación 'moscardón', con un procesador Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz y 3994340 kB de memoria

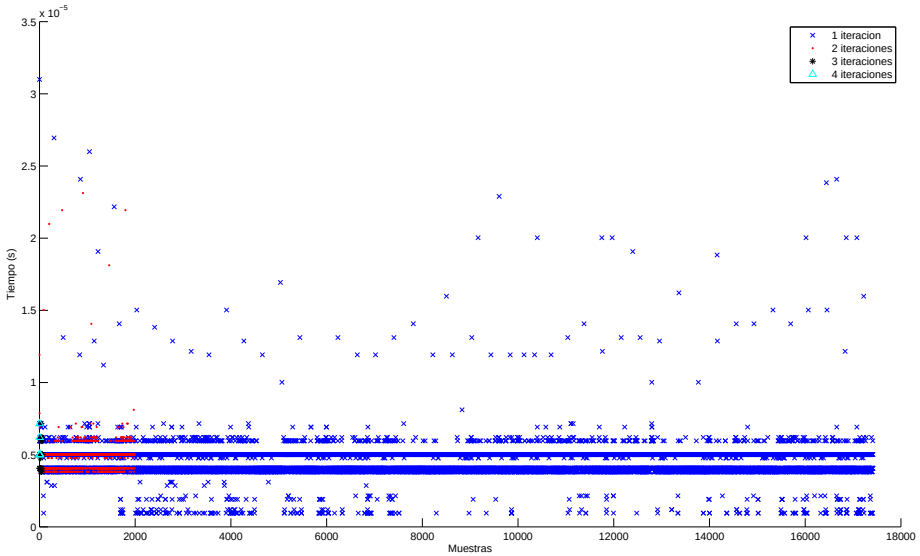
[1]Decimos que una entrada de nuestra base de datos contiene a un prefijo cuando la longitud de la entrada es menor que la del prefijo anunciado en el mensaje BGP y todos los bits a lo largo de dicha longitud coinciden.

RAM. De esta forma podemos emular cual sería la respuesta de un router de gama media y uno de gama alta.

Para medir el tiempo de ejecución de BGP-PFX hemos realizando dos medidas de tiempos con el mismo juego de prefijos a buscar. En la primera medida hemos obtenido los datos sobre el tiempo que tarda en función de si el algoritmo clasifica un prefijo como válido, inválido o desconocido. En la segunda medida hemos obtenido información sobre el tiempo en obtener una salida en función del número de veces que se ha ejecutado la función next_lookup_result. A continuación podemos ver los resultados obtenidos al ejecutar el algoritmo en lapa, así como los tiempos medios de ejecución del algoritmo.



(a) Tiempo en función del resultado del algoritmo



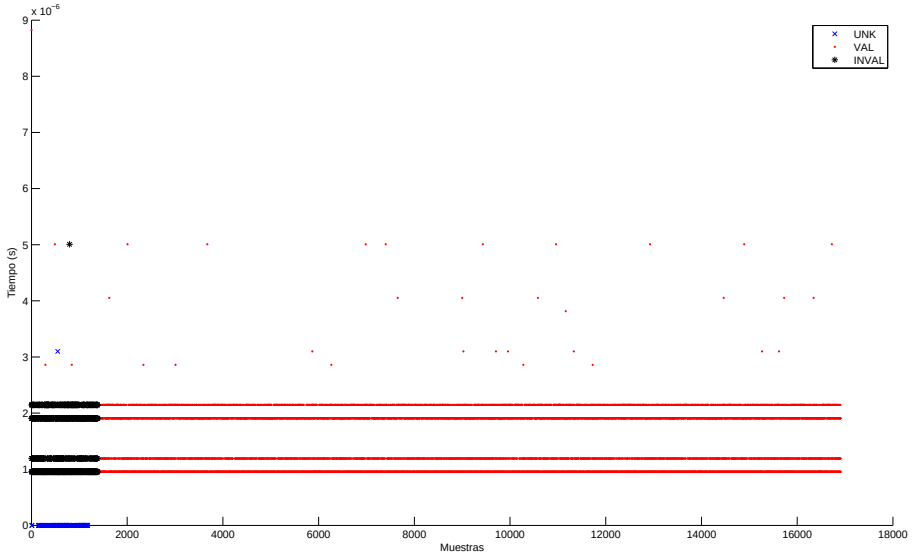
(b) Tiempo en función del número de ejecuciones de next_lookup_result

Figura 4.1: Medidas de tiempo en lapa

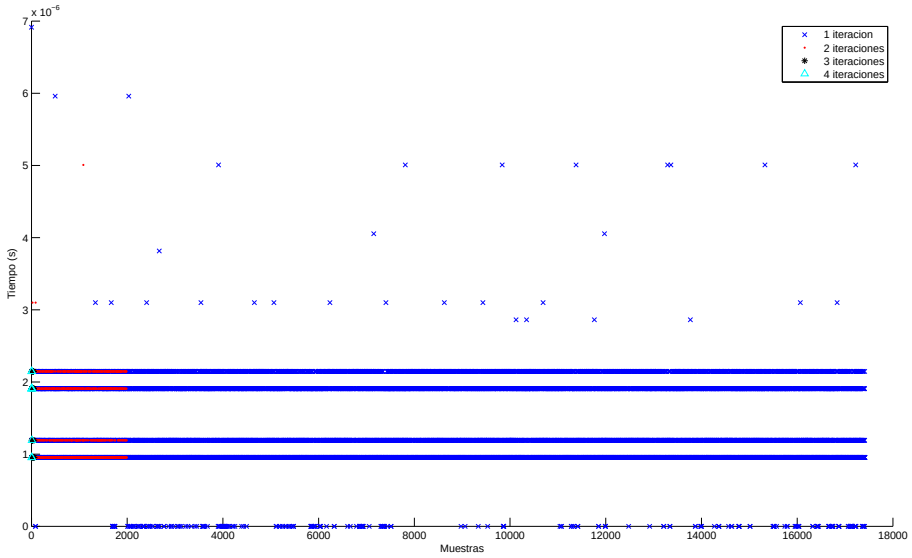
El tiempo medio de procesado en lapa es de $4,4324 \cdot 10^{-06}$ segundos para el caso en el

que hemos agrupado los datos en función del resultado y de $4,4417 \cdot 10^{-06}$ segundos para el caso en que los hemos estudiado en función del número de ejecuciones antes de devolver un resultado.

A continuación, presentamos los datos de moscardón bajo las mismas condiciones que en lapa, así como los tiempos medios de ejecución del algoritmo.



(a) Tiempo en función del resultado del algoritmo



(b) Tiempo en función del número de ejecuciones de next_lookup_result

Figura 4.2: Medidas de tiempo en moscardón

El tiempo medio de procesado es de segundos en el caso en el que hemos agrupado los datos en función del resultado y de en el caso en que los estudiamos en función del número de ejecuciones antes de devolver un resultado. Si ahora pasamos a agrupar los tiempos en función de cada categoría, es decir, procesamos por separado las muestras para cada categoría, obtenemos

4,39718541475692e-06	Lapa	Moscardón
1 iteración	$4{,}3972 \cdot 10^{-06}$	$1,311110^{-06}$
2 iteraciones	$4{,}8128 \cdot 10^{-06}$	$1,421010^{-06}$
3 iteraciones	$4{,}9538 \cdot 10^{-06}$	$1,5806 \cdot 10^{-06}$
4 iteraciones	$5,3102 \cdot 10^{-06}$	$1,7123 \cdot 10^{-06}$
UNK	$3,3876 \cdot 10^{-06}$	$9,619910^{-07}$
VAL	$4,4781 \cdot 10^{-06}$	$1,362510^{-06}$
INVAL	$4,7603 \cdot 10^{-06}$	$1,408610^{-06}$

Tabla 4.4: Tiempo medio de ejecución de cada conjunto de muestras

	Valores medios
UPDATES por segundo (medio)	2,4242
UPDATES por segundo (pico)	5051,1
Tiempo entre updates (medio) (s)	0,4125
Tiempo entre updates (mínimo) (s)	0,0002

Tabla 4.5: Valores medios de las medidas semanales de tráfico BGP.

Como podemos observar en las figuras 4.1 y 4.2, los tiempos son independientes tanto del resultado del algoritmo como del número de veces que se ejecute la función next_lookup_result. Sin embargo, en la tabla 4.4 podemos ver como en media los resultados que implican un mayor número de operaciones son ligeramente más lentas que aquellas que requieren un menor número de estas. Pese a este dato, podemos ver como no podemos predecir el resultado del algoritmo en función del número de iteraciones.

Si observamos la tabla 4.5[2], podemos ver como ambos equipos cumplen los requisitos temporales para que no exista una degradación de servicio al utilizar BGP-PFX.

4.2.2. Propagación de prefijos inválidos o desconocidos

En la sección 2.2 mencionamos como el tratamiento de prefijos inválidos o desconocidos es distinto de AS a AS dado que viene determinado por las políticas locales a cada entidad. Por tanto, es posible que un prefijo que da como resultado del algoritmo el que es inválido, es decir, que es originado por un AS que no esta autorizado a originarlo, puede propagarse al ser la única ruta posible.

