Seguridad en un entorno multidominio de redes programables

Bernardo Alarcos[1], Marifeli Sedano[2], María Calderón[3]

[1] Área de Ingeniería Telemática, Universidad de Alcalá,
Crta. Madrid-Barcelona, Km 33.600, 28871, Alcalá de Henares (Madrid)

[2] Departamento de Ingeniería de Sistemas Telemáticos,
UPM, ETSI Telecomunicación, UPM, Ciudad Universitaria, 28040, Madrid

[3] Departamento de Ingeniería Telemática, Universidad Carlos III de Madrid,
Av. Universidad 30, 28911, Leganés (Madrid)

[1]bernardo@aut.uah.es, [2]marifeli@gsi.dit.upm.es, [3]maria@it.uc3m.es

Resumen

En este artículo se describe una arquitectura de seguridad para redes programables multidominio. El escenario en el que se aplica esta arquitectura está compuesto por varias redes programables que procesan paquetes de señalización para dar servicios a usuarios. Estas redes programables pueden pertenecer a proveedores de servicios de Internet, operadores de telefonía móvil GPRS o UMTS, o Intranets, teniendo cada una de estas redes programables su propio dominio administrativo. Esto implica que un dominio envía paquetes que van a ser procesados por los nodos programables de otro dominio, para lo cual deben establecer acuerdos de autorización. En el artículo se presenta una solución de negociación dinámica y recursiva de acuerdos entre dominios. La protección de los paquetes se realiza basándose en una solución de seguridad monodominio que se extiende al entorno multidominio.

1. Introducción

Las *redes programables* [1] permiten insertar dispositivos programables en determinados puntos de la arquitectura de red (asociados a nodos). Estos dispositivos tienen la capacidad de ejecutar códigos para procesar determinados paquetes, ofreciendo con ello nuevos servicios o mejorando los ya existentes.

La introducción de programabilidad en la red introduce nuevos riesgos de seguridad. Un importante aspecto que complica más el escenario y la solución de seguridad, es la interacción entre distintos dominios administrativos.

Este trabajo se sitúa en el contexto de redes programables multidominio. Se basa en el prototipo de nodo programable denominado SARA (Simple Active Router Assistant) y la solución de seguridad intradominio denominada ROSA (Realistic Open Security Architecture for Active Networks). El objetivo del artículo es presentar una solución para extender la propuesta de seguridad a un entorno multidominio.

Esta propuesta aporta un mecanismo seguro para extender los servicios de las redes programables a comunicaciones entre dominios pertenecientes a distintos proveedores de servicios (Proveedores de servicios de Internet u operadores GPRS/UMTS).

La estructura del artículo es la siguiente. En la sección 2 se describe la arquitectura del nodo programable SARA (Simple Active Router Assistant) que puede dar soporte a redes IP para usuarios que están en terminales fijos o en terminales móviles (GPRS o UMTS). Además de SARA se describe un escenario de aplicación y la solución de seguridad ROSA adaptada a este escenario. A continuación, en la sección 3 se presenta el escenario multidominio y en la sección 4 se propone una solución de seguridad para extender ROSA a dicho escenario. A continuación se presentan otros trabajos relacionados en la sección 5 y se termina con conclusiones en la sección 6.

2. Red programable basada en SARA

2.1. Nodo programable Sara

SARA[2,3] es una arquitectura de nodo programable desarrollada en la Universidad Carlos III de Madrid bajo el contexto del proyecto IST GCAP[1].

Un nodo programable basado en SARA consiste en un router reprogramado para detectar paquetes de señalización y desviarlos a un *asistente* que es capaz de asociar los paquetes a una aplicación Java que los procesa.

Los paquetes de señalización llevan en su cabecera SARA un código (Ci) que identifica la aplicación que debe procesarlos. Si el asistente no tiene la aplicación la descarga desde un *Servidor de Código*.

[1] GCAP IST Página principal. http://www.laas.fr/GCAP

2.2. Arquitectura de seguridad ROSA

Las redes programables crean un entorno en el que la seguridad [4] juega un papel muy importante. Se hace necesario por lo tanto analizar los riesgos y proponer soluciones. ROSA[5] proporciona servicios de seguridad que permiten:

1) Realizar una descarga segura de código desde los *Servidores de Código (CS)* a los *nodos programables (*PN*)*
2) Los nodos programables pueden comprobar que los paquetes que le llegan son auténticos y están autorizados a ser procesados.

Cada aplicación de red es identificada por un código Ci y tiene asociado un valor secreto Kci que sólo conocen los componentes de la red programables PNs, CS y AS. Estos valores secretos son generados periódicamente por AS, enviados a CS y este a su vez los distribuyen a los PNs cuando estos lo solicitan.

