Implementación de IPsec en una arquitectura *TCP splitting*

Juan Caubet, Jose L. Muñoz, Juanjo Alins, Jorge Mata-Díaz, Oscar Esparza
Universitat Politècnica de Catalunya (UPC)

***Resumen*—El rendimiento de las aplicaciones que utilizan el protocolo de transporte TCP (*Transmission Control Protocol*) sobre enlaces vía satélite tiene una degradación significativa. Esto se debe principalmente a que el algoritmo de control de congestión estándar de TCP no es adecuado para superar las deficiencias de las redes satelitales. *TCP splitting* es una solución prometedora para mejorar el rendimiento general de TCP, incluso en el segmento satelital. La división de la conexión TCP se logra mediante la instalación de dos PEPs (*Performance Enhancement Proxies*) en los extremos del segmento satelital. Sin embargo, la división de TCP entra en conflicto con IPsec. Si el cifrado y/o la autenticación son aplicados sobre los datagramas IP, el PEP no puede manipular las correspondientes cabeceras IP y TCP para dividir las conexiones TCP. En este trabajo presentamos tres propuestas para implementar IPsec en un escenario *TCP splitting*, proporcionando los servicios de seguridad habituales y un buen rendimiento en la conexión vía satélite. La idea básica es permitir a los PEPs manipular las cabeceras IP y TCP en función del nivel de confianza que los usuarios tengan en ellos.**

I. INTRODUCCIÓN

Las redes de banda ancha vía satélite están ganando importancia debido a su alta disponibilidad de ancho de banda y gran cobertura. Estas redes satelitales jugarán un papel crucial en el futuro de Internet debido a la necesidad de servicios de comunicación en cualquier momento y en cualquier lugar. Sin embargo, se ha demostrado que el protocolo TCP (*Transmission Control Protocol*) tiene una degradación significativa sobre enlaces satelitales. Esto se debe principalmente al hecho de que las redes con enlaces satélite presentan grandes retardos de propagación, introducen una alta probabilidad de error de transmisión y disponen de un notable nivel de asimetría entre los anchos de banda de los canales de difusión y de retorno.

La degradación de TCP se debe principalmente a que su algoritmo de control de congestión no es adecuado para superar las deficiencias de los enlaces satelitales [1], [2]. TCP aumenta su ventana de congestión hasta que se produce una pérdida. Entonces, cuando ésta es detectada, el número de paquetes dentro del sistema se reduce a la mitad. En las redes terrestres, las pérdidas de paquetes son causadas principalmente por la congestión en las colas de espera de los dispositivos de red. No obstante, las pérdidas de paquetes también pueden ser causadas por errores de transmisión. Este efecto es especialmente notable en las redes inalámbricas, provocando una reducción innecesaria de la carga del sistema y, por lo tanto, del rendimiento del protocolo TCP. Esta degradación se ve acentuada cuando la red inalámbrica además dispone de un elevado RTT (*Round Trip Time*). Así, en redes con satélites geoestacionarios, las conexiones TCP pueden tardar varias decenas de segundos en restituir la carga de paquetes tras una pérdida de paquetes debido a errores de transmisión.

Se han propuesto algunas soluciones para superar los problemas de TCP sobre enlaces satelitales [3]. Por un lado, algunas extensiones del TCP estándar han sido propuestas especialmente para las redes vía satélite. Y por otra parte, algunas variantes de TCP han sido especialmente diseñadas para optimizar su rendimiento en entornos satelitales, como *TCP Peach*, *TCP Westwood* o *TCP Hybla*.

Sin embargo, la eficacia de estos dos tipos de soluciones está limitada por el hecho de que no están universalmente adoptadas por todos los sistemas finales en Internet.

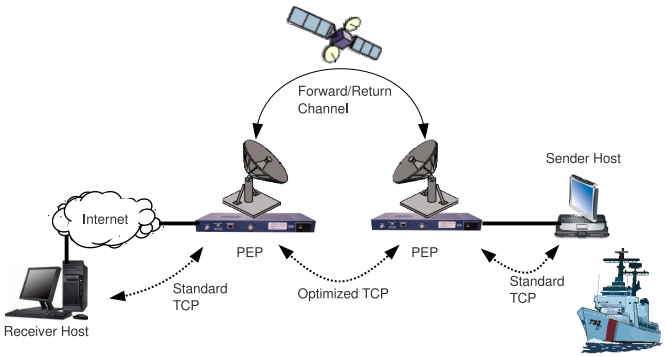

Figura 1. *TCP splitting*.