Es, por tanto, importante notar que la seguridad no dependerá tanto del resultado del algoritmo como de las políticas locales en cada AS para con los resultados de dicho algoritmo. Este hecho se traduce en que, salvo que se unifiquen las políticas de enrutamiento en relación con BGP-PFX, no se garantiza la no propagación de anuncios inválidos al resto de la red. Además, debemos tener en cuenta que no hay establecida ningún tipo de política de implantación de BGP-PFX de forma que es muy probable que su implantación sea gradual a lo largo de la red, haciendo que aquellos AS que no quieran o puedan implantar BGP-PFX serán vulnerables y podrán, en el caso peor, utilizarse para atacar otros AS a priori seguros.

Es interesante reseñar también que, tal y como hemos podido ver en las figuras 4.1(a) y 4.2(a), existen incoherencias en la red BGP ya que no siempre los mismos AS originan los mismos prefijos. Esto puede deberse a que sean AS de las mismas confederaciones[3] o por decisiones de configuración.

Por tanto también hay que tener en cuenta que con la implantación de BGP-PFX los administradores de todos los AS que actualicen sus sistemas para utilizar RPKI y BGP-PFX deberán tener en cuenta que una mala configuración puede suponer el que el resto de la red no reconozca sus anuncios como válidos al no haber indicado, por ejemplo, que todos los AS miembros de una confederación son orígenes válidos de los prefijos.

[2]La tabla contiene los valores medios de los resultados estadísticos obtenidos en 4.2.

[3]Conjuntos de AS que pertenecen al mismo organismo y que están autorizados a anunciar los mismos conjuntos de prefijos IP

4.2.3. Inserciones, modificaciones y ataques MITM

En la figura 2.6 podemos ver como, en ningún momento, BGP-PFX realiza ningún tipo de comprobación sobre la integridad del mensaje. La única comprobación que realiza es que el par AS origen - prefijo, de forma que la información sobre el resto de AS por los que ha circulado el mensaje no se tiene en cuenta. Además, debemos tener en cuenta que el comportamiento de cada AS ante un mismo resultado del algoritmo viene dado por la política de enrutamiento del mismo, de forma que se pueden aceptar prefijos inválidos o desconocidos como rutas a utilizar en caso de que no haya otra alternativa en lugar de descartar dichas rutas directamente.

Es por ello que este algoritmo, utilizado por si solo, es vulnerable a cualquier tipo de ataque que modifique el tráfico, mediante inserción o modificación de mensajes, y sea totalmente incapaz de llegar siquiera a detectar que dichos ataques. Por tanto, un atacante podría insertar tráfico BGP y este no seria automáticamente descartado en caso de resultar inválido en el algoritmo, ya que dependiendo de las políticas de cada AS dichas rutas falsas podrían almacenarse y llegar a utilizarse.

También cabe la posibilidad de que se modifiquen, sin poder detectar dicha modificación, mensajes de un origen válido de forma que se fuerce a que el tráfico pase por los equipos de un tercero, permitiéndole analizar el tráfico IP que circulase una vez establecida la nueva ruta[4].

Por tanto, podemos concluir que debido a la naturaleza del algoritmo, es vulnerable a cualquier ataque MITM ya que no es capaz de identificar cuando un atacante se ha introducido en la red BGP y esta inyectando o escuchando tráfico.

4.3. Problemática BGPSEC

Como ya hemos explicado en la sección 2.3, BGPSEC trata de incrementar la seguridad en BGP en dos aspectos bien diferenciados. En primer lugar verifica que los anuncios BGP provienen de AS que están autorizados para realizarlos, ejecutando BGP-PFX con los problemas anteriormente discutidos. Sin embargo, para protegerse de los ataques discutidos en la sección 4.2.3, introduce el que cada router eBGP por el que pase el anuncio BGP, de forma que se aumente la seguridad. Esto, sin embargo, genera otro tipo de problemas ya que todas estas operaciones no son instantáneas, lo cual disminuye el número de mensajes por segundo que puede gestionar un router.

Además hay que tener en cuenta que las modificaciones del mensaje BGP para permitir la ejecución de este algoritmo hacen que la información útil que pueda transportarse, como el número de sistemas autónomos por el que puede pasar el anuncio, disminuya ya que la longitud máxima de los mensajes BGP es fija.

4.3.1. Velocidad de firma y verificación

En primer lugar debemos discutir si realmente el hecho de firmar los mensajes y, por tanto, tener que verificarlos en cada router eBGP por el que pasen puede generar un retardo cual que la red no fuese capaz de dar un servicio a todos los anuncios que se generan y, por tanto, las prestaciones del servicio empeorarían. Por ello debemos medir cuanto se tarda en firmar un bloque de datos y cuanto se tarda en verificar dicha firma.

Para la medida de las velocidades de firma y verificación de firmas hemos utilizado las funciones proporcionadas a tal efecto en openssl[5] para la utilización de curvas elípticas de la familia ECDSA. En concreto se utiliza la combinación del algoritmo SHA-256 para obtener la función resumen de un mensaje de 4KB (tamaño máximo de los mensajes BGP [10]).

El cálculo se ha efectuado realizando experimentos de treinta minutos de duración durante los cuales se ha medido el número de operaciones de firma o verificación realizadas. A partir de estos datos, y realizando seiscientos experimentos independientes, trescientos

[4]Recordemos que el propósito del protocolo BGP es establecer rutas para acceder a las distintas direcciones IP desde cualquier otro punto de la red.

[5]http://www.openssl.org/

para los parámetros de la operación de firma y trescientos para los de la operación de verificación, podemos obtener datos estadísticos sobre el número firmas y el número de verificaciones que es capaz de realizar el sistema, así como el tiempo de cada operación. Para poder obtener resultados significativos, hemos utilizado equipos con una potencia de cálculo similares a las de los routers del apéndice A. En concreto hemos utilizado un equipo, con denominación 'lapa', con un procesador Intel(R) Pentium(R) 4 CPU 1.70GHz y 774380 kB de memoria RAM y otro equipo, con denominación 'moscardón', con un procesador Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz y 3994340 kB de memoria RAM. Los resultados los podemos observar en las figuras 4.3 a 4.14, con los parámetros estadísticos de media, mediana y desviación típica de cada medida resumidos en la tabla 4.6.

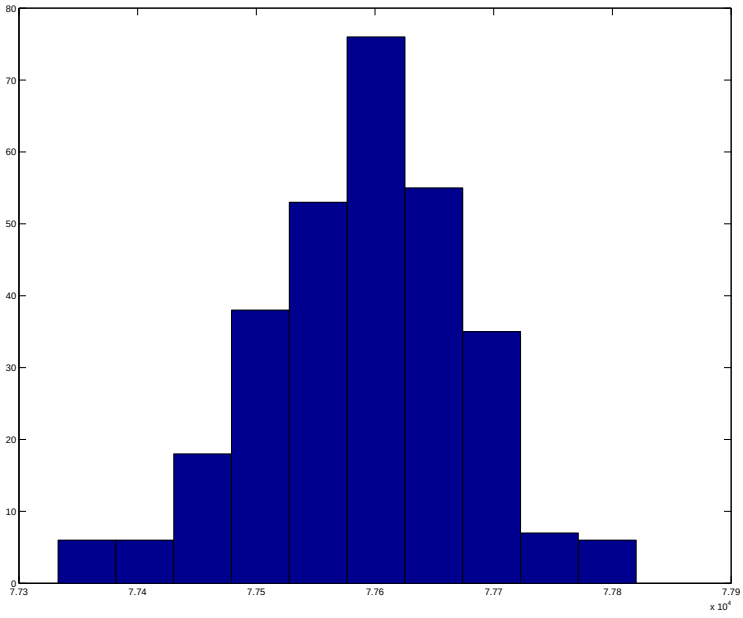


Figura 4.3: Número de firmas en 1800 segundos con ECDSA 256. [Resultados de lapa]

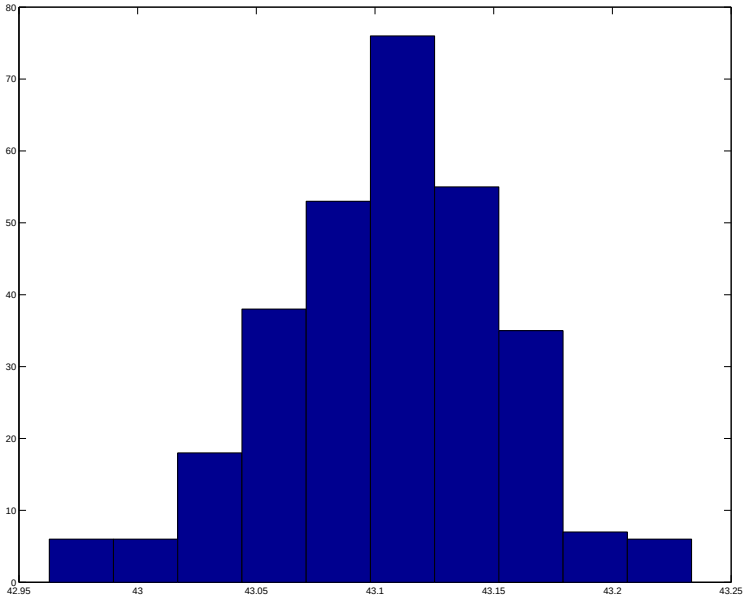


Figura 4.4: Número de firmas por segundo con ECDSA 256. [Resultados de lapa]