Antes de comenzar a enviar paquetes de señalización hacia el destino, el cliente de la red programable solicita permiso (figura 1) a un *Servidor de Autorización* (*AS*). En dicha solicitud el cliente realmente solicita un servicio que se ofrece con una sesión identificada por los siguientes parámetros:

- Ui: identificador del cliente que solicita el servicio. No se permitirá que se utilice un identificador de usuario diferente en los paquetes de señalización de esta sesión.
- L: direcciones IP del origen y destino de la comunicación. No se permitirá paquetes de señalización con direcciones de origen y destino distintas.
- Ts: tiempos de inicio y fin de la sesión. No se admitirá paquete de señalización fuera de estos valores temporales.
- Ci: código que identifica el servicio solicitado, realmente referencia a la aplicación de red que procesa los paquetes de señalización. No se permitirá que el cliente utilice un Ci diferente en sus paquetes de señalización.

Como contestación a la solicitud de servicio, el AS le proporciona al cliente una clave de sesión K. Esta clave es calculada por el AS utilizando una *función de derivación de claves KDF* y es función de los parámetros de sesión y del valor secreto Kci asociado a la aplicación de red.

$$K = KDF\ (Ui, L, Ts, Ci, Kci)$$

El cliente utilizará K para proteger los paquetes de señalización mediante un mecanismo basado en *hmac*. Pero antes de generar el *autenticador* hmac, el cliente introduce en el paquete los parámetros que identifican la sesión (Ui, L, Ts y Ci). Esta información tiene una doble función; permite a los PNs generar la clave de sesión K para comprobar la autenticidad (hmac) del paquete, y a su vez actúa como información de autorización que viaja en el paquete en forma de credencial, permitiendo al PN comprobar si el paquete cumple con los parámetros que define la sesión: dentro de tiempo (Ts), generado por Ui, desde la fuente al destino correctos (L) y solicita el servicio (Ci) correcto.

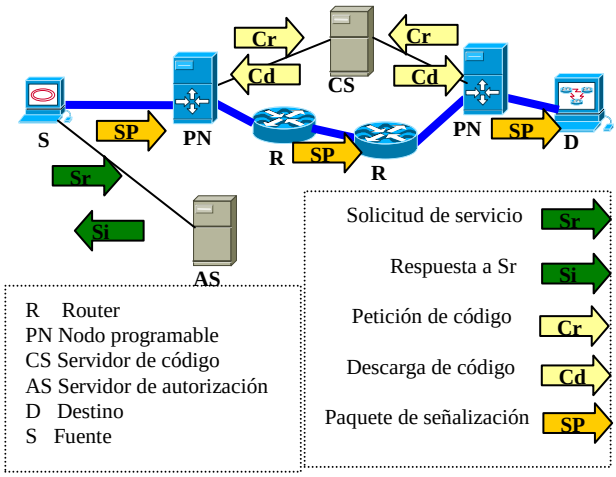

Fig. 1. Arquitectura ROSA

Así cuando un paquete de señalización llega a un PN por primera vez, el PN solicita al CS el código de la aplicación de red identificada por Ci y la clave Kci. Entonces el Pn genera K, verifica que el paquete es auténtico y está autorizado, procesa el paquete con la aplicación de red, y si este ha sido modificado, vuelve a generar el nuevo autenticador usando hmac y K. Tanto Kci como la aplicación de red permanecen en el PN, así para el resto de paquetes de señalización correspondientes a la sesión, no será necesario volver a solicitarlos al CS.

3. Escenario multidominio en redes programables

Planteamos un escenario multidominio con clientes que acceden a los servicios de red desde terminales fijos y móviles (figura 2).

Siguiendo el criterio de colocar los nodos programables cerca del cliente, existen las siguientes posibilidades de dominio en los que puede haber nodos programables:

- ISPs (Internet Service Provider) que dan servicio a clientes.
- Intranets que agrupan un conjunto de clientes pertenecientes a una organización o empresa. Las Intranets a su vez son clientes de los ISPs.
- PLMNs (Public Land-based Mobile Network) que dan servicio a clientes que utilizan terminales móviles GPRS o UMTS.

La figura 3 representa el escenario considerando sólo las redes programables. Podemos ver que en el caso más extremo de comunicación interdominio, intervienen cuatro dominios programables. La red de transporte aglutina el resto de redes troncales de la figura 2 que no tienen nodos programables.

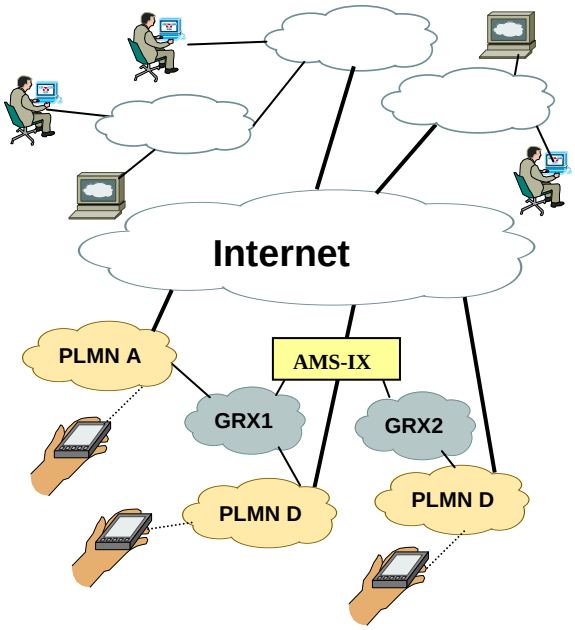

Fig. 2. Escenario multidominio.

