IPsec en Ambientes IPv4 e IPv6

Hugo Adrian Francisconi

adrianfrancisconi@yahoo.com.ar

Primera Edición (Versión 1.0)

Agosto 2005

Datos del Autor/Editor de Esta Obra

Nombre y Apellido del Autor: Hugo Adrian Francisconi
Domicilio: Carril Godoy Cruz 2801, Villa Nueva, Guaymallén, Mendoza Argentina
Código Postal: 5521
Tel: 054-0261-4455427
E-mail: adrianfrancisconi@yahoo.com.ar

Derechos de Copyright sobre esta Obra

ISBN 987-43-9727-6
Elaborado, editado e impreso en Carril Godoy Cruz 2801, Villa Nueva, Guaymallén, Mendoza Argentina.
Fecha de elaboración , edición e impresión en Agosto del 2005.

Marcas Comerciales

Todo los términos en este libro que correspondan a Marcas Comerciales o marcas de Servicio, el autor no puede certificarse la exactitud de la información. No debe considerarse que el uso de un término en este libro afecte a la validez de cualquier marca comercial o marca de servicio. Las marcas comerciales y demás marcas denominadas son propiedad de sus respectivos titulares.

Advertencia y Renuncia a Derechos

Se a realizado el máximo esfuerzo para hacer de este libro una obra tan completa y precisa como sea posible, pero no se ofrece ninguna garantía implícita de adecuación a un fin en particular.

La información se suministra "tal como está". El autor no será responsable ante cualquier persona o entidad con respecto a cualquier pérdida o daño que pudiera resultar emergente de la información contenida en este libro.

Las opiniones expresadas en este libro pertenecen al autor.

A mi Padre

Índice General de Contenidos

Prefacio

Esta completa obra proporciona un análisis exhaustivo sobre Seguridad en Internet en la Capa IP (IPsec), basándose en estándares internacionales (RFC). Esto incluye los servicios de Autentificación, Integridad, Ocultación del contenido (Confidencialidad), Control de Acceso, etc. y el conjunto de protocolos por el cual esto es llevado a cabo (AH, ESP, IKE, ISAKMP, etc). Como así también se profundiza sobre los conceptos y algoritmos criptográficos para poder entender y desarrollar esa tecnología.

Este libro es altamente recomendable para los implementadores de esta tecnología, como para el público en general que requieran un conocimiento profundo del sistema y este familiarizado con el Protocolo de Internet (IP) y la tecnología relacionada a redes.

Prólogo

Este libro surge debido a que para mi trabajo de fin de carrera (tesis), para obtener el título de Ing. en Electrónica en la Universidad Tecnológica Nacional - Facultad Regional Mendoza (UTN-FRM) Argentina (http://www.frm.utn.edu.ar/), se me pidió que investigara sombre IPsec. Al comenzar esta investigación (a principios del año 2003), encontré que la bibliografía en castellano era muy escasa y la que se encontraba resultaba escueta y/o errónea por ende me dispuse a investigar directamente de la fuente "Los RFCs" esto trajo aparejado que bebido a que tenía que realizar un estudio con gran profundidad en el tema es que **traduje íntegramente** los RFC:

RFC 2401: Security Architecture for the Internet Protocol.
RFC 2402: IP Authentication Header.
RFC 2403: The Use of HMAC-MD5-96 within ESP and AH.
RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH.
RFC 2405: The ESP DES-CBC Cipher Algorithm With Explicit IV.
RFC 2406: IP Encapsulating Security Payload (ESP).
RFC 2407: The Internet IP Security Domain of Interpretation for ISAKMP.
RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP).
RFC 2409: The Internet Key Exchange (IKE).
RFC 2410: The NULL Encryption Algorithm and Its Use with IPsec.
RFC 2411: IP Security Document Roadmap.
RFC 2412: The OAKLEY Key Determination Protocol.

los cuales pondré a disposición de cualquier persona que me los pida y estarán disponibles para descargarse de forma gratuita de la pagina del grupo de investigación universitario del que formo parte **Codarec 6**, http://codarec6.frm.utn.edu.ar y también espero que sean publicados en http://www.rfc-es.org. Esa labor junto con con años de investigación de otros RFCs y demás bibliografía consultada derivaron en la creación de este libro.

He realizado considerables esfuerzos para que el contenido de este libro sea totalmente compatible con el formato ".txt", es decir que tenga símbolos (como por ejemplo estilos de texto, subrayados superíndice, etc.) que al pasarlos a texto plano se pierda el contenido y/o el sentido del texto. Esto se ha logrado es todos los capítulos salvo en el **Capítulo 5** que contiene símbolos que al pasarlos a texto plano se perderán.

Este libro ha sido realizado con programas gratuitos; **OpenOffice** v1.1.4, corriendo bajo **Linux** (Ubuntu-hoary).

Capítulo 1

Introducción

1. **Audiencia**

La audiencia de este documento incluye a implementadores de esta tecnología de seguridad IP y a los interesados que quieran un conocimiento profundo del sistema. En particular, los usuarios potenciales de esta tecnología (los usuarios finales o los administradores de sistema) son parte de la audiencia. Un Glosario y un listado de Acrónimos es proporcionado como Apéndice para comprender el vocabulario. Este documento asume que el lector esta familiarizado con el Protocolo Internet (IP), la tecnología relacionada a redes, los términos y conceptos generales de seguridad.

Los capítulos 2, 3, 4, 7, 8, 9 y 10 derivan en gran medida de los trabajos realizados en [ARCH], [AH], [ESP], [ISAKMP], [DOIPsec], [OAKLEY] y de [IKE] respectivamente. El capítulo 6 deriva en gran medida de los trabajos realizados en [HMACMD5], [HMACSHA], [DES], [ESPNULL]. Por ende a esos capítulos se puede aplicar que: Las palabras DEBE, NO DEBE, REQUERIDO, PODER, NO PODER, DEBERÍA, NO DEBERÍA, RECOMENDADO, PUEDE y OPCIONAL, cuando aparezcan en este documento, deben interpretarse como se describe en [Bra97].

2. **Objetivos de Diseño**

IPsec está diseñado para proporcionar seguridad ínter-operable, de alta calidad, basada en criptografía, tanto para IPv4 como para IPv6. El conjunto de servicios de seguridad ofrecidos incluye:

- Control de acceso: previene el uso no autorizado de recursos.
- Integridad sin conexión: detección de modificaciones en un datagrama IP individual.
- Autenticación del origen de los datos.
- Protección anti-replay: una forma de integrabilidad parcial de la secuencia, detecta la llegada de datagramas IP duplicados.
- Confidencialidad: encriptación.
- Confidencialidad limitada del flujo de tráfico: el uso del modo túnel permite encriptar las cabeceras IP internas, ocultando las identidades del origen del tráfico y del (último) destino. También, se puede usar el "relleno en la carga útil" (payload padding) de ESP para ocultar el tamaño de los paquetes, consiguiendo ocultar las características externas del tráfico.

Estos servicios se implementan en la capa IP, y ofrecen protección para este nivel y/o los niveles superiores. El IPsec DOI también soporta negociación de compresión IP [IPCOMP], motivado en parte por la observación de que cuando se emplea encriptación en IPsec, esta impide la compresión eficiente de los datos en los protocolos inferiores.

Estos objetivos se llevan a cabo haciendo uso de dos protocolos de seguridad, la Cabecera de Autentificación (AH) y la Carga de Seguridad Encapsulada (ESP), a través de procedimientos de manejo de claves criptográficas y protocolos. El conjunto de protocolos IPsec empleados en cualquier conexión, y la forma en que se emplean, serán determinados por la seguridad, y los requerimientos del sistema del usuario, aplicaciones y/o sitios o organizaciones.

Cuando estos mecanismos se implementan correctamente y se ejecutan, no afectan negativamente a los usuarios, hosts, y otros componentes de Internet que no empleen estos mecanismos de seguridad para la protección de su tráfico. Estos mecanismos están diseñados para ser independientes del algoritmo. Esta modularidad permite seleccionar diferentes conjuntos de algoritmos sin afectar a las otras partes de la implementación. Por ejemplo, grupos diferentes de usuarios pueden seleccionar grupos diferentes de algoritmos si se necesita.

Un conjunto de algoritmos se especifica para facilitar la interoperatibilidad en la Internet global. El uso de estos algoritmos, en conjunto con la protección del tráfico de IPsec, y los protocolos de manejo de claves, están constituidos para permitir el desarrollo de aplicaciones, sistemas y tecnología criptográfica de seguridad de alta calidad en la capa IP.

3 **Descripción del Sistema**

Esta sección proporciona una descripción de cómo trabaja IPsec, los componentes del sistema, y como se adaptan para proporcionar los servicios de seguridad. La meta de esta descripción es permitirle al lector "comprender" el proceso global del sistema, ver como se adaptan en el ambiente IP, y proporcionar el contexto para capítulos posteriores de este documento, que describen cada uno de los componentes más detalladamente.

Una implementación de IPsec funciona en un host o en un security gateway (SG), proporcionando protección al tráfico IP. La protección ofrecida se basa en requerimientos definidos en el establecimiento de una "Base de Datos de Políticas de Seguridad" (SPD) y mantenidas por un usuario o administrador del sistema o por una aplicación funcionando dentro de las restricciones ya establecidas. En general, los paquetes se seleccionan para uno de tres modos de procesamiento basados en IP y la información de la cabecera de la capa de transporte comparándolas con las entradas en la Base de Datos de Políticas de Seguridad (SPD). Cada paquete es un servicio de seguridad, descartado, desviado, o procesado, de acuerdo con las políticas aplicables en la base de datos identificadas por los selectores.

3.1 Que hace IPsec

IPsec proporciona servicios de seguridad en la capa IP permitiendo a un sistema seleccionar los protocolos de seguridad, determinar el/los algoritmo/s a utilizar para el/los servicio/s, y implementar cualquier algoritmo criptográfico requerido para proporcionar los servicios solicitados. IPsec se puede utilizar para proteger una o más "trayectorias" entre un par de hosts, o entre un par de security gateway, o entre un security gateway y un host. El término security gateway se utiliza en este documento para referirse a un sistema intermedio que implementa los protocolos IPsec. Por ejemplo, un router o un firewall implementando IPsec es un security gateway.

3.2 Como Trabaja IPsec

IPsec utiliza dos protocolos para proporcionar seguridad al tráfico: la Cabecera de Autentificación (AH) y la Carga de Seguridad Encapsulada (ESP).

- La Cabecera de Autentificación (AH): Proporciona integridad sin conexión, autentificación del origen de datos, y un servicio opcional de protección antireplay.

- La Carga de Seguridad Encapsulada (ESP): Puede proporcionar confidencialidad (encriptación), y confidencialidad limitada de flujo de tráfico. También puede proporcionar integridad sin conexión, autentificación del origen de datos, y un servicio de protección antireplay. Uno u otro de estos servicios de seguridad debe ser aplicado siempre que se use ESP.

- AH y ESP son instrumentos para el control de acceso, basados en la distribución de claves criptográficas y en el manejo de flujo de tráfico concerniente a estos protocolos de seguridad.

Estos protocolos pueden aplicarse solos o en conjunto con otros para proporcionar un conjunto de servicios de seguridad en IPv4 e IPv6. Cada protocolo soporta dos modos de uso: modo transporte y modo túnel. En modo transporte los protocolos proporcionan protección sobre todo a los protocolos de capa superiores; en modo túnel, los protocolos son aplicados a paquetes (a los que se hizo un túnel a través de IP).

IPsec permite que el usuario (o el administrador de sistema) controle la granularidad en la cual un servicio de seguridad es ofrecido. Por ejemplo, uno puede crear un único túnel encriptado y llevar todo el tráfico entre dos security gateway o un túnel encriptado separado se puede crear para cada conexión TCP entre cada par de hosts que se comunican a través de un gateways. La gestión de IPsec debe incorporar facilidades para especificar:

- Que servicios de seguridad se utilizar y en que combinaciones.
- La granularidad con la que se debe aplicar una determinada protección de seguridad.
- Los algoritmos usados para efectuar la seguridad basada en criptografía.

Debido a que estos servicios de seguridad usan valores secretos compartidos (claves criptográficas), IPsec se basa en un conjunto de mecanismos separados para que pongan estas claves en su sitio (las claves se utilizan para autentificación/integrabilidad y los servicios de encriptación). Este documento explica la distribución manual y automática de claves. Explica un método basado en clave pública (IKE – [ISAKMP], [OAKLEY], [IKE]) para la gestión automática de claves, pero otras técnicas de distribución automatizadas de claves PUEDEN ser utilizadas. Por ejemplo, los sistemas basados en KDC tales como Kerberos y otros sistemas de clave pública tales como SKIP podrían ser empleados.

3.3 Donde Puede ser Implementado IPsec

Hay varias formas en las cuales se puede implementar IPsec, en un host o en conjunto con un router o un firewall (creando un security gateway). Algunos ejemplos frecuentes son:

a. Integrar IPsec en una implementación nativa IP. Requiere tener acceso al código fuente IP, y se puede aplicar tanto a host como a un security gateway.

b. "Puesto en la Pila" (**BITS**), IPsec se implementa "por debajo" de una implementación existente de una pila IP, entre el IP nativo y los drivers locales de la red. El acceso al código fuente para la pila IP no es requerido en este contexto, este contexto es apropiado para los sistemas antiguos. Este método, cuando se adopta, se emplea generalmente en hosts.

c. "Puesto en el cable" (**BITW**). El uso de un procesador criptográfico externo es una característica de diseño común de los sistemas de seguridad de red usados por los militares, y en algunos sistemas comerciales. Tales implementaciones se pueden diseñar para asistir a un host o un gateway (o a ambos). El dispositivo BITW generalmente tiene una IP direccionable. Cuando asiste a un único host, puede resultar análogo a una implementación BITS, pero en un router o en un firewall debe funcionar como un security gateway.

4. **Observaciones y Advertencias**

El grupo de protocolos IPsec y demás algoritmos asociados permiten proporcionar seguridad de alta calidad para el flujo de tráfico de Internet. Sin embargo, la seguridad ofrecida por estos protocolos depende en última instancia de la calidad de su implementación, que esta fuera del alcance de este libro. Además, la seguridad de un sistema informático o en una red es una función de muchos factores. IPsec solo es una parte de un sistema global de seguridad.

La seguridad proporcionada por el uso de IPsec dependerá bastante de diversos aspectos del ambiente operativo en el cual la implementación de IPsec se ejecuta. Por ejemplo, los defectos en la seguridad del sistema operativo, negligencia en la práctica de manejo de protocolos, etc., todo esto puede degradar la seguridad proporcionada por IPsec. Como se mencionó anteriormente, ninguna de estas características están dentro del alcance de este libro o de algún estándar de IPsec.

Por ultimo, este documento no es una arquitectura global de seguridad para Internet, solo se ocupa de la seguridad en la capa IP.

Capítulo 2

Arquitectura IPsec

1. **Introducción**

1.1 Contenido del Capítulo

Este capítulo especifica la estructura fundamental de IPsec. La meta de la arquitectura es proporcionar diversos servicios de seguridad para el tráfico en la capa IP, en ambientes IPv4 e IPv6. Este capítulo describe las metas de tales sistemas, sus componentes y cómo estos se adaptan en el ámbito IP. Este capítulo también describe el servicio de seguridad proporcionado por los protocolos IPsec, y como estos servicios se pueden emplear en ambientes IP. Los componentes fundamentales de la arquitectura de IPsec se discuten en términos de la funcionalidad subyacente requerida. Capítulos adicionales (véase la Sección 1.2 que comenta otros capítulos) definen los protocolos (a), (c), y (d).

a. Protocolos de seguridad: Cabecera de Autentificación (AH) y la Carga de Seguridad Encapsulada (ESP).

b. Asociaciones de Seguridad SA: que son, como funcionan, como se administran, y como se procesan.

c. Manejo de claves: en forma manual y automática (IKE, Intercambio de Claves en Internet).

d. Algoritmos para autentificación y encriptación.

1.2 Capítulos Relacionados

Otros capítulos proporcionan definiciones detalladas de algunos de los componentes de IPsec y de su interrelación. Los siguientes temas se incluyen en los siguientes capítulos:

a. Protocolos de seguridad: Capítulos que describen la Cabecera de Autentificación (Capítulo 3) y la Carga de Seguridad Encapsulada (Capítulo 4).

b. Algoritmos para autentificación y encriptación: en el Capítulo 5 se brinda los conceptos teóricos, métodos y algoritmos generales. En el Capítulo 6 se brindan los algoritmos que obligatoriamente toda implementación de AH y ESP DEBEN tener.

c. Gestión automática de claves: El capítulo "Intercambio de Clave en Internet (IKE)" (Capítulo 10), " Protocolo de Manejo de Claves y Asociaciones de Seguridad en Internet (ISAKMP)" (Capítulo 7), "Protocolo de Determinación de Claves OAKLEY" (Capítulo 9), y el Dominio de Interpretación de Seguridad IP en Internet para ISAKMP (Capítulo 8).

2. **Asociaciones de Seguridad**

Esta sección define los requisitos para administrar Asociaciones de Seguridad para toda implementación IPv6 y para implementaciones IPv4 que implemente AH, ESP, o ambos. El concepto de Asociación de Seguridad (SA) es fundamental para IPsec. AH y ESP hacen uso de SAs y una función importante de IKE es el establecimiento y el mantenimiento de SAs. Toda implementación de AH o ESP DEBE soportar el concepto de SA como se describe abajo. El resto de esta sección describe los diversos aspectos del manejo de SA, definiendo las características requeridas para la gestión de políticas de SA, procesamiento de tráfico, y las técnicas de gestión de SA.

2.1 Definiciones y Ámbito

Una Asociación de Seguridad (SA) es una "conexión" lógica unidireccional (simplex) que ofrece servicios de seguridad al tráfico transportado por este. Los servicios de seguridad ofrecidos en una SA son usados por AH o ESP, pero no por ambos. Si ambos (AH y ESP) se aplican a un flujo de tráfico, dos (o más) SAs se crearán para generar la protección de flujo del tráfico. Para asegurar la comunicación bidireccional entre dos hosts, o entre dos security gateway, se requieren dos Asociaciones de Seguridad (una en cada sentido).

Una SA es identificada unívocamente por un trío que consiste en: un Índice de Parámetros de Seguridad (SPI), una Dirección IP de Destino, y un identificador de protocolo de seguridad (AH o ESP). En principio, la Dirección de Destino puede ser una dirección unicast, una dirección de difusión IP, o una dirección de grupo multicast. Sin embargo, los mecanismos IPsec para la gestión de SA se definen solamente para unicast. Aun cuando el concepto también es aplicable a conexiones punto a multipuntos.

Según lo observado arriba, se definen dos tipos de SAs: modo transporte y modo túnel. Una SA en modo trasporte es una SA entre dos hosts. En IPv4, una cabecera de protocolo de seguridad en modo transporte aparece inmediatamente después de la cabecera IP y de algunas opciones, y antes que cualquier protocolo de capas superior (por ejemplo, TCP o UDP). En IPv6 las cabeceras del protocolo de seguridad se sitúan después de la cabecera IP y de extensiones pero deben aparecer antes o después de la cabecera de opciones de dirección y antes de los protocolos de capas superiores. En el caso de ESP, una SA en modo transporte proporciona servicios de seguridad solamente para los protocolos de las capas superiores, no para la cabecera IP o cualquier cabecera de extensión precedente a la cabecera ESP. En el caso de AH la protección se extiende a las partes seleccionadas de la cabecera IP, a las partes seleccionadas de las cabeceras de extensión y a las opciones seleccionadas (contenidas en la cabecera de IPv4, la cabecera de extensión Salto-por-Salto de IPv6, o la cabecera de extensión de destino de IPv6). Para más detalles de la cobertura proporcionada por AH, vea la especificación de AH en el Capítulo 3.

Una SA en modo túnel es en esencia una SA aplicada a un túnel IP. Siempre que un extremo de la SA sea un security gateway, la SA DEBE estar en modo túnel. Una SA entre dos security gateway, es siempre una SA en modo túnel, al igual que una SA entre un host y un security gateway. Nótese que para el caso donde el tráfico es destinado para el security gateway, por ejemplo, comandos SNMP, la security gateway actúa como un host y el modo trasporte es permitido. Pero en este caso, la security gateway, no esta actuando como un gateway, es decir, no esta trasportando tráfico. Dos host PUEDEN establecer una SA en modo túnel entre ellos. El requisito para cualquier SA que involucre a una security gateway (trasporte de tráfico) es un túnel SA debido a la necesidad de evitar problemas potenciales con la fragmentación y reensamblaje de paquetes IPsec y en circunstancias donde existan múltiples trayectorias (por ejemplo vía diferentes security gateway) para el mismo destino detrás de un security gateway.

Para una SA en modo túnel, hay una cabecera IP "externa" que especifica el destinatario del proceso IPsec, más una cabecera IP "interna" que especifica el último destinatario (aparente) del paquete. La cabecera del protocolo de seguridad aparece después de otras cabeceras IP externas y antes de las cabeceras IP internas. Si se emplea AH en modo túnel, a otras partes de la cabecera IP se les ofrecen protección así como también a todo

el paquete IP al cual se le hizo el túnel (es decir, toda la cabecera IP interna es protegida, como así también protocolos de capas superiores). Si se emplea ESP, la protección es proporcionada únicamente al paquete IP al cual se le hizo el túnel (al paquete "tunelizado"), no a las cabeceras externas. En resumen:

a) Un host DEBE soportar modo trasporte y túnel.

b) Una security gateway solo debe soportar el modo túnel. Si soporta modo trasporte este debería ser usado únicamente cuando la security gateway actúa como host, por ejemplo para la administración de la red.

2.2 Funcionalidad de las Asociaciones de Seguridad

El conjunto de servicios de seguridad ofrecido por una SA depende del protocolo de seguridad seleccionado, del modo de la SA, de los extremos de la SA, y de la elección de los servicios opcionales seleccionados dentro del protocolo. Por ejemplo, AH proporciona autentificación del origen de los datos e integridad sin conexión para datagramas IP (a partir de ahora equivale a "autentificación"). La "precisión" de estos servicios de autentificación estará en función de la granularidad de la SA con la que se emplea AH. Esto se describe en la Sección 2.4.2 "selectores"

AH ofrece además un servicio de anti-replay (integridad parcial de la secuencia) según el deseo del receptor, esto ayudará a prevenir ataques contra denegación de servicios. AH es un protocolo apropiado para emplearse cuando la confidencialidad no es requerida (o no se permite, por ejemplo, debido a las restricciones gubernamentales en el uso criptográfico). AH también proporciona autentificación para las partes seleccionadas de la cabecera IP, que puede ser necesaria en algunos contextos. Por ejemplo, si la integridad de una opción de IPv4 o una cabecera de extensión de IPv6 se debe proteger en el camino entre el emisor y el receptor, AH puede proporcionar este servicio (a excepción de las partes mutables no predecibles de la cabecera IP).

ESP proporciona de forma opcional confidencialidad para el tráfico. (La robustez del servicio de confidencialidad depende en parte, del algoritmo de encriptación utilizado). ESP también proporciona de forma opcional, autenticación como en el caso anterior. Si la autentificación es negociada por una SA ESP, el receptor también puede elegir implementar el servicio de anti-replay con las mismas características que el servicio de anti-replay de AH. La autentificación ofrecida por ESP abarca menos que la ofrecida por AH, es decir las cabeceras que quedan por fuera de la cabecera ESP no están protegidas. Si solo los protocolos de capas superiores necesitan ser autentificados, entonces la autentificación de ESP es una elección apropiada y es más eficiente en tamaño que usar ESP encapsulado con AH. Note que aunque la confidencialidad y la autenticación son opcionales en ESP, no se pueden omitir ambas, al menos una DEBE ser escogida.

Si se elige el servicio de confidencialidad, entonces una SA ESP (en modo túnel) entre dos security gateway pueden ofrecer confidencialidad parcial del flujo de tráfico. El uso del modo túnel permite encriptar las cabeceras IP internas, ocultando las identidades del origen del tráfico y del (último) destino. También, se puede usar el "relleno en la carga útil" (payload padding) de ESP para ocultar el tamaño de los paquetes, consiguiendo ocultar las características externas del tráfico. Similares servicios de confidencialidad del flujo de tráfico pueden ser ofrecidos cuando un usuario móvil está asignado a una dirección IP dinámica en un contexto de dialup, y establecer una SA ESP (en modo túnel) en un firewall

corporativo (actuando como un security gateway). Observe que SAs con poca granularidad generalmente son más vulnerables al análisis de tráfico que unos con mucha granularidad en el cual se esta llevando el tráfico de muchos suscriptores.

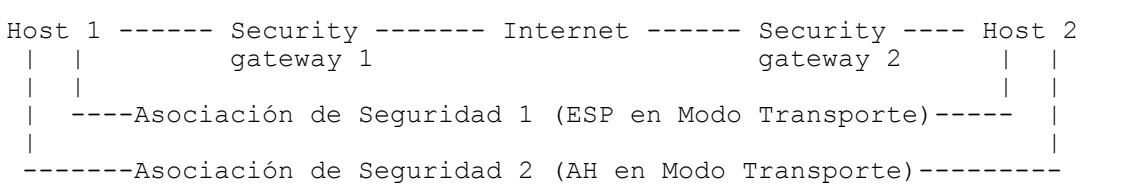
2.3 Combinación de Asociaciones de Seguridad

Los datagramas IP trasmitidos por una SA individual permiten la protección de un protocolo de seguridad, AH o ESP, pero no ambos. En ocasiones una política de seguridad puede determinar una combinación de servicios para un flujo de tráfico especifico que no se puede realizar por una única SA. En estos casos será necesario emplear múltiples SAs para implementar la política de seguridad requerida. El termino "grupo de asociaciones de seguridad" o "grupo de SA" se aplica a una secuencia de SAs las cuales deben procesar el tráfico para satisfacer una política de seguridad. El orden de la secuencia se define en la política de seguridad. Note que las SAs que comprenden un grupo pueden terminar en diferentes extremos. Por ejemplo, una SA puede comprender a un host móvil y a un security gateway, y una segunda SA puede existir entre el host móvil y un host que se encuentra de tras de la security gateway.

extenderse entre un host móvil y un security gateway y una segunda, SA puede extenderse a una host detrás de un gateway.)

Las SAs pueden combinarse entre grupos de dos formas: transporte adyacente (*transport adjacency*) y entre túneles (*iterated tunneling*).

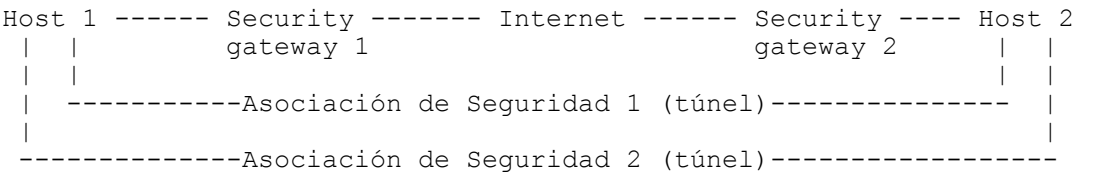
Transporte adyacente: se aplica más de un protocolo de seguridad al mismo datagrama IP, sin utilizar túneles. Este método combina a AH y a ESP permitiendo solamente un nivel de combinación, el anidado adicional no produce beneficio adicional (asumiendo el uso de algoritmos adecuados en cada protocolo) puesto que el proceso se realiza en una instancia de IPsec en el último destino.



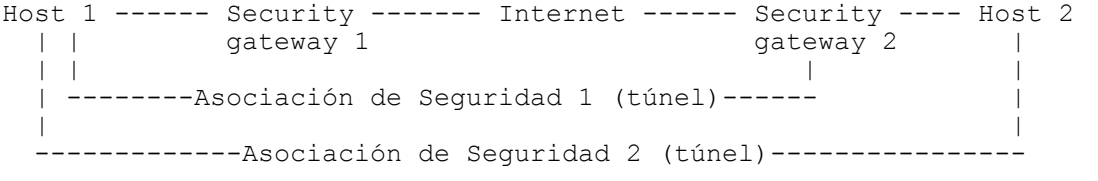
Entre túneles: se refiere a la aplicación de múltiples capas del protocolo de seguridad efectuando múltiples túneles IP. Este método permite múltiples niveles de anidado, puesto que cada túnel se puede originar o terminar en nodos diferentes a lo largo de la trayectoria. No se espera ningún tratamiento especial para el tráfico de ISAKMP en las security gateway intermedias con excepción de que se puede especificar a través de que Base de Datos de Política de Seguridad (SPD) asignada entrar (véase el caso 3 de la Sección 2.5).

Hay tres casos básicos de entre túneles, pero solamente se requiere soporte para el caso 2 y 3:

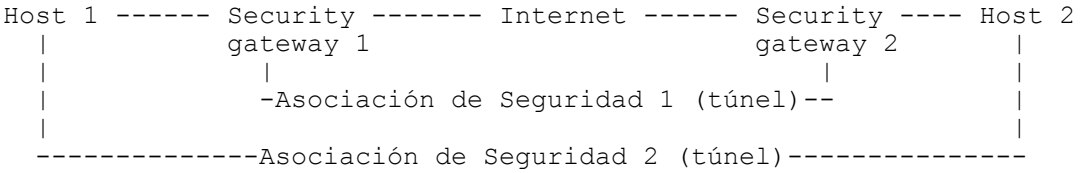
1. Ambos extremos de las SAs son los mismos: Los túneles (interno o externo) pueden ser AH o ESP, aunque es improbable que el host 1 especifique ambos túneles iguales, es decir, AH a dentro de AH, o ESP dentro de ESP.