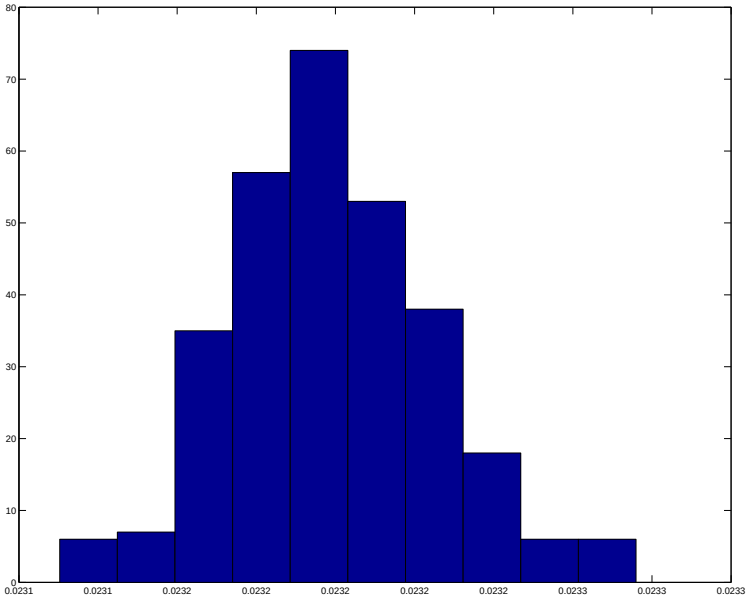


Figura 4.5: Tiempo en realizar una firma (en segundos) con ECDSA 256. [Resultados de lapa]

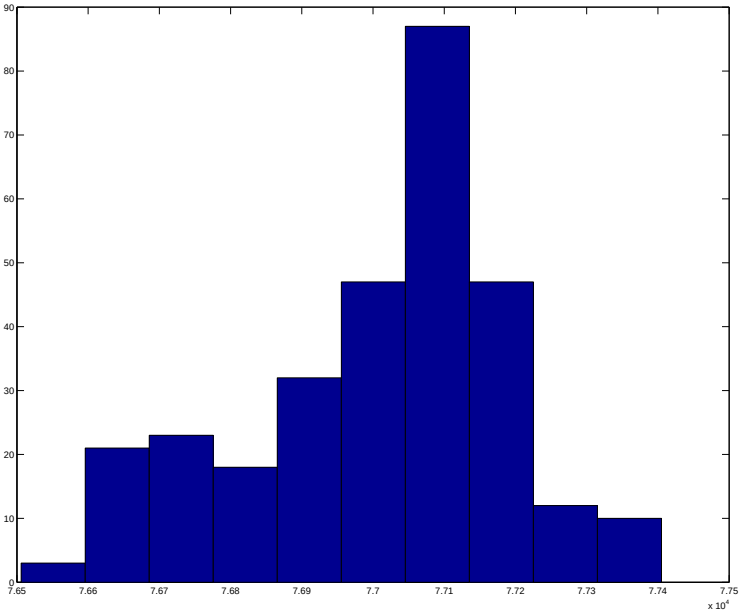


Figura 4.6: Número de verificaciones en 1800 segundos con ECDSA 256. [Resultados de lapa]

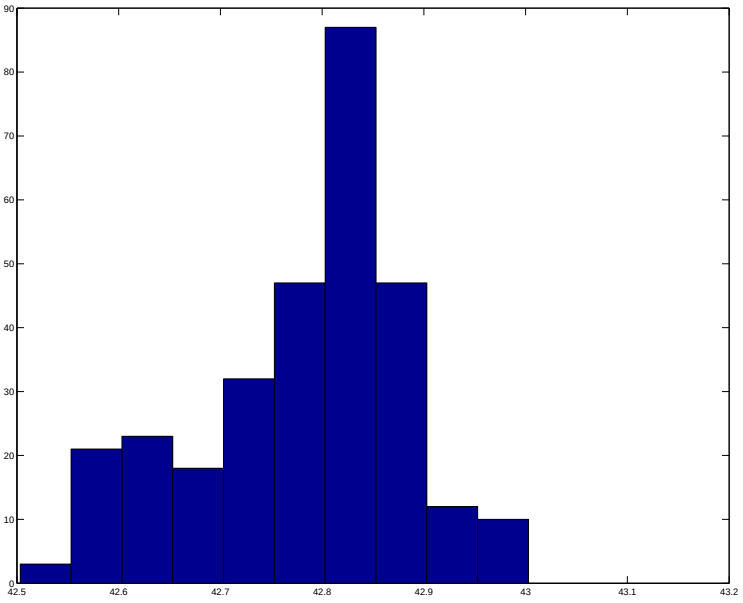


Figura 4.7: Número de verificaciones por segundo con ECDSA 256. [Resultados de lapa]

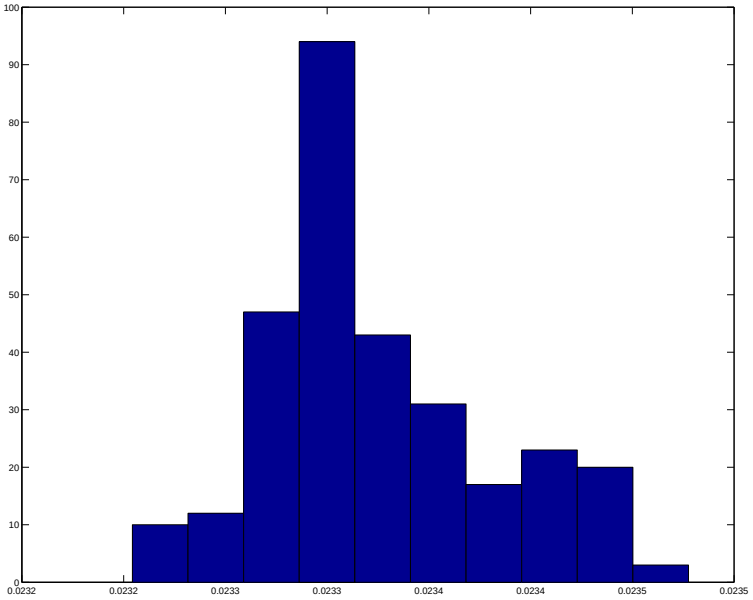


Figura 4.8: Tiempo en realizar una verificación (en segundos) con ECDSA 256. [Resultados de lapa]

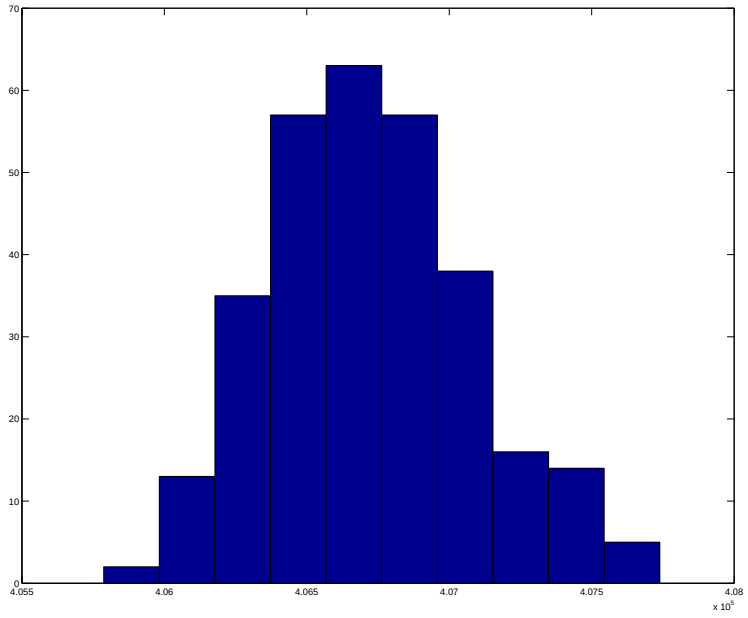


Figura 4.9: Número de firmas en 1800 segundos con ECDSA 256. [Resultados de moscardon]

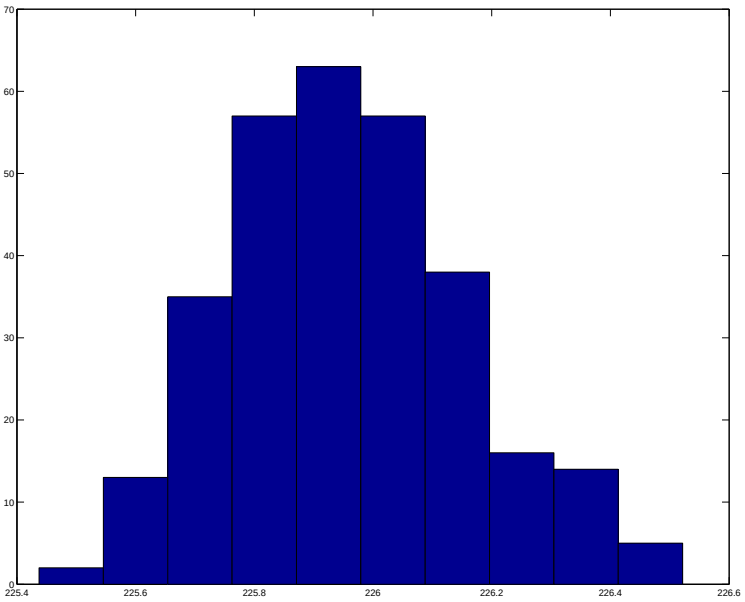


Figura 4.10: Número de firmas por segundo con ECDSA 256. [Resultados de moscardon]