Los PEPs (*Performance Enhancement Proxies*) pueden ser introducidos para utilizar esos protocolos optimizados, o extensiones, sobre enlaces vía satélite, sin variar los TCPs de los segmentos cableados [4]. El objetivo es dividir la ruta completa en segmentos cableados y un segmento satelital. En general, la comunicación se divide en tres partes o conexiones: emisor-PEP, PEP-PEP y PEP-receptor. Este mecanismo también se conoce como *TCP splitting*. La figura 1 muestra un escenario típico de esta técnica, en la que un barco (que actúa como servidor) proporciona contenidos a un host situado en Internet.

El objetivo de la división es aislar el enlace satelital de gran latencia mediante la introducción de agentes intermedios (PEPs), que dividen la conexión TCP. Los PEPs son responsables de la recepción, el almacenamiento, y el reconocimiento de los datos generados por un emisor, y además, del reenvío de éstos hacia el receptor. *TCP splitting* permite implementar un control de congestión optimizado para mejorar el rendimiento

de TCP en la conexión vía satélite, y dejar el protocolo TCP estándar en los segmentos cableados. Esta división es transparente tanto para el origen como para el destino de la comunicación.

Las redes satelitales también son propensas a ataques de seguridad, debido especialmente a su naturaleza *broadcast*. Por esta razón, es necesario proteger estas comunicaciones. IPsec es una solución estándar que proporciona los servicios de seguridad necesarios para prevenir la mayoría de estos ataques. A diferencia de otras soluciones extremo a extremo que operan en la capa de transporte o en la capa de aplicación, IPsec opera en la capa de red. IPsec se utiliza principalmente para crear VPNs (*Virtual Private Networks*), aunque también se puede utilizar para proteger las comunicaciones entre dos máquinas remotas. En nuestro escenario, el problema es que el uso de IPsec afecta negativamente al funcionamiento de los PEPs, ya que ellos necesitan manipular las cabeceras de los protocolos TCP/IP, que están criptográficamente protegidas. Si se utiliza IPsec, los PEPs no pueden dividir las conexiones TCP y, en consecuencia, el rendimiento de TCP en la redes satelitales no puede ser mejorado.

En este trabajo presentamos tres propuestas diferentes para permitir que IPsec y *TCP splitting* puedan trabajar conjuntamente. El objetivo es proporcionar a las comunicaciones los niveles de seguridad adecuados, sin afectar demasiado las operaciones realizadas por los PEPs. En la primera propuesta, que se explica en la sección III-A, consideramos un escenario particular en el que los usuarios finales (emisor y receptor) no confían en el operador satelital (tal vez porque desconocen su existencia), por lo que quieren que todos los datos intercambiados sean protegidos criptográficamente por IPsec. Así pues, no se confía en absoluto en los PEPs, y por ello no pueden ni leer ni manipular los paquetes TCP/IP, y obviamente, no pueden dividir las conexiones TCP. Sin embargo, estos paquetes son encapsulados por un protocolo TCP optimizado con el fin de mejorar el rendimiento en la conexión satelital. En las otras dos propuestas que se explican en las secciones III-B y III-C, respectivamente, consideramos que los usuarios finales conocen al operador satelital y confían en él. Esto permitirá que los PEPs puedan leer/modificar todo el contenido (*Fully-trusted PEPs*), o ciertas partes (*Partial-trusted PEPs*), de los paquetes TCP/IP, dependiendo del grado de confianza que los usuarios puedan tener en el operador de red y la mejora de rendimiento que quieran conseguir.

II. TRABAJOS RELACIONADOS

Las propuestas que utilizan IPsec en escenarios satelitales con PEPs se pueden clasificar en dos grupos, los que modifican el comportamiento estándar de IPsec y los que sólo se adaptan a él.

En cuanto al primer grupo de propuestas, una de las soluciones más interesantes es ML-IPsec [5], que se basa en dividir los paquetes en diferentes zonas donde se aplicarán los diferentes servicios de seguridad de forma independiente. El número de zonas y su tamaño son definidos a priori utilizando un mapa de zonas. La propuesta también rediseña las Asociaciones de Seguridad (SAs) para definir el tipo de seguridad (algoritmos criptográficos, claves, etc.) que se va a utilizar en cada zona. Se crea una nueva Asociación de Seguridad Compuesta (CSA), que consta de dos partes: una que contiene información común a todas las zonas, y otra que contiene una lista de SAs reducidas, una por zona. Tanto el mapa de zonas como la CSA son compartidas por todos los dispositivos que tienen acceso a alguna zona de los paquetes IPsec. En [6] se puede encontrar un análisis crítico de ML-IPsec. [7] y [8] son dos propuestas que también dividen los paquetes en zonas: una zona para las cabeceras TCP/IP y otra para los datos de usuario. En [8], las dos zonas están cifradas con dos claves diferentes, mientras que en [7] también se utilizan diferentes algoritmos criptográficos.