2. Un extremo de las SAs es igual: Los túneles (interno o externo) pueden ser AH o ESP.



3. Ninguno de los extremos es igual: Los túneles (interno o externo) pueden ser AH o ESP



Estos dos métodos podrían ser combinados, por ejemplo, un grupo de SAs se podría construir a partir de un modo túnel SA y uno o dos modo trasporte SAs, aplicados en secuencia (ver Sección 2.5 "Combinaciones Básicas de Asociaciones de Seguridad"). Observe que los túneles anidados también pueden existir donde ni el origen ni los extremos destinatarios de cualquiera de los túneles son los mismos. En este caso no habría host o security gateway con un grupo que corresponda para los túneles anidados.

Para SAs en modo trasporte, sólo una estructura de protocolos de seguridad parece ser apropiada. AH es aplicado tanto a los protocolos de capas superiores como a las partes de la cabecera IP. Así si AH es usado en modo trasporte, en conjunto con ESP, AH DEBERÍA aparecer como la primera cabecera después de la cabecera IP, antes de la cabecera ESP. En este caso, AH se aplica a la salida del texto cifrado de ESP. En cambio, para SAs en modo túnel, uno puede usar varias estructuras (ordenamientos) de AH y de ESP. El conjunto requerido de tipos de grupos de SA DEBE ser soportado por una implementación compatible IPsec como se describe en la Sección 2.5.

2.4 Bases de Datos

Muchos de los detalles relacionados al procesamiento de tráfico IP en una implementación IPsec son en gran parte tema local, no sujetos a estandarización. Sin embargo algunos aspectos externos del proceso deben ser estandarizados para asegurar interoperatibilidad y proporcionar una capacidad de gestión mínima que es esencial para el uso productivo de IPsec. Esta sección describe un modelo general para procesar el tráfico IP referente a asociaciones de seguridad, el soporte de esta interoperatibilidad y el funcionamiento global.

Hay 2 bases de datos en este modelo:

- La Base de Datos de Políticas de Seguridad (SPD): especifica las

políticas que determinan el tratamiento de todo el tráfico IP entrante o saliente en un host, security gateway, o en implementaciones IPsec BITS o BITW.

- La Base de Datos de Asociaciones de seguridad (SAD): contiene los parámetros que se asocian con cada SA activa.

Esta sección también define el concepto de Selector, que es un conjunto de campos con valores de protocolos de capas superiores y de la capa IP que son usados por la SPD para asignar el tráfico a una política, es decir, a una SA (o grupo de SA).

Cada interfaz para la cual se habilite IPsec normalmente requiere, entradas y salidas de la base de datos separadas (SAD y SPD), debido a que la direccionalidad de varios de los campos son usados como selectores. Típicamente hay solo una interfaz, para un host o security gateway. Observe que un security gateway podría tener 2 interfaces, pero una red corporativa interna, usualmente no tendría habilitado IPsec y tan sólo un par de SADs y un par de SPDs sería necesarios. Por otra parte, si un host tiene múltiples interfaces o un security gateway tiene múltiples interfaces externas, puede que sea necesario tener una SAD y una SPD separadas para cada interfaz.

2.4.1 Base de Datos de Políticas de Seguridad (SPD)

En última instancia, una SA es generada por la gestión usada para implementar una política de seguridad en el ambiente IPsec. De esta manera un elemento esencial del proceso de la SA es una SPD subyacente que especifica qué servicios deben ser ofrecidos a los datagramas IP y de qué forma. La forma de la base de datos y su interfaz están fuera del alcance de este libro. Sin embargo, esta sección especifica ciertas funciones mínimas de gestión que deben ser proporcionadas, para permitir que un usuario o administrador del sistema controle cómo se aplica IPsec al tráfico enviado o recibido por un host o una transmisión a un security gateway.

El SPD se debe consultar durante todo el procesamiento del tráfico (ENTRANTE y SALIENTE), incluyendo tráfico no IPsec. Para soportar esto, la SPD requiere entradas distintas para el tráfico de entrada y de salida. Uno puede pensar en esto como SPDs separadas (una de entrada y otra de salida). Una SPD nominal separada se debe proporcionar para cada interfaz IPsec habilitada.

Una SPD debe diferenciar entre el tráfico al que debe ofrecer protección IPsec de al que le esta permitido evitar IPsec. Esto implica que la protección IPsec a ser empleada debe estar presente tanto en el receptor como en el emisor. Para cualquier datagrama de entrada o de salida, hay tres opciones de procesamiento posibles:

- Descartar: es el tráfico al que no se permite salir del host, atravesar una security gateway, o que se entregue a una aplicación.

- Evitar IPsec (no IPsec): se refiere al tráfico que se le permite pasar sin la protección de IPsec.

- Que se aplique IPsec: es el tráfico que es protección producida por IPsec, y para tal tráfico la SPD debe especificar los servicios de seguridad que se proporcionarán, los protocolos que se emplearán, los algoritmos que se utilizarán, etc.

Para cada implementación IPsec, DEBE haber una interfaz administrativa que permita a un usuario o administrador del sistema manejar la SPD. Específicamente, cada paquete de entrada o de salida está sujeto al procesamiento IPsec, y la SPD deberá especificar qué acción será tomada en cada caso. Específicamente, la SPD deberá especificar qué acción se tomará para cada paquete de entrada y de salida. La interfaz administrativa debe permitir que el usuario (o el administrador del sistema) especifique que proceso de seguridad a de ser aplicado a cada paquete entrante o saliente del sistema. (En una implementación host IPsec que utiliza una interfaz socket, la SPD puede no necesitar ser consultada sobre bases de paquetes, pero el efecto sigue siendo igual.) La interfaz de gestión para el SPD DEBE permitir la creación de entradas consistentes con los selectores definidos en la Sección 2.4.2, y DEBE soportar el ordenamiento total de esas entradas. Se espera que con el uso de comodines en varios campos del selector, y puesto que todos los paquetes en una sola conexión UDP o TCP tenderán correspondencia con una sola entrada SPD, este requisito no impondrá un nivel irracionalmente detallado de la especificación de SPD. Los selectores son análogos a los que se encuentran en un firewall o en un filtrado de router los cuáles son actualmente manejados de esa forma.

En un sistema host, las aplicaciones se PUEDEN permitir seleccionar que proceso de seguridad debe ser aplicado al tráfico que generan y consumen. El administrador del sistema DEBE poder especificar si una aplicación puede o no reemplazar la política del sistema. Observe que la aplicación especificó políticas que pueden satisfacer requisitos del sistema, de modo que el sistema puede no necesitar un proceso IPsec adicional que procese más allá de este para resolver los requisitos de una aplicación.

El SPD contiene una lista ordenada de políticas de entrada. Cada política de entrada es introducida por uno o más selectores que definen el conjunto de tráfico IP comprendido por esta política de entrada. Estos definen la granularidad de las políticas o SAs. Cada entrada incluye un indicador para el tráfico coincidente con esta política, si será desviado, desechado, o procesado por IPsec. Si el procesamiento IPsec es aplicado, la entrada incluirá una especificación de SA (o grupo de SA), listado de Protocolos IPsec, los modos, y algoritmos que se emplearán, y incluirán cualquier requisito relacionado. Por ejemplo, una entrada puede demandar proteger todo el tráfico coincidente por ESP en modo trasporte usando 3DES-CBC con un IV explicito, anidado adentro de AH en modo túnel usando HMAC/SHA-1. Para cada selector, la política de entrada especificará cómo obtener los valores correspondientes para una nueva entrada SAD en la SPD y el paquete (Note que actualmente, los rangos son sólo soportados para direcciones IP, pero por medio del comodín puede ser expresado cualquier selector):

a. Usar el valor en el mismo paquete: Esto limita el uso de las SA a paquetes que tienen esos valores en el paquete para el selector, incluso si en selector tiene una política de entrada en un rango de valores de entrada permitidos o en comodín para este selector.

b. Usar el valor asociado con la política de entrada: Si este fuera un solo valor, no habría diferencia entre (a) y (b). Sin embargo si los valores permitidos para el selector son un rango (de direcciones IP) o un comodín, entonces en el caso de un rango, (b) habilitaría el uso de SA para cualquier paquete con un valor del selector dentro de un rango no exacto pero los paquetes con el valor del selector en el paquete provocarán la creación de SA. En el caso de un comodín, (b) podría usarse en la SA para paquetes con cualquier valor para ese selector.

Por ejemplo suponga que hay una SPD de entrada donde el valor permitido

para una dirección de origen es cualquiera de un rango de host (192.168.2.1 a 192.168.2.10). Y suponga que un paquete es enviado a una dirección 192.168.2.3. El valor que se utiliza para la SA podría ser cualquiera de los que esta debajo de ejemplo, dependiendo de la política de entrada para este selector, es decir el origen del valor del selector.

Origen del valor a ser usado en la SA	Ejemplo de un nuevo valor de selector SAD
a. Paquete	192.168.2.3 (un host)
b. Entrada SPD	192.168.2.1 a 192.168.2.10 (rango de host)

Note que si la entrada de SPD tenía un valor permitido de comodín para la dirección de origen, entonces el valor del selector de la SAD podría ser un comodín (cualquier host). Para el caso (a) se puede utilizar la prohibición de compartir, aun entre paquetes que correspondan a la misma entrada de SPD.

Como se describe en la Sección 2.4.3, los selectores pueden incluir entradas "comodín" y por lo tanto selectores de dos entradas pueden superponerse. (Esto es análogo a la superposición que se presenta con ACLs o filtro de entradas en routers o firewalls de filtrado de paquetes). De esta manera se asegura consistencia y procesamiento predecible. Las entradas SPD DEBEN ser ordenadas y el SPD DEBE buscar siempre en el mismo orden, para que la primera entrada coincidente sea seleccionada consistentemente. Este requisito es necesario a los efectos del procesamiento del tráfico entre las entradas SPD que debe ser deterministas, pero no hay posibilidad de que las entradas canónicas de SPD soporten el uso de comodines para algunos selectores). Para más detalles sobre la correspondencia de paquetes entre entradas de la SPD ver la Sección 3.

Note que si ESP es especificado, la autentificación o encriptación pueden ser omitidas (pero no ambos). También DEBE ser posible configurar el valor de la SPD para que los algoritmos de autentificación o encriptación sean "NULL". Sin embargo, por lo menos uno de estos servicios DEBE ser seleccionado, es decir, NO DEBE ser posible configurar a los dos como "NULL".

La SPD puede utilizarse para asignar tráfico especifico de SAs o grupos de SA. Así puede funcionar como base de datos de referencia para la política de seguridad y como asignador de SA (o grupos de SA). (Para organizar el desvió y el descarte de las políticas citadas arriba, la SPD también puede proporcionar un medio para asociar tráfico de estas funciones, aunque no son, de por si, procesamiento IPsec). La forma en la cual funcionan las SPD es diferente para el tráfico de entrada que para el tráfico de salida y también puede ser diferente para implementaciones en un host, un security gateway, BITS o un BITW. En la Sección 3.1 y 3.2 se describe el uso de la SPD para el procesamiento de salida y de entrada respectivamente.

Para que una política de seguridad pueda requerir que más de una SA se aplique a un grupo especifico de tráfico, en un orden especifico, la política de entrada en la SPD debe preservar el orden requerido. Así debe ser posible para una implementación IPsec determinar que paquete de entrada o de salida debe ser completamente procesado por una secuencia de SAs. Conceptualmente para el procesamiento de salida, uno puede imaginar vínculos (hacia la SAD) desde una entrada SPD para la cual hay SAs activas y cada entrada consistirá de una SA o de una lista ordenada de SAs que se

corresponden a un grupo de SA. Cuando un paquete es coincidente con una entrada en la SPD y hay una SA (o grupo de SA) activas que se pueden usar para trasportar el tráfico, el procesamiento del paquete es controlado por la SA (o grupo de SA) de entrada de la lista. Para un paquete IPsec de entrada para el cual múltiples SAs IPsec se aplican, las operaciones de búsqueda se basan en la dirección de destino, protocolo IPsec, y el SPI, los cuales deberían identificar una SA.

La SPD se utiliza para controlar TODO el flujo de tráfico en IPsec incluyendo seguridad, manejo de claves (por ejemplo ISAKMP), tráfico entrante y saliente de entidades detrás de un security gateway. Esto significa que el tráfico ISAKMP se debe referenciar explícitamente en la SPD, sino será descartado. Note que un security gateway podría prohibir la encriptación de paquetes de varias formas, por ejemplo: con una entrada de DESCARTE en la SPD para paquetes con ESP o proporcionando claves en proxys. En este último caso el tráfico estaría internamente ruteado por el módulo manejador de claves en la security gateway.

2.4.2 Selectores de Paquetes

Una SA (o grupo de SA) puede tener mayor o menor modularidad dependiendo de los selectores usados para definir el grupo de tráfico para la SA. Por ejemplo todo el tráfico entre dos host puede ser trasportado por una SA, y ofrecer un conjunto uniforme de servicios de seguridad. Alternativamente, el tráfico entre un par de host puede ser extendido a múltiples SAs, dependiendo de la aplicación donde será usada (definido por el campos Siguiente Protocolo y el Puerto), cuando diferentes servicios de seguridad se ofrecen por diferentes SAs. Similarmente, todo el tráfico entre un par de security gateway puede ser trasportado por una SA, o una SA podría ser asignada para cada par de host que se comunican. Los siguientes parámetros del selector DEBEN ser soportados por la gestión de SA para facilitar el control de la granularidad de SA:

- Dirección IP de Destino (IPv4 o IPv6): Esta puede ser una dirección IP simple (unicast, anycast, broadcast (únicamente para IPv4), o de grupo multicast) o un rango de direcciones (valores altos y bajos (inclusive), dirección + máscara, o una dirección comodín). Estas ultimas tres se usan para soportar más de un destino que comparten la misma SA (por ejemplo detrás de un security gateway). Note que estos selectores se conceptualizan diferente del campo "Dirección IP de Destino" en la tupla <Dirección IP de destino, Protocolo IPsec, SPI> usada para identificar unívocamente a una SA. Cuando llega un paquete tuneliado, la SPI/Dirección de destino/Protocolo se usa para buscar la SA para ese paquete en la SAD. Esta dirección de destino viene desde la cabecera IP encapsulada. Una vez que el paquete se a procesado según el túnel SA y se a llegado a la salida del túnel, este selector busca la SPD de entrada. La SPD de entrada tiene un selector denominado, dirección de destino. Esta dirección IP de destino es la que esta en el interior (encapsulada) de la cabecera IP. En el caso de un paquete en modo trasporte, habrá una sola cabecera IP y esta ambigüedad no existirá. -- REQUERIDO por toda implementación --

- Dirección IP de Origen (IPv4 o IPv6): Esta puede ser una dirección IP simple (unicast, anycast, broadcast (únicamente para IPv4), o de grupo multicast) o un rango de direcciones (valores altos y bajos (inclusive), dirección + máscara, o una dirección comodín). Estas ultimas tres se usan para soportar más de un origen que comparten la misma SA (por ejemplo detrás de un security gateway o en un host multihomed) -- REQUERIDO por toda implementación --

- Nombre: Hay dos casos (note que esta forma de nombre es soportada en el IPsec DOI):
    1. Identificación de usuario (User ID)
        a. Una secuencia de nombre de usuario completamente cuantificado (DNS), por ejemplo: mozart@foo.bar.com
        b. Nombre característico X.500, por ejemplo: C = US, SP = MA, O = GTE Internetworking, CN = Pepe Lopez
    2. Nombre del sistema (host, security gateway, etc.)
        a. Un nombre completamente cuantificado DNS, por ejemplo: foo.bar.com
        b. Nombre característico X.500
        c. Nombre genérico X.500

  Nota: un uso de los valores de este selector es "oculto" ("OPAQUE")

  -- REQUERIDO para los siguientes casos (observe que el soporte para formas de nombres con excepción de la dirección no es requerido para las claves administradas manualmente en las SA):

    • Identificación de usuario
        - implementación en un host nativo
        - Implementación BITW y BITS activas como HOST cuando solamente hay un usuario
        - Implementación en un security gateway para el procesamiento de entrada
    • Nombres de sistemas: todas las implementaciones --

- Nivel de sensibilidad de los datos: (etiquetas IPSO/CIPSO) --REQUERIDO por los sistemas que proporcionan información de flujo de seguridad según la Sección 6, OPCIONAL para el resto de los sistemas--

- Protocolo de la Capa de Trasporte: obtenido del campo "Protocolo" en IPv4 o del campo "Siguiente Cabecera" en IPv6. Este puede ser un número de protocolo individual. Estos campos del paquete pueden no contener el Protocolo de Trasporte debido a la presencia de cabeceras de extensión, por ejemplo: una cabecera de enrutamiento, AH, ESP, fragmentación, opciones de destino, opciones salto-por-salto, etc. Note que el protocolo de trasporte puede no estar disponible en el caso de recibir un paquete con un encabezado ESP, en este caso un valor de "oculto" ("OPAQUE") DEBERÍA ser soportado. --REQUERIDO por toda implementación.--

  Observe que la localización del protocolo de trasporte, en un sistema que tiene cabeceras anidadas, se chequea el campo "protocolo" o "Siguiente Cabecera" hasta que encuentra y reconoce el protocolo de trasporte, o hasta que alcanza uno que no este en la lista de encabezados de extensión, o hasta que encuentra un encabezado ESP que haga al protocolo de trasporte "oculto".

- Puertos de Origen y Destino (por ejemplo, puertos TCP o UDP): Estos pueden ser valores de puertos TCP o UDP individuales o un puerto comodín. El uso del campo Siguiente Protocolo y el campo Puerto de Origen y/o Destino (en conjunto con el campo Dirección de Origen y/o Destino), como un selector de SA a veces se especifican como "claves orientadas a sesión". Note que el puerto origen y destino pueden no estar disponibles en el caso de recibir un paquete con un encabezado ESP, en este caso un valor de "OCULTO" DEBERÍA ser soportado.

La tabla siguiente resume la relación entre el valor "Siguiente Cabecera" en el paquete y la SPD y el valor obtenido del Puerto del Selector para la SPD y SAD.

Siguiente Cabecera en el Paquete	Protocolo de la Capa de Trasporte en la SPD	Valor Derivado del Campo Puerto del Selector en la SPD Y SAD.
ESP	ESP o cualquiera	cualquiera (es decir, no busca aquí)
no importa	cualquiera	cualquiera (es decir, no busca aquí)
fragmento valor especifico	valor especifico	no cualquiera (es decir, parte del paquete)
no fragmento valor especifico	valor especifico	campo del puerto del selector actual

Si el paquete ha sido fragmentado, la información del puerto puede no estar disponible en el fragmento actual. Si es así entonces se descarta el fragmento. Un ICMP PMTU debería ser enviado para el primer fragmento, el cual contendrá la información del puerto. --PUEDE ser soportado--

Note que en el caso de recibir un paquete con una cabecera ESP, por ejemplo en un security gateway o en una implementación BITW, el protocolo de la capa de trasporte, puerto de origen/destino y nombres (si están presente) pueden estar "ocultos", es decir inaccesibles debido a la encriptación o fragmentación.

El contexto de una implementación IPsec determinará que selector se debe utilizar. Por ejemplo una implementación integrada en un host dentro de la pila puede hacer uso de una interfaz socket. Cuando una nueva conexión es establecida la SPD puede ser consultada y una SA (o grupo de SA) unirá al socket. Así el tráfico enviado vía ese socket no necesitará operaciones de búsqueda adicionales en la SPD/SAD. En contraste implementaciones, BITS, BITW o security gateway necesitan mirar cada paquete y realizar operaciones de búsqueda en SPD/SAD basados en los selectores. Los valores permitidos para los campos del selector difieren debido al flujo de tráfico, la SA y la política de seguridad.

La siguiente tabla suministra los tipos de entradas que uno necesitan poder expresar en la SPD y en la SAD. (Nota: La entrada "carácter comodín" ("wild") o "comodín" ("wildcard") para direcciones de origen (src) y destino (dst) incluyen una máscara, un rango, etc.)

Campo	Valor del tráfico	Entrada SAD	Entrada SPD
Dirección de origen	Dirección IP única	Único,rango,wild	Único,rango,comodín
Dirección de destino	Dirección IP única	Único,rango,wild	Único,rango,comodín
Protocolo xpt*	Protocolo xpt	único,comodín	único,comodín
Puerto de origen*	Puerto de origen único	único,comodín	único,comodín
Puerto de destino*	Puerto de destino único	único,comodín	único,comodín
ID de usuario*	ID de usuario único	único,comodín	único,comodín
Segunda etiqueta	Valor único	único,comodín	único,comodín

* Las entradas SAD y SPD para estos campos podrían estar "OCULTAS" debido a que el valor del tráfico esta encriptado.

NOTA: En principio, uno puede tener selectores y/o valores de selectores en la SPD que no pueden ser negociados por una SA o grupos de SAs. Por ejemplo se puede incluir valores de selectores usados para seleccionar el tráfico para descartar o listas enumeradas que causen que SAs separadas puedan ser creadas para cada ítem de la lista. Por ahora esto se deja para versiones futuras de este documento y la lista de selectores requeridos y valores de selectores es el mismo para la SPD y la SAD. Sin embargo es aceptable tener una interfaz administrativa que soporte el uso de valores de selectores que no pueden ser negociados a condición de que ello no confunda al usuario en la creencia de que está creando una SA con estos valores del selector. Por ejemplo, la interfaz puede permitir que el usuario especifique una lista enumerada de valores pero daría lugar a la creación de una política separada y a una SA para cada ítem de la lista. Un vendedor puede soportar tal interfaz para hacérselo más fácil a sus clientes y para especificar políticas.

2.4.3 Base de Datos de Asociaciones de Seguridad (SAD)

Cada entrada de la SAD define los parámetros asociados con una SA y cada SA tiene una entrad en la SAD. Para el procesamiento saliente, las entradas están señaladas por entradas en la SPD. Note que si una entrada SPD actualmente no señala a una SA que es apropiada para el paquete, la implementación creará una SA (o grupo de SAs) y vínculos a las entradas SPD para la entrada SAD (ver Sección 3.1.1). Para el procesamiento de entrada, cada entrada en la SAD es indexada por, una dirección IP de destino, el tipo de protocolo IPsec y SPI. Los siguientes parámetros se asocian con cada entrada en la SAD (esta descripción especifica los ítem de datos mininos requeridos que debe soportar una SA en una implementación IPsec):

Para el procesamiento de entrada: Los siguientes campos del paquete se usan para buscar la SA en la SAD y son REQUERIDOS por toda implementación:

- Dirección IP de Destino, otras cabeceras: La dirección de destino IPv4 o IPv6.

- Protocolo IPsec: AH o ESP, usado como un índice para buscar la SA en esta base de datos. Especifica el protocolo IPsec aplicado al tráfico en esta SA

- SPI: es un valor de 32 bits que se usa para diferenciar SAs diferentes que tienen el mismo destino (la misma dirección IP de destino) y que usan el mismo protocolo IPsec.

Para cada uno de los selectores definidos, la entrada de la SA en la SAD DEBE contener el valor o los valores que fueron negociados para esa SA cuando fue creada. Para el emisor, estos valores se utilizan para decidir si una SA dada es apropiada para usarse con un paquete de salida. Esto es parte de la comprobación para saber si una SA existente puede ser utilizada. Para el receptor, este valor es utilizado para comprobar que los valores de los selectores en un paquete de entrada concuerdan con aquellos para la SA (y así indirectamente aquellos para la política coincidente). Para el receptor esta es parte de la verificación de que la SA fue la correcta para el paquete (véase la Sección 4 para las reglas para los paquetes ICMP). Estos campos pueden contener valores específicos, un rango, comodines, o "oculto" según lo descripto en la Sección 2.4.2. Note que para

una SA ESP, el algoritmo de encriptación o el algoritmo de autentificación podrían ser "NULL". Sin embargo no DEBEN ser ambos "NULL".

Los siguientes campos de la SAD son usados en el procesamiento IPsec:

- Contador de Número de Secuencia: Un valor de 32 bit usado para generar el campo Número de Secuencia de la cabecera AH o ESP. -- REQUERIDO por toda implementación, pero es usado solamente por el tráfico saliente --

- Desbordamiento del Contador de Secuencia: Una bandera (flan) que indica si el desbordamiento del Contador del Número de Secuencia debería generar un acontecimiento auditadle y prevenir la transmisión de los paquetes adicionales en la SA. -- REQUERIDO por toda implementación, pero es usado solamente para el tráfico saliente --

- Ventana de Anti-Replay: un contador de 32 bits y un asignador de bits (bit-map) (o equivalente) usado para determinar si un paquete AH o ESP es un paquete duplicado. -- REQUERIDO por toda implementación, pero es usado solamente para el tráfico entrante. Nota: Si el anti-replay ha sido desactivado por el receptor, por ejemplo en el caso de una clave SA manual, la ventana de anti-replay no se utilizará. --

- Algoritmo de Autenticación AH, claves, etc. -- REQUERIDO por implementaciones AH --

- Algoritmos de encriptación ESP, claves, modo IV, IV, etc. -- REQUERIDO por implementación de ESP --

- Algoritmo de autenticación ESP, Claves, etc. Si el servicio no se selecciona este campo será null -- REQUERIDO por implementaciones ESP --

- Tiempo de Vida de la SA: un intervalo de tiempo después del cual una SA debe ser reemplazada por una nueva SA (y un nuevo SPI) o debe ser terminada, más un indicador de cuando esta acción debe ocurrir. Esta puede ser expresada como un tiempo o como un contador de byte, o un uso simultáneo de ambos, el primer tiempo de vida que expire tiene prioridad. Una implementación completa DEBE soportar ambos tipos de tiempo de vida. Si se emplea el tiempo de vida y si IKE emplea certificados X.509 para el establecimiento de SA, el tiempo de vida de la SA se debe acotar (restringir) para los intervalos de validez de los certificados y el NextIssueDate (fecha inmediatamente importante) usada en la lista de renovación de certificados (CRLs Certificate Revocation List) del intercambio IKE para las SA. Tanto el que inicia como el que responde son responsables de la restricción del periodo del tiempo de vida de la SA en estos modos. -- REQUERIDO por toda implementación --

  Nota: los detalles de cómo manejar la renovación de las claves cuando expiran las SAs es un tema local. Sin embargo una aproximación razonable es:

  (a) Si se usa el contador de bytes, entonces la implementación DEBERÍA contar el número de bytes a los cuales se le aplica el algoritmo IPsec. Para ESP, este es el algoritmo de encriptación (incluyendo encriptación NULL) y para AH, este es el algoritmo de autentificación. Esto incluye los bytes de relleno, etc. Note que las implementaciones DEBERÍAN ser capaces de manejar los

contadores de los extremos de una SA para tener contadores sincronizados, por ejemplo, por la pérdida de paquetes o porque las implementaciones en los extremos de la SA no hacen las cosas de la misma manera.

(b) DEBERÍAN haber dos clases de tipos de tiempo de vida: un tiempo de vida suave que advierte a la implementación que es necesario iniciar una acción de reemplazo de la SA y un tiempo de vida duro cuando la SA termina.

(c) Si el paquete entero no puede ser entregado durante el tiempo de vida de la SA, el paquete DEBERÍA ser descartado.

- Modo del protocolo IPsec: túnel, trasporte o comodín. Indica el modo de AH o ESP que se aplica al tráfico en esa SA. Note que si este campo es un "comodín", el extremo emisor de la SA, de la aplicación especificará el modo para la implementación IPsec. Este uso del comodín permite que la misma SA sea usada para trasportar el tráfico en modo túnel o en modo trasporte en un paquete, por ejemplo por diferentes sockets. El receptor no necesita conocer el modo para procesar correctamente las cabeceras IPsec del paquete.
  --REQUERIDO en (salvo que se defina explícitamente en el contexto):
  - Las implementaciones en host debe soportar todos los modos.
  - Las implementaciones en gateway debe soportar el modo túnel. --

  NOTA: el uso de comodines para el modo del protocolo de una SA de entrada puede añadir complejidad a la situación en el receptor (solamente a host). Desde tales paquetes una SA puede ser entregada en modo túnel o trasporte, la seguridad de un paquete entrante podría depender en parte del modo que haya sido utilizado para entregarlo. Si por ende una aplicación se ocupa del modo de la SA de un paquete dado, entonces la aplicación necesitará un mecanismo para obtener información del modo aplicado.

- MTU de la Trayectoria: cualquier trayectoria MTU observada y incluyendo el envejecimiento de la variables. Ver Sección 4.1.2.4 -- REQUERIDO por toda implementación, pero es usado solamente para el tráfico saliente --

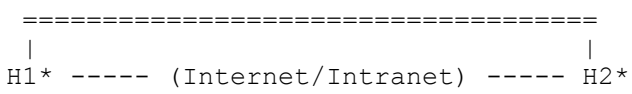
2.5 Combinaciones Básicas de Asociaciones de Seguridad

Esta sección describe cuatro ejemplos de combinaciones de SA que DEBEN ser completamente soportados por hosts IPsec o security gateways. Combinaciones adicionales de HA y/o ESP en modo trasporte y/o túnel PUEDEN ser soportadas a criterio del implementador. Una adecuada implementación DEBE ser capaz de generar estas cuatro combinaciones y un acuse de recibo de procesamiento, pero DEBERÍA ser posible recibir y procesar cualquier combinación. Los diagramas y textos debajo describen los casos básicos. La leyenda para los diagramas es:

```
    ==== = una o más SA (AH o ESP, en modo tunes o trasporte)
    ---- = Conectitividad (o también puede indicar un limite
administrativo)
    Hx   = host x
    SGx  = security gateway x
    X*   = X soporta IPsec
```

NOTA: Las SAs debajo pueden ser AH o ESP. El modo (túnel/trasporte) es determinado por la naturaleza de los extremos. Para SA host a host, el modo puede ser trasporte o túnel.