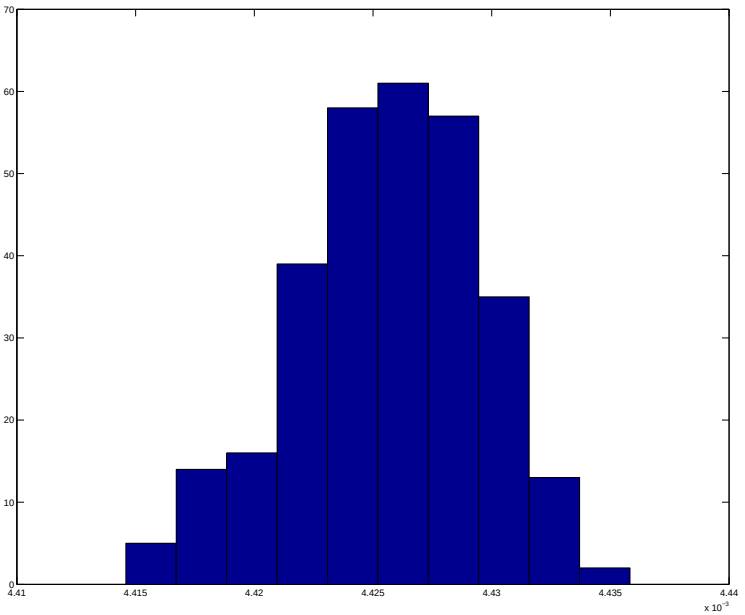


Figura 4.11: Tiempo en realizar una firma (en segundos) con ECDSA 256. [Resultados de lapa]

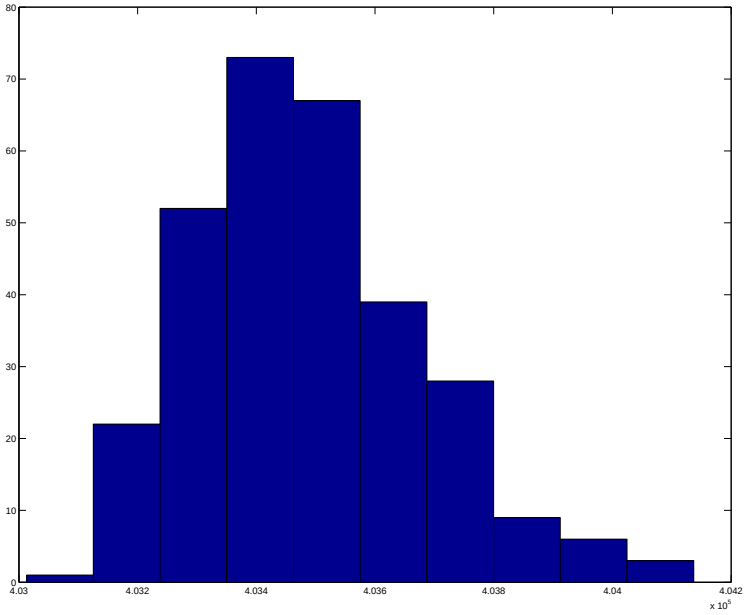


Figura 4.12: Número de verificaciones en 1800 segundos con ECDSA 256. [Resultados de moscardon]

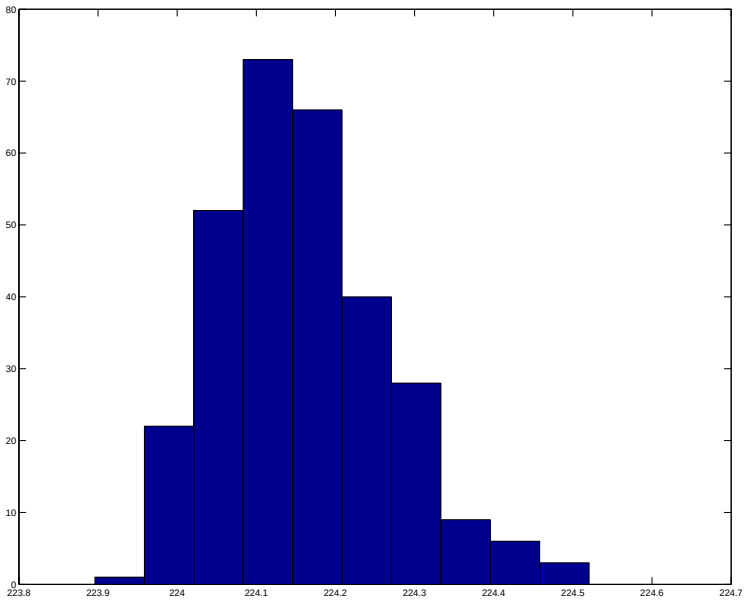


Figura 4.13: Número de verificaciones por segundo con ECDSA 256. [Resultados de moscardon]

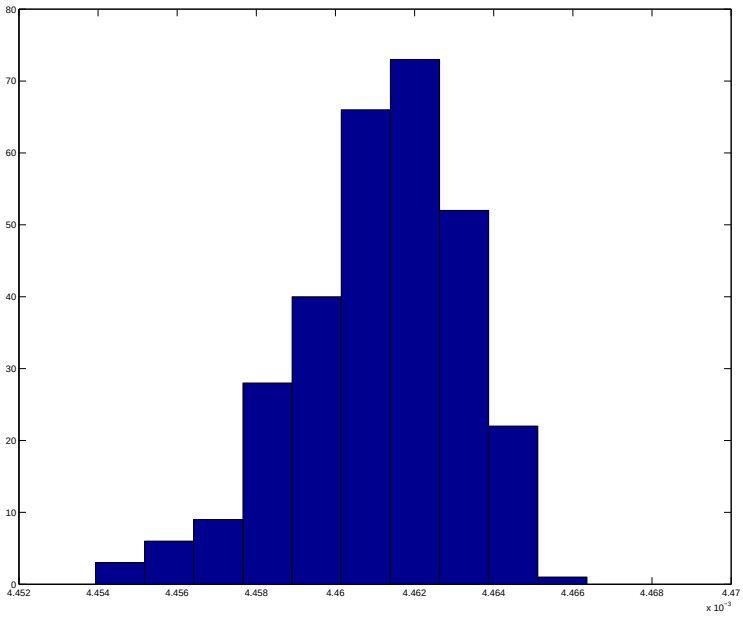


Figura 4.14: Tiempo en realizar una verificación (en segundos) con ECDSA 256. [Resultados de moscardon]

Longitud media del AS Path	4.5
Longitud máxima del AS Path	31

Tabla 4.7: Datos sobre la longitud del AS Path. Fuente: http://thyme.rand.apnic.net/

Debemos tener en cuenta que el sistema autónomo k-ésimo por el que pase un mensaje BGP debe de realizar $k-1$ operaciones de verificación de firma y una única operación de firma. De esta forma, en general, el tiempo de procesado será el que se indica en la ecuación 4.1.

$$tiempo_{procesado} = tiempo_{firma} + (k-1) \cdot tiempo_{verificacion} \tag{4.1}$$

Con esta información, podemos calcular cuanto se tardaría en procesar un caso en el que el anuncio ha pasado por el número medio de AS, redondeándolo hacia arriba ya los AS o se recorren o no, pero no se pasa por medio AS, como el caso peor donde se encuentra con el número máximo de saltos. Los resultados de dichos cálculos se pueden ver en la tabla 4.8.

	Lapa		Moscardón	
	(k-1)=5	(k-1)=31	(k-1)=5	(k-1)=31
Tiempo en procesar un mensaje (en segundos)	0,1400719	0,7478115	0,0267311	0,1427197
Mensajes por segundo	7,13919066	1,33723539	37,409609	7,00674119

Tabla 4.8: Tiempos de procesado y mensajes procesados por segundo.

Si ahora comparamos estos resultados con las mediciones actuales sobre el tráfico BGP mostradas en los tabla 4.1 y 4.2 podemos ver como, aunque se cumplen los tiempos medios,

el sistema no sería capaz de dar servicio en un momento de tráfico intenso en que se alcanzase un pico de mensajes, lo cual resultaría en una congestión de la red.

Finalmente, debemos reseñar que estos cálculos se han realizado asumiendo que el router pudiese acceder de manera instantánea a las claves públicas de los routers por los que haya pasado el paquete BGPSEC, es decir, no se perdiese tiempo en la búsqueda de dichas claves en una estructura de datos. Sin embargo, este tiempo de búsqueda puede llegar a ser una componente crítica ya que, aunque el número total de routers eBGP es desconocido, podemos asumir que se trata, como mínimo, de un router por AS.

Por tanto utilizando algoritmos de búsqueda de tipo 'binary search' la complejidad computacional sería $O(log\, n)$[43] lo cual, aunque mejor que con otro tipo de algoritmos, es no despreciable debido al gran tamaño de la búsqueda. Es cierto que habrá determinadas firmas que el router podrá tener siempre en memoria para no tener que emplear tiempo en buscarlas, como son aquellas de sus vecinos más próximos, pero hay que tener en cuenta que a partir de cierto número de saltos, el origen puede estar en cualquier punto de la red, por lo que es inevitable que tenga que realizar dichas búsquedas.

4.3.2. Limitación en el tamaño de los mensajes BGPSEC

Otra posible limitación en las prestaciones de BGPSEC frente a las de BGP reside en el hecho de que, al introducir algoritmos de firma, el paquete contiene más cabeceras para la misma cantidad de información. Dado que el tamaño de un paquete BGP viene limitado por las características de TCP[6], tanto el contenido del NRLI como las cabeceras y toda la información extra pueden ocupar, como máximo, cuatro mil noventa y séis octetos, en otras palabras, cuatro kilobytes.