En referencia al segundo grupo de propuestas, en [9], los dispositivos IPsec establecen una sesión con el PEP para proporcionarle la información necesaria para generar los ACKs prematuros. El PEP utiliza un Identificador de Conexión (CI) para relacionar cada paquete TCP con la información proporcionada por el remitente. Otras soluciones proponen generar un hash de la información de flujo TCP e incluirlo en el campo de opciones de la cabecera IP [10], [11]. En este caso, los PEPs pueden distinguir diferentes flujos de tráfico TCP sin necesidad de modificar los paquetes, por lo que es posible retransmitir los paquetes perdidos en el enlace satelital (*TCP snooping*). En [12], el mecanismo de recuperación de pérdidas de TCP está explícitamente informado sobre la naturaleza de las mismas. En [13], los temporizadores TCP se pueden congelar obligando al emisor TCP a pasar al modo persistente.

III. SOLUCIONES DE SEGURIDAD IPSEC EN ARQUITECTURAS TCP SPLITTING

Como ya se ha comentado, nuestro objetivo es implementar servicios de seguridad con IPsec en arquitecturas *TCP splitting*. El problema es que en estas arquitecturas los PEPs necesitan tener acceso a la información transportada por los paquetes en las cabeceras TCP/IP. Como veremos más adelante, existe un compromiso entre la mejora del rendimiento de la comunicación y la aplicación de la seguridad. También queremos mencionar que no hemos considerado las soluciones que se basan en no aplicar los servicios de seguridad a ciertas partes del paquete o en copiar algunos datos en áreas no protegidas por IPsec. No consideramos este tipo de soluciones porque crean críticas vulnerabilidades de la seguridad. En este caso, las comunicaciones seguras tienen lugar entre entidades de confianza, mientras que los dispositivos intermedios que no son de confianza no tendrán acceso a los datos transportados por los paquetes IPsec.

En este trabajo, proponemos tres maneras de lograr seguridad. La primera propuesta considera que los PEPs no son entidades de confianza. Por otra parte, la segunda y la tercera consideran que los nodos finales tienen cierto grado de confianza en los PEPs. Para estos dos casos, hemos desarrollado un sencillo protocolo de intercambio seguro para que los dispositivos IPsec compartan las Asociaciones de Seguridad (SAs) con los PEPs.

III-A. *Untrusted PEPs*

En esta propuesta los PEPs no se consideran entidades de confianza. Esta consideración hace que no puedan manipular las cabeceras TCP/IP. Esto evitará que los PEPs puedan aplicar los mecanismos correspondientes para mejorar el rendimiento del enlace satelital, como el envío de confirmaciones prematuras a los usuarios finales (*TCP spoofing*). Por esta razón, la única manera de mejorar el rendimiento sin revelar ningún dato sobre el paquete, es encapsularlos en un protocolo TCP optimizado. Con esta solución, los PEPs, al menos, pueden controlar las retransmisiones necesarias sobre el enlace satelital. Más concretamente, este control de retransmisión lo llevan a cabo los PEPs mediante el intercambio de reconocimientos, teniendo la propiedad de gestionar las retransmisiones de paquetes debidas a pérdidas en el enlace satelital de forma transparente a los usuarios finales. Por lo tanto, se evitan los efectos de una reducción innecesaria de la carga en el sistema. Observe que esta propuesta se comportará como un mecanismo típico de *snooping*, pero manteniendo los paquetes IPsec sin cambios.

El hecho de que el PEP deba establecer una conexión TCP optimizada antes de poder enviar información, hace que esta solución requiera del establecimiento de dos conexiones: una conexión TCP estándar entre los usuarios finales y otra conexión TCP optimizada entre los PEPs. El inconveniente es que para establecer conexiones, TCP utiliza un *Three-Way Handshake* que introduce cierto retardo, y por consiguiente, una reducción del tiempo de respuesta de la aplicación de usuario. En la figura 2 se muestra el intercambio de paquetes requeridos por esta propuesta.