Caso 1: Proporciona seguridad extremo a extremo entre 2 host a través de Internet (o una Intranet).



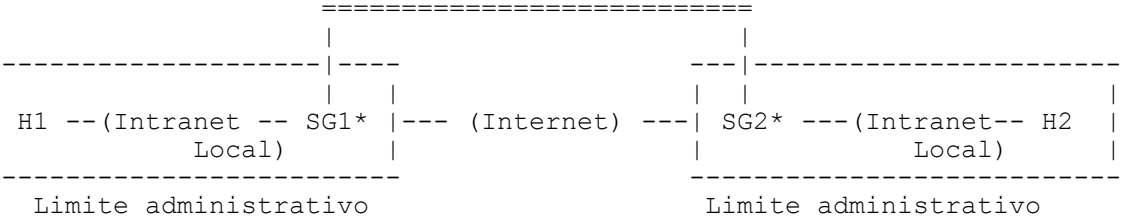
Note que tanto el modo trasporte como el modo túnel pueden ser seleccionados en los host. Las cabeceras de un paquete entre H1 y H2 podrían aparecer en alguna del las siguientes formas:

Modo Trasporte	Modo Túnel
1. [IP1][AH][upper]	4. [IP2][AH][IP1][upper]
2. [IP1][ESP][upper]	5. [IP2][ESP][IP1][upper]
3. [IP1][AH][ESP][upper]	

upper = cabecera de nivel supero/res

Note que no hay ninguna condición para el soporte de enrutamiento (de cabeceras), en modo trasporte, AH y ESP se pueden aplicar al paquete. En este caso, el procedimiento para establecer la SA DEBE aplicar primero ESP, y luego aplicar AH al paquete.

Caso 2: Este caso ilustra el soporte para una Red Privada Virtual simple VPN (Virtual Private Networks).

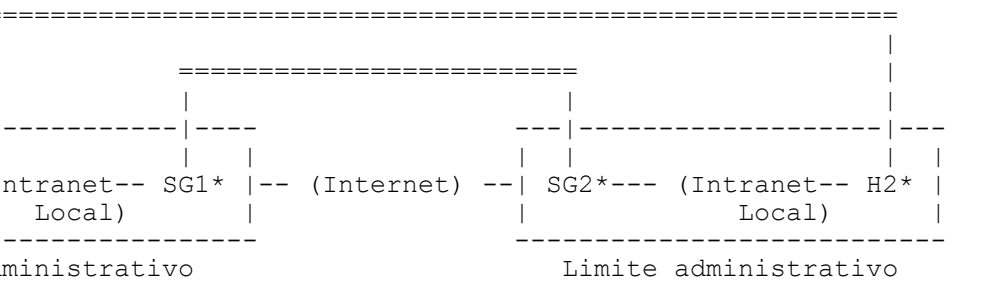


Solo el modo túnel es requerido. Las cabeceras en un paquete entre SG1 y SG2 podrían aparecer en cualquiera de las siguientes formas:

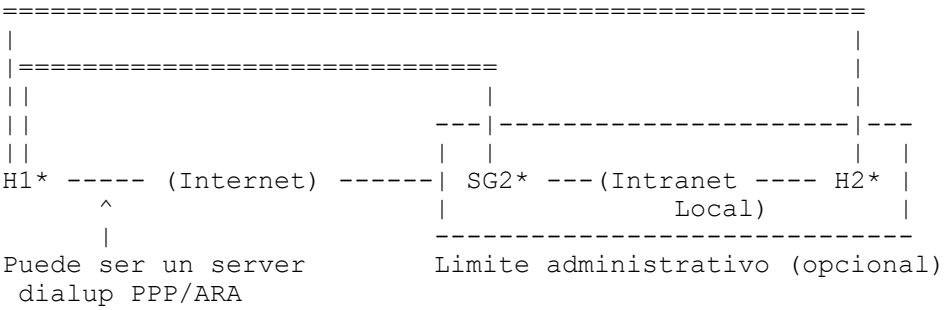
Modo Túnel
4. [IP2][AH][IP1][upper]
5. [IP2][ESP][IP1][upper]

upper = cabecera de nivel supero/res

Caso 3: En este caso se combina el caso 1 y el caso 2, agregando seguridad extremo a extremo entre el host emisor y el receptor. No impone ningún nuevo requisito en host o en la security gateway, con excepción de que el security gateway sea configurado para dejar pasar tráfico IPsec (incluyendo tráfico ISAKMP) a los host que están detrás de él.



Caso 4: Este caso cubre la situación en donde un host remoto (H1) utiliza Internet para alcanzar el firewall de una organización (SG2) y después para acceder a algún servidor o a otro host (H2). El host remoto podría ser un host móvil (H1) que marca hasta un servidor local PPP/ARA (no demostrado) en Internet y luego cruza la Internet hacia el firewall de la organización (SG2), etc. Los detalles para el soporte para este caso, (como H1 localiza a SG2, lo autentifica y verifica su autorización para representar H2) se discuten en la Sección 2.6.3 "Localizando a una Security Gateway"



Solo el modo túnel es requerido entre H1 y H2. Las opciones para la SA entre H1 y SG2 pueden ser una de las opciones del caso 2. Las opciones para la SA entre H1 y H2 pueden ser una de las opciones del caso 1.

Note que en este caso, el emisor DEBE aplicar la cabecera de trasporte antes que la cabecera del túnel. Por lo tanto la interfaz administrativa para la implementación de IPsec DEBE soportar la configuración de la SPD y la SAD para asegurar este orden de las cabeceras de aplicación en IPsec.

Según la nota de arriba, el soporte para combinaciones adicionales de AH y ESP es opcional. El uso de otras combinaciones de opciones puede afectar negativamente la interoperatibilidad.

2.6 Asociaciones de Seguridad y Gestión de Claves

IPsec exige dar soporte para SA tanto manuales como automatizadas y la gestión criptográfica de claves. El protocolo IPsec, AH y ESP son en gran medida independientes de las técnicas de gestión asociadas a las SA, aunque las técnicas empleadas afectan un poco el servicio de seguridad ofrecido por los protocolos. Por ejemplo los servicios opcionales de anti-replay disponibles en AH y ESP requieren una administración automática de SA. Por otra parte la granularidad empleada para la distribución de claves con IPsec determinará la granularidad de la autentificación proporcionada (véase también una discusión de este tema en la Sección 2.7). En general la autentificación del origen de los datos en AH y ESP esta limitada por el alcance en el uso del secreto del algoritmo de autentificación (o por un

protocolo de manejo de claves que cree tales secretos) que se comparten entre las múltiples fuentes posibles.

El siguiente texto describe los requisitos mínimos para ambos tipos de manejo de SAs.

2.6.1 Técnicas Manuales

La forma más simple de gestión es administrando en forma manual, en la cual una persona configura manualmente, cada sistema con material clave y administra los datos de las SA relevantes a las comunicaciones seguras con otros sistemas. Las técnicas manuales se usan en ambientes estáticos pequeños, pero la escalabilidad es mala. Por ejemplo una compañía puede crear una VPN usando IPsec en security gateway en varios sitos. Si el número de sitios es pequeño y como todos los sitios están bajo el mismo dominio administrativo, este es un contexto factible para las técnicas administrativas manuales. En este caso el security gateway puede proteger selectivamente el tráfico a y desde otros sitios dentro de la organización usando una configuración manual de claves, mientras que no proteja tráfico para otros destinos. También puede ser apropiado cuando solo se selecciona comunicaciones que necesitan ser seguras. Un argumento similar puede aplicarse al uso de IPsec entrante dentro de una organización para un número pequeño de host y/o gateway. Las técnicas administrativas manuales emplean a menudo configuraciones estáticas y clave simétricas, aunque también existen otras opciones.

2.6.2 Gestión de Claves y Asociaciones de Seguridad Automatizadas

El uso de implementaciones IPsec en general requiere de un estándar para Internet, escalable, automatizado y con protocolos para la administración de SAs. Este soporte es requerido para facilitar el uso de las características anti-replay de AH y ESP y para una adecuada creación de SA bajo demanda, por ejemplo, para el uso de claves orientadas a usuarios o a sesiones. (Un "recambio de claves" en una SA actual implica la creación de una nueva SA con un nuevo SPI, un proceso que generalmente implica el uso automatizado de protocolos de gestión de claves/SA).

El protocolo de gestión de claves automáticas por defecto que usa IPsec es IKE (véase el Capítulo 7, Capítulo 9, y el Capítulo 10) bajo el Domino de Interpretación (DOI) de IPsec , a través de ISAKMP (véase el Capítulo 8). Se PUEDEN emplear otros protocolos para el manejo automatizado de SA.

Cuando los protocolos de gestión de claves/SA se emplean, la salida de estos protocolos pueden ser empleados para crear múltiples claves, por ejemplo, para una SA ESP. Esto puede originarse debido a:

- Los algoritmos de encriptación usan múltiples claves (por ejemplo, Triple DES).
- Los algoritmos de autentificación usan múltiples claves.
- Se emplean tanto el algoritmo de encriptación como el algoritmo de autentificación.

El Sistema de Gestión de Claves puede proporcionar una cadena separada de bits para cada clave o puede generar una cadena de bits de la cual se extraigan todas las claves. Si una sola cadena de bits es proporcionada, hay que tener en cuenta que las partes del sistema que asignen la cadena de bit a las claves requeridas lo hagan en la misma forma en ambos extremos de

la SA. Para garantizar que las implementaciones IPsec en cada extremo de la SA usan los mismos bits para las mismas claves, independientemente de que parte del sistema divide la cadena de bits entre las claves individuales, la clave o claves encriptadas DEBEN ser extraídas de los primeros bits (los de más a la izquierda, de orden superior) y la clave de autentificación DEBE ser tomada de los bits restantes. El número de bit para cada clave es definido en el Capítulo 6 que especifica los algoritmos requeridos para AH y ESP. En el caso de claves de encriptación múltiple o claves de autentificación múltiple, la especificación del algoritmo debe especificar el orden en el cual beben ser seleccionados de una cadena de bits provistos para el algoritmo.

2.6.3 Localizando un Security Gateway

Esta sección discute asuntos referentes a como un host aprende sobre la existencia de security gateway relevantes y una vez que el host ha contactado a este security gateways, como sabe que este es el security gateway correcto. Los detalles de donde se almacena la información requerida es un tema local.

Considere una situación en la cual un host remoto (H1) es utilizado en Internet para acceder a un servidor o a otro host (H2) y hay un security gateway (SG2), por ejemplo, un firewall, a través del cual el tráfico de H1 se debe pasar. Un ejemplo de esta situación seria un host móvil que cruza la Internet al firewall de una organización (SG2). (Véase el caso 4 de la Sección 2.5) Esta situación plantea varios interrogantes:

1. ¿Cómo H1 sabe/aprende sobre la existencia del security gateway SG2?

2. ¿Cómo se autentifica SG2 y una vez que se haya autentificado SG2, como él confirma que SG2 ha sido autorizado para representar H2?

3. ¿cómo SG2 autentifica a H1 y verifica que H1 esté autorizado para entrar en contacto con H2?

4. ¿Como H1 sabe/aprende sobre los gateways de respaldo que proporcionan las trayectorias a H2?

Para tratar estos problemas, un host o un security gateway DEBE tener una interfaz administrativa que permita al usuario (o al administrador del sistema) configurar la dirección del security gateway para cualquier dirección de destino que se requiera para el uso. Esto incluye la capacidad de configurar:

- La información requerida para localizar y autentificar al security gateway y verificar su autorización de representar al host de destino.

- La información requerida para localizar y autentificar cualquier gateways de respaldo y verificar su autorización de representar al host de destino.

Se asume que la SPD también esta configurada con la información de la política que cubre cualquier otro requisito IPsec para la trayectoria del security gateway y del host de destino.

2.7 Asociaciones de Seguridad y Multicast

La orientación del receptor para la SA implica que, en el caso del tráfico unicast, el sistema de destino seleccionará normalmente el valor del SPI.

Teniendo el destino seleccionado en el valor del SPI, no hay ningún problema potencial para que la SA manualmente configurada este en conflicto con la SA automáticamente configurada (por ejemplo, vía un protocolo de gestión de claves) o para que la SA de múltiples fuentes este en conflicto con alguna otra. Para el tráfico multicast, hay sistemas de destino múltiples a través de grupos multicast. Algún sistema o persona se necesitará para coordinar todos los grupos multicast para seleccionar una SPI o SPIs en representación de cada grupo multicast y después comunicar la información IPsec de grupo a todos los miembros legítimos de ese grupo multicast a través de mecanismos no definidos en este documento.

Múltiples emisores en un grupo multicast DEBERÍAN utilizar una sola SA (y por lo tanto un solo SPI) para todo el tráfico en ese grupo cuando se emplea un algoritmo de encriptación o de autentificación de clave simétrico. En tales circunstancias el receptor, sabe que el mensaje vino de un sistema que poseía la clave para ese grupo multicast pero el receptor generalmente no podrá autentificar que sistema envió el tráfico multicast.

Para los grupos multicast que tienen relativamente pocos miembros, la distribución de claves manuales o el uso múltiple de, algoritmos de distribución de claves unicast existentes tales como Diffie-Hellman modificado parecen ser factibles. Un ejemplo del trabajo actual en esta área es el Protocolo de Gestión de Claves para Grupos GKMP (Group Key Management Protocol) [HM97].

3. **Procesamiento del Tráfico IP**

Según lo mencionado en la Sección 2.4.1 La SPD, se debe consultar durante todo el procesamiento del tráfico (de entrada y de salida), incluyendo el tráfico no IPsec. Si no se encuentra ninguna política en el SPD que corresponda con el paquete (para el tráfico de entrada o de salida), el paquete debe ser desechado.

NOTA: Todos los algoritmos criptográficos usados en IPsec guardan su entrada en orden canónico de byte de red (véase el Apéndice del RFC 791) y generan su salida en orden canónico de byte de red. Los paquetes IP también se transmiten en orden de byte de red.

3.1 Procesamiento del Tráfico IP Saliente

3.1.1 Seleccionando y Usando una SA o Grupo de SAs

En un security gateway o una implementación BITW (y en muchas implementación BITS), cada paquete saliente se compara con la SPD para determinar que procesamiento se requiere para el paquete. Si el paquete va a ser descartado, esto es un evento auditadle. Si al tráfico se le esta permitido evitar el procesamiento IPsec, el paquete continúa con el procesamiento "normal" para las condiciones en la cual el procesamiento IPsec está ocurriendo. Si se requiere el procesamiento IPsec, el paquete es asociado con una SA existente (o grupo de SA), o una nueva SA (o grupo de SA) se crea para el paquete. Puesto que un selector de paquetes pueden coincidir con múltiples políticas o con múltiples SAs existentes y puesto que la SPD esta ordenada, pero la SAD no lo esta, IPsec debe:

1. Hacer Corresponder los campos del selector de paquetes con las políticas de salida en el SPD para localizar la primera política apropiada, la cual apuntará a cero o más grupos de SAs en la SAD.

2. Hacer Corresponder los campos del selector de paquetes con ésos grupos

de SA encontrados en (1) para localizar el primer grupo de SA que coincida. Si no se encontró ninguna SAs o ninguna que coincida, se creará un grupo de SA apropiados y vínculos de entrada en la SPD hacia la entrada de la SAD. Si no se encuentra entrada para la gestión de claves, deseche el paquete.

3. Utilizar el grupo de SA encontradas/creadas en (2) para realizar el procesamiento IPsec requerido, por ejemplo, autentificar y encriptar.

En un host basado en sockets que implementación IPsec, la SPD será consultada siempre que se cree un nuevo socket, para determinar, si existe, un procesamiento IPsec que será aplicado al tráfico que fluirá en ese socket.

NOTA: Una adecuada implementación no DEBE permitir una SA ESP que emplee encriptación NULL y un algoritmo de autentificación NULL. Una tentativa de negociar tal SA es un acontecimiento auditable.

3.1.2 Construcción de Cabeceras para el Modo Túnel

Esta sección describe como manipular las cabeceras internas y externas IP, las cabeceras de extensión, y las opciones para túneles AH y ESP. Esto incluye cómo construir la cabecera (externa) de encapsulado IP, cómo manejar los campos en la cabecera IP interna, y que acciones deben ser tomadas. La idea general esta modelada en el RFC 2003, "IP con Encapsulación IP":

- La Dirección de Origen y la Dirección de Destino en la cabecera IP externa identifican los "extremos" del túnel (al encapsulador y desencapsulador). La Dirección de Origen y la Dirección de Destino en la cabecera IP interna identifican al verdadero emisor y receptor del datagrama, (respectivamente para ese túnel), (véase la nota del punto 3 de la Sección 3.1.2.1 para más detalles de la dirección IP de origen encapsulada.)

- La cabecera IP interior no se puede modificar excepto para decrementar el TTL según se observa debajo, y permanece inmutable durante la entrega hacia el otro extremo del túnel.

- Ningún cambio en las cabeceras opcionales o en las cabeceras de extensión IP internas ocurre durante la entrega del datagrama encapsulado a través del túnel.

- De ser necesario, otras cabeceras de protocolos tales como la cabecera de Autentificación se pueden insertar entre la cabecera IP externa y la cabecera IP interna.

Las tablas de las subsiguientes secciones muestran el manejo de los diferentes campos de la cabecera de IPv4 e IPv6, los de opciones, y los de las cabeceras de extensión (en IPv6).

Nota: En la siguientes tablas el termino "la genera" significa que el valor en el campo exterior se construye independientemente del valor del campo interno.

3.1.2.1 Construcción de Cabeceras en Modo Túnel para IPv4

	Como otras Hdr se relacionan en el interior de la Hdr	
Campos de la cabecera IPv4	Hdr externa en el Encapsulador	Hdr interna en el Desencapsulador
Versión	4 (1)	no cambia
Longitud de la Hdr	La genera	no cambia
TOS	Lo copia de la Hdr interna (5)	no cambia
Longitud total	La genera	no cambia
ID	La genera	no cambia
Banderas (DF,MF)	La genera, DF (4)	no cambia
Offset de fragmento	La genera	no cambia
TTL	La genera (2)	Lo decrementa (2)
Protocolo	AH, ESP, Hdr de enrutamiento	no cambia
Checksum	La genera	La genera (2)
Dirección origen	La genera (3)	no cambia
Dirección destino	La genera (3)	no cambia
Opciones	Nuca copiar	no cambia

Hdr=Cabecera

1. La versión IP en la cabecera encapsulada puede ser diferente que el valor de la cabecera interna.

2. TTL en el interior de la cabecera es decrementado por el encapsulador antes de reenviarlo y por el desencapsulador en caso de reenviar el paquete. (El checksum cambia cuando el TTL cambia.)

   NOTA: El decremento del TTL es una de las acciones usuales que tienen lugar cuando se reenvía un paquete. Los paquetes que se originan en el mismo nodo que los encapsula no tienen su TTL decrementado, pues el nodo que envía está originando el paquete en lugar de reenviarlo.

3. La dirección de origen y destino dependen de la SA, la cual se usa para determinar la dirección de destino, la cual determinará a su vez que dirección de origen (de interfaz de red) se usará para reenviar el paquete.

   NOTA: En principio, la dirección IP de origen encapsulada puede ser alguna de las direcciones de la interfaze del encapsulador o incluso una dirección diferente de alguna de las direcciones IP del encapsulador, (por ejemplo, si se actuá como un nodo NAT) siempre que la dirección sea accesible por el encapsulador desde el entorno dentro del cual el paquete es enviado. Esto no causa problema porque actualmente IPsec no tiene ningún requisito de procesamiento de ENTRADA que involucre la Dirección de Origen de la cabecera IP encapsulada. Por lo tanto mientras que los extremos receptores del túnel examinan la Dirección de Destino en la cabecera IP encapsulada, este solo considera la Dirección de Origen en la cabecera IP interna (encapsulada).

4. La configuración determinará si se copia en el encabezado interno (sólo en IPv4), o si coloca un 1 o un cero en el bit DF.

5. Si la cabecera interna es IPv4 (Protocolo = 4), copia el campo TOS. Si la cabecera interna es IPv6 (Protocolo = 41) asigna el campo Class (Clase de Tráfico) al campo TOS

3.1.2.2 Construcción de Cabeceras en Modo Túnel para IPv6

	Como otras Hdr se relacionan en el interior de la Hdr	
Campos de la cabeceras IPv6	Hdr externa en el Encapsulador	Hdr interna en el Desencapsulador
Versión	6 (1)	no cambia
Clase de Tráfico	La copia o configura (6)	no cambia
Tipo de Flujo	La copia o configura	no cambia
Longitud de la Carga Útil	La genera	no cambia
Cabecera Siguiente	AH, ESP, Hdr de enrutamiento	no cambia
Limite de Saltos	La genera (2)	Lo decrementa (2)
Dirección Origen	La genera (3)	no cambia
Dirección de Destino	La genera (3)	no cambia
Cabeceras de Extensión	Nuca copiar	no cambia

Hdr=Cabecera

Ver la Sección 3.1.2 para las notas de 1 al 5 que se indican allí

6. Si la cabecera interna es IPv6 (Cabecera Siguiente = 41), se copia el campo Clase de Tráfico (Class). Si la cabecera interna es IPv4 (Cabecera Siguiente = 4), asigna el campo TOS (tipo de servicio) al campo Clase de Tráfico (Class).

3.2 Procesamiento del Tráfico IP Entrante

Antes de ejecutar el procesamiento de AH o ESP, cualquier fragmento IP es reensamblado. Cada datagrama IP de entrada al cual se le aplicó el procesamiento IPsec es identificado por los valores característicos de AH o de ESP en el campo Protocolo Siguiente (o por la cabecera de extensión AH o ESP en IPv6).

Nota: la Sección 10 contiene un código simple para chequear la máscara de bits (bismask) para una ventana de 32 paquetes que puede ser usado para implementar el servicio de anti-replay.

3.2.1 Seleccionando y Usando una SA o Grupo de SAs

Asociar el datagrama IP a la SA apropiada es simple debido a la presencia del SPI en la cabecera de AH o de ESP. Observe que las comprobaciones que realiza el selector se hacen en las cabeceras internas, no en las cabeceras externas (las del túnel). Los pasos a seguir son:

1. Usar la dirección de destino de los paquetes (cabecera externa IP), protocolo IPsec, y el SPI para buscar la SA en la SAD. Si la búsqueda de la SA falla, se desecha el paquete y se registra e informa el error.

2. Utilice la SA encontrada en (1) para realizar el procesamiento IPsec, por ejemplo, autentifique y desencripte. Este paso incluye hacer

corresponder a los selectores del paquete (de estar tuneliado el paquete debe usar la cabecera interna) con los selectores de la SA. La política local determinará la especificidad de los selectores de la SA (valor único, lista, rango, comodín). En general, la dirección de destino del paquete DEBE coincidir con el valor del selector SA. Sin embargo, un paquete ICMP recibido en una SA en modo túnel puede tener una dirección de origen diferente que la que se tiene en la SA y tales paquetes se deben permitir como excepción en esta comprobación. Para un paquete ICMP, los selectores incluyen el paquete problemático (la dirección de origen y destino y puertos; los cuales se deberían intercambiar) que debería ser chequeado con el selector de la SA. Note que algunos o todos estos selectores pueden ser inaccesibles debido a limitaciones en alguno de los bits del paquete problemático que el paquete ICMP permite llevar, o debido a la encriptación. Ver Sección 4.

Realice (1) y (2) para cada cabecera IPsec hasta que una Cabecera de Protocolo de Trasporte o una cabecera IP que NO sea parte de este sistema sea encontrada. Mantener un registro de que SAs han sido usadas y el orden en que se usaron.

3. Encuentre una política entrante en la SPD que coincida con el paquete. Esto puede hacerse, por ejemplo, a través de punteros invertidos de las SAs hacia la SPD o haciendo corresponder a los selectores del paquete (cabecera interna si esta tunelizada) con las políticas de entrada en la SPD.

4. Comprobar si el procesamiento IPsec requerido ha sido aplicado, es decir, verificar que las SAs encontradas en (1) y (2) concuerdan con el tipo y orden de SAs requeridas por la política encontrada en (3).

   NOTA: La política adecuada que "concuerda" no necesariamente es la primer política de entrada encontrada. Si la comprobación del paso (4) falla, los pasos (3) y (4) se repiten hasta que todas las políticas de entrada hayan sido comprobadas o hasta que la comprobación sea exitosa.

Después de haber realizado esos pasos, pase el paquete resultante a la Capa de Transporte o reenvié el paquete. Observe que cualquier cabecera IPsec procesada en estos pasos puede haber sido retirada, a excepción de esa información, es decir, qué SAs fueron utilizadas y de que forma se usaron, puede ser necesaria para el procesamiento subsiguiente de IPsec o del firewall.

Observe que en el caso de un security gateway, si el reenvío causa un paquete saliente vía una interfaz IPsec habilitada, entonces el proceso adicional de IPsec puede ser aplicado.

3.2.2 Manejo de HA y ESP en Túneles

El manejo de las cabeceras IP internas y externas, de las cabeceras de extensión, y de las opciones para túneles AH y ESP deberían ser realizadas según lo descripto en las tablas de la Sección 3.1.

4. **Procesamiento ICMP (Relativo a IPsec)**

El enfoque de esta sección es la manipulación de mensajes ICMP de errores. Otro tráfico ICMP, por ejemplo, Echo/Reply deberían ser tratados como tráfico normal y pueden ser protegidos de extremo a extremo usando SAs

normalmente.

Un mensaje de error ICMP protegido por ESP o AH y generado por un router DEBERÍA ser procesado y enviado por una SA en modo túnel. La política local determina si está o no subordinado a la comprobación de la dirección de origen por el router en el extremo destinatario del túnel. Note que si el router en el extremo iniciador del túnel esta reenviando un mensaje de error ICMP para otro router, la comprobación de la dirección de origen podría fallar. Un mensaje ICMP protegido por AH o ESP y generado por un router NO DEBE ser enviado en una SA en modo trasporte (a menos que la SA haya sido establecida para el router actuando como un host, por ejemplo una conexión telnet usada para gestionar un router). Un mensaje ICMP generado por un host DEBERÍA realizar comprobaciones entre los selectores de direcciones IP de origen vinculados a la SA dentro de la cual el mensaje llega. Note que por más que el origen de un mensaje ICMP de error sea autentificado, la cabecera IP reenviada podría no ser válida. Por consiguiente los valores del selector en la cabecera IP DEBERÍAN ser comprobados para asegurar que son consistentes con los selectores de la SA por la cual el mensaje ICMP fue recibido.

La tabla de la Sección 11 caracteriza los mensajes ICMP como generados por el host, generados por el router, ambos, desconocidos/no asignados. Los mensajes ICMP que no están dentro de estas dos últimas categorías deberían ser manipulados según lo determine la política del receptor.

Un mensaje ICMP no protegido por AH o ESP sin autentificado, su procesamiento y/o envío puede resultar en denegación de servicio. Esto sugiere que, en general sería aconsejable ignorar tales mensajes. Sin embargo, se espera que muchos router (versus security gateways) no implementarán IPsec para transportar el tráfico y así estricta adhesión a esta regla causaría que muchos mensajes ICMP sean descartados. El resultado es que algunas funciones criticas de IP podrían ser perdidas, por ejemplo, redirección y procesamiento PMTU. De esta manera se DEBE configurar una implementación IPsec para aceptar o rechazar tráfico ICMP (de router) según la política de seguridad local.

Lo que queda de esta sección habla de cómo se DEBE realizar procesamiento PMTU en hosts y en security gateways. Esta sección también trata el procesamiento de mensajes ICMP PMTU autentificados y no autentificados. Sin embargo, como se dijo anteriormente, los mensajes ICMP no autentificados PUEDEN ser descartados según la política local.

4.1 Procesamiento PMTU/DF

4.1.1 Bit DF

Cuando un sistema (host o gateway) agrega una cabecera de encapsulación (túnel ESP o túnel AH), DEBE soportar la opción de copiar el bit DF del paquete original a la cabecera de encapsulación (y procesar los mensajes ICMP). Esto significa que DEBE ser posible configurar un tratamiento del sistema del bit DF (fijar, limpiar, copiar la cabecera encapsulada) para cada interfaz. (Ver Sección 9).

4.1.2 Descubrimiento de la Ruta MTU (PMTU)

Esta sección trata el manejo de IPsec para mensajes de Descubrimiento de la ruta MTU. ICMP PMTU es usado aquí para referirse a un mensaje ICMP para:

IPv4 (ICMPv4-RFC 792):
- Tipo = 3 (Destino inalcanzable).
- Código = 4 (Fragmentación necesaria y el DF esta establecido).
- El siguiente salto MTU dentro de 16 bits de menor orden de la segunda palabra de la cabecera ICMP (etiquetado "no usado" en el RFC 792), con los 16 bit de mayor orden puestos en cero.