Para que los paquetes de señalización puedan ser procesados por todos los dominios, debe existir autorización de estos. Esto implica que los dominios deben negociar para llegar a acuerdos sobre autorización y posiblemente tarificación. El modelo de negociación propuesto consiste en que cada dominio negocia con el siguiente que encuentra en el camino, estableciéndose una cadena de como máximo tres negociaciones (si hay 4 dominios).

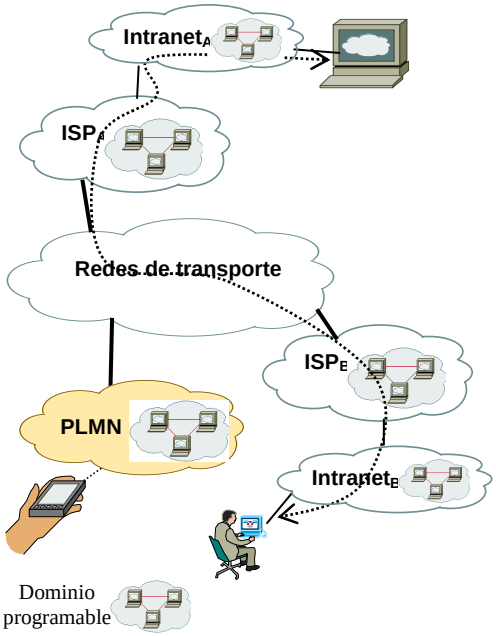

Fig. 3. Escenario multidominio programable.

4. Descripción de seguridad en el escenario multidominio

En cada dominio existe una relación de confianza entre los componentes de la red programable (PN, AS y CS). Esta relación de confianza permite que estos componentes puedan compartir la clave secreta Kci asociada a cada código.

Las Kcis no deben extenderse más allá de un dominio debido a que esto crearía una relación de confianza excesivamente amplia y poco realista. Esto nos lleva a plantear un escenario con varios dominios en los cuales cada uno tiene sus propias Kcis. Así las claves de sesión que generan los nodos programables para una sesión son distintas en cada dominio.

Podemos distinguir dos fases en la propuesta de seguridad multidominio:

1) *Negociación entre dominios para establecimiento de sesión.*
2) *Envío de paquetes de señalización protegidos por el camino multidominio.*

4.1. Proceso de negociación entre dominios para establecimiento de sesión

Este proceso a su vez se divide en dos subprocesos:

1) *Descubrimiento de dominios en un camino*: permite al AS de cada dominio saber con qué AS debe negociar.
2) *Negociación entre dominios*: permite obtener autorización de todos los dominios por los que pasa el paquete y la información de autorización para poder procesar los paquetes.

La figura 4 ilustra los dos subprocesos en un entorno con 4 dominios A, B C y D (el caso más extenso posible). En cada uno de estos dominios hay una clave Kci diferente: Kcia, Kcib, Kcic y Kcid para una determinada aplicación de red y por lo tanto se generará una clave de sesión diferente Ka, Kb, Kc y Kd para una determinada solicitud.

Cuando la fuente solicita el establecimiento de una sesión, genera un valor **IDs** que **identifica la solicitud de forma única** en la red. Este valor se va propagando por las entidades (ASs y PNs) de la figura 4 en los mensajes y es utilizado para identificar los mensajes que le llegan a estas entidades correspondientes a una solicitud de servicio en progreso. Más adelante veremos más detalles del uso de este parámetro.

Explicaremos en los siguientes apartados los dos subprocesos y los mecanismos de seguridad utilizados en cada uno de ellos.

Subproceso de descubrimiento de dominios en un camino.

Cuando un cliente de una red programable solicita un servicio desde la fuente S (mensaje 1, figura 4), al

comunicar al AS los datos del servicio, le indica entre otros parámetros, el destino con el que va a comunicar. El AS, que tiene conocimiento de las direcciones de red que pertenecen a su dominio, determina si el destino pertenece a otro dominio y por lo tanto se trata de una comunicación interdominio. Si la comunicación es interdominio hay que lanzar un proceso recursivo que permita a cada dominio en el camino negociar con el siguiente, hasta alcanzar el destino.