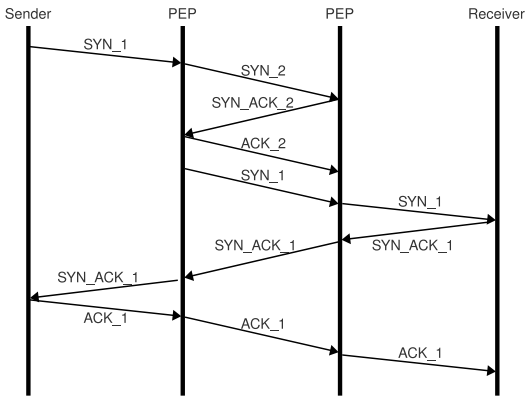

Figura 2. *Three-Way Handshake* en la propuesta *Untrusted PEPs*.

Los paquetes marcados con el "1" son los que se utilizan para establecer una conexión TCP estándar entre los usuarios finales. Por otra parte, los paquetes marcados con el "2" son los que se usan para establecer la conexión TCP optimizada. Un simple análisis permite observar que la fase de establecimiento de la conexión extremo a extremo se incrementa en un RTT del segmento satelital. Este incremento está dentro del margen de operación del protocolo TCP estándar que admite un tiempo de espera del primer paquete SYN de hasta 3 segundos. Una descripción detallada de la operación de los PEPs en esta propuesta sería la siguiente:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red y los pasa a la capa de transporte.
2. Una vez en la capa de transporte, el PEP añade la cabecera del protocolo TCP optimizado y, a continuación, pasa los paquetes a la capa de red.
3. La capa de red añade la cabecera IP y transmite los paquetes a través del enlace satelital.
4. El otro PEP recibe los paquetes a través del enlace satelital, y una vez en la capa de red, les extrae la cabecera IP y los pasa a la capa de transporte.
5. En la capa de transporte, el PEP genera un paquete ACK para cada paquete recibido.
6. A continuación, el PEP elimina la cabecera del protocolo TCP optimizado y devuelve el paquete a la capa de red.
7. Por último, la capa de red añade la cabecera IP y envía el paquete a su destino final.

Hay que tener en cuenta que los pasos anteriores tienen que ser realizados para cada uno de los dos sentidos de la comunicación, y tanto para paquetes de datos como para paquetes ACK.

Por otra parte, también notar que los paquetes permanecen protegidos desde el inicio de la comunicación hasta el final mediante IPsec. Como resultado, en caso de utilizar el protocolo de seguridad *Authentication Header* (AH) se proporciona integridad de los datos y autenticación, y en caso de utilizar el *Encapsulating Security Payload* (ESP) la confidencialidad está asegurada. Por lo tanto, el nivel de seguridad de esta propuesta es equivalente al proporcionado por el IPsec estándar.

Los dos principales inconvenientes de esta propuesta son:

- El overhead introducido en los paquetes debido al hecho de añadir una nueva cabecera TCP en el enlace satelital.
- Un aumento del retardo debido a la creación y al cierre de las conexiones TCP adicionales entre PEPs.

La propuesta presentada en esta sección puede ser conveniente para escenarios en los que los dispositivos IPsec finales no saben de antemano que sus comunicaciones van a pasar a través de un enlace vía satélite. En tal escenario, los usuarios pueden utilizar IPsec normalmente sin tener que tomar ninguna decisión adicional. Un ejemplo de este tipo de escenario puede ser un *host* ubicado en una red de área local (LAN) que no es consciente de que se está conectando a Internet a través de un enlace satelital.

III-B. *Fully-trusted PEPs*

En esta propuesta se supone que uno de los usuarios finales que utiliza IPsec puede establecer una relación de confianza con uno de los PEPs. Este puede ser el caso de un usuario cuyo ISP (*Internet Service Provider*) es un operador satelital. Aquí suponemos que el usuario puede confiar completamente en los PEPs. Con este tipo de propuesta es posible poner a su disposición la información contenida en las cabeceras. En este caso, el PEP podrá crear/modificar paquetes IPsec

válidos, posibilidad que puede aprovecharse para mejorar el rendimiento de la conexión vía satélite.