IPv6 (ICMPv6-RFC 2463):
- Tipo = 2 (Paquete demasiado grande).
- Código = 0 (Fragmentación necesaria).
- Siguiente salto MTU en el campo MTU de 32 bit del mensaje ICMP6.

4.1.2.1 Transmisión del PMTU

La cantidad de información retornada con un mensaje ICMP PMTU (IPv4 o IPv6) es limitada y esto afecta a los selectores que están disponibles para usarse en la futura transmisión de información PMTU. (Vea la Sección 9 para una discusión más detallada de este tema.)

- Un mensaje PMTU de 64 bits de la cabecera IPsec: si el mensaje ICMP PMTU contiene solamente 64 bits de la cabecera IPsec (mínimo para IPv4) una security gateway DEBE soportar las siguientes opciones para las SPI/SA:

  a. Si el host originador puede ser determinado (o los host de origen posibles están limitados a un número manejable), enviar la información PMTU a todos los host originadores posibles.

  b. Si el host originador no puede ser determinado, almacene el PMTU con la SA y espere a que el siguiente paquete llegue del host originador para la SA relevante. Si el paquete o los paquete son más grandes que el PMTU, descarte los paquetes, y cree un mensaje ICMP PMTU con un nuevo paquete y el PMTU actualizado, y envié el mensaje ICMP sobre el problema al host originador. Guarde la información PMTU para cualquier mensaje que pueda llegar posteriormente. (ver la Sección 4.1.2.4, " Envejecimiento de la PMTU").

- Mensaje PMTU con más de 64 bits de la cabecera IPsec: Si el mensaje ICMP contiene más información del paquete original, entonces, puede haber suficiente información no oculta para determinar inmediatamente que host trasmitió el mensaje ICMP/PMTU y para proporcionar un sistema con 5 campos (dirección de origen, dirección de destino, puerto de origen, puerto de destino, protocolo de trasporte) necesarios para determinar donde almacenar/actualizar el PMTU. Bajo tales circunstancias, una security gateway DEBE generar inmediatamente un mensaje ICMP PMTU al recibir un ICMP PMTU de un camino más lejano.

- La Distribución del PMTU para la Capa de Trasporte: El mecanismo del host para conseguir el PMTU actualizado para la capa de trasporte no tiene cambios, según lo especificado en el RFC 1191 (Descubrimiento de la ruta MTU).

4.1.2.2 Cálculo del PMTU

El cálculo de PMTU para un ICMP PMTU DEBE tener en cuenta el agregado de cualquier cabecera IPsec: trasporte AH, trasporte ESP, trasporte AH/ESP, túnel ESP, túnel AH. (Vea la Sección 9 para una discusión de los asuntos

relacionados con la implementación).

Nota: el agregado de la cabecera de IPsec podría resultar en un PMTU (visto por el host o aplicación) que es inaceptablemente pequeño. Para evitar este problema la implementación puede establecer un umbral bajo el cual no se reportará un PMTU reducido. En tales casos, la implementación aplicaría IPsec y después fragmentaría el paquete resultante de acuerdo al PMTU. Esto proporcionará un uso más eficiente del ancho de banda disponible.

4.1.2.3 Granularidad del Procesamiento de PMTU

En host, la granularidad con la cual el procesamiento ICMP PMTU puede ser realizado se diferencia dependiendo de la situación de la implementación. Mirando a un host, hay 3 situaciones que son de interés con respecto a cuestiones PMTU (ver la Sección 9 para más detalles adicionales de este tema):

a. Integración de IPsec en la implementación nativa IP.

b. Implementación BITS (Bump-in-the-stack), donde IPsec esta implementado por "debajo" de una implementación existente de una pila de protocolo TCP/IP, entre el IP nativo y los drivers de red.

c. No hay implementación IPsec: Este caso es incluido porque es relevante en el caso donde una security gateway esta enviando información PMTU devuelta a un host.

Solamente en el caso (a) los datos PMTU pueden ser mantenidos en la misma granularidad que las asociaciones de comunicación. En (b) y en (c), la capa IP solo podrá mantener los datos PMTU a la granularidad de la direcciones de origen y destino IP (y opcionalmente TOS), como se describe en el RFC 1191. Esto es una diferencia importante porque más de una asociación de comunicación puede asignarse a las mismas direcciones de origen y destino IP, y cada asociación de comunicación puede tener un diferente costo computacional en la cabecera IPsec (por ejemplo, debido al uso de diferentes trasformaciones o diferentes algoritmos).

La implementación del calculo PMTU y el soporte de PMTUs en la granularidad de asociaciones de comunicaciones es un tema local. Sin embargo, una implementación IPsec basada en socket en un host DEBERÍA mantener la información para cada socket. Los sistemas BITS DEBEN pasar un ICMP PMTU al host de implementación IP, después de adaptarla para cualquier cabecera IPsec con costos computacionales adicionales para esos sistemas. El cálculo de los costos computacionales DEBERÍA ser determinado por la inspección del SPI y cualquier otro selector de información presente en un mensaje ICMP PMTU devuelto.

4.1.2.4 Envejecimiento de la PMTU

Todos los sistemas (host o gateway) que implementan IPsec y mantienen información de la PMTU, la PMTU asociada a una SA (trasporte o túnel) DEBE "envejecer" y algún mecanismo se debe poner en funcionamiento para actualizar la PMTU dentro de un tiempo razonable, especialmente para descubrir si el PMTU es mas pequeño de lo que necesita ser. Una PMTU tiene que permanecer activa por un lapso de tiempo suficiente para que un paquete llegue de un extremo de origen de la SA de un sistema al otro extremo de la SA y propague un mensaje de error ICMP si la PMTU actual es demasiado grande. Observe que si hay túneles anidados, múltiples paquetes y los tiempos de viaje de ida y vuelta podrían ser requeridos para conseguir que

un mensaje ICMP vuelva a un encapsulador o host de origen.

Los sistemas DEBERÍAN usar la metodología descripta en documento "Descubrimiento de la Ruta MTU" (RFC 1191, Sección 6.3), el cual sugiere el receteo periódico del PMTU para el vínculo de datos del primer salto MTU y dejar que los procesos de descubrimientos normales de PMTU actualicen la PMTU cuando sea necesario. Este período DEBERÍA ser configurable.

5. **Auditoría**

No todos los sistemas que implementan IPsec implementarán auditoría. Gran parte de la granularidad de la auditoría es de incumbencia local. No obstante varios eventos auditadles están identificados en las especificaciones de AH y ESP y para cada uno de estos eventos un conjunto mínimo de información DEBERÍA ser incluido en un registro de auditoría, si es definido. Información adicional también PUEDE ser incluida en el registro de auditoría para cada uno de estos eventos, y eventos adicionales, no explícitamente tratados en esta especificación, también PUEDEN registrarse en el registro de auditoría. No existe requisitos para el receptor de trasmitir ningún mensaje al trasmisor pretendido en respuesta a la detección de un evento auditadle, debido al potencial de inducir denegación de servicio a través de tal acción.

6. **Uso de la Información de Flujo de Seguridad en Soportes Informáticos**

La información de varios niveles de sensibilidad puede ser trasportada en una sola red. Las etiquetas de información (por ejemplo, no clasificada, propiedad de la compañía, secreto) [DoD85], [DoD87] son frecuentemente empleadas para distinguir tal información. El uso de etiquetas facilita la clasificación de información, y el soporte a los modelos de seguridad de flujo de información, por ejemplo el modelo Bell-LaPadula [BL73]. Tales modelos, y la tecnología para el soporte correspondiente, están diseñados para prevenir el flujo no autorizado de información sensible, aun frente a ataques de tipo "Caballo de trolla" (Trojan Horse). Convencionalmente los mecanismos de control de acceso (DAC), por ejemplo, mecanismos basados en listas de control de acceso, generalmente no son suficientes para soportar tales políticas y por lo tanto las instalaciones tales como el SPD no son suficientes en tales ambientes.

En el contexto militar la tecnología que soporta tales modelos se denomina "Múltiples Niveles de Seguridad o Seguridad Multinivel (MLS)". Las computadoras y las redes se designan a menudo como "Seguridad de múltiples niveles" si soportan la separación de datos etiquetados junto con políticas de seguridad del flujo de información. Aunque tal tecnología es más ampliamente aplicable que solamente aplicaciones militares, este documento usa el acrónimo "MLS" para señalar la tecnología de acuerdo con bastante de la literatura actual.

Los mecanismos de IPsec pueden fácilmente soportar conexiones de redes MLS. Las conexiones de redes MLS requieren el uso de fuertes Controles de Acceso Obligatorios (MAC), que los usuarios no privilegiados o los procesos no privilegiados son incapaces de controlar o violar. Esta sección concierne solamente al uso de mecanismos de seguridad IP en habientes MLS (Política de seguridad de flujo de información). Nada en esta sección se aplica a los sistemas que no proporcionan MLS.

Según lo utilizado en esta sección, "la información sensible" puede incluir implementaciones definidas en niveles jerárquicos, categorías, y/o divulgación de información.

AH puede ser usado para proporcionar autentificación fuerte como apoyo a las decisiones de control de acceso obligatorios en ambientes MLS. Si la información de sensibilidad IP explícita se utiliza (por ejemplo IPSO [Ken91]) y la confidencialidad no se considera necesaria dentro de un ambiente operacional particular, AH puede ser usado para autentificar el enlace entre las etiquetas de sensibilidad en la cabecera IP y la carga IP (incluyendo datos del usuario). Esto es un avance significativo de las redes etiquetadas de IPv4 donde se confía en la información de la sensibilidad aunque no hay enlaces de autentificación o criptográficos de información en la cabecera IP y los datos del usuario. Las redes IPv4 pueden o no usar etiquetamiento explícito. Envés de usar la información explicita de la sensibilidad, IPv6 normalmente usa la información implícita de la sensibilidad que es parte de la SA IPsec pero no trasmitida con cada paquete. Toda la información explícita de la sensibilidad IP DEBE ser autentificada usando ESP, AH, o ambos.

La encriptación es útil y puede ser deseable aun cuando todos los host están dentro de un ambiente protegido, por ejemplo, detrás de un firewall o que no tengan conexión externa. ESP puede ser usado, conjuntamente con adecuados algoritmos de gestión de claves y de encriptación, soportando DAC y MAC. (La elección de los algoritmos de encriptación y autentificación y el nivel de aseguramiento de una implementación IPsec determinarán los ambientes en los que una implementación puede ser considerada suficiente para satisfacer los requerimientos MLS.) La administración de claves puede hacer uso de la información de la sensibilidad para proporcionar MAC. Las implementaciones IPsec en los sistemas que demandan proporcionar MLS DEBERÍAN ser capaces de usar IPsec para proporcionar MAC a comunicaciones basadas en IP.

6.1 Relación Entre SA y la Sensibilidad de los Datos

La Carga de Seguridad Encapsulada y la Cabecera de Autentificación se pueden combinar con apropiadas políticas de Asociaciones de Seguridad para proporcionar una red con múltiples niveles de seguridad. En cada caso cada SA (o grupo de SA) es normalmente usada para una única instancia de información de sensibilidad. Por ejemplo, "PROPRIETARY - Internet Engineering" debe estar asociaba con una SA diferente (o grupo de SA) que la "PROPRIETARY - Finance".

6.2 Control de la Consistencia de Sensibilidad

Una implementación de Seguridad Multinivel (en host y en router) PUEDE asociar la información de sensibilidad, o un rango de información de sensibilidad con una interfaz, o con una dirección IP configurada con su prefijo asociado (este último se refiere algunas veces como una interfaz lógica o como un alias de interfases). Si tales propiedades existen la implementación DEBERÍA comparar la información de sensibilidad asociada con el paquete, con la información de la sensibilidad asociada a la interfaz o a la dirección/prefijo desde la cual el paquete llegó, o a través de la cual el paquete saldrá. Esta comprobación verificará que la sensibilidad corresponda, o que la sensibilidad del paquete está dentro del rango de interfases o dirección/prefijo.

Esta comprobación DEBERÍA ser realizada en el procesamiento entrante y saliente.

6.3 Atributos Adicionales de la Seguridad Multinivel (MLS) para las SADs

La Sección 2.4 discute dos bases de datos de Asociaciones de Seguridad (la Base de datos de Políticas de Seguridad (SPD) y la Base de Datos de Asociaciones de Seguridad (SAD)) y los selectores de la política asociada y los atributos de las SAs. La red MLS introduce un selector/atributo adicional:

- Información de sensibilidad

La Información de sensibilidad ayuda a seleccionar los algoritmos apropiados y las fuerzas de claves, de modo que el tráfico obtenga un nivel de protección apropiado a su importancia o sensibilidad como se describe en la Sección 6.1. La sintaxis exacta de la información de sensibilidad depende de la implementación.

6.4 Etapas Adicionales del Procesamiento de Entrada para Redes de Seguridad Multinivel

Después que un paquete entrante a pasado por el procesamiento IPsec, una implementación MLS DEBERÍA primero controlar la sensibilidad del paquete (según lo definido por la SA (o grupo de SA) usada para el paquete) con la interfaz o direccionamiento/prefijo según se describe en la Sección 6.2 antes de enviar el datagrama a un protocolo de capa superior o reenviarlo.

El sistema MLS DEBE retener el enlace de los datos recibidos en un paquete protegido por IPsec y la información de la sensibilidad en una SA o en SAs usadas para el procesamiento, por lo tanto las decisiones apropiadas de la política pueden ser realizadas cuando se envía el datagrama a una aplicación o es reenviado. Estas formas de mantener este enlace son específicos de la implementación.

6.5 Etapas Adicionales del Procesamiento de Salida para Redes de Seguridad Multinivel

Una implementación MLS IPsec DEBE realizar dos controles adicionales a parte de los pasos normales detallados en la Sección 3.1.1. Al consultar la SPD o la SAD para encontrar una SA saliente, la implementación MLS DEBE usar la sensibilidad de los datos para seleccionar una apropiada SA (o grupo de SA) saliente. El segundo control se origina antes de enviar el paquete a su destino, y es el control de la consistencia de la sensibilidad descripta en la Sección 6.2.

6.6 Procesamiento Adicional para la Seguridad Multinivel para Security Gateways

Una security gateway con Seguridad multinivel DEBE seguir las reglas de procesamiento entrante y saliente mencionadas anteriormente así como también realizar un procesamiento adicional especifico para la protección intermedia de los paquetes en un ambiente MLS.

Una security gateway PUEDE actuar como proxy saliente, creando SAs para sistemas MLS que originan paquetes reenviados por el gateway. Estos sistemas MLS pueden etiquetar explícitamente los paquetes que se enviarán, o la red entera de origen puede tener características de sensibilidad asociadas con el paquete. La security gateway debe crear y usar apropiadas SAs para AH, ESP o ambas, para proteger el tráfico que envía.

De la misma forma un gateway DEBERÍA aceptar y procesar paquetes salientes

AH y/o ESP y reenviarlos apropiadamente, usando etiquetamiento del paquete explicito, o confiando en las características de sensibilidad de la red de destino.

7. **Consideraciones de Desempeño**

El uso de IPsec impone costos computacionales en los host o security gateway que implementen estos protocolos. Estos costos están relacionados con la memoria necesaria para el código IPsec, el cálculo de los valores de control de integridad, encriptación y desencriptación y la manipulación de cada paquete. Los costos computacionales por cada paquete serán manifestados por la latencia creciente, que luego podría llegar ha ser reducido a lo largo del proceso. El uso de protocolos de administración de SA/claves, especialmente aquellos que emplean criptografía de clave publica, también agregan costos computacionales de desempeño al uso de IPsec. Estos costos computacionales por asociación estarán manifestados en términos de latencia creciente en el establecimiento de asociaciones. Para muchos host se anticipa que la criptografía basada en software no reducirá el rendimiento, pero la de hardware podrá ser requerida para las security gateways (puesto que representan puntos de agregación), y para ciertos hosts.

El uso de IPsec también impone costos de utilización de ancho de banda en la transmisión, intercambio, y componentes de enrutamiento en la estructura de Internet, componentes no implementados por IPsec. Esto se debe al tamaño creciente del paquete como resultado de agregarle las cabeceras AH y/o ESP, realizando túnel AH y/o ESP (que agrega una segunda cabecera IP) y el tráfico de paquetes incrementado asociado con los protocolos de administración de claves. Se anticipa que, en la mayoría de los casos, esta demanda de incremento de ancho de banda no afectará perceptiblemente la estructura de Internet. Sin embargo, en algunos casos los efectos pueden ser significantes, por ejemplo, transmitir tráfico ESP encriptado sobre un enlace dialup el cual podría ser comprimido.

Nota: La sobrecarga del establecimiento inicial de SA se sentirá en el primer paquete. Este retardo podría impactar en la capa de trasporte y aplicación. Por ejemplo podría causar que TCP trasmitiera el SYN antes de que el intercambio ISAKMP se haga. El efecto del retraso sería diferente en UDP que en TCP porque TCP no debe trasmitir ninguna otra cosa que no sea el SYN hasta que la conexión halla sido establecida, mientras que UDP seguirá adelante y trasmitirá los datos además del primer paquete.

Nota: Como se discute anteriormente, la compresión se puede emplear todavía en capas superiores a la capa IP. Existe un grupo de trabajo en la IETF (Protocolo de Compresión de la carga IP (IPPCP)) trabajando en "especificaciones del protocolo que hacen posible realizar compresión sin perdida en cargas individuales antes de que la carga sea procesada por un protocolo que la encripte. Estas especificaciones permitirán que las operaciones de compresión se realicen antes de la encriptación de la carga por protocolos IPsec".

8. **Requisitos de Conformidad**

Todos los sistemas IPv4 que demandan implementar IPsec DEBEN cumplir con todos los requisitos de este documento. Todos los sistemas IPv6 DEBEN cumplir con todos los requisitos de este documento.

9. **Análisis/Discusión de PMTU/DF/Cuestiones de Fragmentación**

9.1 Bit DF

¿En el caso donde un sistema (host o gateway) agregue una cabecera de encapsulación (por ejemplo, túnel ESP), el bit DF en el paquete original debería ser copiado en la cabecera de encapsulación?

La fragmentación parece ser adecuada en algunas situaciones, por ejemplo, puede ser apropiado fragmentar paquetes sobre una red con un MTU muy pequeña, por ejemplo, en redes inalámbricas (packet radio network) o en un salto de un teléfono celular a un nodo móvil, en vez de volver a trasmitir PMTU muy pequeñas para usarse sobre el resto de la trayectoria. En otras situaciones, puede ser apropiado fijar el bit DF para conseguir realimentación de routers posteriores sobre las restricciones de PMTU que requieren fragmentación. La existencia de estas situaciones permite a un sistema decidir si fragmenta o no sobre el "enlace" determinado de red, es decir, se necesita que una implementación sea capaz de copiar el bit DF (y procesar los mensajes ICMP PMTU), pero elaborando una opción que será seleccionada sobre la base de la información. En otras palabras, un administrador debería poder configurar el tratamiento del router del bit DF (fijar, limpiar, copiar de la cabecera encapsulada) para cada interfaz.

Nota: Si una implementación BITS intenta aplicar diferentes algoritmos IPsec basado en los puertos de origen/destino, será difícil aplicar ajustes en la Trayectoria MTU.

9.2 Fragmentación

Si se requiere, la fragmentación IP ocurre después del procesamiento IPsec dentro de una implementación IPsec. Así como, en modo trasporte AH o ESP se aplica solamente a los datagramas no a los fragmentados. Un paquete IP al cual se le a aplicado AH o ESP pueden ser fragmentados por router en la trayectoria, y tales fragmentos DEBEN ser reensamblados antes que se realice el procesamiento IPsec en el receptor. En modo túnel, AH o ESP se aplica a un paquete IP, cuya carga puede ser un paquete IP fragmentado. Por ejemplo, implementaciones IPsec BITS o BITW en security gateway, pueden aplicar AH en modo túnel a tales fragmentos. Observe que las implementaciones BITW o BITS son ejemplos en donde una implementación IPsec en host puede recibir un fragmento al cual se le aplica modo túnel, sin embargo, si se aplica al modo trasporte, estas implementaciones DEBEN reensamblar los fragmentos antes de aplicar IPsec.

Nota: IPsec siempre tiene que determinar a que campos de la cabecera IP encapsular. Esto es independiente de donde se haya insertado IPsec y esta intrínsicamente en la definición de IPsec. Por lo tanto cualquier implementación IPsec que no esta integrada dentro de una implementación IP debe incluir un código para construir la cabecera IP necesaria (por ejemplo IP2):

1. AH-túnel ---> IP2-AH-IP1-Trasporte-Datos
2. ESP-túnel --> IP2-Cabecera_ESP-IP1-Trasporte-Datos-trailer_ESP

En resumen, el método de fragmentación/reensamblaje descriptos arriba sobre la construcción para todos los casos examinados es:

	AH PuertoX		AH Túnel		ESP PuertoX		ESP Túnel	
Método de Implementación	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
Hosts (Integrado en la pilas IP)	Si	Si	Si	Si	Si	Si	Si	Si
Hosts (entre los drivers y la pila IP)	Si	Si	Si	Si	Si	Si	Si	Si
Security Gateway (Integrado en la pilas IP)			Si	Si			Si	Si
Procesador criptográfico externo*								

* Si un sistema de procesamiento criptográfico tiene su propio direccionamiento IP, entonces esta cubierto por el caso del security gateway. Este dispositivo recibe el paquete de un host y realiza el procesamiento IPsec. Tiene que poder manejar el mismo AH, ESP, y el procesamiento relativo a IPv4/IPv6 en modo túnel que una security gateway tendría que manejar. Si no tiene su propio direccionamiento, es similar a la implementación BITS entre los drivers de red y la pila IP.

El siguiente análisis asume que:

1. Hay solamente un modulo IPsec en una pila del sistema. No hay un modulo A de IPsec (añadiendo encriptación/ESP y por lo tanto) ocultando el protocolo de trasporte, puerto de origen, y el puerto de destino del modulo B de IPsec.

2. Hay varios lugares donde IPsec podría ser implementado:

   a. Hosts con integración de IPsec en la implementación nativa de IPsec. El implementador tiene acceso al código fuente de la pila IP.

   b. Hosts con implementaciones BITS donde IPsec es implementado entre los drivers de la red local y la pila IP. El acceso al código fuente de la pila IP no esta disponible; pero existe interfases bien definidas que permiten al código de IPsec incorporarse en el sistema.

   c. Security gateways y procesamiento criptográfico externo con integración de IPsec en la pila IP.

3. No todos los métodos descriptos arriba son factibles en todos los host. Pero se asume que para cada método, hay ciertos host para los cuales el método es factible.

Para cada una de las 3 categorías descriptas arriba, hay IPv4 y IPv6 en modo trasporte y túnel de AH y modo trasporte y túnel de ESP que dan un total de 24 casos (3x2x4).

Algunos campos de la cabecera y campos de interfaz se enumeran aquí para una fácil referencia. No están en el orden en el que van las cabeceras, sino que están listadas para permitir la comparación entre las columnas.

IPv4	IPv6	IP/interfaz de transporte (RFC 1122 – Sección 3.4)
Versión = 4	Versión = 6	
Longitud de la Cabecera		
TOS (*1)	Clase, Etiqueta de Fuljo	TOS
Longitud del Paquete	Longitud de la Carga	Longitud
Identificador		Identificador (Opcional)
Banderas(*1)		DF
Desplazamiento(*1)		
TTL(*1)	Limite de Saltos(*1)	TTL
Protocolo	Cabecera Siguiente	
Suma de control(*1)		
Dirección de Origen	Dirección de Origen	Dirección de Origen
Dirección de Destino	Dirección de Destino	Dirección de Destino
Opciones (*2)	Opciones(*2)	Opciones

(*1)= No cubierto por la autentificación de AH. La autentificación de ESP no cubre ninguna cabecera que la preceda.

(*2)= AH cubre Tipo-Opción y Longitud de la Opción, pero no cubre los Datos de la Opción.

Los resultados de cada uno de los 20 casos se muestran debajo ("se construye" = funcionará si el sistema fragmenta después del procesamiento IPsec saliente y reensambla antes que se realice el procesamiento IPsec entrante). Observe las cuestiones de implementaciones indicadas.

a. Hosts (Integrado dentro de la pilas IP)
   - AH-trasporte --> (IP1-AH-Transporte-Datos)
     - IPv4 -- se construye
     - IPv6 -- se construye
   - AH-túnel --> (IP2-AH-IP1-Transporte-Datos)
     - IPv4 -- se construye
     - IPv6 -- se construye
   - ESP-transporte --> (IP1-Cabecera_ESP-Transporte-Datos-trailer_ESP)
     - IPv4 -- se construye
     - IPv6 -- se construye
   - ESP-túnel --> (IP2-Cabecera_ESP-IP1-Transporte-Datos-trailer_ESP)
     - IPv4 -- se construye
     - IPv6 -- se construye

b. Host (BITS): coloque IPsec entre la capa IP y los drivers de red. En este caso el módulo IPsec tendría que hacer algo como lo siguiente para la fragmentación y el reensamblaje.

   - Realice el trabajo de fragmentación/reensamblaje y envié/reciba el paquete directamente a/de la capa de red. En modo trasporte AH o ESP esto es correcto. En modo túnel AH o ESP donde el extremo del túnel es el último destino, esto es correcto. Pero en los modos túneles AH o ESP donde el extremo del túnel es diferente del último destino y donde el host de origen es multi-homed, este meto podría resultar en un camino no tan óptimo porque el módulo IPsec no podría obtener la información necesaria (interfaz LAN y gateway del siguiente salto) para dirigir el paquete a la apropiada interfaz de red. Esto no es un problema si la interfaz y la gateway del siguiente salto son las mismas para el último destino y para el extremo del túnel. Pero si son diferentes, IPsec necesitaría saber

la interfaz LAN y la gateway del siguiente salto para el extremo del túnel. (Nota: El extremo del túnel (security gateway) es altamente probable que este en una trayectoria habitual al último destinó. Pero podría existir más de una trayectoria para el destino, por ejemplo, el host podría estar en una organización con dos firewalls. Y la trayectoria que esta siendo usada podría involucrar al firewall normalmente menos seleccionado) O

- Pase el paquete IPsec de nuevo a la capa IP donde se añadirá una cabecera IP extra y el módulo IPsec debería comprobarlo y dejar los fragmentos ahí. O

- Pase los contenidos del paquete a la capa IP de tal forma que la capa IP recree una cabecera IP apropiada.

En la capa de red, el modulo IPsec tendrá acceso a los siguientes selectores de el paquete: dirección de origen, dirección de destino, Protocolo Siguiente, y si hay una cabecera de capa de trasporte entonces, el puerto de la dirección de origen y el puerto de la dirección de destino. Uno no puede asumir que IPsec tiene acceso al Nombre. Se asume que la información del selector disponible es suficiente para calcular la entrada a la Política de Seguridad relevante y a la/s Asociación/es de Seguridad.

- AH-transporte --> (IP1-AH-Transporte-Datos)
  - IPv4 -- se construye
  - IPv6 -- se construye
- AH-túnel --> (IP2-AH-IP1-Transporte-Datos)
  - IPv4 -- se construye
  - IPv6 -- se construye
- ESP-transporte --> (IP1-Cabecera_ESP-Transporte-Datos-ESP_trailer)
  - IPv4 -- se construye
  - IPv6 -- se construye
- ESP-túnel --> (IP2-Cabecera_ESP-IP1-Transporte-Datos-ESP_trailer)
  - IPv4 -- se construye
  - IPv6 -- se construye

c. Security gateways –IPsec incorporado dentro de la pila IP.

Nota: El módulo IPsec tendrá acceso a los siguientes selectores del paquete: dirección de origen, dirección de destino, Protocolo Siguiente, y si hay una cabecera de capa de trasporte entonces, el puerto de la dirección de origen y el puerto de la dirección de destino. No tendrá acceso al Identificador de Usuario (solamente los hosts tienen acceso a la información de Identificador de Usuario.) Algunas implementaciones BITS son diferentes, las security gateways son capaces de buscar la Dirección de Origen en los DNS para proporcionar un Nombre de Sistema, por ejemplo, dentro de una situación que involucre el uso de direcciones IP asignadas dinámicamente en conjunto con entradas DNS dinámicas. Tampoco tendrá acceso a la información de la capa de trasporte si hay una cabecera ESP, o si no es el primer fragmento de un mensaje fragmentado. Se asume que la información del selector disponible es suficiente para calcular la entrada a la Política de Seguridad relevante y a la/s Asociación/es de Seguridad.

- AH-túnel --> (IP2-AH-IP1-Transporte-Datos)
  - IPv4 -- se construye
  - IPv6 -- se construye

• ESP-túnel -->(IP2-Cabacera_ESP-IP1-Transporte-Datos-ESP_trailer)
- IPv4 -- se construye
- IPv6 -- se construye

9.3 Descubrimiento de la Trayectoria MTU

Como se mencionó antes, "ICMP PMTU" hace referencia a un mensaje ICMP usado para el descubrimiento de la trayectoria MTU.

La leyenda para los diagramas que están debajo en la Sección 9.3.1 y 9.3.3 es:

==== = SA (AH o ESP, trasporte o túnel)
---- = conectitividad (o si esta etiquetado, limite administrativo)
.... = mensaje ICMP (de aquí en adelante designado como ICMP PMTU) para :

IPv4 (ICMP para IPv4, [ICMPv4]):
- Tipo = 3 (Destino inalcanzable)
- Código = 4 (fragmentación necesaria, fijar DF)
- MTU del Siguiente Salto de los 16 bits de menor orden de la segunda palabra de la cabecera ICMP (etiquetado como no usado en [ICMPv4]), con los 16 bits de mayor orden puestos a cero.