El proceso recursivo tiene una iteración por cada nuevo dominio que se encuentre en el camino hasta un total de 3 (caso de que haya 4 dominios). Cada una de las iteraciones está compuesta por cuatro mensajes:

- **Solicitud de descubrimiento**: con este mensaje un AS inicia una iteración de descubrimiento del siguiente dominio, solicitando a su nodo programable (o a la fuente en la primera iteración) que envíe un paquete *explorador* hacia el destino del camino.
- **Explorador**: es un paquete de señalización enviado hacia el destino por la fuente o nodo programable que recibe la *solicitud de descubrimiento*. El paquete *explorador* lleva información de identidad del AS que ha iniciado esta iteración.
- **Solicitud de identidad**: el primer nodo programable de un dominio distinto al del AS que ha iniciado la iteración, que captura el *explorador*, le envía una *solicitud de identidad* a su AS. Este mensaje lleva la identidad del AS que ha iniciado la iteración (recordemos que esta información va en el *explorador*).
- **Información de identidad**: con este mensaje, el AS que ha recibido la *solicitud de identidad* envía su identidad al AS que inició la iteración. La identidad del AS que inició la iteración va apuntada en los tres mensajes anteriores y por lo tanto llega al AS que recibe la solicitud de identidad.

En la figura 4 podemos ver que estos 4 mensajes son respectivamente 2, 3, 4 y 5 para la primera iteración en la que el AS del dominio A descubre al AS del dominio B. Los mensajes 7, 8, 9 y 10 para la segunda iteración en la que el AS del dominio B descubre al AS del dominio C. Y los mensajes 12, 13, 14 y 15 para la tercera iteración en la que el AS del dominio C descubre al AS del dominio D.

Si nos fijamos en la secuencia que sigue todo el proceso en el ejemplo de la figura 4, el proceso recursivo comienza cuando el AS del dominio A (AS_A) le envía una *solicitud de descubrimiento* (mensaje 2) a la fuente (S), indicándole que inicie un procedimiento de descubrimiento del siguiente dominio (que será el dominio B).

La fuente envía un paquete *explorador* (mensaje 3), hacia el destino. El paquete *explorador* lleva la identidad de AS_A. Cuando el explorador pasa por los nodos programables del dominio A, estos detectan que pertenece a su dominio local (el AS referenciado en el *explorador* es el de su propio dominio) y lo reenvían hacia el destino. Cuando el *explorador* pasa por el primer nodo programable del siguiente dominio (domino B), este nodo lo identifica (el AS del explorador no es el de su dominio) y envía a su AS (AS_B) una *solicitud de identidad* (mensaje 4). Cuando AS_B recibe la *solicitud de identidad*, envía un mensaje de *información de identidad* (mensaje 5) a AS_A indicándole su identidad. De esta forma AS_A conoce la identidad de AS_B que representa el siguiente dominio en el camino hacia el destino. Así AS_A podrá contactar con AS_B para negociar la sesión.

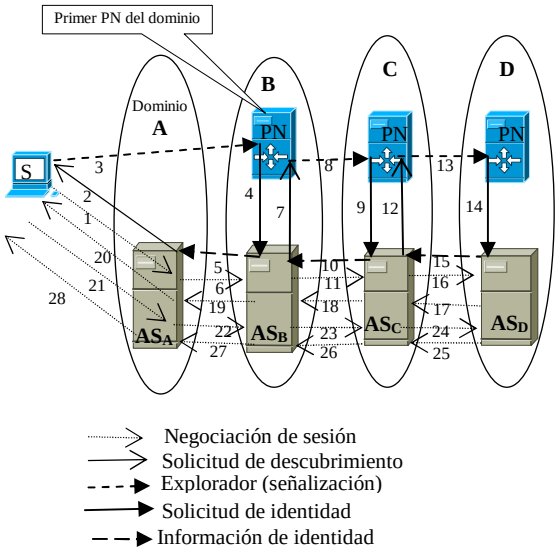

Fig. 4. Proceso de negociación de sesión en un escenario multidominio.

Este proceso se repite en la siguiente iteración entre el segundo dominio y el siguiente (AS_B y AS_C). Pero ahora es AS_B quien origina el proceso y utiliza el nodo programable que le envió la *solicitud de identidad* en la iteración anterior, para enviarle la nueva *solicitud de descubrimiento* (mensaje 7). Cuando este nodo recibe la solicitud genera un nuevo paquete *explorador* con la identidad de AS_B y lo envía (mensaje 8) hacia el destino. El primer nodo programable del dominio D detecta que el explorador es de un dominio distinto al suyo y le envía un mensaje de *solicitud de identidad* (mensaje 9) a su AS (AS_C) indicándole la identidad del AS que ha iniciado la iteración (AS_B). Entonces AS_C termina la iteración enviando un mensaje de *información de identidad* (mensaje 10) a AS_B. Ahora AS_B sabe que tendrá que negociar con AS_C.

El mismo proceso se repite en la tercera iteración entre AS_C y AS_D con los mensajes 12, 13, 14 y 15.

Terminada esta iteración AS_C sabe que tiene que negociar con AS_D. Además AS_D detecta que el destino pertenece a su dominio y termina el proceso no lanzando más paquetes de *solicitud de descubrimiento*.