La limitación por tamaño del mensaje al número de saltos debe tener en cuenta el tamaño de las cabeceras fijas así como el incremento que tiene lugar por cada salto. Además, tal y como comentamos en la sección 2.3.3, puede hacer uno o dos bloques de firmas, lo que hace que en el caso en que estén conviviendo dos algoritmos de firmas el número máximo de saltos variará.

Sabiendo que las cabeceras del mensaje BGPSEC ocupan cuarenta y tres octetos y, suponiendo que se utiliza un único bloque de firma, que, por cada salto, el mensaje crece en noventa y nueve octetos debido a los incrementos en el campo AS PATH, así como a la firma que se concatena en el mensaje, el cálculo indica que, como máximo, el mensaje puede recorrer cuarenta sistemas autónomos distintos.

Sin embargo, si suponemos dos bloques de firmas, y suponiendo que en ambos casos se utilizan firmas del mismo tamaño (setenta y un octetos), obtenemos que el número máximo de saltos es de veintiuno. Esto se debe a que el tamaño de las cabeceras se incrementa en tres octetos, como consecuencia de tener el segundo bloque de firmas, pero el incremento por cada salto es ahora de ciento noventa y dos octetos, dado que hay que tener en cuenta el hecho de que por cada sistema autónomo por el que pase el mensaje se firma utilizando dos protocolos distintos. En teoría, esta situación se daría solo cuando se estuviese migrando de un algoritmo de firma a otro, pero es importante señalarlo.

Por tanto, si comparamos esto con los datos de la tabla 4.7 podemos ver que, en el caso de que se utilice como único algoritmo de firma ECDSA 256, el rendimiento se mantiene, ya que es capaz de proporcionar un mayor número de saltos que el requerido en la práctica, aunque menor que el que proporciona BGP. Sin embargo, si se utilizan dos algoritmos de firma, el rendimiento baja al sesenta y cuatro con cinco por ciento en el caso peor.

En cambio, si comparamos con el tamaño máximo posible de un paquete BGP, el rendimiento disminuye en ambos casos. En un anuncio BGP transmitido en las mismas condiciones que los anuncios BGPSEC, es decir, con un NRLI de longitud cinco y un atributo WITHDRAW ROUTES de longitud 1 tenemos que la longitud fija del paquete es de, teniendo en cuenta los atributos obligatorios como ORIGIN y NEXT HOP es de cuarenta y tres octetos, lo que deja una longitud máxima teórica del AS PATH de cuatro mil cincuenta y tres octetos. Por tanto, en teoría, un anuncio BGP podría recorrer, como máximo, mil trece AS distintos.

[6]Recordemos que BGP se envía como PDU de un paquete TCP

Comparando esta cifra con las obtenidas en este apartado, nos encontramos con que en el caso de utilizar un único algoritmo de firma tenemos que el rendimiento se sitúa en el tres con noventa y cinco por ciento, mientras que si utilizamos dos algoritmos de firmas el rendimiento es del uno con noventa y siete por ciento.

4.3.3. Almacenamiento de las claves públicas de los routers

Como ya hemos mencionado, en BGPSEC la clave privada que se utiliza para firmar los mensajes en BGPSEC es única para cada router [40], lo que hace que un router que valide mensajes BGPSEC deba tener acceso a las claves públicas de todos los demás routers que utilicen el protocolo. Esta decisión de diseño se ha tomado para que, en caso de que un router se vea comprometido, se pueda revocar únicamente su certificado, sin afectar al resto de routers del mismo sistema autónomo.

Sin embargo, dicha decisión tiene un inconveniente ya que obliga a que bien sea necesario disponer de dos sistemas de clave pública, RPKI por un lado y otro sistema que almacene la información de los routers por otro, bien RPKI tenga un número de objetos firmado tan grande que pueda disminuir el rendimiento del mismo y su manipulación se haga muy compleja debido al tamaño.

A pesar de la falta de datos, para poder hacer los cálculos, realizaremos una comparación para el caso en que haya un único router por sistema autónomo, que haya diez o que haya cien para poder comprender porque el número de routers con capacidad de firma puede ser un problema para BGPSEC ya que, como hemos mencionado en la sección 4.3.1, dicho número tiene una influencia directa sobre el tiempo que tardan en gestionarse los paquetes entrantes en un router que deba realizar verificaciones de firma.

4.3.4. El problema de los routers iBGP

En el diseño de BGPSEC se plantea el que solo firmen los mensajes los routers eBGP por los que pasen los mismos. Esta decisión de diseño plantea el problema de que dentro de un sistema autónomo se pierde la seguridad que otorga BGPSEC para el protocolo BGP.

Esta decisión hace que la red BGP no sea segura en su totalidad ya que un atacante que pudiese llegar a controlar un router iBGP podría utilizar dicha ventaja para realizar ataques de tipo MITM sobre sus vecinos inmediatos.

4.3.5. Implantación gradual de BGPSEC en la red BGP

El diseño tanto de BGPSEC como de BGP-PFX permiten el que no se implante en toda la red BGP al mismo tiempo sino que se vaya migrando de manera escalonada en función de las posibilidades de los distintos AS. Esta implantación gradual de BGPSEC da lugar a que sea necesario, para el correcto funcionamiento de Internet, el que los sistemas que utilicen BGPSEC puedan utilizar rutas que el protocolo ha detectado como inválidas cuando no exista una alternativa mejor.

Lo expuesto anteriormente implica que los ataques de tipo secuestro de prefijos puedan seguir llevándose a cabo, afectando incluso a sistemas que utilicen BGPSEC, si el atacante se aprovecha de que la seguridad podrá no implantarse de manera uniforme sobre toda la red, permitiendo la convivencia de partes seguras con partes inseguras.

	Lapa			Moscardón		
	Media	Mediana	Desviación típica	Media	Mediana	Desviación típica
Firmas en 1800 segundos	77590	77595	87,9247	406730	406708	359,6438
Firmas por segundo	43,1055	43,1083	0,0488	225,9609	225,9489	0,1998
Velocidad de firma (en segundos)	$23{,}1989 \cdot 10^{-3}$	$23{,}1974 \cdot 10^{-3}$	$26{,}2975 \cdot 10^{-6}$	$4{,}4256 \cdot 10^{-3}$	$4{,}4258 \cdot 10^{-3}$	$3{,}9121 \cdot 10^{-6}$
Verificaciones en 1800 segundos	77007	77052	181,6627	403490	403465	194,2343
Verificaciones por segundo	42,7818	42,8067	0,1009	224,1590	224,1469	0,1079
Velocidad de verificación (en segundos)	$23{,}3746 \cdot 10^{-3}$	$23{,}3608 \cdot 10^{-3}$	$55{,}2130 \cdot 19^{-6}$	$4{,}4611 \cdot 10^{-3}$	$4{,}4614 \cdot 10^{-3}$	$2{,}1469 \cdot 10^{-6}$

Tabla 4.6: Datos estadísticos de los experimentos

Capítulo 5

Conclusiones y trabajo futuro

A lo largo de este documento hemos estudiado, en primer lugar, el estado de las tres tecnologías que se están desarrollando para dotar de seguridad a las redes interdominio para, después, centrarnos en el estudio de los principales problemas que presenta cada una de ellas.

Para realizar este estudio hemos analizado cuantitativamente los problemas de RPKI partiendo de los datos conocidos de la red BGP y realizando suposiciones de aquellos datos de los que no disponemos. También hemos realizado simulaciones de rendimiento del algoritmo BGP-PFX y de los algoritmos de firma de BGPSEC. Además, hemos analizado en profundidad los posibles problemas de seguridad de las tres tecnologías.

Podemos, por tanto, extraer conclusiones al respecto así como sugerir posibles líneas para el trabajo futuro en el área de la seguridad de las redes interdominio.

5.1. Políticas de enrutamiento

Como hemos podido ver en las secciones 2.2 y 2.3, los resultados que se obtienen de BGP-PFX y BGPSEC no implican que exista un protocolo de actuación homogéneo en toda la red. Esto tiene como consecuencia que pueda darse el caso de que habiendo un mensaje que se identifica como inválido este no solo no sea descartado, sino que la ruta propuesta por el mismo se utilice en caso de no existir una alternativa mejor. Esta falta de criterios comunes de hace que pueda darse el caso en que, si las políticas de enrutamiento no son lo suficientemente restrictivas, se puedan propagar rutas erróneas de forma que se ocasionen nuevos secuestros de prefijos.

Es por ello que una línea de investigación posible es el desarrollo de políticas que prevengan estos casos de forma que haya un marco común de prácticas recomendadas para toda la red BGP.

5.2. Estados de implantación

Es importante notar que BGP-PFX y BGPSEC están diseñados para poder ser implantados gradualmente, de forma que es posible que en un mismo momento convivan AS con BGP junto a otros que tengan implantado BGP-PFX o BGPSEC. Por tanto, debe asegurarse el buen funcionamiento de la red bajo esta situación, garantizando que un AS que no tenga implantada ningún tipo de seguridad siga recibiendo el tráfico que le corresponde de la red BGP.

Para esto hay que tener en cuenta que BGPSEC marca como inválido todo anuncio que no haya resultado válido en BGP-PFX, es decir, tanto los que hayan resultado desconocidos como los que hayan resultado inválidos, por lo cual habría que estudiar cómo conjugar ambos resultados en la toma de decisiones BGP para no descartar mensajes de AS que no hayan implantado seguridad pero, al mismo tiempo, no propagar mensajes inválidos.