IPv6 (ICMP para IPv6,[ICMPv6]):
- Tipo = 2 (paquete demasiado grande)
- Código = 0 (Fragmentación necesaria y fijar el DF)
- MTU del Siguiente Salto del campo MTU de 32 bits del ICPM6

Hx = host x
Rx = router x
SGx = security gateway x
X* = X soporta IPsec

9.3.1 Identificando al Host de Origen

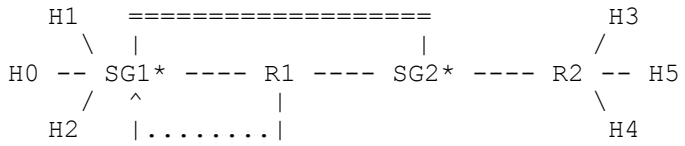
La cantidad de información devuelta por el mensaje ICMP es limitada y esto afecta a los selectores que están disponibles para identificar a la SA, host de origen, etc. para usarse en transmisiones futuras de la información PMTU.

En resumen ... Un mensaje ICMP debe contener la siguiente información del paquete "problemático":

- IPv4 [ICMPv4] –cabecera IP más un mínimo de 64 bits.

Por consiguiente en IPv4, un ICMP PMTU puede identificar solamente a la primera SA (externa). Esto se debe a que el ICMP PMTU puede contener solamente 64 bits del paquete "problemático" mas allá de la cabecera IP, que capturará solamente el primer SPI de AH o de ESP. En IPv6, el ICMP PMTU probablemente proporcionará todos los SPIs y los selectores en la cabecera IP, pero quizás no proporcione los puertos de la Dirección de Origen/Destino (en la cabecera de trasporte) o el protocolo encapsulado (TCP, UDP, etc.). Por otra parte, si se utiliza ESP, los puertos de trasporte y los selectores del protocolo pueden estar encriptados.

Analizando el diagrama de abajo de un túnel entre dos security gateways (las security gateways no usan modo trasporte)...



Suponiendo que la política de seguridad del SG1 (security gateway 1) es usar una única SA hacia el SG2 para todo el tráfico entre los host H0, H1, H2 y los host H3, H4, H5. Y suponiendo que H0 envía un paquete de datos hacia H5 el cual causa que R1 envié un mensaje ICMP PMTU al SG1. Si el mensaje PMTU tiene solamente el SPI, el SG1 podrá buscar la SA y encontrar la lista de posibles host (H0 , H1, H2, comodín); pero SG1 no tendrá forma de saber que H0 envió el tráfico que activó el mensaje ICMP PMTU.

Paquete Original	Procesamiento IPsec posterior	Paquete ICMP
		Cabecera IP-3 (Origen = R1, Destino = SG1)
		Cabecera ICMP (contiene el PMTU)
	Cabecera IP-2	Cabecera IP-2 (Origen = SG1, Destino = SG2)
	Cabecera ESP	Un mínimo de 64 bits de la cabecera ESP (*)
Cabecera IP-1	Cabecera IP-1	
Cabecera TCP	Cabecera TCP	
Datos TCP	Datos TCP	
	Trailer ESP	

(*) Los 64 bits incluirán bastante de la cabecera ESP (o AH) para incluir el SPI.

- ESP: SPI (32 bits), Número de Secuencia (32 bits)

- AH : Cabecera Siguiente (8 bits), Longitud de la Carga (8 bits), Reservado (16 bits), SPI (32 bits)

Esta limitación en la cantidad de información que vuelve con un mensaje ICMP crea un problema en la identificación de los host de origen para el paquete (para saber a quien trasmitir la futura información ICMP PMTU). Si el mensaje ICMP contiene solamente 64 bits de la cabecera IPsec (mínimo para IPv4), los selectores de IPsec (por ejemplo, direcciones de origen y destino, Protocolo Siguiente, puertos de Origen y de Destino, etc.) se perderían. Pero el mensaje de error ICMP aun proporcionará al SG1 el SPI, la información PMTU y las gateways de origen y destino para la SA relevante.

La security gateway de destino y el SPI definen únicamente una SA que a su vez define un conjunto de posibles host de origen. En este punto, la SG1 podría:

a. Enviar la información PMTU a todo los posibles host de origen. Esto no funcionaría bien si la lista de host es un comodín o si muchos o la mayoría de los host no estuvieran enviando hacia el SG1; pero esto funcionaría si el SPI/destino/etc. estuvieran asociados a un número pequeño de host.

b. Almacenar el PMTU con el SPI/etc. y esperar hasta que el próximo o los próximos paquetes lleguen del host(s) de origen para la SA relevante. Si el paquete o los paquetes son más grandes que el PMTU descarte los paquetes, y compare el o los mensajes ICMP PMTU con el o los nuevos

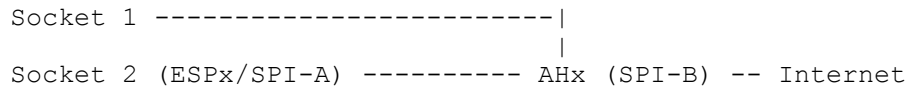
paquetes y el PMTU actualizado y envíe el o los mensaje ICMP sobre el problema a el o los host de origen. Esto implica un retraso en la notificación al host(s) de origen, pero evita los problemas de (a).

Puesto que solamente el último método es factible en todos los casos, una security gateway DEBE proporcionar tal método como una opción. Sin embargo, si el mensaje ICMP contiene más información del paquete original, habrá suficiente información para determinar inmediatamente a que host trasmitir el mensaje ICM/PMTU y proporcionar a este sistema con los 5 campos (dirección de origen, dirección de destino, puerto de destino, puerto de origen, y protocolo de trasporte) necesarios para determinar donde almacenar/actualizar el PMTU. Bajo tales circunstancias, una security gateway DEBE generar un mensaje ICMP PMTU inmediatamente al recibir un ICMP PMTU de una trayectoria futura.

NOTA: El campo Protocolo Siguiente no estará contenido en el mensaje ICMP y el uso de encriptación ESP puede ocultar los campos de los selectores que han sido encriptados.

9.3.2 Cálculo del PMTU

El cálculo del PMTU de un ICMP PMTU tiene que considerar la adición de cualquier cabecera IPsec por H1 – trasporte AH y/o ESP, o túnel ESP o AH. Dentro de un único host, múltiples aplicaciones pueden compartir un SPI y la concatenación de SA puede ocurrir. (Ver la Sección 2.5 Combinaciones Básicas de SA, para la descripción de las combinaciones que DEBEN ser soportadas.) El diagrama que sigue ilustra un ejemplo de SA entre un par de host (visto desde la perspectiva de uno de los host.) (ESPx o AHx = modo trasporte)



Para averiguar el PMTU para cada socket que se asocia a SPI-B, será necesario tener punteros invertidos de SPI-B para cada una de las dos trayectorias que conducen al socket 1 y al Socket 2/SPI-A.

9.3.3 Granularidad para Mantener Datos PMTU

En host, la granularidad con la cual el procesamiento ICMP PMTU puede ser realizada difiere dependiendo de la situación de implementación. Mirando a un host, hay tres situaciones que son de interés para cuestiones PMTU:

a. Integración de IPsec dentro de la implementación IP nativa

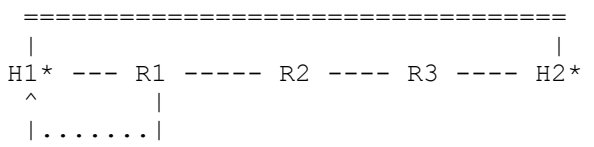
b. Implementaciones BITS, donde IPsec es implementado "por debajo" de una implementación existente de una pila de protocolo TCP/IP, entre el IP nativo y los drivers de red locales.

c. No hay implementación IPsec: este caso esta incluido por que es relevante en los casos donde una security gateway esta enviando de vuelta la información PMTU a un host.

Solamente en el caso (a) se puede mantener los datos PMTU al mismo nivel de granularidad que las asociaciones de comunicación. En los otros casos, la capa IP mantendrá los datos PMTU en la granularidad de las direcciones de Origen y de Destino IP (y opcionalmente TOS/Clase), según lo descripto en el RFC 1191. Esto es una diferencia importante, por que más de una

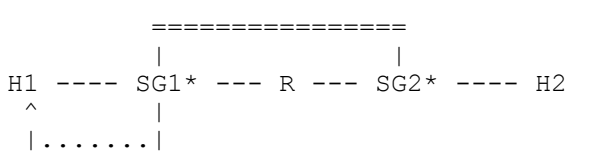
asociación de comunicación puede estar asociada a las mismas direcciones de origen y destino IP, y cada asociación de comunicación puede tener cantidades diferentes de sobrecarga informática en la cabecera IPsec (por ejemplo debido al uso de trasformaciones diferentes o algoritmos diferentes.) Esto esta ilustrado en los siguientes ejemplos.

En los casos (a) y (b)... Suponga que usted tiene la siguiente situación. H1 esta enviando hacia H2 y el paquete que sé envía de R1 hacia R2 excede el PMTU del salto de red entre ellos.



Si R1 es configurado para no fragmentar el tráfico del suscriptor, R1 envía un mensaje ICMP PMTU con el adecuado PMTU hacia H1. El procesamiento de H1 variaría con la naturaleza de la implementación. En el caso (a) (IP nativo), los servicios de seguridad están ligados a los sockets o al equivalente. Aquí la implementación IP/IPsec en H1 puede almacenar/actualizar el PMTU para el socket asociado. En el caso (b), la capa IP en H1 puede almacenar/actualizar el PMTU solo para la granularidad de las direcciones de Origen y Destino y posiblemente TOS/Clase, según lo observado arriba. Por lo tanto puede que el resultado no sea tan óptimo, puesto que la PMTU para una SRC/DST/TOS/Clase dada será la sustracción de mayor cantidad de cabecera IPsec usada por cualquier asociación de comunicación entre un origen y un destino.

En el caso (c), bebe haber una security gateway para hacer cualquier procesamiento IPsec. Entonces suponiendo que usted tenga la siguiente situación. H1 esta enviando hacia H2 y el paquete para ser enviado de SG1 hacia R excede el salto de red PMTU entre ellos.



Como se describe arriba para el caso (b), la capa IP en H1 puede almacenar/actualizar el PMTU para solamente la granularidad de las direcciones de Origen y Destino, y posiblemente TOS/Clase. Por lo tanto puede que el resultado no sea tan óptimo, puesto que la PMTU para una SRC/DST/TOS/Clase dada será la sustracción de mayor cantidad de cabecera IPsec usada para cualquier asociación de comunicación entre un origen y un destino.

9.3.4 Mantenimiento de Socket a Través de Datos PMTU

La implementación del cálculo de PMTU (Sección 9.3.2) y el soporte para PMTUs en la granularidad de "asociaciones de comunicación" individuales (Sección 9.3.3) es un tema local. No obstante una implementación IPsec en un host basada en socket DEBERÍA mantener la información a través de la base de socket. Sistemas BITS DEBEN comunicar un ICMP PMTU a la implementación IP del host, después de adaptarla a alguna cabecera IPsec supletoria que se agregue a estos sistemas. La determinación de la cabecera supletoria DEBERÍA estar determinada por el análisis del SPI y cualquier otra información del selector presente en un mensaje ICMP PMTU reenviado.

9.3.5 Entrega de Datos PMTU a la Capa de Trasporte

Los mecanismos del host para trasmitir el PMTU actualizado a la capa de trasporte son invariables, según lo especificado en el RFC 1191 (Descubrimiento de la Trayectoria MTU)

9.3.6 Envejecimiento de los Datos PMTU

Este tema fue visto en la Sección 4.1.2.4

10. **Ejemplo de Código de Secuencia de Espacio de Ventana**

Este capítulo contiene una rutina [JaHu] que implementa un control de máscara de bit (bitmask) para una ventana de 32 paquetes. Observe que este código controla si hay reenvíos y actualiza la ventana. Así el algoritmo, como se muestra, debería ser solamente convocado después de que el paquete haya sido autentificado. Los implementadores pueden desear considerar dividir el código para realizar el control de reenvíos antes de calcular el ICV. Si el paquete no es un reenvío, el código calcularía el ICV, (descarte cualquier paquete defectuoso) y si el paquete es correcto, actualice la ventana.

```
#include <stdio.h>
#include <stdlib.h>
typedef unsigned long u_long;

enum {
    ReplayWindowSize = 32
};

u_long bitmap = 0;                 /* session state - must be 32 bits */
u_long lastSeq = 0;                     /* session state */

/* Returns 0 if packet disallowed, 1 if packet permitted */
int ChkReplayWindow(u_long seq);

int ChkReplayWindow(u_long seq) {
    u_long diff;

    if (seq == 0) return 0;             /* first == 0 or wrapped */
    if (seq > lastSeq) {                /* new larger sequence number */
        diff = seq - lastSeq;
        if (diff < ReplayWindowSize) {  /* In window */
            bitmap <<= diff;
            bitmap |= 1;                /* set bit for this packet */
        } else bitmap = 1;          /* This packet has a "way larger" */
        lastSeq = seq;
        return 1;                       /* larger is good */
    }
    diff = lastSeq - seq;
    if (diff >= ReplayWindowSize) return 0; /* too old or wrapped */
    if (bitmap & ((u_long)1 << diff)) return 0; /* already seen */
    bitmap |= ((u_long)1 << diff);              /* mark as seen */
    return 1;                           /* out of order but good */
}

char string_buffer[512];

#define STRING_BUFFER_SIZE sizeof(string_buffer)
```

```
int main() {
    int result;
    u_long last, current, bits;

    printf("Input initial state (bits in hex, last msgnum):\n");
    if (!fgets(string_buffer, STRING_BUFFER_SIZE, stdin)) exit(0);
    sscanf(string_buffer, "%lx %lu", &bits, &last);
    if (last != 0)
    bits |= 1;
    bitmap = bits;
    lastSeq = last;
    printf("bits:%08lx last:%lu\n", bitmap, lastSeq);
    printf("Input value to test (current):\n");

    while (1) {
        if (!fgets(string_buffer, STRING_BUFFER_SIZE, stdin)) break;
        sscanf(string_buffer, "%lu", &current);
        result = ChkReplayWindow(current);
        printf("%-3s", result ? "OK" : "BAD");
        printf(" bits:%08lx last:%lu\n", bitmap, lastSeq);
    }
    return 0;
}
```

11. **Categorización de mensajes ICMP**

La tabla siguiente caracteriza los mensajes ICMP como generado por el host, generado por el router, por ambos, en disponibilidad/desconocido. El primer conjunto es de IPv4 y el segundo es de IPv6.

IPv4

Tipo	Nombre/código	Referencia
GENERADO POR EL HOST:		
3	Destino Inaccesible	
2	Protocolo Inaccesible	[ICMPv4]
3	Puerto Inaccesible	[ICMPv4]
8	Origen del Host Incomunicado	[ICMPv4]
14	Violación de la Precedencia del Host	[RFC1812]
10	Selección del Router	[RFC1256]

Tipo	Nombre/código	Referencia
GENERADO POR EL ROUTER:		
3	Destino Inaccesible	
0	Red Inaccesible	[ICMPv4]
4	Fragmentación Necesaria, no fue Fijada la Fragmentación	[ICMPv4]
5	Error en la Ruta de Origen	[ICMPv4]
6	Red de Destino Desconocida	[ICMPv4]
7	Destino del host Desconocido	[ICMPv4]
9	Comm. W/Red esta Administrativamente Prohibido	[ICMPv4]
11	Destino de Red Inaccesible para el Tipo de Servicio	[ICMPv4]
5	Redireccionamiento	
0	Datagrama de Redireccionamiento de Red (o Subred)	[ICMPv4]
2	Datagrama de Redireccionamiento para el Tipo de Servicio & Red	[ICMPv4]
9	Anuncio de Router	[RFC1256]
18	Respuesta a la Máscara de Dirección	[RFC950]

Tipo	Nombre/código	Referencia
GENERADOS POR EL ROUTER Y EL HOST:		
0	Echo Reply	[ICMPv4]
3	Destino Inaccesible	
1	Host Inaccessible	[ICMPv4]
10	Comm. W/Host de Destino Administrativamente Prohibido	[ICMPv4]
12	Host de Destino Inaccesible por el Tipo de Servicio	[ICMPv4]
13	Comunicación Administrativamente Prohibida	[RFC1812]
15	Precedencia de Limite en Efecto	[RFC1812]
4	Apaciguando al Origen (Source Quench)	[ICMPv4]
5	Redireccionamiento	
1	Datagrama de Redireccionamiento para el Host	[ICMPv4]
3	Datagrama de Redireccionamiento para el Tipo de Servicio y Host	[ICMPv4]
6	Dirección de Host Alternativa	[IAMA-3]
8	Echo	[ICMPv4]
11	Tiempo Excedido	[ICMPv4]
12	Problema de Parámetros	[ICMPv4], [IAMA-3]
13	Marca de Tiempo	[ICMPv4]
14	Respuesta de Marca de Tiempo	[ICMPv4]
15	Solicitud de Información	[ICMPv4]
16	Respuesta de Información	[ICMPv4]
17	Solicitud de Dirección de Máscara	[RFC950]
30	Traceroute	[RFC1393]
31	Error de Conversión de Datagrama	[RFC1475]
32	Redireccionamiento del Host Móvil	[IAMA-3]
39	SKIP	[IAMA-3]
40	Photuris	[IAMA-3]

Tipo	Nombre/código	Referencia
TIPO DISPONIBLE O GENERADOR POR DESCONOCIDO:		
1	Disponible	[IAMA-3]
2	Disponible	[IAMA-3]
7	Disponible	[IAMA-3]
19	Reservado (para Seguridad)	[IAMA-3]
20-29	Reservado (Para Experimento de Fuerza)	[IAMA-3]
33	IPv6 Donde Estas	[IAMA-3]
34	IPv6 Acá Estoy	[IAMA-3]
35	Solicitud De Registración Móvil	[IAMA-3]
36	Repuesta De Registración Móvil	[IAMA-3]
37	Solicitud de Nombre de Dominio	[IAMA-3]
38	Respuesta de Nombre De Dominio	[IAMA-3]
41-255	Reservado	[IAMA-3]

IPv6

Tipo	Nombre/código	Referencia
GENERADO POR EL HOST:		
1	Destino Inaccesible	[ICMPv6]
	4 Puerto Inaccesible	

Tipo	Nombre/código	Referencia
GENERADO POR EL ROUTER:		
1	Destino Inaccesible	[ICMPv6]
	0 Sin Ruta para el Destino	
	1 Comm. w/esta Administrativamente Prohibido	
	2 Sin un Vecino	
	3 Dirección Inaccesible	
2	Paquete Demasiado Grande	[ICMPv6]
	0	
3	Tiempo Excedido	[ICMPv6]
	0 Limite de Salto Excedido en Transito	
	1 Limite de Reensamblaje de Fragmento Excedido	

Tipo	Nombre/código	Referencia
GENERADOS POR EL ROUTER Y EL HOST:		
4	Parámetro Problema	[ICMPv6]
	0 Encuentro de Campo de Cabecera Errónea	
	1 Encuentro de Tipo de cabecera siguiente no Reconocido	
	2 Encuentro de Opción IPv6 no Reconocida	

Capítulo 3

Cabecera de Autentificación

1. **Introducción**

La Cabecera de Autentificación IP (AH) se usa para proporcionar integridad sin conexión y autenticación del origen de datos para datagramas IP ("autentificación" a partir de ahora), y para proporcionar protección contra reenvíos. Este último servicio es opcional y puede seleccionarse una vez que se ha establecido la Asociación de Seguridad (SA). Aunque se establece por defecto que el emisor incremente el Número de Secuencia usado en el anti-replay, el servicio es efectivo solamente si el receptor controla el Número de Secuencia. HA proporciona autentificación a las partes de la cabecera IP que se les pueda brindar este servicio, así como también a los datos de los protocolos de las capas superiores. Sin embargo, algunos campos de la cabecera IP pueden cambiar durante el transporte, y el valor de estos campos, cuando el paquete llega al receptor, puede que no sea previsible para el emisor. Los valores de tales campos no pueden ser protegidos por AH. Así la protección proporcionada a la cabecera IP por AH se proporciona solo a partes de la cabecera IP.

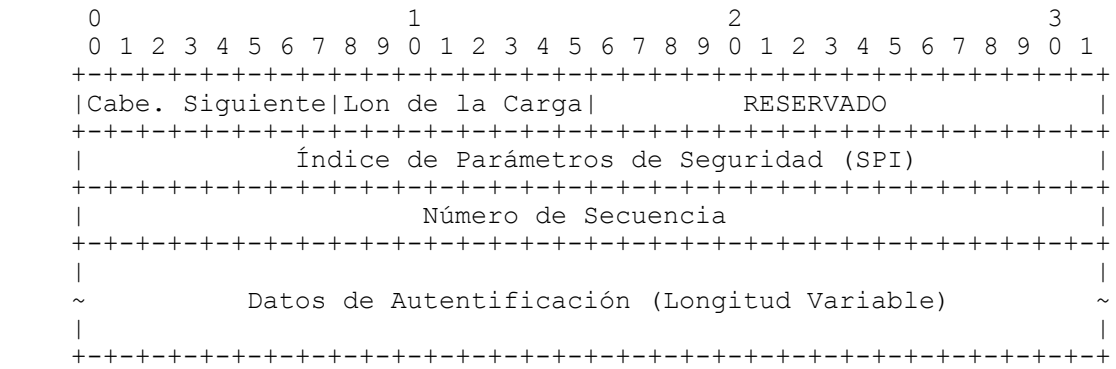
AH se puede aplicar solo, o en combinación con la Carga de Seguridad Encapsulada IP (ESP) (véase el capítulo 4), o a través de la modalidad anidada usando el modo túnel (véase el capítulo anterior). Los servicios de seguridad pueden ser suministrados a comunicaciones, entre un par de hosts, o entre un par de security gateway (SG), o entre security gateway y un host. ESP puede ser usado para proporcionar los mismos servicios de seguridad, y también para proporcionar un servicio de confidencialidad (encriptación). La diferencia principal entre la autentificación proporcionada por ESP y la de AH es la extensión de la cobertura. Específicamente, ESP no protege ninguno de los campos de la cabecera IP a menos que esos campos sean encapsulados por ESP (en modo túnel). Para más detalles en cómo utilizar AH y ESP en varios ambientes de red, vea el capítulo de la Arquitectura de Seguridad.

Nota: las opciones requeridas actualmente para el manejo de claves tanto para AH como para ESP son el modo manual y en el modo automatizado por medio de IKE (véase el capítulo 10).

Nota: Para poder llegar a comprender íntegramente este capítulo es necesario que se haya comprendido el capítulo de la Arquitectura de Seguridad, el capítulo de ISAKMP, el de IKE (y por consiguiente el de OAKLEY), lo que ocurre es que para comprender los capítulos antes mencionados es necesario haber "visto" este capítulo antes.

2. **Formato de la cabecera de Autentificación**

La cabecera del protocolo (IPv4, IPv6, o de Extensiones) inmediatamente antes de la cabecera de AH contendrá el valor 51 en el Protocolo (IPV4) o en el campo Cabecera Siguiente (de Extensión, en IPv6) [STD-2].



Las siguientes subsecciones definen los campos que comprenden el formato de AH. Todos los campos descriptos aquí son obligatorios, es decir, están siempre presentes en el formato de AH y se incluyen en el cálculo del Valor de Comprobación de Integridad (Integrity Check Value - ICV), (ver las Sección 2.6 y 3.3.3).

2.1 Cabecera Siguiente

La Cabecera Siguiente es un campo de 8 bits que identifica el tipo de carga siguiente después de la Cabecera de Autentificación. El valor de este campo se elige del conjunto de Números de Protocolo IP definidos en el más resiente RFC de "Números Asignados" [STD-2] por la Autoridad de Números de Asignación de Internet (IANA).

2.2 Longitud de la Carga

Este campo de 8 bits especifica la longitud de AH en palabras de 32 bit (en unidades de 4 byte), menos "2". Todas las cabeceras de extensión de IPv6, según el RFC 2460, codifican el campo "Longitud de la Cabecera de Extensión" primero restando uno (palabra de 64-bit) a la longitud de la cabecera (medido en palabras de 64-bit). AH es una cabecera de extensión IPv6. Sin embargo, puesto que su longitud se mide en palabras de 32 bit, la "longitud de la carga" es calculada restando 2 (palabras de 32 bit). En el caso "estándar" de un valor de autentificación de 96 bits positivos divididos en 3 palabras de 32 bits de tamaño fijo, este campo tendrá una longitud de "4". Un algoritmo de autentificación "NULL" puede ser usado solamente para propósitos de depuración (puesta a punto del sistema). Su uso daría lugar a un valor "1" para este campo en IPv4 o "2" en IPv6, puesto que no habría Datos de Autentificación en el correspondiente campo (ver Sección 3.3.3.2.1 "Carga de Datos de Autentificación").

2.3 Reservado

Este campo de 16 bits esta reservado para uso futuro. Se DEBE fijar a "cero." Observe que el valor está incluido en el cálculo de los datos de autentificación, pero es ignorado por el receptor.

2.4 Índice de Parámetros de Seguridad (SPI)

El SPI es un valor arbitrario de 32 bits que, conjuntamente con la dirección de destino IP y el protocolo de seguridad (AH), identifican unívocamente a la Asociación de Seguridad para este datagrama. El conjunto de valores de SPI en el rango de 1 a 255 están reservados por la IANA para uso futuro; un valor reservado de SPI no será destinado normalmente por el IANA a menos que el uso del valor destinado de SPI se especifique en un

RFC. Este es seleccionado por el sistema de destino sobre el establecimiento de una SA (véase el capítulo de la Arquitectura de Seguridad para más detalles).

El valor de SPI cero (0) esta reservado para usarse localmente, las implementaciones no deben trasmitir este valor por la red. Por ejemplo, una implementación de administración de clave PUEDE utilizar el valor cero de SPI para denotar que "No Existe Asociación de Seguridad" durante el período en el cual la implementación IPsec ha solicitado a la entidad administradora de claves que se establezca una nueva SA, pero la SA todavía no se ha establecido.

La necesidad del SPI se hace evidente cuando tenemos más de una comunicación con la misma dirección IP de destino y protocolo de seguridad (AH o ESP). En teoría entonces, podemos tener (2^32)-256 Asociaciones de Segundad con la misma dirección IP de destino y protocolo de seguridad.

2.5 Número de Secuencia

Campo de 32 bits sin signo que contiene un valor creciente y único del contador (del número de secuencia). Es obligatorio y debe estar siempre presente incluso si el receptor elige no habilitar el servicio de anti-replay para una SA específica. El procesamiento del campo Número de Secuencia esta a criterio del receptor, es decir, el emisor DEBE transmitir siempre este campo, pero el receptor no necesita actuar sobre él (véase la discusión de la Verificación del Número de Secuencia en "Procesamiento de Paquetes Entrantes" en la sección posterior).

El contador del emisor y del receptor se inicializan a 0 cuando se establece una SA. (El primer paquete que se envíe bajo esa SA tendrá el Número de Secuencia 1; vea la Sección 3.3.2 para más detalles de cómo se genera el Número de Secuencia.) Si se habilita el anti-replay (por defecto), la transmisión del Número de Secuencia nunca debe permitir que el Número de Secuencia retorne a cero. Por ende, el contador del emisor y del receptor DEBEN ser resetiados (para el establecimiento de una nueva SA y de esta manera también una nueva clave) antes de que se trasmitan 2^32 paquetes sobre una SA.

2.6 Datos de Autentificación

Este campo es de longitud variable y contiene el Valor de Comprobación de Integridad (ICV) para este paquete. Este campo debe contener un múltiplo entero de 32 bits de longitud. Los detalles del cálculo de ICV se describen en la Sección 3.3.2. Este campo puede incluir relleno explícito (apreciable). Este relleno se incluye para asegurarse de que la longitud de la cabecera de AH sea múltiplo entero de 32 bits (en IPv4) o de 64 bits (en IPv6). Todas las implementaciones DEBEN soportar tales rellenos. Los detalles de cómo calcular la longitud del relleno se proporcionan abajo. El algoritmo de autentificación DEBE especificar la longitud ICV y las reglas de comparación y los pasos de procesamiento para la validación.

3. **Procesamiento de la Cabecera de Autentificación**

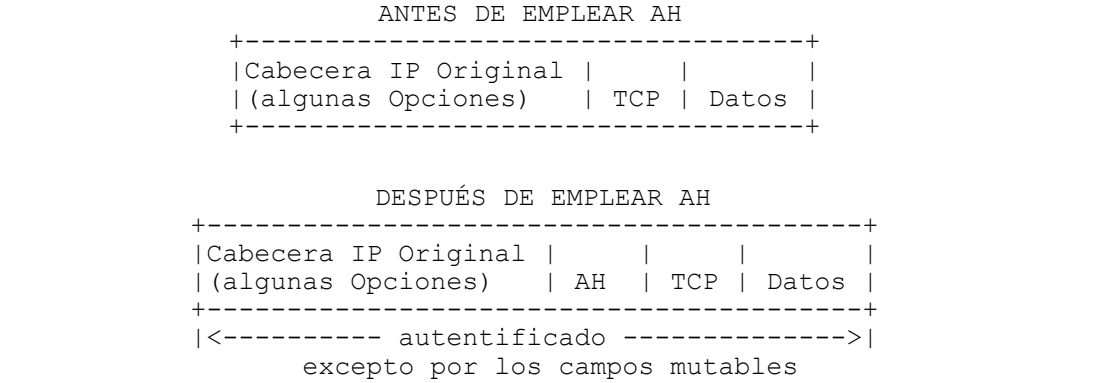
3.1 Localización de la Cabecera de Autentificación

Al igual que ESP, HA se puede emplear en modo trasporte o en modo túnel. El modo trasporte es aplicable solamente a implementaciones host y proporciona protección para los protocolos de capa superiores, además de los campos seleccionados de la cabecera IP. En este modo, observe que para las

implementaciones BITS o BITW, según lo definido en el capítulo anterior, fragmentos IP entrantes y salientes se pueden precisar en una implementación IPsec para realizar el reensamblaje/fragmentación IP conforme a esta especificación para proporcionar soporte IPsec transparente. Cuidado especial se requiere para realizar tales operaciones dentro de estas implementaciones cuando múltiples interfaces se están usando.