Mecanismos de seguridad usados en el subproceso de descubrimiento de dominios en un camino.

Los cuatro mensajes de este subproceso necesitan servicios de autenticación y protección contra repetición.

Para implementar los servicios de autenticación entre ASs que pertenecen a distintos dominios administrativos y no tiene ninguna relación previa entre si, utilizaremos mecanismos de firma digital que utilizan sistemas de clave pública. Para ello supondremos que los ASs y PNs tiene una pareja de claves privada/pública y existe una infraestructura PKI (Public Key Infrastruture) que permite que cada componente que pueda conocer la clave pública del otro de forma fiable. Este mecanismo se implementará en los mensajes entre ASs o entre PNs, que como hemos comentado anteriormente, pertenecen a dominios administrativos distintos. Estos mensajes son respectivamente *paquete explorador* y *solicitud de identidad*.

Los servicios de autenticación en los mensajes entre PNs y sus AS, que pertenecen a un mismo dominio administrativo, se pueden realizar con mecanismos basados en hmac que ofrece una mayor eficiencia que la firma digital. Como sabemos, estos mecanismos necesitan del uso de clave simétrica y podemos utilizar la facilidad que proporciona ROSA para que los PNs puedan compartir una clave simétrica con su AS, según vimos en la sección 2.2. En este caso, las aplicaciones de red que manejan los paquetes exploradores tiene asociada una clave Kce, que se distribuye de la misma forma que las Kcis y permite a los AS y PNs de un mismo dominio poder generar una misma clave (llamémosle Ke) para una solicitud concreta de servicio. Ke se genera utilizando la siguiente expresión:

$$Ke = KDF(Kce, IDs).$$

Ke es utilizada por PNs y AS de un mismo dominio para generar y verificar el autenticador utilizando hmac. Los mensajes que se protegen con hmac son *Solicitud de descubrimiento* y *Solicitud de identidad*.

Una vez que hemos resuelto el problema de la autenticación de los mensajes, hay que resolver el problema de la protección contra repetición de estos mensajes que en este caso no es trivial. Este problema habitualmente se resuelve añadiendo información de unicidad (que garantiza que el mensaje es único) al mensaje antes de protegerlo con mecanismos de autenticación (firma o hmac). Vamos a utilizar como información de unicidad en los mensajes el valor IDs (identificador único de la solicitud del servicio que generó la fuente al solicitar el servicio). Por lo tanto los ASs y PNs deben comprobar que no reciben dos mensajes del mismo tipo con el mismo valor de IDs. Esto les obliga a mantener información de estado sobre las solicitudes de sesión que tienen pendientes, el estado en el que están y su identificador IDs.

Podemos encontrar dos situaciones distintas que nos obligan a utilizar mecanismos distintos:

- **Mensajes no esperados**: mensajes que llegan a una entidad por primera vez sin que esta los estuviese esperando. En este caso están el paquete *explorador* que llega a un PN y el mensaje de *Solicitud de identidad* que llega a un AS.

- **Mensajes esperados**: mensajes que llegan a una entidad como respuesta a un mensaje que había enviado anteriormente, y por lo tanto son esperados. En este caso están el mensaje *Solicitud de descubrimiento* que llega a un PN como respuesta una *Solicitud de identidad* (si no es el último dominio), y el mensaje de *Información de identidad* que llega a un AS como respuesta a una solicitud de descubrimiento.

En el caso de los **mensajes esperados** la entidad que espera el mensaje se comporta como una máquina de estado que espera un mensaje con un formato concreto y un valor IDs concreto. Cuando lo recibe ya no lo espera pasando a otro estado, por lo tanto si vuelve a llegar el mismo mensaje un tiempo más tarde lo descarta.

En el caso de los **mensajes no esperados** la solución es un poco más compleja. Estos mensajes, además de llevar el valor de unicidad IDs, llevan una marca de tiempo (figura 5). Cuando los mensajes llegan a su receptor, este comprueba que no había llegado un mensaje de este tipo y con un valor de IDs igual, desde hace un tiempo determinado dentro de una ventana temporal (por ejemplo en los últimos 60 segundos). Para que esto funcione el receptor del mensaje necesita guardar en memoria los valores de IDs recibidos dentro de la ventana de tiempo. Si la marca de tiempo del mensaje indica que el mensaje se genero en un tiempo anterior al final de la ventana de tiempo, el mensaje se descarta igualmente por considerarse que es antiguo.

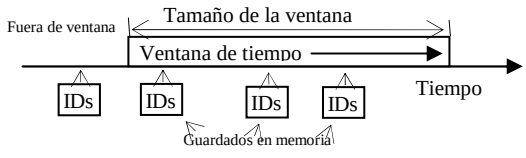

Fig. 5. Ventana de tiempo.

En la figura 6 se puede ver un resumen de los mensajes y los mecanismos de protección usados en cada uno de ellos.