Entrando más en detalle, nuestra propuesta funciona de la siguiente forma. En un principio, los usuarios origen y destino establecen una conexión IPsec extremo a extremo mientras los PEP se encuentra en modo pasivo (no están involucrados en esta fase de la comunicación). Una vez que la comunicación IPsec está establecida, y por tanto las SAs se han negociado, uno de los usuarios finales (el que tiene una relación de confianza con los PEP) se las envía a los PEPs[1]. Después de eso, ya se pueden empezar a transferir datos.

En la fase de transferencia de datos de la comunicación, los PEPs operan de la siguiente forma:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red.
2. Extrae la protección criptográfica de los paquetes antes de enviar los datos de usuario a la capa de transporte.
3. Una vez en la capa de transporte, envía un TCP ACK al usuario origen por cada paquete TCP recibido. Ya que el PEP conoce las SAs, puede crear TCP ACKs válidos para todos los paquetes recibidos (cifrados y/o autenticados). De hecho, el PEP suplanta al destino final.
4. El PEP genera nuevos segmentos usando los datos de usuario extraídos de los paquetes recibidos y los pasa a la capa de red. Sin embargo, estos segmentos son de un TCP optimizado para la conexión vía satélite.
5. En la capa de red les añade la protección criptográfica correspondiente y la cabecera IP. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
6. El PEP transmite estos paquetes IP a través del enlace satelital.
7. El otro PEP recibe los paquetes, les extrae la cabecera IP y la protección criptográfica, y obtiene los datos de usuario para enviarlos a la capa de transporte.
8. Este segundo PEP le envía un TCP ACK al primero por cada paquete recibido.
9. En la capa de transporte, el PEP genera segmentos TCP estándar con los datos de usuario. Estos segmentos se transmiten a la capa de red.
10. La capa de red les añade las protección criptográfica correspondiente y la cabecera IP. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
11. Por último, el PEP transmite los paquetes sobre el enlace terrestre.

La solución propuesta es equivalente al típico *TCP splitting* excepto por el hecho de que tenemos una protección adicional en los paquetes IPsec de los usuarios. Tenga en cuenta que se establecen tres conexiones diferentes: emisor-PEP, PEP-PEP y PEP-receptor. En cada una de estas conexiones, se puede utilizar el TCP más adecuado. Por ejemplo, un TCP estándar para las conexiones terrestres y un TCP optimizado para la conexión vía satélite.

La figura 3 muestra el *Three-Way Handshake* que se realiza para establecer estas conexiones. Como se puede observar, el tiempo requerido para establecer la conexión extremo a extremo en esta propuesta es prácticamente el mínimo posible sobre que queda determinado por el retardo de propagación del enlace satelital.

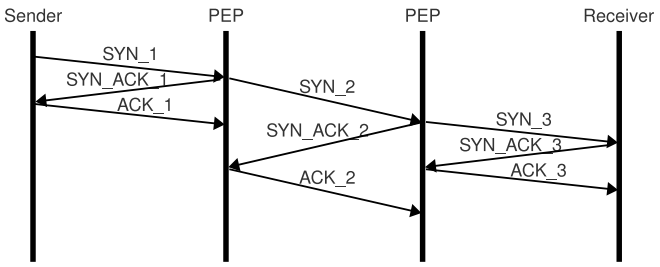

Figura 3. *Three-Way Handshake* en la propuesta *Fully-trusted PEPs*.

En resumen, esta propuesta mejora el rendimiento extremo a extremo porque, por un lado, el PEP puede enviar ACKs prematuros al emisor TCP (evitando reducciones innecesarias de la ventana de congestión), y por otro lado, el PEP puede utilizar un TCP optimizado para la transmisión sobre el enlace satelital. Además, la solución es completamente transparente para el usuario que no es cliente del operador satelital.

Esta propuesta también tiene algunos inconvenientes. Los más importantes son los siguientes:

- Una vez que los PEPs tienen las correspondientes SAs, pueden acceder a toda la información transmitida en los paquetes IPsec (incluyendo los datos de usuario).
- Los PEPs tienen que manipular la protección criptográfica correspondiente (por ejemplo, descifrar cada paquete recibido y cifrarlo antes de transmitirlo).

Los inconvenientes anteriores implican que los PEPs ahora son considerados como terceras partes de confianza (es decir, son nuevos puntos de vulnerabilidad) y que se produce un aumento de la carga de procesamiento en ellos.