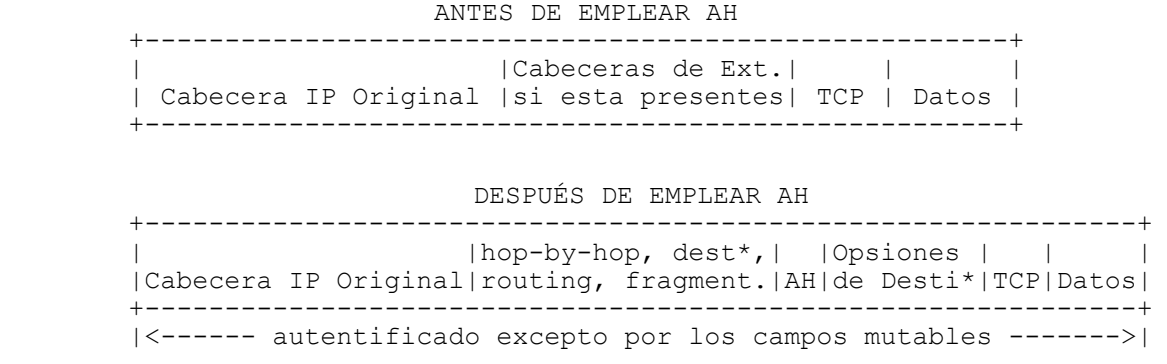
En modo trasporte, AH se inserta después de la cabecera IP y antes del protocolo de capa superior, por ejemplo, TCP, UDP, ICMP, etc. o antes de cualquier otra cabecera IPsec que ya se haya incluido. En el contexto de IPv4, esto requiere colocar AH después de la cabecera IP (y de cualquier opción que esta contenga), pero antes del protocolo de capa superior. (Observe que el término modo "transporte" no debería ser mal interpretado restringiendo su uso solamente a TCP y UDP. Por ejemplo, un mensaje ICMP se PUEDE enviar usando modo "transporte" o modo "túnel".) El diagrama siguiente ilustra a AH en modo trasporte para un paquete típico IPv4, "antes y después" de haberle aplicado AH en modo trasporte.

Paquete IPv4



En el contexto IPv6, el AH se ve como carga extremo a extremo (end-to-end), y debe aparecer después de las cabeceras de extensión: salto-por-salto (hop-by-hop), de encaminamiento (routing), y de fragmentación. Las cabecera (s) de extensión opciones de destino podrían aparecer antes o después de la cabecera AH dependiendo de la semántica deseada. El diagrama siguiente ilustra AH en modo trasporte colocado en un paquete típico de IPv6.

Paquete IPv6

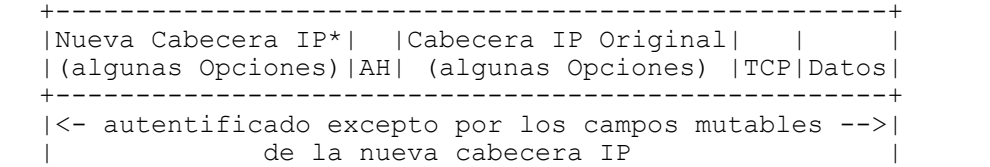


* = si están presentes, pueden estar antes que AH, después de AH, o en ambos.

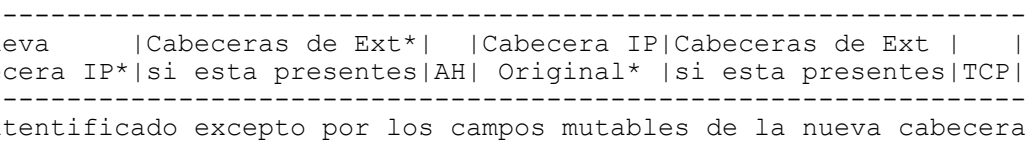
Las cabeceras ESP y AH se pueden combinar de varias formas. El capítulo anterior se describió las combinaciones de Asociaciones de Seguridad que deben ser soportadas.

AH en modo túnel puede ser empleado en host o securitys gateway (o en implementaciones BITS o BITW, según lo definido en el capítulo anterior. Cuando AH se implementa en una security gateway (protege tráfico en tránsito), el modo túnel debe ser utilizado. En modo túnel, la cabecera IP "interna" lleva la última dirección de origen y de destino, mientras que la cabecera IP "externa" puede contener distintas direcciones IP, por ejemplo, direcciones de securitys gateway. En modo túnel, AH protege el paquete IP interno completamente, incluyendo la cabecera IP interna entera. La posición de AH en modo túnel, concerniente a la cabecera IP exterior, es igual que para AH en modo trasporte. El diagrama siguiente ilustra AH en modo túnel colocado en paquetes típicos IPv4 y IPv6.

Paquete IPv4



Paquete IPv6



* = Construcción de otras cabeceras IP y/o de extensión y modificación de la cabecera IP interna y/0 de extensión según lo debatido mas abajo.

3.2 Algoritmos de Autentificación

El algoritmo de autentificación empleado para el cálculo de ICV esta especificado por la SA. Para las comunicaciones punto a punto, los algoritmos de autentificación más aptos incluyen claves con Código de Autentificación de Mensaje (MACs) basados en algoritmos de encriptación simétricos (por ejemplo, 3DES) o funciones hash unidireccionales (por ejemplo, MD5 o SHA-1). Para comunicaciones multicast, los algoritmos hash unidireccionales combinados con algoritmos de firmas asimétricas son apropiados, aunque las consideraciones de funcionamiento y de espacio actual imposibilitan el uso de tales algoritmos. Los algoritmos de autentificación que deben implementarse obligatoriamente se describen en la Sección 5 "Requerimientos de Conformidad". Otros algoritmos PUEDEN ser empleados.

3.3 Procesamiento de Paquetes Salientes

En modo trasporte, el emisor inserta la cabecera AH después de la cabecera IP y antes de la cabecera del protocolo de capa superior, como se describió anteriormente. En modo túnel, la cabeceras externas y internas IP/de extensiones se pueden interrelacionar de varias formas. La construcción de

las cabeceras externas IP/extensiones llevadas a cabo durante el proceso de encapsulación se describieron en el capítulo anterior.

Si se requiere más de una cabecera IPsec/extensión, el orden de aplicación de las cabeceras de seguridad DEBE estar definido por la política de seguridad. Para la simplicidad del procesamiento, cada cabecera de IPsec DEBERÍA ignorar la existencia (es decir, no fijar a cero los contenidos o no intentar predecir los contenidos) de las cabeceras de IPsec que se aplicarán después. Mientras que una implementación IP nativa o BITS podría predecir los contenidos de las últimas cabeceras IPsec a las que esta implementación se aplicó, no será posible que esta implementación prediga ninguna de las cabeceras IPsec agregadas por una implementación BITW entre el host y la red.

3.3.1 Búsqueda de Asociaciones de Seguridad

AH se aplica a un paquete saliente solamente después de que una implementación IPsec determine que el paquete está asociado con una SA la cual requiere el procesamiento de AH. El proceso de determinar qué (si existe alguno) procesamiento IPsec se aplica al tráfico saliente, se describe en el Capítulo 2.

3.3.2 Generación del Número de Secuencia

El contador del emisor es inicializado a 0 cuando se establece una SA. El emisor incrementa el Número de Secuencia para esta SA e inserta el nuevo valor dentro del Campo Número de Secuencia. Así, el primer paquete enviado usando una SA dada tendrá un valor de Número de Secuencia de 1.

Si se habilita el anti-replay (por defecto), el emisor controla para asegurarse que el contador no ha completado un ciclo antes de insertar el nuevo valor en el campo Número de Secuencia. Es decir, el emisor NO DEBE enviar un paquete en una SA, si al hacerlo haría que el Número de Secuencia complete un ciclo. Una tentativa de transmitir un paquete que resultaría en un desbordamiento del Número de Secuencia es un evento auditable. Observe que este método de administración del Número de Secuencia no requiere el uso de la aritmética modular (ver el Capítulo 5).

El emisor asume que el anti-replay es habilitado por defecto, a menos que sea notificado de otra cosa por el receptor (véase la Sección 3.4.3). Así, si el contador ha completado un ciclo, el emisor establecerá una nueva SA y una clave (a menos que la SA haya sido configurada con administración manual de claves).

Si el anti-replay esta deshabilitado, el emisor no necesita monitorear o volver a cero el contador, por ejemplo, en el caso de administración manual de claves (véase la Sección 5). Sin embargo, el emisor incrementa el contador y cuando alcanza el valor máximo, el contador vuelve nuevamente a cero.

3.3.3 Cálculo del Valor de Comprobación de Integridad

El ICV de AH es calculado sobre:

- Los campos de la cabecera IP que son inmutables en tránsito o que son predecibles en valor al momento de la llegada del paquete IP en los extremos para la SA AH.
- La cabecera de AH: Cabecera Siguiente, Longitud de la Carga,

Reservado, SPI, Número de Secuencia, y los Datos de Autentificación (se fijan a cero para este cálculo), y los bytes explícitos de relleno (si los hay).

- Los datos del protocolo de nivel superior, que se asumen son inmutables en tránsito

3.3.3.1 Manipulación de los Campos Mutables

Si un campo puede ser modificado durante el tránsito, el valor del campo se fija a cero para los propósitos del cálculo del ICV. Si un campo es mutable, pero su valor en el receptor (IPsec) es predecible, entonces ese valor es insertado en el campo para los propósitos del cálculo del ICV. El campo Datos de Autentificación también se fija a cero en preparación para este cálculo. Observe que reemplazando el valor de cada campo por cero, en lugar de omitir el campo, la alineación es preservada para el cálculo del ICV. También, el método de colocar el valor cero asegura que la longitud de los campos que son manipulados no se pueda cambiar durante el tránsito, aun cuando sus contenidos no son cubiertos explícitamente por el ICV.

Si se crea una nueva cabecera de extensión o de opción en IPv4, esta será definida en su propio RFC y DEBERÍA incluir (en la sección de Consideraciones de Seguridad) la forma de cómo se debería manipular el cálculo del ICV de AH. Si la implementación IP (IPv4 o IPv6) encuentra una cabecera de extensión que no reconoce, desechará el paquete y enviará un mensaje ICMP. IPsec nunca verá el paquete. Si la implementación IPsec encuentra una opción IPv4 que no reconoce, debería poner a cero la opción entera, usando el segundo byte de la opción como la longitud. Las opciones de IPv6 (en las cabeceras de extensión de Destino o la de Salto por Salto) contienen una bandera que indica la mutabilidad, que determina el procesamiento apropiado para tales opciones.

3.3.3.1.1 Cálculo de ICV para IPv4

3.3.3.1.1.1 Campos de la Cabecera Base

Los campos de la cabecera de IPv4 se clasifican de la siguiente manera:

Inmutables:
- Versión
- Longitud de la Cabecera Internet (IHL)
- Longitud Total
- Identificación
- Protocolo (éste debería tener el valor para AH.)
- Dirección de Origen
- Dirección de Destino (sin una ruta de destino estricta o libre)

Mutable pero predecible
- Dirección de Destino (con ruta de destino estricta o libre)

Mutable (se colocan a cero antes del cálculo del ICV)
- Tipo de Servicio (TOS)
- Banderas (Flags)
- Desplazamiento del Fragmento
- Tiempo de Vida (TTL)
- Suma de Verificación de la Cabecera

TOS: Este campo es excluido porque se sabe que algunos routers cambian el valor de este campo, aunque la especificación de IP no considera al

TOS como un campo mutable de la cabecera.

Banderas: Este campo es excluido puesto que routers intermedios puede fijar el bit de DF, incluso si el origen no lo seleccionó.

Desplazamiento del Fragmento: Puesto que AH se aplica solamente a paquetes IP no a fragmentados, el Campo Desplazamiento debe ser siempre cero, y así excluido (aunque es predecible).

TTL: Éste es cambiado en ruta como curso normal del procesamiento por routers, y así su valor en el receptor no es predecible por el emisor.

Suma de Verificación de la Cabecera: Esta cambiará si alguno de estos otros campos cambian, y así su valor en la recepción no se puede predecir por el emisor.

3.3.3.1.1.2 Opciones

Para IPv4 (no así para IPv6), no hay mecanismos para marcar opciones como mutables en tránsito. Por lo tanto las opciones IPv4 se en listan explícitamente en la Sección 6 y se clasifican como: inmutables, mutable pero predecible, o mutables. Para IPv4, la opción entera se ve como una unidad; por lo tanto el tipo y longitud de los campos dentro de la mayoría de las opciones son inmutables en tránsito, si una opción se clasifica como mutable, la opción entera se pone en cero para los propósitos del cálculo del ICV.

3.3.3.1.2 Cálculo de ICV para IPv6

3.3.3.1.2.1 Campos de la Cabecera Base

Los campos de la cabecera base IPv6 se clasifican de la siguiente manera:

Inmutable
- Versión
- Longitud de la Carga
- Cabecera Siguiente (ésta debería tener el valor para AH.)
- Dirección de Origen
- Dirección de Destino (sin la Cabecera de Extensión de Ruteo)

Mutable pero predecible
- Dirección de Destino (con la Cabecera de Extensión de Ruteo)

Mutable (puesto a cero para el cálculo de ICV)
- Clase
- Etiqueta de Flujo
- Límite de Saltos

3.3.3.1.2.2 Cabeceras de Extensión que Contienen Opciones

Las opciones IPv6 de las Cabeceras de Extensión de, Salto por Salto y de Destino contienen un bit que indican si la opción puede o no cambiar (de forma impredecible) durante el tránsito. Para cualquier opción para la cual los contenidos puedan cambiar en tránsito, todo el campo "Datos Opcionales" debe ser tratado con valor de cero octetos al calcular o verificar del ICV. El Tipo de Opción y la Longitud de los Datos Opcionales se incluyen en el cálculo del ICV. Todas las opciones para las cuales el bit indica inmutabilidad se incluyen en el cálculo del ICV. Vea la especificación de IPv6 [IPv6] para más información.

3.3.3.1.2.3 Cabeceras de Extensión que no Incluyen Opciones

Las cabeceras de extensión de IPv6 que no contienen opciones se incluyen explícitamente en la Sección 6 y se clasifican como: inmutables, mutable pero predecibles, o mutables.

3.3.3.2 Relleno

3.3.3.2.1 Relleno de los Datos de Autentificación

Según lo mencionado en la sección 2.6, el campo Datos de Autentificación incluye explícitamente el relleno para asegurarse de que la cabecera de AH es un múltiplo de 32 bits (para IPv4) o de 64 bits (para IPv6). Si se requiere el relleno, su longitud es determinada por dos factores:

- la longitud del ICV
- la versión del protocolo IP (IPv4 o IPv6)

Por ejemplo, si la salida del algoritmo seleccionado es de 96 bits, no se requerirá ningún relleno para IPv4 o para IPv6. Sin embargo, si se genera una longitud ICV distinta, debido al uso de un algoritmo diferente, entonces el relleno puede ser requerido dependiendo de la longitud y de la versión del protocolo IP. El contenido del campo de relleno es seleccionado arbitrariamente por el emisor. El relleno es arbitrario, pero necesita no ser aleatorio para lograr seguridad. Estos bytes de relleno se incluyen en el cálculo de los Datos de Autentificación, se cuentan como parte de la Longitud de la Carga y se transmiten al final del campo Datos de Autentificación para permitir al receptor realizar el cálculo del ICV.

3.3.3.2.2 Relleno Implícito del Paquete

Para algunos algoritmos de autentificación, la cadena de bytes sobre la cual el cálculo del ICV se realiza debe ser un múltiplo de un tamaño de bloque especificado por el algoritmo. Si la longitud del paquete IP (incluido AH) no coincide con los requisitos del tamaño de bloque para el algoritmo, el relleno implícito DEBE ser aplicado al final del paquete, antes del cálculo del ICV. Los octetos de relleno DEBEN tener un valor de cero. El tamaño del bloque (y por lo tanto la longitud del relleno) es especificado por la especificación del algoritmo. Este relleno no se transmite con el paquete. Observe que MD5 y SHA-1 tienen un tamaño de bloque de un byte debido a sus convenciones internas del relleno.

3.3.4 Fragmentación

Si se requiere, la fragmentación IP ocurre después del procesamiento de AH dentro de una implementación IPsec. Así, en modo trasporte AH se aplica solamente a los datagramas IP enteros (no a los fragmentos IP). Un paquete IP al cual se ha aplicado AH se puede fragmentar por routers en ruta, y tales fragmentos se deben reensamblar antes de que AH sea procesado por el receptor. En modo túnel, AH se aplica a un paquete IP, el cual la carga puede ser un paquete IP fragmentado. Por ejemplo, en una security gateway o en implementaciones IPsec BITS o BITW (véase el capítulo 2 para más detalles) se puede aplicar AH en modo túnel a tales fragmentos.

3.4 Procesamiento de Paquetes Entrantes

Si hay más de una cabecera/extensión de IPsec presente, el procesamiento para cada una ignorará (no pone a cero, no usa) cualquier cabecera IPsec

subsiguiente aplicada a la cabecera que esta siendo procesada.

3.4.1 Reensamblaje

Si se requiere, el reensamblaje se realiza antes del procesamiento de AH. Si un paquete brindado a AH para procesamiento parece ser un fragmento IP, es decir, el campo de desplazamiento (OFFSET) es diferente a cero o la bandera de MAS FRAGMENTOS (MORE FRAGMENTS) está en uno, el receptor DEBE desechar el paquete; esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, Fecha/Hora, Dirección de Origen, Dirección de Destino, y (en IPv6) el Identificador de Flujo (Flow ID).

Nota: Para el reensamblaje del paquete, IPv4 NO requiere que el campo DESPLAZAMIENTO (OFFSET) sea cero o que este en cero la bandera de MAS FRAGMENTOS. Para que un paquete reensamblado pueda ser procesado por IPsec (contrariamente a descartar un aparente fragmento), el código IP debe hacer dos cosas después de reensamblar un paquete.

3.4.2 Buscando la Asociación de Seguridad

Al recibir un paquete que contiene una Cabecera de Autentificación, el receptor determina la SA (unidireccional) apropiada, basándose en la Dirección de Destino IP, el Protocolo de Seguridad (AH), y el SPI. La SA indica si: se controlará el campo Número de Secuencia, especifica el/los algoritmo/s empleados para el cálculo del ICV, y indica la/s clave/s requerida/s para validar el ICV.

Si no existe ninguna SA válida para esta sesión (por ejemplo, el receptor no tiene ninguna clave), el receptor DEBE desechar el paquete; esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, Fecha/Hora, Dirección de Origen, Dirección de Destino, y (en IPv6) el Identificador de Flujo (Flow ID).

3.4.3 Verificación del Número de Secuencia

Todas las implementaciones de AH DEBEN soportar el servicio de anti-replay, aunque su uso puede estar habilitado o deshabilitado por el receptor sobre la base de una SA. Observe que no hay soporte para administrar los valores de los Números de Secuencia transmitidos entre múltiples emisores que dirigen el tráfico a una única SA (independientemente de que si la dirección de destino es unicast, broadcast, o multicast). Así el servicio de anti-replay NO DEBERÍA ser usado en ambientes múlti-emisor que emplee una única SA.

Si el receptor no habilita el anti-replay para una SA, no se realizará las comprobaciones entrantes en el Número de Secuencia. Sin embargo, desde la perspectiva del emisor el valor por defecto es asumir que el anti-replay esta habilitado en el receptor. Para evitar que el emisor haga un monitoreo innecesario del número de secuencia y el establecimiento de una SA (ver Sección 3.3.2), si un protocolo de establecimiento de SA tal como IKE se emplea (ver el Capítulo 10), el receptor DEBERÍA notificar al emisor, durante el establecimiento de una SA, si el receptor no proporcionará la protección anti-replay.

Si el receptor tiene habilitado el servicio de anti-replay para esta SA, el contador de recepción de paquetes para la SA, se debe inicializar en cero cuando la SA es establecida. Para cada paquete recibido, el receptor DEBE verificar que el paquete contiene un Número de Secuencia que no es igual al

Número de Secuencia de ningún otro paquete recibido durante la vida de esa SA. Este DEBERÍA ser el primer control de AH aplicado a un paquete después de que haya sido correspondido a una SA, para acelerar el rechazo de paquetes duplicados.

Los paquetes duplicados son rechazados a través del uso de una ventana de recepción deslizable. Un tamaño de ventana mínimo de 32 DEBE ser soportado; pero un tamaño de ventana de 64 es más aconsejable y DEBERÍA ser empleado como valor por defecto. Otro tamaño de ventana (más grande que el mínimo) PUEDE ser elegido por el receptor. El receptor no notifica al emisor del tamaño de ventana.

El lado "Derecho" de la ventana representa el valor del Número de Secuencia más alto autentificado y recibido en esta SA. Los paquetes que contienen Números de Secuencias menores que el lado "izquierdo" de la ventana son rechazados. Los paquetes que caen dentro de la ventana son controlados con una lista de paquetes recibidos dentro de la ventana. Un modo eficiente de realizar este control, basado en el uso de un bit mask, se describió en el capítulo anterior.

Si el paquete recibido cae dentro de la ventana y es nuevo, o si el paquete esta a la derecha de la ventana, el receptor procede con la verificación del ICV. Si la verificación ICV falla, el datagrama IP recibido no es válido y el receptor DEBE descartar el paquete. Esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, fecha/hora recibido, Dirección de Origen, Dirección de Destino, Número de Secuencia, y (en IPv6) el Identificador de Flujo. La ventana de recepción es actualizada solo si la verificación del ICV tiene éxito.

Observe que si el paquete esta dentro de la ventana y es nuevo, o si esta fuera de la ventana en el lado "derecho", el receptor DEBE autentificar el paquete antes de actualizar el valor de la ventana del Número de Secuencia.

3.4.4 Verificación del Valor de Comprobación de Integridad

El receptor calcula el ICV sobre los campos apropiados del paquete, usando el algoritmo de autentificación especificado, y verifica que es el mismo que el ICV incluido en el campo Datos de Autentificación de el paquete.

Si el ICV calculado y recibido concuerdan, el datagrama es válido, y es aceptado. Si el control falla, el receptor debe descartar el datagrama IP recibido porque no es válido; esto es un evento auditadle. Los datos del registro de auditoría deberían incluir el valor del SPI, fecha/hora recibido, Dirección de Origen, Dirección de Destino, Número de Secuencia, y (en IPv6) el Identificador de Flujo.

El procedimiento para el cálculo de comprobación del valor del ICV es:

> Comience guardando el valor ICV y reemplácelo con cero. Ponga a cero el resto de los campos que han sido modificados durante el tránsito. (Ver la Sección 3.3.3.1 para una discusión sobre que campos son puestos a cero antes de realizar el cálculo del ICV.) Controle la longitud total del paquete, y si se requiere relleno implícito basado en los requerimientos del algoritmo de autentificación, se agregan los bytes de relleno con valor cero en el extremo del paquete como es requerido. Realice el cálculo del ICV y compare el resultado con el valor guardado, usando las reglas de comparación definidas por la especificación del algoritmo. Si una firma digital y un hash unidireccional se utilizan

para el cálculo del ICV, el proceso de correspondencia es más complejo.

4. **Auditoría**

No todos los sistemas que implementan AH implementarán auditoría. Sin embargo, si AH es incorporado a un sistema que soporta auditoría, la implementación AH debe también soportar auditoría y debe permitirle a un administrador de sistema habilitar o deshabilitar la auditoría para AH. La granularidad de la auditoría es un tema local. Sin embargo, varios eventos auditadles se identifican en esta especificación y para cada uno de estos eventos un conjunto mínimo de información debería ser incluido en el registro de auditoría. Información adicional también puede ser incluida en el registro de auditoría para cada uno de estos eventos, y los eventos adicionales, no explícitamente exigidos en esta especificación, también pueden resultar en entradas del registro de auditoría. No hay requisito para el receptor de trasmitir ningún mensaje al emisor pretendido en respuesta a la detección de un evento auditadle, debido al potencial de inducir la Denegación de Servicio a través de tal acción.

5. **Requerimiento de Conformidad**

Las implementaciones deben implementar la síntesis de AH, los procesos descriptos aquí y cumplir con todos los requisitos del capítulo 2. Si la clave usada para calcular un ICV es distribuida manualmente, la correcta provisión del servicio anti-replay requerirá el correcto estado del contador en el emisor, hasta que la clave es reemplazada y no habría probablemente disponibilidad automatizada de recuperación si el desbordamiento del contador fuera inminente. Así, una implementación no DEBERÍA proporcionar este servicio en conjunto con SAs que generan claves manuales. Una implementación AH debe soportar e implementar obligatoriamente los siguientes algoritmos (Véase el Capítulo 6, que explica estos algoritmos):

- HMAC con MD5
- HMAC con SHA-1

6. **Mutabilidad de Opciones IP/Cabeceras de Extensión**

6.1 Opciones de IPv4

Esta tabla muestra como las opciones de IPv4 están clasificadas de acuerdo a la "mutabilidad". Donde dos referencias son proporcionadas, la segunda sustituye a la primera. Esta tabla está basada en la información proporcionada en el RFC 1700, NÚMEROS ASIGNADOS, (octubre 1994).

Copiar	Clase	N° de Opción	Nombre	Referencia
INMUTABLE ------ Incluidos en el cálculo de ICV				
0	0	0	Final de la lista de opciones	[IPv4]
0	0	1	No operación	[IPv4]
0	0	2	Seguridad	[Ken91]
1	0	5	Extensión de seguridad	[Ken91]
1	0	6	Seguridad comercial	Ahora uso militar
1	0	20	Alerta de Router	[RoutAlert]
1	0	21	dirección del emisor de entrega multi-destino	[RFC1770]
MUTABLE -------- Poner a cero				
1	0	3	Ruta de origen no fija	[IPv4]
0	2	4	Fecha de registro	[IPv4]
0	0	7	Registrar ruta	[IPv4]
1	0	9	Ruta de origen estricta	[IPv4]
0	2	18	Traceroute	[RFC1393]
EXPERIMENTAL, SUSTITUIR, ---- Poner a cero				
1	0	8	Identificador de Flujo	[IPv4], [RFC1122]
0	0	11	Prueba de MTU	[PMTU]*
0	0	12	MTU Reply	[PMTU]*
1	0	17	Extended Internet Proto	[RFC1063, [PMTU]
0	0	10	Medición experimental	[IAMA-4]
1	2	13	Control de Flujo experimental	[IAMA-4]
1	0	14	Control de Acceso Experimental	[IAMA-4]
0	0	15	?	[IAMA-4]
1	0	16	Descriptor de tráfico IMI	[IAMA-4]
1	0	19	Extensión de direcciones	[IAMA-4]

* = según [IAMA-4] actualmente obsoleto.

Nota: El uso de la opción de alerta de router es incompatible con el uso de IPsec. Aunque la opción es inmutable, su uso implica que cada router a través de la trayectoria del paquete "procesará" el paquete y consecuentemente podría cambiar el paquete, esto podría pasar, en las bases de un salto por salto a medida de que el paquete vaya de router a router. Antes de ser procesado por la aplicación por la cual los contenidos están controlados, por ejemplo, RSVP/IGMP, el paquete debería ser procesado por AH.

Sin embargo el procesamiento de AH, requerirá que cada router a través de la trayectoria sea miembro de una SA multicast definida por el SPI. Esto puede plantear problemas para los paquetes que no están encaminados a un origen estricto, y requiere que las técnicas de soporte multicast no estén disponibles.

NOTA: el agregado o el removido de cualquier etiqueta de seguridad (BSO, ESO, CIPSO), por sistemas a través de la trayectoria de un paquete esta en conflicto con la clasificación de inmutables de estas Opciones IP y es incompatible con el uso de IPsec.

NOTA: Las opciones Final de la Lista de Opciones DEBERÍA ser repetida como sea necesario para asegurar que la cabecera IP termina en un limite de 4 bytes para asegurar que no hay bytes no especificados que se podrían utilizar para un canal secreto.

6.2 Cabeceras de Extensión de IPv6

Esta tabla muestra como las Cabeceras de Extensión de IPv6 están clasificadas de acuerdo a la "mutabilidad".