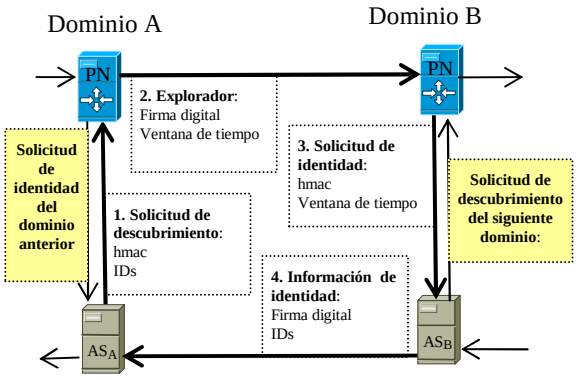

Fig. 6. Resumen de mensajes y mecanismos de protección.

Subproceso de negociación entre dominios.

Una vez que un AS conoce al siguiente en la cadena, comienza el subproceso de negociación de sesión entre dominios. Se producen negociaciones entre las siguientes entidades de la figura 4: S con AS_A, AS_A con AS_B, AS_B con AS_C, AS_C y AS_D. Este subproceso, al igual que el anterior, es recursivo. De esta forma la negociación entre dos entidades está condicionada a los resultados de las negociaciones que hay por delante. Es decir si la negociación entre AS_C y AS_D no terminan con éxito, las negociaciones entre AS_B y AS_C tampoco lo harán.

El objetivo de la negociación es que cada dominio autorice a procesar los paquetes de señalización que pasen por el y le de al otro dominio información de autorización (K) para poder comprobar la información que viene del otro dominio.

La negociación consta de cuatro mensajes que se intercambian entre S y su AS o entre dos ASs contiguos. Estos mensajes son:

- **Solicitud de servicio** (mensajes 1, 6, 11 y 16): lleva los parámetros que definen el servicio como son identificador de servicio solicitado (Ui, Ts, L, Ci).
- **Respuesta a la solicitud** (mensajes 17, 18, 19 y 20)**:** estos mensajes llevan información sobre el coste del servicio solicitado, una vez que el dominio ha verificado si su cliente está autorizado a solicitar el servicio, en los términos indicados en la solicitud. El coste de cada dominio se va acumulando en los mensajes, de esta forma llega a S el coste total en el mensaje 20.
- **Confirmación** (mensajes 21, 22, 23 y 24)**:** S responde diciendo si acepta el servicio con los costes recibidos en el mensaje 20. Esta contestación se propaga por los otros dominios en los mensajes 22, 23 y 24. Cada dominio además utiliza este mensaje para informar de la clave que usará el dominio para esa sesión. Así el dominio A le da al dominio B su clave Ka, el dominio B le da al C su clave Kb y el C le da al D su clave Kc. Estas claves se utilizarán en la fase de señalización como veremos en la sección 4.2. Si un dominio no dispone de la *aplicación de red* para ofrecer el servicio solicitado, se lo comunicó al AS vecino en el mensaje de *respuesta de confirmación*, y en estos mensajes de *confirmación* se distribuye la aplicación correspondiente desde el dominio A al resto de dominios que la hayan solicitado.
- **Clave de sesión** (mensajes 25, 26, 27 y 28): este mensaje es utilizado por los ASs de destino de cada negociación para enviar a los ASs de origen su clave de sesión. De esta forma el AS del dominio D envía Kd al AS del dominio C; el AS del dominio C envía Kc al AS del dominio B; y el AS del dominio B envía Kb al AS del dominio A.

Una vez terminado el proceso cada AS tendrá la clave de sesión (K) de sus ASs vecinos y esto le permitirá poder comprobar los paquetes de señalización que llegan desde estos dominios.

Mecanismos de seguridad usados en el subproceso de negociación entre dominios.

Las entidades (fuente y ASs) que intercambian los mensajes de negociación entre si necesitan tener garantías de que están hablando con la entidad correcta y que los mensajes no son repeticiones de mensajes que corresponden a otras negociaciones anteriores. Además la clave de sesión que envía AS_A a la fuente y las claves de sesión que se intercambian los ASs, deben enviarse de forma confidencial.

Para garantizar la autenticidad de los mensajes se utiliza la firma digital, acompañada de información que se introduce en los mensajes antes de firmarlos y permite identificarlo de forma única. La firma digital además confiere un valor contractual al ofrecer el servicio de no repudio. Todos los mensajes de este subproceso son del tipo de **mensajes esperados**, dado que cuando un AS recibe la *solicitud de identidad* después de contestar, espera la *solicitud de servicio* y cuando contesta a dicha solicitud espera la *confirmación*. Por lo tanto basta con utilizar IDs como información de unicidad en todos los mensajes según vimos en el subproceso anterior y esperar en cada momento recibir el mensaje con el formato correcto y auténtico, de la fuente esperada y con el valor apropiado de IDs.