III-C. Partially-trusted PEPs (2L-IPsec)

Se propone una tercera propuesta que ofrece un buen equilibrio entre seguridad y rendimiento. Modifica el protocolo IPsec con el fin de proporcionar una protección criptográfica extremo a extremo de los datos de usuario, pero permitiendo que los PEPs puedan manipular las cabeceras TCP/IP para mejorar el rendimiento de la conexión satelital. Se supone que uno de los usuarios de IPsec tienen una relación de confianza con los PEPs. Al igual que en el caso anterior, este puede ser el caso de un usuario cuyo ISP (*Internet Service Provider*) es un operador satelital. Sin embargo, a diferencia del caso anterior, es suficiente una relación de confianza más débil entre los usuarios y los PEPs, ya que el PEP no manipula los datos de usuario, sólo las cabeceras TCP/IP.

Nuestra propuesta se basa parcialmente en las propuestas ML-IPsec [5], LES [7] y [8], ya que también dividen los paquetes en diferentes zonas. Sin embargo, estas partes corresponden a diferentes capas en lugar de a distintas zonas. La idea es usar capas de cifrado, ya que nos permite divulgar

[1]Observe que estas SAs se han de volver a enviar a los PEPs siempre que se hayan renegociado.

selectivamente diferentes partes de un paquete a los distintos usuarios sin poner en peligro la seguridad de las otras partes. En nuestra propuesta, llamada 2L-IPsec (*two-layer IPsec*), una clave se utiliza para cifrar las cabeceras TCP/IP y otra para cifrar los datos de usuario. Los usuarios finales distribuirán la primera clave para que los PEPs puedan manipular las cabeceras TCP/IP y mantendrán en secreto la segunda, por lo que los PEPs no serán capaces de romper la protección criptográfica de los datos de usuario.

En primer lugar, los usuarios finales de IPsec deben negociar la Asociación de Seguridad (SAs), mediante el protocolo estándar *Internet Key Exchange* (IKEv2) que proporciona IPsec, mientras los PEPs permanecen en modo pasivo. Estas SAs contendrán la clave criptográfica K_D, que se utilizará para proteger los datos de usuario. Una vez que tengan las SAs, tienen que generar una nueva clave de cifrado para proteger las cabeceras TCP/IP, K_H, y enviársela a los PEPs[2].

Entonces, los usuarios finales tienen acceso a los paquetes enteros, y los PEPs sólo pueden manipular las cabeceras TCP/IP y no los datos de usuario. Esto es suficiente para aplicar las técnicas de *splitting*.

En la fase de transferencia de datos de la comunicación, los PEPs operan de la siguiente forma:

1. El primer PEP extrae la cabecera IP de los paquetes que le llegan a la capa de red.
2. Extrae la protección criptográfica de las cabeceras TCP/IP, y envía los datos de usuario (que son protegidos criptográficamente usando K_D) a la capa de transporte.
3. También envía un TCP ACK al usuario origen por cada paquete TCP que recibe. Ya que el PEP conoce K_H, puede crear ACKs criptográficamente protegidos.
4. En la capa de transporte, el PEP genera nuevos segmentos usando los datos de usuario (protegidos), pero utilizando un TCP optimizado para la conexión vía satélite. Estos nuevos segmentos se transmiten a la capa de red.
5. La capa de red les añade la protección criptográfica correspondiente y la cabecera IP (usando K_H). En estos nuevos paquetes IP, la dirección IP de destino no cambia.
6. El PEP transmite los paquetes IP sobre el enlace satélite.
7. El otro PEP recibe los paquetes, extrae la cabecera IP y la protección criptográfica (utilizando K_H), y obtienen los datos de usuario (protegidos) para enviarlos a la capa de transporte.
8. En la capa de transporte, envía un TCP ACK al primer PEP por cada paquete que recibe.
9. Genera segmentos TCP estándar con los datos de usuario (protegidos). Después los transmite a la capa de red.
10. La capa de red les añade la protección criptográfica correspondiente y la cabecera IP, usando K_H. En estos nuevos paquetes IP, la dirección IP de destino no cambia.
11. Por último, el PEP transmite los paquetes sobre el enlace terrestre.

[2]Igual que en el caso anterior, esta operación se debe repetir cada vez que las SAs sean renegociadas.

Observe que en el caso de que un paquete se pierda en cualquiera de las tres partes de la ruta, el paquete sólo se retransmitirá en dicha parte. Por otro lado, la definición de las zonas ha sido específicamente diseñada teniendo en cuenta que para aplicar *TCP splitting* el PEP sólo necesita acceso a las cabeceras TCP/IP. Por otra parte, el tamaño de estas cabeceras es variable, ya que pueden incluir información opcional. Así, hemos definido una zona que contiene las cabeceras TCP/IP y otra zona que cubre el *payload* de TCP, ambas de longitud variable. De esta manera, los PEPs siempre tienen acceso a la zona deseada sin importar el tamaño de las cabeceras, ya que su tamaño se gestiona de forma dinámica.