5.3. ¿Firmas software o firmas hardware?

En la sección 4.3.1 se explica como el tiempo de firma y verificación de firma utilizando procedimientos software no llega a cumplir los requisitos temporales en los casos en que existan picos de tráfico. Esto se debe a que el cifrado software, al realizarse en un microprocesador de propósito general, pese a ser igual de efectivo que el cifrado hardware desde el punto de vista de seguridad, es mucho más lento ya que los microprocesadores de propósito general están pensados para poder realizar tareas de muy diversa índole y seguir siendo económicos. Sin embargo, esta versatilidad tiene como contrapartida el hecho de que no son tan eficientes como los equipos específicos. Es por ello que la firma hardware, al utilizar tarjetas específicas para cada protocolo de firma, hace que los procesos tanto de firma como de verificación sean mucho más rápidos.

Es importante notar que al ser los algoritmos ECDSA de desarrollo reciente, todavía no se han fabricado tarjetas de cifrado hardware para esta familia de algoritmos, abre una posible línea de trabajo futuro: el diseño y la fabricación de tarjetas hardware que sean capaces de firmar y verificar las firmas de manera que se cumplan los requisitos temporales para un adecuado funcionamiento de BGPSEC.

También debe tenerse en cuenta que no todos los routers son compatibles con tarjetas hardware, así como algunos RIR que están estudiando el uso de routers software como XORP, por lo que otra posible línea de trabajo consiste en la integración de tarjetas de firma hardware compatibles con dichos sistemas y que sean igual de eficientes que en el caso de que el router si este preparado para utilizarlas. Además, es necesario el actualizar el software de dichos equipos para permitir la inclusión de los resultados de BGPSEC en sus algoritmos de decisión.

5.4. Algoritmos de firma

Uno de los principales inconvenientes del incremento de seguridad en un protocolo es que su eficiencia disminuye. En este caso, dicha disminución de la eficiencia se traduce tanto en un aumento del tiempo que se tarda en procesar cada mensaje BGP, disminuyendo así la capacidad del router, como en un aumento del tamaño de las cabeceras y otra información de control del mensaje BGP, haciendo que el ancho de banda disminuya al poder transmitir menos información útil por paquete[1].

Es por estos motivos que es importante, y de hecho BGPSEC prevé esta posibilidad, seguir investigando en nuevos algoritmos de firma para poder compactar la información de la firma, de forma que la relación entre la cantidad de información útil transmitida en el mensaje y el tamaño del mensaje sea la mayor posible.

5.5. Inversión en estructuras

A la hora de estudiar la inversión que deberá llevarse a cabo, tenemos que distinguir entre las tres tecnologías estudiadas en este documento ya que cada una tiene unos requisitos muy distintos. Es por ello que, en función de qué desee implantar cada sistema autónomo, la inversión podrá ir desde la necesaria para realizar una simple actualización software a la requerida para cambiar todos los equipos de red del AS.

5.5.1. RPKI

La inversión para establecer una red RPKI es la necesaria para establecer las cachés y las bases de datos necesarias para almacenar toda la información, además de la requerida para mantener dicha información al día. Dado que todas estas tareas se puede realizar sin necesidad de comprar equipos de propósito específico, es una inversión razonable para un AS. Además, hay que tener en cuenta que en aquellos AS donde el enrutamiento se realice mediante routers software como XORP es posible que las propias máquinas que

[1] Recordemos que el tamaño del paquete BGP viene fijado por el tamaño máximo del payload del protocolo TCP que es de cuatro kilobytes.

almacenan el router software puedan realizar también parte de las funciones requeridas por RPKI, haciendo que la inversión disminuya en gran medida. A esta inversión en infraestructuras habría que añadirle la inversión en seguridad necesaria para asegurar el buen funcionamiento de RPKI.

Es importante notar que los RIR están empezando a ofrecer RPKI como un servicio más a sus usuarios, de forma que la inversión en infraestructuras no es obligatoria al ser posible contratarlo en caso de que el AS no desee o no pueda administrarlo.

5.5.2. BGP-PFX

Como hemos podido ver en la sección 2.2, para poder implantar BGP-PFX son necesarias tres condiciones:

- Una infraestructura RPKI para obtener la información necesaria para el algoritmo.
- Una estructura de datos que almacene la información sobre los prefijos obtenida de RPKI.
- Un equipo que permita modificaciones en su proceso de decisión para incluir los resultados del algoritmo.

En caso de que cumplamos estos tres requisitos, la implantación BGP-PFX pasa por una actualización del software del router, sin necesidad de mayores modificaciones en hardware. En el peor de los casos, de que el router no tenga espacio para almacenar la información de la estructura de datos, sería necesario conectar un sistema de almacenaje al equipo. Como puede verse en el apéndice A, distintos modelos de routers comerciales permiten dicha ampliación. Sin embargo, hay que tener en cuenta que es necesaria la inversión para mantener o contratar RPKI, de forma que se deben tener en cuenta las consideraciones de la sección 5.5.1.

Si los routers utilizados por el AS no permitiesen las modificaciones software necesarias, la inversión requerida sería mucho mayor, ya que a los costes de la sección 5.5.1 y a los de la modificación de la lógica de BGP para que tenga en cuenta los resultados de BGP-PFX, habría que sumarles la renovación de todos los routers BGP del AS.

5.5.3. BGPSEC

Para evaluar la inversión necesaria para la implantación de BGPSEC hay que tener en cuenta que BGPSEC necesita que estén implantados en el mismo AS tanto RPKI como BGP-PFX. A estos costes hay que añadirles la necesidad del uso de tarjetas de firma hardware, tal y como hemos concluido en la sección 5.3. Dichas tarjetas deben poder aceptar nuevos algoritmos ya que, como se explica en la sección 5.4, los algoritmos de firma seguirán evolucionando, de forma que el sistema debe poder actualizarse sin necesidad de modificar el hardware.

En el caso de que los routers no permitan la instalación de tarjetas software, como ocurre con los de gama baja, los costes implicarían una modernización del parque de routers a modelos que sí admitan dichas tarjetas, lo que implica una inversión aún mayor.

5.6. Implantación gradual

A la vista de las secciones 5.5.1, 5.5.2 y 5.5.3, cabe pensar que la manera más lógica de implantar la seguridad en BGP es partir de una primera fase donde se implante RPKI, ya que tanto BGP-PFX, BGPSEC o las soluciones como RIVET, S-BGP o soBGP coinciden en la necesidad de una infraestructura PKI como base de la seguridad. Es, por tanto, lógico plantear que una implantación gradual posible sería aquella en la que, en primer lugar, toda la red BGP implantase RPKI para, en un segundo paso, implantar BGP-PFX y, finalmente, BGPSEC.

Es interesante notar que una implantación de este tipo, aunque lenta, disminuye los posibles agujeros de seguridad en la red ya que el punto de implantación es uniforme en

toda ella, evitando las situaciones en las que algunos AS tengan ya implantado BGPSEC mientras que otros no tengan implantado nada. Pese a todo, es necesario el diseño de algún tipo de calendario de implantación para poder coordinar esfuerzos y que no se den situaciones como la de la implantación de IPv6 [44] o de los AS de treinta y dos bits [45].

5.7. Trabajo futuro

Una de las líneas posibles de investigación que se abren a partir de este proyecto consiste en hacer un análisis de prestaciones experimental de RPKI. Debemos tener en cuenta que, en la elaboración de este proyecto, no hemos podido realizar un análisis práctico sobre las prestaciones de RPKI ni lo hemos podido simular. Es por ello interesante el realizar dichos estudios antes de la implantación total de la tecnología para analizar su comportamiento y posibles mejoras de la misma.

También debe tenerse en cuenta para un futuro la necesidad de, antes de implantar BGP-PFX, analizar como de correctos son actualmente los anuncios BGP. Para ello sería necesario, a partir de los datos proporcionados por los RIRs sobre qué entidades pueden anunciar qué prefijos, estudiar el porcentaje de los anuncios BGP actuales que tienen resultados válidos, inválidos o desconocidos. A partir de estos resultados se pueden realizar otros estudios sobre la coherencia de los anuncios BGP y su relación con la seguridad de la red.

Queda pendiente también, como ya hemos mencionado, la elaboración de políticas recomendadas en el tratamiento de los resultados de BGP-PFX y BGPSEC para tratar de unificar los criterios y mejorar la seguridad, desde un punto de vista global, de la red BGP.

Por otra parte, es necesario desarrollar tarjetas hardware para la firma y la verificación de firmas utilizando algoritmos de la familia ECDSA, así como investigar algoritmos más ligeros para la firma y verificación de firmas ya que, como hemos podido ver a lo largo de este documento, el tiempo de estas operaciones es esencial.