Las claves de sesión se intercambian entre los ASs de forma confidencial utilizando cifrado asimétrico en un sentido y ocultación con XOR en sentido contrario. Así por ejemplo AS_A le envía a AS_B su clave de sesión Ka cifrada con la clave pública de B. Una vez que B descifra Ka le envía a B su clave de

sesión Kb ocultada de la siguiente forma: Kab= Kb XOR Ka. Cuando A recibe Kab obtiene la clave del dominio B de la siguiente forma: Ka = Kab XOR Kb.

4.2. Envió de señalización por el camino multidominio

Una vez alcanzado un acuerdo entre todos los dominios, puede comenzar el proceso de envío de paquetes de señalización. La fuente comienza a enviar paquetes (mensaje 1, figura 7) protegidos con hmac usando la clave de la sesión (Ka) para su dominio (dominio A). Los routers de este dominio lo procesarán según se especifica en ROSA. Cuando el paquete entra en el siguiente dominio (domino B) y es capturado por el primer PN, este nodo genera la clave de sesión de su dominio (Kb) por lo que la comprobación de autenticación (hmac) del paquete falla. Entonces el nodo descarta el paquete, pero encapsula una copia de este y lo envía a su AS (mensaje 2). El AS comprueba si el paquete corresponde a algún acuerdo de sesión alcanzado y si es así devuelve al nodo el paquete (mensaje 3) junto con la clave del dominio A (Ka). El nodo programable reenvía el paquete hacia el destino protegido ahora con la nueva clave Kb y guarda Ka para usarla en la comprobación del resto de paquetes de esta sesión que recibirá desde la fuente. Este mismo proceso se repite para el resto de los dominios. De esta forma el primer nodo de cada dominio hará funciones de ***pasarela de seguridad*** entre dominios.

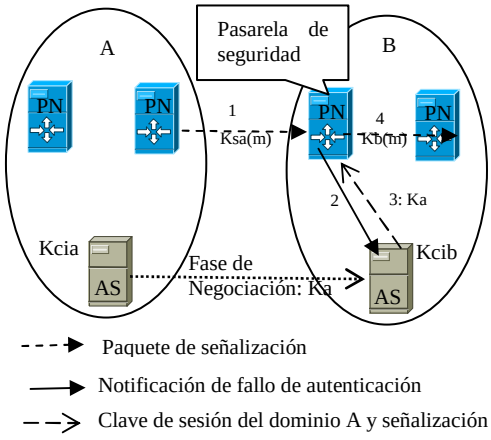

Fig. 7. Procesamiento de los paquetes de señalización al cambiar de dominio.

5. Otros trabajos relacionados

En el entorno de las redes activas, se ha especificado la arquitectura de seguridad denominada ANSA [4] que presenta un enfoque del problema con una solución flexible sin dar soluciones concretas. En esta arquitectura se han basado algunas de las propuestas [6] concretas de seguridad para la red ABone (red experimental basada en tecnología de redes activas). Una de estas propuestas de seguridad es SANTS [7,8], que ofrece seguridad a ANTS, uno de los principales entornos de ejecución usados en Abone.

En estas propuestas, la seguridad está basada básicamente en mecanismos de autorización a las aplicaciones a realizar acciones en los nodos programables, cuando procesan los paquetes de señalización. Se propone un sistema de autorización basado en credenciales que son transportadas por los propios paquetes. Las políticas de seguridad que puede haber en cada dominio pueden ser totalmente distintas. Debido a esto las credenciales pueden ser distintas para cada dominio. Esto obliga a transportar cadenas de credenciales, una para cada dominio. Debido a que las credenciales pueden contener una gran cantidad de información, una opción es transportar referencias a credenciales que serán distribuidas por medio de una infraestructura de distribución segura de credenciales (DNSSEC, Kerberos, LDAP seguro, PKI, etc). Las soluciones planteadas en este entorno se basan en que los nodos conocen a sus vecinos y ya existen relaciones de confianza entre ellos.

La solución multidominio planteada en ROSA se basa en un sistema multidominio homogéneo en el que una única credencial siempre lleva los mismos parámetros (Ui, L, Ts, Ci) y estos pueden ser utilizados por los distintos dominios para comprobación de autorización. Unos dominios pueden comprobar unos parámetros y otros no, pero la credencial siempre es la misma. Esto permite tener un sistema más escalable y con unos atributos suficientes para comprobar si se tiene autorización a ejecutar el código, en un periodo de tiempo, generado por un usuario Ui y que procesa un flujo de paquetes desde una fuente a un destino determinado. Esto es aplicable a una red programable en donde los códigos que se ejecutan son códigos controlados y previamente auditados. En este caso no se habilita mecanismos de control de acceso fino basados en acciones que dejan hacer los códigos. El código en conjunto está autorizado o no lo esta. Por lo tanto podemos decir que la solución planteada en ROSA no permite hacer un control fino de autorización pero se gana en sencillez que a su vez se traduce en una mejora de escalabilidad. Por otro lado los paquetes se protegen con clave simétrica incluso en el entorno multidominio dejando la clave asimétrica reservada al proceso de solicitud de servicio y negociación entre dominios. Esto mejora también la escalabilidad del sistema con respecto otras soluciones que utilizan clave asimétrica y simétrica.