Opción/Extensión Nombre	Referencia
MUTABLE PERO PREDECIBLE --- Incluidos en el cálculo de ICV	
Ruteo (Tipo 0)	[IPv6]
EL BIT INDICA SI LA OPCIÓN ES MUTABLE (CAMBIA EN FORMA IMPREDECIBLE DURANTE EL TRÁNSITO	
Opción Salto por Salto	[IPv6]
Opciones de Destino	[IPv6]
NO APLICABLE	
Fragmentación	[IPv6]

Opciones: Las cabeceras de opción de Salto por Salto y de Destino de IPv6 contienen un bit que indican si la opción puede cambiar impredeciblemente durante el tránsito. Para cada opción cuyo contenido puede cambiar en ruta, el campo entero "Datos de Opción" debe ser tratado como octetos con valor cero al momento del cálculo o verificación del ICV. El campo Tipo de Opción y la Longitud de los Datos de la Opción están incluidos en el cálculo del ICV. Todas las opciones cuyos bits indiquen inmutabilidad están incluidas en el cálculo del ICV. Ver la especificación de IPv6 [IPv6] para más información.

Ruteo (Tipo 0): La Cabecera de Ruteo de IPv6 "Tipo 0" cambiará solo los campos de dirección dentro del paquete durante el tránsito del origen al destino. Sin embargo, los contenidos del paquete como aparecerán en el receptor y todos los saltos intermedios, son conocidos por el emisor. Por lo tanto, la Cabecera de Ruteo de IPv6 "Tipo 0" esta incluida en el cálculo de los Datos de Autentificación como mutable pero predecible. El emisor debe ordenar el campo de tal forma de que aparezca como se verá en el receptor, antes de realizar el cálculo del ICV.

Fragmentación: Ocurre después del procesamiento IPsec saliente (Sección 3.3) y el reensamblaje ocurre antes del procesamiento IPsec de entrada (Sección 3.4). Por lo tanto la Cabecera de Extensión de Fragmentación, si existe, no es vista por IPsec.

Observe que en el lado del receptor, la implementación IP podría dejar una Cabecera de Extensión de Fragmentación en su sitio al momento de hacer el reensamblaje. Si esto pasa, cuando AH recibe el paquete, antes de realizar el proceso ICV, AH debe "quitar" (o saltarla) esta cabecera y cambiar la cabecera anterior del campo "Cabecera Siguiente" para que sea el campo "cabecera siguiente" en la Cabecera de Extensión de Fragmentación.

Observe que en el lado del emisor, la implementación IP podría dar al código IPsec un paquete con una Cabecera de Extensión de Fragmentación con un Desplazamiento de cero (primer fragmento) y una Bandera de Más Fragmentos de cero (ultimo fragmento). Si esto pasa, antes de hacer el proceso ICV, AH primero debe "quitar" (o saltar sobre) esta cabecera y cambiar la cabecera anterior del campo "Cabecera Siguiente" para que sea el campo "cabecera siguiente" en la Cabecera de Extensión de Fragmentación.

Capítulo 4

Cabecera de Encriptación

1. **Introducción**

La cabecera de Carga de Seguridad Encapsulada (ESP) está diseñada para proporcionar un conjunto de servicios de seguridad en IPv4 y en IPv6. ESP puede ser aplicado solo, o en conjunto con la Cabecera de Autentificación (AH) (véase el capítulo 3), o en forma anidada, por ejemplo, a través del uso del modo túnel (véase el capítulo 2). Los servicios de seguridad pueden ser suministrados a comunicaciones, entre un par de hosts, o entre un par de security gateway (SG), o entre security gateway y un host. Para más detalles de cómo se usa ESP y AH en varios ambientes de red, vea el capítulo 2.

La cabecera ESP se inserta antes que la cabecera IP y después que la cabecera de protocolo de capa superior (en modo transporte) o después de una cabecera IP encapsulada (en modo túnel).

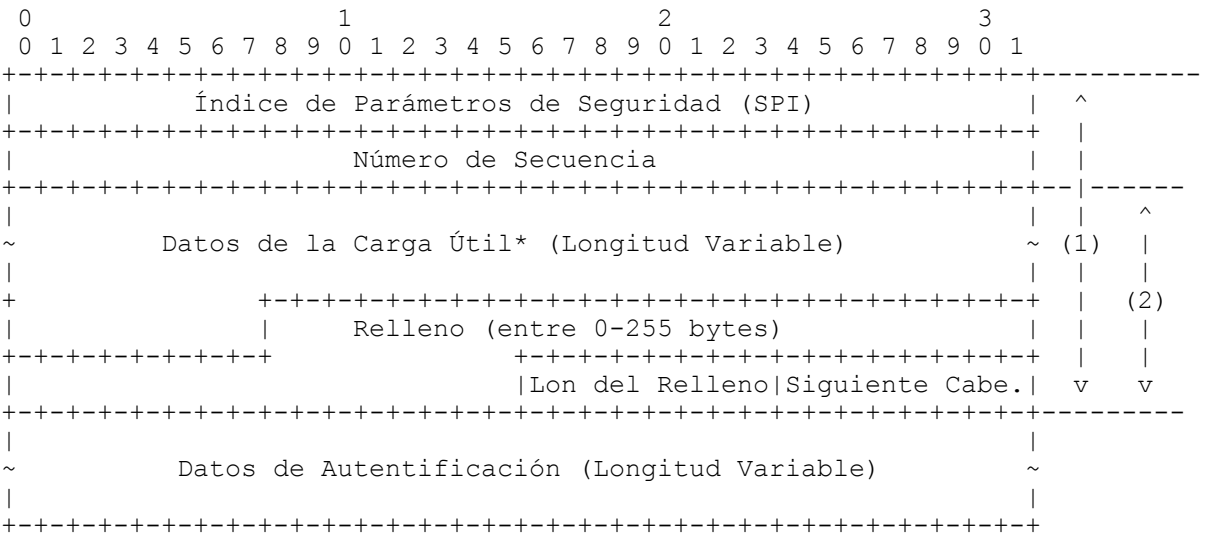
ESP es usado para proporcionar confidencialidad, autentificación del origen de los datos, integridad sin conexión, un servicio de anti-replay (una forma parcial de integrabilidad de secuencia) y confidencialidad limitada del flujo de tráfico. El conjunto de servicios proporcionados depende de las opciones seleccionadas al momento del establecimiento de la Asociación de Seguridad (SA) y de dónde esté localizada la implementación. La confidencialidad puede ser seleccionada independientemente del resto de los servicios. No obstante el uso de la confidencialidad sin integridad/autentificación (en ESP o en AH) puede subordinar tráfico hacia ciertos tipos de ataques activos que podrían socavar el servicio de confidencialidad (ver [Bell96]). La autentificación del origen de los datos y la integridad sin conexión son servicios que están unidos (de aquí en adelante a ambos servicios se los denominará como "autentificación") y son ofrecidos como una opción junto con la confidencialidad (opcional). El servicio de anti-replay puede ser seleccionado únicamente si la autentificación del origen de los datos es seleccionado, y esta elección esta supeditada solamente al albedrío del receptor. (Aunque el valor por defecto exige que el emisor incremente el Número de Secuencia usado para el anti-replay, el servicio es efectivo solamente si el receptor controla el Número de Secuencia.) La confidencialidad del flujo de tráfico requiere de la selección del modo túnel, y es más efectiva si esta implementada en una security gateway donde la agregación de tráfico puede encubrir patrones verdaderos del originador y del destinatario. Observe que aunque la confidencialidad y la autentificación son opcionales, al menos una de ellas debe ser seleccionada.

Nota: las opciones requeridas actualmente para el manejo de claves tanto para AH como para ESP son el modo manual y en el modo automatizado por medio de IKE (véase el Capítulo 10).

Nota: Para poder llegar a comprender íntegramente este capítulo es necesario que se haya comprendido el capítulo de la Arquitectura de Seguridad, el capítulo de ISAKMP, el de IKE (y por consiguiente el de OAKLEY), lo que ocurre es que para comprender los capítulos antes mencionados es necesario haber "visto" este capítulo antes.

2. **Formato del Paquete de la Carga de Seguridad Encapsulada**

La cabecera del protocolo (IPv4, IPv6, o de Extensión) inmediatamente antes de la cabecera de ESP contendrá el valor 50 en su Protocolo (IPv4) o en el campo Siguiente Cabecera (de IPv6, o de Extensión) [STD-2].



(1) Alcance de la Autentificación

(2) Alcance de la Confidencialidad

* Si se incluye en el campo Carga útil los datos de sincronización criptográfica, (por ejemplo, un Vector de Inicialización (IV), ver Sección 2.3) usualmente estos no estarán encriptados, aunque a menudo se lo hace referencia como parte del texto cifrado.

Las secciones subsiguientes definen el formato de los campos en la cabecera. "Opcional" significa que el campo es omitido si la opción no es seleccionada, es decir, no esta presente ni en el paquete ni como trasmitido ni como formateado para el cálculo del Valor de Comprobación de Integridad (ICV, ver Sección 3.7). Si una opción es o no seleccionada es definido como parte del establecimiento de la Asociación de Seguridad (SA). Así, el formato de los paquetes ESP para una SA dada es fijo, para la duración de la SA. En cambio, los campos "obligatorios" están siempre presentes en el formato del paquete ESP, para todas las SAs.

2.1 Índice de Parámetros de Seguridad (SPI) (SPI)

El SPI es un valor arbitrario de 32 bits que, conjuntamente con la dirección de destino IP y el protocolo de seguridad (ESP), identifican unívocamente a la Asociación de Seguridad para este datagrama. El conjunto de valores de SPI en el rango de 1 a 255 está reservado por la IANA para uso futuro; un valor reservado de SPI no será destinado normalmente por el IANA a menos que el uso del valor destinado de SPI se especifique en un RFC. Este es seleccionado por el sistema de destino sobre el establecimiento de una SA. El campo SPI es obligatorio.

El valor de SPI cero (0) esta reservado para usarse localmente, las implementaciones no deben trasmitir este valor por la red. Por ejemplo, una implementación de administración de clave PUEDE utilizar el valor cero de SPI para denotar que "No Existe Asociación de Seguridad" durante el período en el cual la implementación IPsec ha solicitado a la entidad administradora de claves que se establezca una nueva SA, pero la SA todavía no se ha establecido.

La necesidad del SPI se hace evidente cuando tenemos más de una comunicación con la misma dirección IP de destino y protocolo de seguridad (AH o ESP).

2.2 Número de Secuencia

Campo de 32 bits sin signo que contiene un valor crecientes y único del contador (del número de secuencia). Es obligatorio y debe estar siempre presente incluso si el receptor elige no habilitar el servicio de anti-replay para una SA específica. El procesamiento del campo Número de Secuencia esta a criterio del receptor, es decir, el emisor DEBE transmitir siempre este campo, pero el receptor no necesita actuar sobre él.

El contador del emisor y del receptor se inicializan a cero (0) cuando se establece una SA. (El primer paquete que se envíe bajo esa SA tendrá el Número de Secuencia 1; vea la Sección 3.3.3 para más detalles de cómo se genera el Número de Secuencia.) Si se habilita el anti-replay (por defecto), la transmisión del Número de Secuencia nunca debe permitir que el Número de Secuencia retorne a cero. Por ende, el contador del emisor y del receptor DEBEN ser resetiados (para el establecimiento de una nueva SA y de esta manera también una nueva clave) antes de que se trasmitan 2^32 paquetes sobre una SA.

2.3 Datos de la Carga Útil

Los Datos de la Carga Útil es un campo de longitud variable que contiene los datos descriptos por el campo Siguiente Cabecera. El campo Datos de la Carga Útil es obligatorio y su longitud es un número de bytes enteros. Si el algoritmo usado para encriptar a la carga útil requiere datos de sincronización criptográficos, por ejemplo, de un Vector de Inicialización (IV), estos datos se PUEDEN llevar explícitamente en el campo Carga Útil (vea la Sección 4.1.2 del Capítulo 5 para una descripción detallada del Vector de Inicialización). Cualquier algoritmo de encriptación que requiera tales datos explícitos, BEBERÍA enviar una paquete previamente de sincronización indicando: la longitud, la estructura para tales datos, y la localización de estos datos de sincronización criptográficos como parte de la especificación del RFC del algoritmo que se utiliza con ESP. Si tales datos de sincronización son implícitos, el algoritmo para derivar los datos DEBE ser parte del RFC.

Note que en cuanto a la certeza de alinear el (verdadero) texto cifrado en presencia de un IV se observa que:

- Para alguno de los modos de operación basados en IV, el receptor trata el IV como el comienzo del texto cifrado, introduciéndolo dentro del algoritmo directamente. En estos modos, la alineación del comienzo del (verdadero) texto cifrado no es asunto del receptor.

- En algunos casos, el receptor lee el IV por separado del texto cifrado. En estos casos, la especificación del algoritmo DEBE tratar la forma de alinear el (verdadero) texto cifrado.

2.4 Relleno (para la Encriptación)

Varios factores requieren o motivan el uso del campo Relleno, algunos de ellos son:

- Si se emplea un algoritmo de encriptación que requiere que el texto plano sea un múltiplo de un cierto número de bytes, por ejemplo, el

tamaño de bloque de un bloque cifrado, el campo Relleno es usado para rellenar el texto plano (el cual consta de los Datos de la Carga Útil, y los campos Longitud del Relleno y Siguiente Cabecera, así como también del Relleno) para el tamaño requerido por el algoritmo.

- El relleno también puede ser requerido, independientemente de los requisitos del algoritmo de encriptación, para asegurarse de que el texto cifrado resultante termine en un límite de 4 bytes. Específicamente, los campos Longitud del Relleno y Siguiente Cabecera deben estar alineados correctamente dentro de una palabra de 4 bytes, según lo ilustrado en la figura del formato del paquete ESP, para asegurarse de que el campo Datos de Autentificación (si está presente) esté alineado en un límite de 4 bytes.

- Más allá del relleno requerido para el algoritmo o por las razones de alineación citadas arriba, se puede utilizar para encubrir la longitud real de la carga, en respaldo de la confidencialidad (parcial) del flujo de tráfico. Sin embargo, la inclusión de tal relleno adicional tiene implicaciones adversas en el ancho de banda y su uso bebe ser emprendido con cautela.

El emisor PUEDE agregar de 0 a 255 bytes de relleno. La inclusión del campo Relleno en un paquete ESP es opcional, pero todas las implementaciones DEBEN soportar la generación y el uso del relleno.

a. Con el fin de asegurarse de que los bits que se encriptarán sean múltiplo del tamaño del bloque del algoritmo (primer punto de arriba), el cómputo del relleno se aplica a los campos: Datos de la Carga Útil (no incluyendo los del IV), al campo Longitud del Relleno, y al campo Siguiente Cabecera.

b. Para los propósitos de asegurarse de que los Datos de Autentificación estén alineados en un límite de 4 bytes (segundo punto de arriba), el cómputo del relleno se aplica a los campos: Datos de la Carga Útil (incluyendo los del IV), al campo Longitud del Relleno, y al campo Siguiente Cabecera.

Si son necesarios los bytes de relleno pero el algoritmo de encriptación no especifica el contenido del relleno, entonces el proceso que se describe a continuación (proceso por defecto) DEBE ser utilizado:

Los bytes de Relleno se inicializan con una serie de (bytes sin signo) de valores de números enteros. El primer byte del relleno añadido al texto plano se lo numera como 1, con los bytes subsiguientes de relleno formado por una secuencia sucesiva creciente: 1, 2, 3,...

Cuando se emplea este esquema de relleno, el receptor DEBERÍA examinar el campo Relleno. Este esquema fue seleccionado debido a su simplicidad relativa, fácil implementación en hardware, y porque ofrece protección limitada contra ciertas formas de ataques del tipo "copiar y pegar" en ausencia de otras medidas de integridad, si el receptor controla los valores del relleno sobre la desencriptación.

Cualquier algoritmo de encriptación que requiera Relleno con excepción del valor por defecto descripto arriba, DEBE definir el contenido del Relleno (por ejemplo, ceros o datos aleatorios) y cualquier proceso requerido por el receptor de estos bytes de Relleno debe estar especificado en un RFC que especifique como se usa el algoritmo con ESP. En tales circunstancias, el contenido del campo Relleno será determinado por el algoritmo de

encriptación y el modo seleccionado y definido en el RFC correspondiente del algoritmo. El RFC relevante del algoritmo PUEDE especificar que un receptor DEBE examinar el campo Relleno o que un receptor DEBE informar al emisor cómo el receptor manejará el campo Relleno.

2.5 Longitud del Relleno

El campo Longitud del Relleno indica el número de bytes de relleno inmediatamente precedentes a este campo. El rango de valores válidos es de 0 a 255 bytes, donde un valor de cero indica que no hay bytes de Relleno presentes. El campo Longitud del Relleno es obligatorio.

2.6 Siguiente Cabecera

El campo Siguiente Cabecera es un campo de 8 bits que identifica el tipo de datos contenidos en el campo Datos de la Carga Útil, por ejemplo, una cabecera de extensión IPv6 o un identificador de protocolo de capa superior. El valor de este campo se elige del conjunto de Números de Protocolo IP definidos en [STD-2] por la IANA.

2.7 Datos de Autentificación

El campo Datos de Autentificación es de longitud variable y contiene el Valor de Comprobación de Integridad (ICV) calculado sobre el paquete ESP menos, los Datos de Autentificación. La longitud del campo es especificada por la función de autentificación seleccionada. El campo Datos de Autentificación es opcional, y se incluye solamente si el servicio de autentificación se ha seleccionado para la SA en cuestión. La especificación del algoritmo de autentificación DEBE especificar la longitud del ICV y las reglas de comparación y los pasos para el procesamiento de validación.

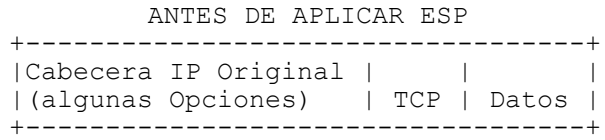
3. **Procesamiento del Protocolo ESP**

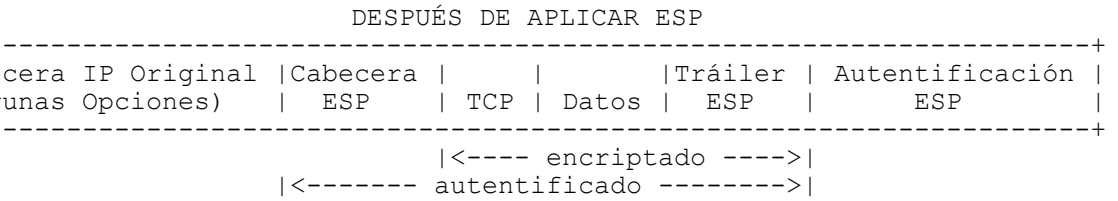
3.1 Localización de la Cabecera ESP

Como AH, ESP puede ser empleado de dos maneras: modo transporte o modo túnel. El primer modo es aplicable solamente a implementaciones host y proporciona la protección para los protocolos de capa superiores, pero no a la cabecera IP. En este modo, observe que para las implementaciones BITS o BITW, los fragmentos IP entrantes y salientes pueden requerir que una implementación IPsec realice un reensamblaje/fragmentación IP adicional a fin de que ambos cumplan con las especificaciones y proporcionen un soporte IPsec transparente. Cuidado especial se requiere para realizar tales operaciones dentro de estas implementaciones cuando múltiples interfaces se están usando.

En modo transporte, ESP se inserta después de la cabecera IP y antes del protocolo de capa superior, por ejemplo, TCP, UDP, ICMP, etc. o antes que cualquier otra cabecera IPsec que se haya insertado. En el contexto de IPv4, esto se traduce en la colocación de ESP después de la cabecera IP (y de cualquiera de las opciones que contenga), pero antes del protocolo de capa superior. (Observe que el término modo "transporte" no debería ser mal interpretado restringiendo su uso solamente a TCP y UDP. Por ejemplo, un mensaje ICMP SE PUEDE enviar usando modo "transporte" o modo "túnel".) El diagrama siguiente ilustra el "antes y el después" de ESP en modo transporte ubicado en un paquete típico de IPv4. El "tráiler" de ESP incluye los campos: Relleno, Longitud del Relleno, y Siguiente Cabecera.

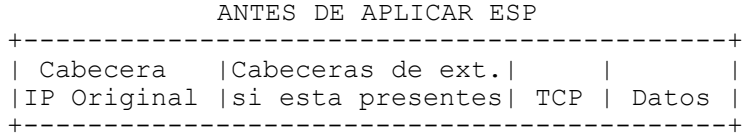
Paquete IPv4

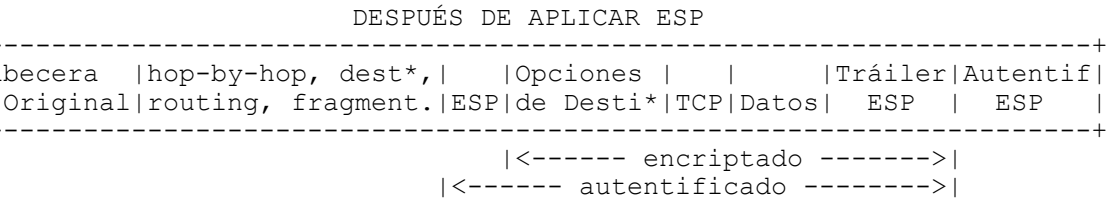




En el contexto de IPv6, ESP se ve como carga útil extremo a extremo, y por ende debe aparecer después de las cabeceras de extensión de salto-por-salto (hop-by-hop), ruteo (routing) y de fragmentación. La cabecera de extensión opciones de destino podría aparecer antes o después de la cabecera ESP dependiendo de la semántica deseada. Sin embargo, puesto que ESP protege solamente a los campos que están después de la cabecera ESP, generalmente puede ser deseable colocar la cabecera Opciones de Destino después de la cabecera ESP. El diagrama siguiente ilustra a ESP en modo transporte ubicado en un paquete típico de IPv6.

Paquete IPv6



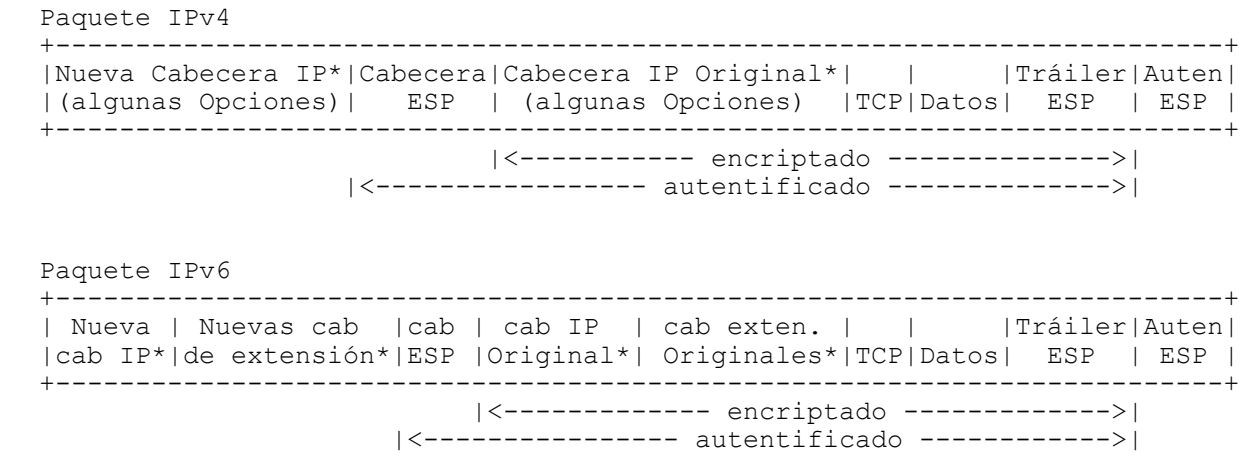


* = la cabecera Opciones de Destino si esta presente, podría estar antes de ESP, después de ESP, o en ambos

Las cabeceras ESP y AH se pueden combinar de varias formas. El Capítulo 2 describe las combinaciones de asociaciones de seguridad que deben ser soportadas.

ESP en modo túnel puede ser empleado en hosts o security gateways. Cuando se implementa ESP en una security gateway (para proteger el tráfico en tránsito del suscriptor), se debe utilizar el modo túnel. En modo túnel, la cabecera IP "interna" lleva las últimas direcciones de origen y de destino, mientras que una cabecera IP "externa" puede contener direcciones IP distintas, por ejemplo, las direcciones de las security gateways. En modo túnel, ESP protege a todo el paquete IP interno, incluyendo toda la cabecera IP interna. La posición de ESP en modo túnel, concerniente a la

cabecera externa IP, es igual que para ESP en modo transporte. El diagrama siguiente ilustra ESP en modo túnel ubicado en un paquete típico de IPv4 y de IPv6.



* = si está presente, construir las cabeceras externas IP/de extensión y modificar las cabeceras internas IP/de extensión según lo discutido posteriormente.

3.2 Algoritmos

Los algoritmos que se deben implementar obligatoriamente se describen en la Sección 5, "Requerimientos de Conformidad". Otros algoritmos PUEDEN ser soportados. Observe que aunque la confidencialidad y la autentificación son opcionales, por lo menos uno de estos servicios DEBE ser seleccionado, por lo tanto, ambos algoritmos NO DEBEN ser NULL simultáneamente.

3.2.1 Algoritmos de Encriptación

El algoritmo de encriptación empleado es especificado por la SA. ESP esta diseñado para usarse con algoritmos de encriptación simétricos (vea el Capítulo 5). Debido a que los paquetes IP pueden llegar en desorden, cada paquete debe llevar necesariamente algún tipo de dato para permitir que el receptor establezca la sincronización criptográfica para la desencriptación. Estos datos se pueden llevar explícitamente en el campo carga útil, por ejemplo, un IV (como se describió anteriormente), o los datos pueden ser derivados de la cabecera del paquete. Puesto que ESP establece normas para el relleno del texto plano, los algoritmos de encriptación empleados con ESP pueden exhibir características de encriptación en modo bloque o de flujo (secuencial). Observe que puesto que la encriptación (confidencialidad) es opcional, este algoritmo puede ser "NULL".

3.2.2 Algoritmos de Autentificación

El algoritmo de autentificación empleado para el cálculo del ICV esta especificado por la SA. Para la comunicaciones punto a punto, los algoritmos de autentificación más aptos incluyen claves con Código de Autentificación de Mensaje (MACs) basados en algoritmos de encriptación simétricos (por ejemplo, 3DES) o funciones hash unidireccionales (por ejemplo, MD5 o SHA-1). Para comunicación multicast, los algoritmos hash

unidireccionales combinados con algoritmos de firmas asimétricas son apropiados, aunque las consideraciones de funcionamiento y de espacio actual imposibilitan el uso de tales algoritmos. Observe que puesto que la autentificación es opcional, este algoritmo puede ser "NULL".

3.3 Procesamiento de Paquetes Salientes

En modo transporte, el emisor encapsula la información del protocolo de la capa superior dentro de ESP, y mantiene la cabecera IP especificada (y cualquiera de las cabeceras IP de extensión, en el contexto de IPv6). En modo túnel, la cabecera/extensiones IP externas e internas se pueden interrelacionar de diversas formas. La construcción de las cabecera/extensiones IP externas realizada durante el proceso de la encapsulación se describió en el Capítulo 2. Si se requiere más de una cabecera/extensión IPsec debido a la política de seguridad, el orden de aplicación de las cabeceras de seguridad se DEBE definir en la política de seguridad.

3.3.1 Buscando la Asociación de Seguridad

ESP se aplica a un paquete saliente solamente después que una implementación IPsec determine que el paquete está asociado con una SA la cual requiere el procesamiento de ESP. El proceso de determinar qué (si existe alguno) procesamiento IPsec se aplica al tráfico saliente, se describió en el Capítulo 2.

3.3.2 Encriptación del Paquete

En esta sección, hablamos del término encriptación siempre aplicado a las implicaciones del formato. Esto se hace con la comprensión que no se ofrece "confidencialidad" usando el algoritmo de encriptación NULL. Por consiguiente, el emisor:

1. Encapsula (dentro del campo Carga Útil de ESP):
   - Para el modo transporte: solo la información primitiva del protocolo de la capa superior.
   - Para el modo túnel: el datagrama IP primitivo entero.
2. Agregar cualquier relleno necesario.
3. Encriptar el resultado (Datos de la Carga Útil, Relleno, Longitud del Relleno, y Siguiente Cabecera) usando la clave, el algoritmo de encriptación, el modo del algoritmo indicado por la SA y los datos de sincronización criptográficos (si hay).
   - Si los datos de sincronización criptográficos son explícitos, por ejemplo, un IV, es indicado, estos se ingresan en el algoritmo de encriptación según la especificación del algoritmo y se colocan en el campo Carga Útil.
   - Si los datos de sincronización criptográficos son implícitos, por ejemplo, un IV, es indicado, estos se construyen y se ingresan en el algoritmo de encriptación según la especificación del algoritmo.

Los pasos exactos para la construcción de la cabecera IP externa dependen del modo (transporte o túnel) y se describieron en el Capítulo 2.

Si se selecciona la autentificación, la encriptación se realiza primero (antes que la autentificación) y la encriptación no abarca el campo Datos de Autentificación. Este orden de procesamiento facilita la rápida detección y rechazo de paquetes re-enviados o falsos para el receptor, antes de desencriptar el paquete, por lo tanto reduciendo potencialmente el impacto de ataques de denegación de servicio. También permite la posibilidad de procesamiento en paralelo de paquetes en el receptor, es decir, la desencriptación puede ocurrir paralelamente a la autentificación. Observe que debido a que los Datos de Autentificación no están protegidos por la encriptación, un algoritmo de autentificación de claves debe ser empleado para calcular el ICV.