Capítulo 6

Presupuesto

6.1. Descripción del proyecto

Autora: Paula González Muñoz
Departamento: Ingeniería Telemática
Título: Estudio sobre la seguridad en routing interdominio.
Duración (meses): 9
Tasa de costes indirectos: 20 %
Presupuesto total del Proyecto (valores en Euros): 44040

6.2. Desglose presupuestario (costes directos)

Personal

Apellidos y nombre	Categoría	Dedicación (hombres mes)	Coste hombre mes	Coste (Euro)
Valera Pintor, Francisco	Ingeniero Senior	1	4.289,54	8.579,08
González Muñoz, Paula	Ingeniero	1	2.694,39	18.860,73
Hombres mes		**2**	**Total**	27.439,81

Notas: 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas). Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

Equipos

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable (Euro)
Polilla(equipo principal)	600,00	100	9	60	90,00
Lapa (eq.pruebas1)	100,00	100	4	60	6,67
Moscardón (eq.pruebas2)	200,00	100	4	60	13,33
Total					110,00

Fórmula de cálculo de la Amortización: $\frac{A}{B} \cdot C \cdot D$
A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100

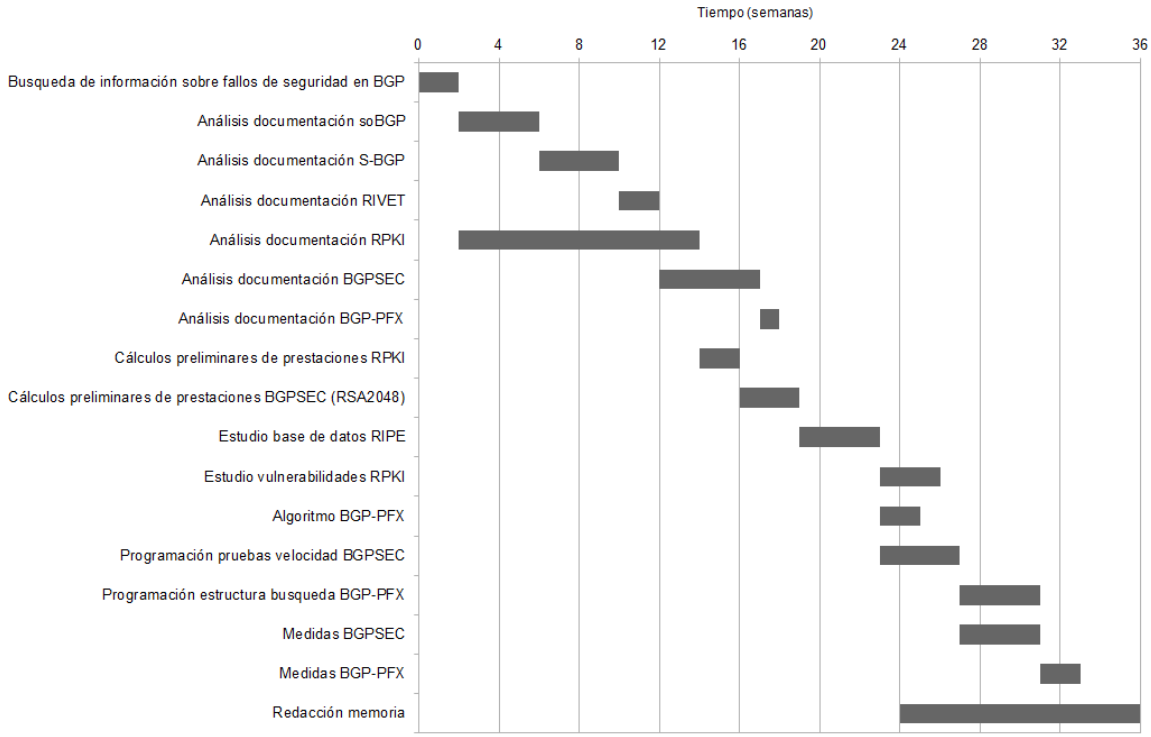
Otros costes directos del proyecto[1]

Descripción	Costes imputable
Alquiler	5.400,00
Luz+agua	1.800,00
Alquiler mobiliario	1.800,00
fungible	150,00
Total	9.150,00

6.3. Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	27.440
Amortización	110
Costes de funcionamiento	9.150
Costes Indirectos	7.340
Total	44.040

6.4. Diagrama de Gantt



[1]Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

Apéndice A

Especificaciones técnicas de los routers BGP de JUNIPER

A.1. Serie T

Specifications (continued)

Routing Engine Options	
RE-A-2000	• 2 GHz supported on TX Matrix, (redundancy required), T1600, and T640 • 2 GHz Intel Celeron processor with integrated 256 KB, level 2 cache • 4 GB DRAM, 1 GB compact flash drive for primary storage • 40 GB IDE hard drive for secondary storage, 128 MB PC card for tertiary storage • 10/100ASE-T auto-sensing RJ-45 Ethernet port for out-of-band management • Two RS-232 (DB9 connector) asynchronous serial ports for console and remote management
RE-DUO-C1800-8G-BB	• Supported on TX Matrix Plus line-card chassis (LCC), T1600, T4000 • Dual core, 1.8 GHz Intel Celeron processor • 8 GB DRAM DIMM, 4 GB compact flash drive, and 4 GB USB • Front pluggable slots for two 64 GB SSD hard drives
RE-DUO-C1800-16G-BB	• Supported on TX Matrix Plus LCC, T1600, T4000 • Dual core, 1.8 GHz Intel Celeron processor • 16 GB DRAM DIMM, 4 GB compact flash drive, and 4 GB USB • Front pluggable slots for two 64 GB SSD hard drives
RE-DUO-C2600-16G-BB	• Supported on TX Matrix Plus • Dual core, 2.66 GHz Intel Celeron processor • 16 GB DRAM DIMM, front pluggable 4 GB compact flash drive, and 4 GB USB • Front pluggable slots for two 64 GB SSD hard drives

Juniper Networks Services and Support

Juniper Networks is the leader in performance-enabling services that are designed to accelerate, extend, and optimize your high-performance network. Our services allow you to maximize operational efficiency while reducing costs and minimizing risk, achieving a faster time to value for your network. Juniper Networks ensures operational excellence by optimizing the network to maintain required levels of performance, reliability, and availability. For more details, please visit **www.juniper.net/us/en/products-services**.

8

A.2. Serie M

Ordering Information

Model Number	Description
M7i Bundles with RE-400	
M7iE-AC-RE400-2FETX-B	M7i base unit: 4 PIC slot chassis, 2 built-in Fast Ethernet ports (RJ connectors), cooling, midplane, 1 AC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 400 MHz; optional flash media kit sold separately
M7iE-AC-RE400-1GE-B	M7i base unit: 4 PIC slot chassis, 1 built-in Gigabit Ethernet port (optics sold separately), cooling, midplane, 1 AC power supply, Junos OS, 1 Enhanced Forwarding Engine, 1 Routing Engine 400 MHz; optional flash media kit sold separately
M7iE-DC-RE400-2FETX-B	M7i base unit: 4 PIC slot chassis, 2 built-in Fast Ethernet ports (RJ connectors), cooling, midplane, 1 DC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 400 MHz; optional flash media kit sold separately
M7iE-DC-RE400-1GE-B	M7i base unit: 4 PIC slot chassis, 1 built-in Gigabit Ethernet port (optics sold separately), cooling, midplane, 1 DC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 400 MHz; optional flash media kit sold separately
M7i Bundles with RE-400	
M7iE-AC-5GE-MS-1800-B	M7i base unit, 5 GbE ports, multi-services module, enhanced IQ2 PIC, NAT/FW license, cooling, midplane, 1 AC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz; optional flash media kit sold separately
M7iE-AC-RE1800-1GE-B	M7i base unit: 4 PIC slot chassis, 1 built-in Gigabit Ethernet port (optics sold separately), cooling, midplane, 1 AC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz; optional flash media kit sold separately
M7iE-AC-RE1800-2FETX-B	M7i base unit: 4 PIC slot chassis, 2 built-in Fast Ethernet ports (RJ connectors), cooling, midplane, 1 AC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz; optional flash media kit sold separately
M7iE-DC-RE1800-1GE-B	M7i base unit: 4 PIC slot chassis, 1 built-in Gigabit Ethernet port (optics sold separately), cooling, midplane, 1 DC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz; optional flash media kit sold separately
M7iE-DC-RE1800-2FETX-B	M7i base unit: 4 PIC slot chassis, 2 Built in Fast Ethernet ports (RJ connectors), cooling, midplane, 1 DC power supply, Junos OS, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz; optional flash media kit sold separately
M10i Bundles with RE-400	
M10iE-AC-RE400-B	M10i base unit: 8 PIC slot chassis, cooling, midplane, 2 AC power supplies, 1 enhanced Forwarding Engine, 1 Routing Engine 400 MHz, 1 fan tray, Junos OS and two chassis managers; optional flash media kit sold separately
M10iE-DC-RE400-B	M10i base unit: 8 PIC slot chassis, cooling, midplane, DC power, 1 enhanced Forwarding Engine, 1 Routing Engine 400 MHz, 1 fan tray, Junos OS and two chassis managers; optional flash media kit sold separately