En el proyecto FAIN se propone una arquitectura abierta y flexible de nodo programable que soporte distintos entornos de ejecución y entornos virtuales. En [9] se plantea una arquitectura de seguridad para

FAIN[2] basada en las ideas de las propuestas en ANSA (protección de parte dinámica del paquete con MAC y la estática con firma digital, uso de PKI, credenciales para autorización). Además aporta una idea interesante que consiste en el uso de auditorias de seguridad. El uso de clave simétrica se basa en relaciones de confianza entre nodos vecinos que comparten la clave para generar y verificar el MAC de los paquetes. Esto va en contra de la idea de transparencia de camino que propone SARA. En los entornos multidominio no tiene muy desarrollada la solución y hace referencia al uso de credenciales al igual que ANSA.

En general ninguna de las soluciones mencionadas referentes a redes programables trata en profundidad el tema de seguridad en un entrono multidominio. Otras aplicaciones en las que sí se utilizan negociación entre dominios, aportan ideas que se podrían trasladar a nuestro escenario. Cabe destacar las soluciones de negociación basadas en *Clearing House* (CH) que se utilizan en [10]. En este caso el objetivo es negociar la reserva de recursos a lo largo de un camino para ofrecer QoS. Si el camino pasa por varios dominios la solución CH proporciona la negociación entre dominios en cuanto reserva de recursos y tarificación. En un escenario sencillo (un máximo de cuatro dominios) como el nuestro, no está justificado utilizar un sistema basado en CH ya que no nos beneficiamos de la escalabilidad que proporciona esta arquitectura. Evitamos además tener que delegar la negociación en una entidad externa que nos da el servicio a cambio de un coste adicional. Un último argumento a razón de no usar CH, es que la propia tecnología de SARA nos permite de forma sencilla averiguar los dominios por los que pasará un flujo de paquetes y por lo tanto nos permite negociar con ellos.

6. Conclusiones

Hemos comprobado que un escenario multidominio basado en redes programables puede estar formado como máximo por cuatro dominios. La solución planteada de acuerdos entre dominios está basada en clave asimétrica lo cual permite la escalabilidad en el escenario multidominio. Al mismo tiempo, la solución de seguridad respeta el procedimiento monodominio basado en clave simétrica, con lo que se siguen manteniendo las buenas prestaciones en el procesamiento de seguridad del flujo de señalización.

Referencias

[1] Andrew T. Campbell, Herman G. De Meer, Michael E. Kounavis, Kazuho Miki, John B. Vincente, and Daniel Villela. *A survey of programmable networks.* Computer Communication Review, 29(2):7-23, April 1999.

[2] Proyecto MCYT AURAS. http://matrix.it.uc3m.es/~auras/.

[3] D. Larrabeiti, M. Calderón, A. Azcorra and M. Urueña. A practical approach to Network-based processing. IEEE 4th International Workshop on Active Middleware Services. IEEE Computer Society, pp. 3-10, ISBN: 0-7695-1721-8. Edinburgh, Scotland. July 2002.

[4] AN Security Working Group, Security Architecture for Active Nets, 2001.

[5] Marcelo Bagnulo, Bernardo Alarcos, María Calderón, Marifeli Sedano. "ROSA: Realistic Open Security Architecture for Active Networks". IWAN 2002, LNCS 2546, pp. 204-215. Zurich, Switzerland, December, 4-6 2002. ISBN: 3-540-00223-5.

[6] Faber, T., Braden, B., Lindell, B., Berson, S., Bhaskar, K.: Active Network Security for the ABone. November 30, 2001.

[7] Murphy, S., Lewis, E., Puga, R., Watson, R., Yee, R.: Strong Security for Active Networks. Proceedings IEEE OPENARCH01. April, 27 2001.

[8] NAI Labs ANETS Group, SANTS Security Overview (May 18, 2000), URL:ftp://ftp.tislabs.com/pub/activenets/SANTSsecurityoverview.doc.

[9] Arso Savanovic, Dušan Gabrijelcic, Borka Jerman Blažic and Stamatis Karnouskos An active networks security architecture Informatica International Journal of Computing and Informatics, Volume 26, Number 2, pp 211-221, 2002 (ISSN 0350-5596)

[10] Chen-Nee Chuah, Lakshminarayanan Subramanian, Randy H. Katz and Anthony D. Joseph, "QoS Provisioning Using A Clearing House Architecture," *International Workshop on Quality of Service (IWQoS)*, Pittsburgh, PA, pp. 115-124, June 5-7, 2000.

[2] Más información: http://www.ist-fain.org/