La figura 4 muestra el formato de las cabeceras IPsec (AH y ESP) en los paquetes de 2L-IPsec (en modo de transporte). La longitud de las cabeceras, el *offset* a la siguiente cabecera y el tipo de protocolo de transporte encapsulado en el datagrama IP se determinan mediante cálculos matemáticos simples.

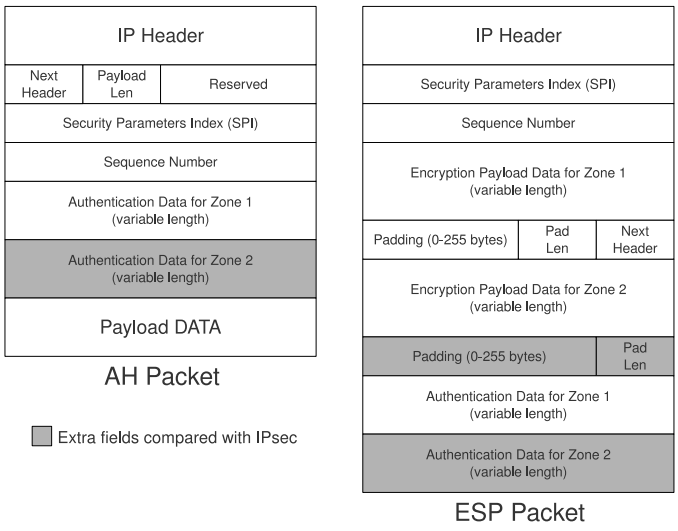

Figura 4. Formato de los paquetes 2L-IPsec (en modo transporte).

Algunas de las ventajas de utilizar 2L-IPsec se resumen a continuación:

- *No compromete la seguridad de los datos de usuario*. Esta propuesta preserva la seguridad de los datos de usuario entre los dispositivos 2L-IPsec. También proporciona confidencialidad, integridad y autenticidad del origen de los datos.
- *Seguridad reforzada de los paquetes*. Ahora se tienen que romper dos claves diferentes para acceder a todos los datos contenidos en el paquete.
- *Preservar la confidencialidad extremo a extremo*. Por ejemplo, si un PEP no necesita modificar ningún dato de los paquetes, entonces las SAs únicamente tendrán que incluir la clave de autenticación, pero no la clave de cifrado, para la zona correspondiente.
- *Son posibles las comunicaciones seguras utilizando TCP splitting*. Este es uno de nuestros principales objetivos. Tenga en cuenta que con esta propuesta podemos implementar un mecanismo que controle qué partes específicas

del paquete pueden ser leídas, modificadas, etc. por terceras partes involucradas en la comunicación. Además, nos permite generar los paquetes ACK prematuros requeridos por *TCP splitting*. Por último, también se pueden retransmitir los paquetes perdidos de manera independiente en cada parte de la ruta.

Como se indica en [7], se ha demostrado que el rendimiento del cifrado de seguridad por capas es comparable al de IPsec.

Nuestra propuesta también tiene algunas desventajas que deben ser consideradas, así:

- *2L-IPsec no es una solución estándar*. La mayoría de los dispositivos conectados a Internet no tendrán este nuevo protocolo. Esto limitará su uso a determinados escenarios, en los que es posible su implantación. Por ejemplo, un escenario real podría ser la comunicación de dos sedes de la misma empresa que utiliza un enlace vía satélite para sus comunicaciones. En este caso, los *routers* de las sedes son los usuarios IPsec finales. Estos dispositivos son los que podrían tener instalado el protocolo 2L-IPsec para mejorar el rendimiento de dichas comunicaciones.
- *Overhead*. Nuestra propuesta aumenta el tamaño de los paquetes, cosa que afecta negativamente al rendimiento. Además, 2L-IPsec requiere más operaciones de cifrado/descifrado, así que son necesarias más CPU y más memoria. Sin embargo, añadir seguridad siempre significa que hay que añadir algo de sobrecarga al sistema.
- *Complejidad*. Los usuarios finales deben descifrar y cifrar los paquetes en dos zonas diferentes utilizando dos claves en vez de una.
- *Distribución de las claves*. Las claves tienen que distribuirse a los PEPs de forma segura.
- *Generación de nuevas claves*. Los usuarios finales tienen que generar claves adicionales. Una solución bastante simple podría ser el uso de funciones hash sobre la clave negociada, H_D, es decir, $K_H = h(K_D)$.