Model Number	Description
M10i Bundles with RE-B-1800X1-4G	
M10iE-AC-RE1800-B	M10i base unit: 8 PIC slot chassis, cooling, midplane, 2 AC power supplies, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz, 1 fan tray, Junos OS and two chassis managers; optional flash media kit sold separately
M10iE-DC-RE1800-B	M10i base unit: 8 PIC slot chassis, cooling, midplane, DC power, 1 enhanced Forwarding Engine, 1 Routing Engine 1,800 MHz, 1 fan tray, Junos OS and two chassis managers; optional flash media kit sold separately
M7i/M10i Spares	
FEB-M10i-M7i-S	M7i/M10i forwarding engine spare
FEB-M7i-SVCS-S	M7i forwarding engine spare with built-in services module
FEB-M10i-M7i-E-S	M7i enhanced forwarding engine spare
FEB-M7i-SVCS-ASM-E-S	M7i enhanced forwarding engine spare with built-in Adaptive Services module
FEB-M7i-SVCS-MS-E-S	M7i enhanced forwarding engine spare with built-in Multiservices module
RE-400-768-S	Routing Engine board spare (400 MHz Celeron, 768 MB DRAM), optional 1 GB compact flash needs to be ordered separately
RE-400-768-WW-S	Routing Engine board spare (400 MHz Celeron, 768 MB DRAM), optional 1 GB compact flash needs to be ordered separately - Junos OS worldwide
RE-B-1800X1-4G-BB	M10i, M7i 1.73 GHz Routing Engine with 4 GB memory, 64 GB SSD, 4 GB CF
RE-B-1800X1-4G-R	M10i, M7i redundant 1.73 GHz Routing Engine with 4 GB memory, 64 GB SSD, 4 GB CF
RE-B-1800X1-4G-S	M10i, M7i spare 1.73 GHz Routing Engine with 4 GB memory, 64 GB SSD, 4 GB CF
RE-B-1800X1-4G-WW-S	M10i, M7i WW 1.73 GHz Routing Engine with 4 GB memory, 64 GB SSD, 4 GB CF
CHAS-MP-M10i-S	M10i chassis spare
CHAS-MP-M7i-1GE-S	M7i chassis spare, 1 built-in GbE port
CHAS-MP-M7i-2FE-S	M7i chassis spare, 2 built-in FE ports
HCM-M10i-S	High availability chassis manager board for M10i
PWR-M10i-M7i-AC-S	M7i/M10i AC power supply spare
PWR-M10i-M7i-DC-S	M7i/M10i DC power supply spare
FANTRAY-M7i-S	M7i fan tray spare
FANTRAY-M10i-S	M10i fan tray spare
CF-UPG2-1G-S	1 GB compact flash upgrade kit for internal media usage on RE-400 or RE-850
RE-CF-1G-S	Compact flash media upgrade for RE
MEM-FEB-256-S	Optional M10i, M7i Forwarding Engine Board (FEB) memory upgrade: 256 MB DRAM module
MEM-RE-256-S	Optional RE memory upgrade: 256 MB DRAM module

Apéndice B

Agradecimientos

Este proyecto es la culminación de un sueño que empezó hace ya más de diez años. Nada de esto hubiera sido posible sin la ayuda de muchísima gente, tantos que me resulta imposible nombrar a todos. Sin embargo, me gustaría agradecer especialmente a algunas personas:

- A Paco, mi tutor en este proyecto, ya que sin su ayuda, su paciencia y su buen humor dudo que hubiese podido acabarlo.
- A mi familia, que ha estado acompañándome y animándome en cada paso de este largo camino.
- A mis compañeros de la Delegación de Estudiantes por esas terapias de grupo cuando todos estábamos demasiado agobiados por todo.
- A Nuria, Rodrigo e Iria por las conspiraciones nocturnas con las que sacábamos fuerzas para seguir adelante aunque fuera un semestre más.
- A David, Samuel, Sandra, Silvia y Victor por su apoyo y ayuda constante.

Bibliografía

[1] Leonard Kleinrock. Information Flow in Large Communication Nets. RLE Quarterly Progress Report, July 1961.

[2] J. C. R. Licklider and W. Clark. On-Line Man Computer Communication. Memorandums, August 1962.

[3] Robert Kahn. Communications Principles for Operating Systems. Internal Memorandum, January 1972.

[4] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A Brief History of the Internet. *ACM SIGCOMM Computer Communication Review*, 39(5):22 a 31, October 2009.

[5] P. Mockapetris. Domain Names - Concepts and Facilities, RFC1034. Technical report, Internet Engineering Task Force, November 1987.

[6] P. Mockapetris. Domain Names - Implementation and Specification. Technical report, Internet Engineering Task Force, November 1987.

[7] Eric C. Rosen and Bolt Beranek. Exterior Gateway Protocol, RFC827. Technical report, Internet Engineering Task Force, October 1982.

[8] Paul Tsuchiya. An Architecture for Network-Layer Routing in OSI. Technical report, Association for Computer Machinery, 1987.

[9] Yakov Rekhter and Tony Li. A Border Gateway Protocol 4 (BGP-4), RFC1771. Technical report, Internet Engineering Task Force, March 1995.

[10] Yakov Rekhter, Tony Li, and Susan Hares. A Border Gateway Protocol 4 (BGP-4), RFC4271, January 2006.

[11] Iljitsch van Beijnum. *BGP*. O'Reilly, September 2002.

[12] Vince Fuller and Tony Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, RFC4632. Technical report, Internet Engineering Task Force, August 2006.

[13] Peter G. Neumann. Internet Routing Black Hole. *The Risks Digest*, 19(12), May 1997.

[14] RIPE. YouTube Hijacking: A RIPE NCC RIS case study. Technical report, RIPE Network Coordination Centre, March 2008.

[15] Andre Toonk. China BGP Hijack. Technical report, BGPmon, April 2010.

[16] RIPE. Analysis of Egyptian Internet outage 27th January - 2nd February 2011. Technical report, RIPE Network Coordination Centre, 2011.

[17] Iván de Moneo. Egipto desaparece del mapa de Internet. El País, 28 January 2011.

[18] Joanne Mikkelson and Karen Seo. Secure BGP (S-BGP). Internet Draft, June 2003.

[19] Russ White. Architecture and Deployment Considerations for Secure Origin BGP (soBGP). Internet draft, Junio 2006.

[20] Brian K. Haberman. *Routin Information Verification Tool for Securing Inter-Domain Routing Information*. PhD thesis, The Johns Hopkins University, July 2011.

[21] Matt Lepinski and Stephen Kent. An infrastructure to support secure internet routing, RFC6480. Rfc, Internet Engineering Task Force, February 2012.

[22] David Cooper, Stefan Santesson, Stephen Farrel, Sharon Boeyen, Russell Housley, and Tim Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC5280. Technical report, Internet Engineering Task Force, May 2008.

[23] Charles Lynn, Stephen Kent, and Karen Seo. X.509 Extensions for IP Addresses and AS Identifiers, RFC3779. Technical report, Internet Engineering Task Force, June 2004.

[24] Russell Housley. Cryptographic Message Syntax (CMS), RFC5652. Technical report, Internet Engineering Task Force, September 2009.

[25] Geoff Huston, George Michaelson, and Robert Loomans. A Profile for X.509 PKIX Resource Certificates, RFC6487. Technical report, Internet Engineering Task Force, February 2012.

[26] Matt Lepinski, Andrew Chi, and Stephen Kent. Signed Object Template for the Resource Public Key Infrastructure (RPKI), RFC6488. Technical report, Internet Engineering Task Force, February 2012.

[27] Yakov Rekhter, Robert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear. Address Allocation for Private Internets, RFC1918. Technical report, Internet Engineering Task Force, February 1996.

[28] Geoff Huston, Samuel Weiler, George Michaelson, and Stephen Kent. Resource Public Key Infrastructure (RPKI) Trust Anchor Locator, RFC 6490. *Internet Engineering Task Force*, February 2012.

[29] Russ Housley, Sam Ashmore, and Carl Wallace. Trust Anchor Format, RFC 5914. Technical report, Internet Engineering Task Force, June 2010.

[30] Samuel Weiler, Dave Ward, and Russ Housley. The rsync URI Scheme, RFC 5781. Technical report, Internet Engineering Task Force, February 2010.

[31] Matt Cooper, Yuriy Dzambasow, Peter Hesse, Susan Joseph, and Richard Nicholas. Internet X.509 Public Key Infrastructure: Certification Path Building. RFC 4158. Technical report, Internet Engineering Task Force, September 2005.

[32] Rob Austein, Geoff Huston, Stephen Kent, and Matt Lepinski. Manifests for the Resource Public Key Infrastructure (RPKI). RFC 6486. Technical report, Internet Engineering Task Force, February 2012.

[33] Telecommunication Standarization Sector of ITU. X.690. Technical report, International Telecommunication Union, July 2002.

[34] Matt Lepinski, Stephen Kent, and Derrick Kong. A Profile for Route Origin Authorizations (ROAs). RFC 6482. Technical report, Internet Engineering Task Force, February 2012.

[35] Andrew Tridgell. *Efficient Algorithms for Sorting and Synchronization*. PhD thesis, The Australian National University, February 1999.

[36] Harshvardhan Tiwari and Krishna Asawa. Cryptographic Hash Function: An Elevated View. *European Journal of Scientific Research*, 43(4):452–465, 2010.

[37] Randy Bush and Rob Austein. The RPKI/Router Protocol. Internet Draft, August 2012.

[38] Pradosh Mohapatra, John Scudder, David Ward, Randy Bush, and Rob Austein. BGP Prefix Origin Validation. Internet Draft, May 2012.

[39] Randy Bush. BGPSEC Operational Considerations. Internet Draft, May 2012.

[40] Kotikalapudi Sriram. BGPSEC Design Choices and Summary of Supporting Discussions. Internet Draft, January 2012.

[41] Mark J. Handley and Eric Rescorla. Internet Denial-of-Service Considerations, RFC 4732. Technical report, Internet Engineering Task Force, November 2006.

[42] XORP. www.xorp.org.

[43] Kyle Loudon. *Mastering Algorithms with C*. O'Reilly Media, Inc, August 1999.

[44] Protocolo de Internet versión 6. www.ipv6.es.

[45] Estadísticas de la tabla de rutas de rrc03.ripe.net.