IV. CONCLUSIONES

En este trabajo hemos analizado y propuesto soluciones para utilizar dos mecanismos que al ser aplicados conjuntamente provocan un conflicto. Estos mecanismos son la seguridad de red extremo a extremo proporcionada por IPsec y la optimización del rendimiento de las redes satelitales mediante *TCP splitting*. Por una parte la seguridad extremo a extremo normalmente utiliza criptografía en la capa de red para proteger los datagramas de usuario, y por otra *TCP splitting* requiere que los nodos intermedios puedan realizar operaciones "inteligentes" sobre los paquetes TCP para mejorar el rendimiento. El problema es que algunas partes de los paquetes que son necesarias para lograr las mejoras de rendimiento podrían no ser accesibles debido a la protección aplicada por IPsec.

En general, la situación anterior es un problema difícil de tratar y hay que asumir que no existe una solución adecuada para todos los escenarios. En este sentido, lo mejor que podemos hacer es encontrar un conjunto de soluciones que ofrezcan diferentes compromisos entre seguridad y rendimiento.

En este trabajo, hemos analizado las que consideramos las tres principales propuestas para solucionar el problema. Estos planteamientos conducen a tres ventajas y desventajas diferentes, que se han discutido a lo largo del trabajo. Por último, las diferentes propuestas también se pueden relacionar con varios escenarios posibles, también descritos en el documento.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia gracias a los proyectos CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", TSI2007-65393-C02-02 "ITACA" y TEC2008-06663-C03-01 "P2PSEC", y por la Generalitat de Catalunya gracias al grupo de investigación consolidado 2009 SGR 1362.

REFERENCIAS

[1] T. R. Henderson and R. H. Katz. Transport Protocols for Internet-Compatible Satellite Networks. *IEEE Journal on Selected Areas in Communications*, 17(2):326–344, 1999.

[2] C. Caini, R. Firrincieli, M. Marchese, T. de Cola, N. Celandroni M. Luglio, C. Roseti, and F. Potortì. Transport Layer Protocols and Architectures for Satellite Networks. *International Journal of Satellite Communications and Networking*, 25:1–26, 2007.

[3] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. A comparison of mechanisms for improving tcp performance over wireless links. *Computer Communication*, 26(4):256–269, 1996.

[4] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135 (Informational), June 2001.

[5] Yongguang Zhang. A multilayer ip security protocol for tcp performance enhancement in wireless networks. *Selected Areas in Communications, IEEE Journal on*, 22(4):767–776, may 2004.

[6] J. Sing and B. Soh. A critical analysis of multilayer ip security protocol. *Proceedings. Third International Conference on Information Technology and Applications*, 2:683–8, 2005.

[7] M. Karir and J. Baras. Les: Layered encryption security. *Proceedings of the Third International Conference on Networking (ICN'04)*, 2004.

[8] A. Roy-Chowdhury and J.S. Baras. Performance-aware security of unicast communication in hybrid satellite networks. *ICC 2009 - 2009 IEEE International Conference on Communications*, pages 6 pp. –, 2009.

[9] N. Thanthry, M. Deshpande, and R. Pendse. A novel mechanism for improving performance and security of tcp flows over satellite links. *Proceedings - International Carnahan Conference on Security Technology*, pages 197–202, 2006.

[10] D.D. Isci, F. Alagoz, and M.U. Caglayan. Ipsec over satellite links: a new flow identification method. *Proceedings of ISCN'06 7th International Symposium on Computer Networks (IEEE Cat. No.06EX1429)*, pages 140–5, 2006.

[11] A. Parichehreh and B. Eliasi. Vpn over satellite: performance improving of e2e secured tcp flows. *2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08)*, pages 40–3, 2008.

[12] V. Obanaik, L. Jacob, and A.L. Ananda. Secure performance enhancing proxy: to ensure end-to-end security and enhance tcp performance over ipv6 wireless networks. *Computer Networks*, 50(13):2225–38, 2006.

[13] G. Ciccarese, M. De Blas, L. Patrono, P. Marra, and G. Tomasicchio. An ipsec-aware tcp pep for integrated mobile satellite networks. *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, 4:2362–6, 2004.