3.3.3 Generación del Número de Secuencia

El contador del emisor es inicializado a cero (0) cuando se establece una SA. El emisor incrementa el Número de Secuencia para esta SA e inserta el nuevo valor dentro del Campo Número de Secuencia. Así, el primer paquete enviado usando una SA dada tendrá un valor de Número de Secuencia de 1.

Si se habilita el anti-replay (por defecto), el emisor controla para asegurarse que el contador no ha completado un ciclo antes de insertar el nuevo valor en el campo Número de Secuencia. Es decir, el emisor NO DEBE enviar un paquete en una SA, si al hacerlo haría que el Número de Secuencia complete un ciclo. Una tentativa de transmitir un paquete que resultaría en un desbordamiento del Número de Secuencia es un evento auditable. (Observe que este método de administración del Número de Secuencia no requiere el uso de la aritmética modular, vea el Capítulo 5.)

El emisor asume que el anti-replay es habilitado por defecto, a menos que sea notificado de otra cosa por el receptor (véase la Sección 3.4.3). Así, si el contador ha completado un ciclo, el emisor establecerá una nueva SA y una clave (a menos que la SA haya sido configurada con administración manual de claves).

Si el anti-replay esta deshabilitado, el emisor no necesita monitorear o volver a cero el contador, por ejemplo, en el caso de administración manual de claves (véase la Sección 5). Sin embargo, el emisor incrementa el contador y cuando alcanza el valor máximo, el contador vuelve otra vez a cero.

3.3.4 Cálculo del Valor de Comprobación de Integridad (ICV)

Si la autentificación es seleccionada para la SA, el emisor calcula el ICV sobre el paquete ESP menos los Datos de Autentificación. Así el SPI, el Número de Secuencia, los Datos de la Carga Útil, el Relleno (si esta presente), la Longitud del Relleno, y la Siguiente Cabecera son abarcados por el cálculo del ICV. Observe que los últimos 4 campos estarán en forma de texto cifrado, puesto que la encriptación se realiza antes de la autentificación.

Para algunos algoritmos de autentificación, la cadena de byte sobre la cual se calcula el ICV debe ser un múltiplo de un tamaño de bloque especificado por el algoritmo. Si la longitud de esta cadena de bytes no corresponde con los requisitos del tamaño de bloque para el algoritmo, el relleno implícito DEBE ser añadido al final del paquete ESP, (después del campo Siguiente Cabecera) antes del cálculo del ICV. Los octetos de relleno DEBEN tener un valor de cero. El tamaño del bloque (y por lo tanto la longitud del relleno) es especificado en la especificación del algoritmo. Este relleno no es transmitido con el paquete. Observe que MD5 y SHA-1 son vistos como

que tienen un tamaño de bloque de 1 octeto debido a sus convenciones internas de relleno.

3.3.5 Fragmentación

Si se requiere, la fragmentación se realiza después del procesamiento ESP dentro de una implementación de IPsec. Así, en ESP en modo transporte se aplica solamente a datagramas IP enteros (no a fragmentos IP). Un paquete al cual se le ha aplicado ESP puede ser fragmentado por routers en ruta, y tales fragmentos se deben volver a reensamblar antes de que se lleve a cabo el procesamiento de ESP en el receptor. En modo túnel, ESP se aplica a un paquete IP, el cual la carga útil puede ser un paquete IP fragmentado. Por ejemplo, en un security gateway o en implementaciones BITS o BITW (según lo definido en el capítulo 2) se puede aplicar ESP en modo túnel a tales fragmentos.

Para el modo transporte (según lo mencionado al principio de la Sección 3.1) las implementaciones BITS y BITW puede que primero tengan que volver a reensamblar el paquete fragmentado por la capa IP local, después aplicar IPsec, y luego fragmentar los paquetes resultantes.

En IPv6: Para las implementaciones BITS y BITW, será necesario recorrer a través de todas las cabeceras de extensión para determinar si hay una cabecera de fragmentación y por lo tanto que el paquete necesita reensamblarse antes de realizar el procesamiento IPsec.

3.4 Procesamiento de Paquetes Entrantes

3.4.1 Reensamblaje

Si se requiere, el reensamblaje se realiza antes del procesamiento de ESP. Si un paquete brindado a ESP para procesamiento parece ser un fragmento IP, es decir, el campo de desplazamiento (OFFSET) es diferente a cero o la bandera de MAS FRAGMENTOS (MORE FRAGMENTS) está en uno, el receptor DEBE desechar el paquete; esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, Fecha/Hora, Dirección de Origen, Dirección de Destino, Número de Secuencia y (En Ipv6) el Identificador de Flujo (Flow ID).

Nota: Para el reensamblaje del paquete, IPv4 NO requiere que el campo DESPLAZAMIENTO (OFFSET) sea cero o que este en cero la bandera de MAS FRAGMENTOS. Para que un paquete reensamblado pueda ser procesado por IPsec (contrariamente a descartar un aparente fragmento), el código IP debe hacer dos cosas después de reensamblar un paquete.

3.4.2 Buscando la Asociación de Seguridad

Al recibir un paquete (reensamblado) conteniendo una Cabecera ESP, el receptor determina la SA (unidireccional) apropiada, basándose en la dirección IP de destino, protocolo de seguridad (ESP), y el SPI. La SA indica si se controlará el campo Número de Secuencia, si el campo Datos de Autentificación debería estar presente, y especificará los algoritmos y claves que se emplearán para la desencriptación y el cálculo del ICV (si es aplicado).

Si no existe ninguna SA válida para este paquete (por ejemplo, el receptor no tiene ninguna clave), el receptor DEBE desechar el paquete; esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, Fecha/Hora, Dirección de Origen,

Dirección de Destino, Número de Secuencia, y (en IPv6) el Identificador de Flujo del texto plano.

3.4.3 Verificación del Número de Secuencia

Todas las implementaciones ESP DEBEN soportar el servicio de anti-replay, aunque su uso puede estar habilitado o deshabilitado por el receptor sobre la base de una SA. Este servicio NO DEBE ser habilitado si no esta habilitado el servicio de autentificación para esa SA, puesto que de otra forma el campo Número de Secuencia no tendrá protección de integridad. Observe que no hay soporte para administrar los valores de los Números de Secuencia transmitidos entre múltiples emisores que dirigen el tráfico a una única SA (independientemente de que si la dirección de destino es unicast, broadcast, o multicast). Así el servicio de anti-replay NO DEBERÍA ser usado en ambientes múlti-emisor que emplee una única SA.

Si el receptor no habilita el anti-replay para una SA, no se realizarán comprobaciones salientes en el Número de Secuencia. Sin embargo, desde la perspectiva del emisor el valor por defecto es asumir que el anti-replay esta habilitado en el receptor. Para evitar que el emisor haga un monitoreo innecesario del número de secuencia y el establecimiento de una SA (ver Sección 3.3.3), si un protocolo de establecimiento de SA tal como IKE se emplea (ver el Capítulo 10), el receptor DEBERÍA notificar al emisor, durante el establecimiento de una SA, si el receptor no proporcionará la protección anti-replay.

Si el receptor tiene habilitado el servicio de anti-replay para esta SA, el contador de recepción de paquetes para la SA, se debe inicializar en cero cuando la SA es establecida. Para cada paquete recibido, el receptor DEBE verificar que el paquete contiene un Número de Secuencia que no es igual al Número de Secuencia de ningún otro paquete recibido durante la vida de esa SA. Este DEBERÍA ser el primer control de ESP aplicado a un paquete después de que haya sido correspondido a una SA, para acelerar el rechazo de paquetes duplicados.

Los paquetes duplicados son rechazados a través del uso de una ventana de recepción deslizable. Un tamaño de ventana mínimo de 32 DEBE ser soportado; pero un tamaño de ventana de 64 es más aconsejable y DEBERÍA ser empleado como valor por defecto. Otro tamaño de ventana (más grande que el mínimo) PUEDE ser elegido por el receptor. El receptor no notifica al emisor del tamaño de ventana.

El lado "Derecho" de la ventana representa el valor del Número de Secuencia más alto autentificado y recibido en esta SA. Los paquetes que contienen Números de Secuencias menores que el lado "izquierdo" de la ventana son rechazados. Los paquetes que caen dentro de la ventana son controlados con una lista de paquetes recibidos dentro de la ventana. Un modo eficiente de realizar este control, basado en el uso de una máscara de bits (bitmask), se describió en el Capítulo 2.

Si el paquete recibido cae dentro de la ventana y es nuevo, o si el paquete esta a la derecha de la ventana, el receptor procede con la verificación del ICV. Si la verificación ICV falla, el datagrama IP recibido no es válido y el receptor DEBE descartar el paquete. Esto es un evento auditable. La entrada del registro de auditoría para este evento DEBERÍA incluir el valor del SPI, fecha/hora recibido, Dirección de Origen, Dirección de Destino, Número de Secuencia, y (en IPv6) el Identificador de Flujo. La ventana de recepción es actualizada solo si la verificación del ICV tiene éxito.

Observe que si el paquete esta dentro de la ventana y es nuevo, o si esta fuera de la ventana en el lado "derecho", el receptor DEBE autentificar el paquete antes de actualizar el valor de la ventana del Número de Secuencia.

3.4.4 Verificación del Valor de Comprobación de Integridad

Si la autentificación a sido seleccionada, el receptor calcula el ICV sobre el paquete ESP menos los Datos de Autentificación usando el algoritmo de autentificación especificado y verifica que es el mismo que el ICV incluido en el campo Datos de Autentificación del paquete.

Si el ICV calculado y recibido concuerdan, el datagrama es válido, y es aceptado. Si el control falla, el receptor debe descartar el datagrama IP recibido porque no es válido; esto es un evento auditadle. Los datos del registro de auditoría deberían incluir el valor del SPI, fecha/hora recibido, Dirección de Origen, Dirección de Destino, Número de Secuencia, y (en IPv6) el Identificador de Flujo del texto plano.

El procedimiento para el cálculo de comprobación del valor del ICV es:

Comience por quitar y guardar el valor ICV (campo Datos de Autentificación). Luego controle la longitud total del paquete ESP menos los Datos de Autentificación. Si se requiere relleno implícito, basado en el tamaño del bloque del algoritmo de autentificación se agregan los bytes de relleno con valor cero al final del paquete ESP directamente después del campo Siguiente Cabecera. Realice el cálculo del ICV y compare el resultado con el valor guardado, usando las reglas de comparación definidas en las especificaciones del algoritmo.

3.4.5 Desencriptación del Paquete

Como en la Sección 3.3.2 "Encriptación de Paquetes", hablamos aquí en términos de encriptación que son siempre utilizados debido a las implicaciones del formato. Esto se hace con la comprensión de que la "confidencialidad" no es ofrecida usando el algoritmo de encriptación NULL. Por consiguiente el receptor:

1. Desencripta los Datos de la Carga Útil de ESP, el Relleno, la Longitud del Relleno, y Siguiente Cabecera, usando la clave, el algoritmo de encriptación, el modo del algoritmo y datos de sincronización criptográficos (si existen), indicados por la SA.
   - Si los datos de sincronización criptográficos son explicitos, por ejemplo, un IV (es indicado), se toman del campo Carga Útil y se coloca en el algoritmo de desencriptación según la especificación del algoritmo.
   - Si los datos de sincronización criptográficos son implícitos, por ejemplo, un IV (es indicado), una versión local de IV es construida y es colocada en el algoritmo de desencriptación según la especificación del algoritmo.
2. Procesar cualquier relleno según lo especificado en la especificación del algoritmo de encriptación. Si se a empleado el esquema de relleno de valor por defecto (ver la Sección 2.4) el receptor DEBERÍA examinar el campo Relleno antes de quitar el relleno y antes de pasar los datos desencriptados a la siguiente capa.
3. Reconstruir el datagrama IP original de:

- Para el modo trasporte: cabecera IP original más la información del protocolo original de la capa superior dentro del campo Carga Útil de ESP.
- Para el modo túnel: la cabecera IP del túnel más el datagrama IP entero dentro del campo Carga Útil de ESP.

Los pasos exactos para reconstruir el datagrama original dependen del modo (trasporte o túnel) y están descriptos en el capítulo 2. Como mínimo, en el un contexto de IPv6 el receptor DEBERÍA asegurarse que los datos desencriptados estén alineados a 8 bytes, para facilitar el procesamiento realizado por el protocolo identificado en el campo Siguiente Cabecera.

Si la autentificación a sido seleccionada, la verificación y la desencriptación pueden realizarse en serie o en paralelo. Para realizarla en serie, la verificación del ICV DEBERÍA realizarse primero. Si se realiza en paralelo, la verificación debe ser completada antes que el paquete desencriptado pase a un proceso posterior. El orden del proceso facilita una rápida detección y rechazo de paquetes reenviados o falsos para el receptor, antes de desencriptar el paquete, por lo tanto reduciendo potencialmente el impacto de ataques de denegación de servicio. Observe que si el receptor realiza la desencriptación en paralelo con la autentificación, se debe tener cuidado para evitar posibles condiciones con relación al acceso de paquetes y a la reconstrucción del paquete desencriptado.

Observe que existe varias causas por las que la desencriptación puede "fallar":

a. La SA seleccionada puede no ser la correcta: La SA puede ser mal seleccionada debido a tampering (ver Sección 8 del Capitulo 5) con los campos SPI, dirección de destino, o tipo de protocolo IPsec. Tales errores, si asocian el paquete a otra SA existente, serán indistinguibles de un paquete corrompido, (caso c). Tampering con el SPI puede ser detectado por medio del uso de la autentificación. Sin embargo, una mala correspondencia con la SA podría aún ocurrir debido a tampering con el campo Dirección IP de Destino o el campo tipo de protocolo IPsec.

b. La longitud del relleno o los valores del relleno pueden ser erróneos: longitud del relleno y valores del relleno deficientes pueden ser detectados independientemente del uso de la autentificación.

c. El paquete ESP encriptado podría ser corrompido: Esto puede ser detectado si la autentificación es seleccionada para la SA.

En el caso (a) o (c), el resultado erróneo de la operación de desencriptación (datagrama IP o capa de trasporte no valida) no será necesariamente detectado por IPsec, y es responsabilidad del procesamiento del siguiente protocolo.

4. **Auditoría**

No todos los sistemas que implementan ESP implementarán auditoría. Sin embargo, si ESP es incorporado a un sistema que soporta auditoría, la implementación ESP debe también soportar auditoría y debe permitirle a un administrador de sistema habilitar o deshabilitar la auditoría para ESP. La granularidad de la auditoría es un tema local. Sin embargo, varios eventos auditadles se identifican en esta especificación y para cada uno de estos

eventos un conjunto mínimo de información debería ser incluido en el registro de auditoría. Información adicional también puede ser incluida en el registro de auditoría para cada uno de estos eventos, y los eventos adicionales, no explícitamente exigidos en esta especificación, también pueden resultar en entradas del registro de auditoría. No hay requisito para el receptor de trasmitir ningún mensaje al emisor pretendido en respuesta a la detección de un evento auditadle, debido al potencial de inducir la Denegación de Servicio a través de tal acción.

5. **Requerimiento de Conformidad**

Las implementaciones deben implementar la síntesis de ESP, los procesos descriptos aquí y cumplir con todos los requisitos del capítulo de la Arquitectura de Seguridad. Si la clave usada para calcular un ICV es distribuida manualmente, la correcta provisión del servicio de anti-replay requerirá el correcto estado del contador en el emisor, hasta que la clave es reemplazada y no habría probablemente disponibilidad automatizada de recuperación si el desbordamiento del contador fuera inminente. Así, una implementación no DEBERÍA proporcionar este servicio en conjunto con SAs que generan claves manuales. Una implementación ESP debe soportar e implementar obligatoriamente los siguientes algoritmos:

- DES en modo CBC [DES]
- HMAC con MD5 [MG97a]
- HMAC con SHA-1[HMACSHA]
- Algoritmo de Autentificación NULL
- Algoritmo de Encriptación NULL

Todos esos algoritmos se pueden encontrar en el Capítulo 6. Puesto que la encriptación y la autentificación son opcionales, el soporte para los dos algoritmos "NULL" se requieren para mantener la consistencia con el modo en que estos servicios son negociados. Observe que a pesar de que la autentificación y la encriptación pueden ser NULL, estos NO DEBEN ser conjuntamente ambos NULL.

Capítulo 5

Criptografía

1. **Introducción**

En este capítulo, veremos definiciones, términos y conceptos básicos de la criptografía, como así también tipos de ataques y algoritmos criptográficos usuales. Debido a que IPsec hace uso continuo y permanente de los algoritmos criptográficos, para poder elegir adecuadamente los algoritmos criptográficos a usarse con los protocolos de seguridad de IPsec es necesario tener conocimientos de los algoritmos criptográficos, por ende este capítulo es fundamental. El Apéndice Glosario y el de Acrónimos le serán de gran utilidad en este capítulo.

2. **Conceptos Básicos sobre Criptografía**

Aquí se incluyen todos los conceptos básicos y se introduce la terminología empleada en el resto del capítulo. Su lectura es recomendable incluso para las personas que ya conocen el tema, puesto que puede evitar cierta confusión en los términos empleados a lo largo de esta obra. A parte de los términos aquí presentados se recomienda que el lector lea el Glosario suministrado como Apéndice en este libro [itu], [hispa], [Principi], [Cripseg], [win2003].

Nota: Las palabras encriptar, cifrar, codificar se usan indistintamente en este libro y son sinónimos, como así también las palabras desencriptar, descifrar, decodificar y descodificar se usan indistintamente en este libro y son sinónimos.

Criptografía

Según el Diccionario de la Real Academia, la palabra Criptografía proviene de dos palabras griegas una que significa oculto, y la otra que significa escritura, y su definición es: "Arte de escribir con clave secreta o de un modo enigmático". Obviamente la criptografía hace años que dejó de ser un arte para convertirse en una técnica, o más bien un conglomerado de técnicas, que tratan sobre la protección (ocultamiento frente a observadores no autorizados) de la información. Finalmente, el término criptología, aunque no está reconocido aún en el diccionario, se emplea habitualmente para agrupar tanto la criptografía como al criptoanálisis.

Cifrado

Proceso de camuflar un mensaje o datos de forma que se oculte su contenido. Método para formar un mensaje oculto. El cifrado se utiliza para transformar un mensaje legible, denominado texto plano (también denominado texto no cifrado o texto sin formato) en un mensaje ilegible, codificado u oculto, denominado texto cifrado. Solamente aquel usuario con una clave de descodificación puede convertir dicho texto en el texto original.

Criptosistema

Definiremos un criptosistema como una quíntupla $(M,\ C,\ K,\ E,\ D)$, donde:

- M, representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto claro, o plaintext) que pueden ser enviados.
- C, representa el conjunto de todos los posibles mensajes cifrados, o criptogramas.
- K, representa el conjunto de claves que se pueden emplear en el criptosistema.
- E, es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un

elemento de C. Existe una transformación diferente E_K para cada valor posible de la clave k.

- D, es el conjunto de transformaciones de descifrado, análogo a E.

Todo criptosistema ha de cumplir la siguiente condición:

$$D_K(E_K(m)) = m$$

Es decir, que si tenemos un mensaje m, lo ciframos empleando la clave K y luego lo desciframos empleando la misma clave, obtenemos de nuevo el mensaje original m.

Criptoanálisis
El criptoanálisis consiste en comprometer la seguridad de un criptosistema.

Criptografía Fuerte
Este tipo de criptografía es muy segura y es prácticamente imposible desencriptar mensajes encriptados con este tipo de criptografía.

Firma Digital
Es una secuencia de caracteres calculados a partir del documento original mediante funciones de resumen (digest) o Hash que acompaña a un documento (o fichero), acreditando quién es su autor ("autenticación") y que no ha existido ninguna manipulación posterior de los datos ("integridad"). Para firmar un documento digital, su autor utiliza su propia clave secreta, cualquier persona puede verificar la validez de una firma si dispone de la clave pública del autor.

Clave (Key)
Llave que permite cifrar o descifrar la información recibida de forma correcta.

Clave Privada
Mitad secreta de una pareja de claves criptográficas que se utiliza con un algoritmo de clave pública. Las claves privadas se usan, normalmente, para descifrar una clave de sesión simétrica, firmar datos digitalmente o descifrar datos que han sido cifrados con la clave pública correspondiente.

Clave Pública
Mitad no secreta de una pareja de claves criptográficas que se utiliza con un algoritmo de clave pública. Las claves públicas se utilizan normalmente para cifrar una clave de sesión o comprobar una firma digital, etc.

3. **Fundamentos Teóricos de la Criptografía**

En esta sección se desarrollan los fundamentos y resultados teóricos sobre los que se basa el resto del capítulo [Cripseg], [hispa].

3.1 Aritmética Modular

La aritmética modular maneja un conjunto finito de números enteros. Mucha gente la conoce como la aritmética del reloj, debido a su parecido con la forma que tenemos de contar el tiempo. Por ejemplo, si son las 19:13:59 y pasa un segundo, decimos que son las 19:14:00, y no las 19:13:60. Como vemos, los segundos (al igual que los minutos), se expresan empleando sesenta valores cíclicos, de forma que tras el 59 viene de nuevo el 0.

Desde el punto de vista matemático diríamos que los segundos se expresan en *módulo* de 60. Empleemos ahora un punto de vista más formal y riguroso:

Dados tres números a, b, n, que pertenezca a $\mathbb{N}$ decimos que a es congruente con b módulo n, y se escribe:

$$a \equiv b \pmod{n}$$

si se cumple:

$a=b+kn$, para algún k pertenezca a $\mathbb{Z}$

Por ejemplo si tenemos $n=5$ tendremos

0	1	2	3	4	5	6	7	8	En $\mathbb{Z}$
0	1	2	3	4	0	1	2	3	En $\mathbb{Z}_5$

$5 \equiv 0 \pmod{5}$, esto es porque 5/5 me da de resto de la división 0.
$6 \equiv 1 \pmod{5}$, esto es porque 6/5 me da de resto de la división 1.
$7 \equiv 2 \pmod{5}$, esto es porque 7/5 me da de resto de la división 2.

Para expresar cualquier elemento de $\mathbb{Z}$ como un elemento de $\mathbb{Z}_5$ bastará con dividirlo por n y quedarnos con su resto. Diremos que dos elementos de $\mathbb{Z}$ son *equivalentes* en $\mathbb{Z}_n$ siempre y cuando tengan el mismo resto al dividir por n. Para nuestro ejemplo lo son -11, -6, 6, 11... Por razones de simplicidad, representamos cada clase de equivalencia por un número comprendido entre 0 y n-1. De esta forma, en nuestro ejemplo (módulo 5) tendremos el conjunto de clases de equivalencia {0, 1, 2, 3, 4, 5}, al que denominaremos $\mathbb{Z}_5$. Llamaremos orden de un "grupo" G y lo denotaremos como $(\mathrm{mod}\ G)$ al número de elementos que posee el grupo en nuestro caso posee 5 elementos por ende el orden del grupo es de 5.

3.2 Función Unidireccional o de un Solo Sentido

Supongamos que $f(x)$ es una función de un sentido (o unidireccional), entonces:

1. Es fácil el cálculo $y=f(x)$, conociendo x.

2. Conocido y es computacionalmente imposible el cálculo de $x=f^{-1}(y)$.

Un ejemplo típico de una función de este tipo es:

$$y \equiv g^x \pmod{p}$$

donde g y x son números reales y p es un número primo con más de 200 dígitos. Esta función es conocida como "exponenciación modular". Su función inversa será:

$$x \equiv \log_g y \pmod{p}$$

Cuando p tiene un tamaño como el que se ha dicho antes es prácticamente imposible el cálculo de esta función. Esta función se conoce como "logaritmo discreto" y es de gran importancia en la criptografía asimétrica.

3.3 El Problema de los Logaritmos Discretos

El problema inverso de la exponenciación es el cálculo de logaritmos discretos. Dados dos números a, b y el módulo n, se define el logaritmo discreto de a en base b módulo n como:

$$c = \log_b (a) \pmod n \iff a \equiv b^c \pmod n$$

En la actualidad no existen algoritmos eficientes que sean capaces de calcular en tiempo razonable logaritmos de esta naturaleza, y muchos esquemas criptográficos basan su resistencia en esta circunstancia. El problema de los logaritmos discretos está íntimamente relacionado con el de la factorización, de hecho está demostrado que si se puede calcular un logaritmo, entonces se puede factorizar fácilmente (el recíproco no se ha podido demostrar).

3.4 El Problema de Diffie-Hellman

Antes de enunciarlo definiremos el término **generador**. Dado el conjunto $\mathbb{Z}_p^*$, con p primo, diremos que α pertenezca a $\mathbb{Z}_p^*$ y es un generador de $\mathbb{Z}_p^*$, si se cumple con:

Cualquiera sea b que pertenezca a $\mathbb{Z}_p^*$, existiera un i tal que $\alpha^i = b$

El enunciado del problema es el siguiente: dado un número primo p, un número α que sea un generador de $\mathbb{Z}_p^*$, y los elementos α^a y α^b, encontrar $\alpha^{a\,b} \pmod p$.

Note que nosotros conocemos α^a y α^b, pero no el valor de a ni el de b. De hecho, si pudiéramos efectuar de forma eficiente logaritmos discretos, sería suficiente con calcular a y luego $(\alpha^b)^a = \alpha^{a\,b}$.

4. **Criptografía Simétrica o Privada**

Un intercambio de claves proporciona simetría si cualquier parte puede iniciar el intercambio, y los mensajes intercambiados pueden cruzarse en la trayectoria sin afectar la clave generada [ISAKMP]. En este tipo de criptografía tanto el emisor como el receptor del mensaje han de conocer la clave y esta clave sirve tanto para encriptar como para desencriptar los mensajes.

Sun principales ventajas son:

- Presentan una longitud de clave considerablemente menor que los algoritmos asimétricos (exceptuando los basados en corvas elípticas).
- Requieren menos recursos computacionales que los algoritmos asimétricos.

- Usa una clave única que sirve tanto para desencriptar como para encriptar.
- El cálculo de la clave no requiere que cada parte sepa quien inició el intercambio [ISAKMP].
- Requiere menos recursos de ancho de banda que los algoritmos asimétricos.

Sun principales desventajas son:

- La simetría en el protocolo de administración de claves puede proporcionar vulnerabilidad a los ataques de reflexión (reflection attacks)[ISAKMP].
- La clave es generada en uno de los extremos de la comunicación, por ende si no se confía en él, este método no serviría.
- Para ser empleados en comunicaciones la clave debe estar tanto en el emisor como en el receptor, lo cual nos lleva a preguntarnos cómo transmitir la clave de forma segura si tenemos un canal inseguro como Internet. Una solución puede ser:

  Emplear un algoritmo asimétrico (como por ejemplo, el algoritmo Diffie-Hellman) para encriptar la clave simétrica (denominada comúnmente clave de sesión).

Este timo de algoritmos es utilizado por [ESP] para encriptar la información. La Figura 1, muestra un esquema conceptual de un sistema simétrico.

$C_K(m)$ = Cifrar el mensaje con la clave K.

$D_K(m)$ = Descifrar el mensaje con la clave K.

Emisor **Receptor**

$C_K(m)$ $C_K(m)$

canal inseguro

---------->

$$D_K(\ C_K(mensaje)) = \ mensaje$$

Figura 1: Sistemas de clave privada o simétricos

Los sistemas de claves simétricos los podemos clasificar en cifrado por Bloque o por Flujo.

4.1 El Cifrado en Bloque

La gran mayoría de los algoritmos de cifrado simétricos se apoyan en los conceptos de confusión (consiste en tratar de ocultar la relación que existe entre el texto claro, el texto cifrado y la clave) y difusión (trata de repartir la influencia de cada bit del mensaje original lo más posible entre el mensaje cifrado) que se combinan para dar lugar a los denominados *cifrados de producto*. Estas técnicas consisten básicamente en trocear el mensaje en bloques de tamaño fijo, y aplicar la función de cifrado a cada uno de ellos. Por ende hemos de tener en cuenta lo que ocurre cuando la

longitud de la cadena que queremos cifrar no es un múltiplo exacto del tamaño de bloque. Entonces tenemos que añadir información al final para que sí lo sea. El mecanismo más sencillo consiste en rellenar con ceros (o con algún otro patrón) el último bloque que se codifica. El problema ahora consiste en saber cuando se descifra por dónde hay que cortar. Lo que se suele hacer es añadir como último byte del último bloque el número de bytes que se han añadido. Esto tiene el inconveniente de que si el tamaño original es múltiplo del bloque, hay que alargarlo con otro bloque entero. Por ejemplo, si el tamaño de bloque fuera 64 bits, y nos sobraran cinco bytes al final, añadiríamos dos ceros y un tres, para completar los ocho bytes necesarios en el último bloque. Si por el contrario no sobrara nada, tendríamos que añadir siete ceros y un ocho [Cripseg].

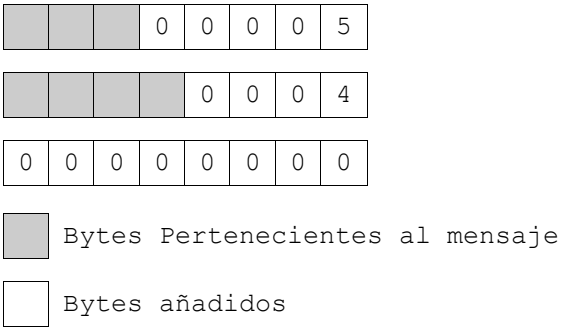


Figura 2: Relleno (padding) de los bytes del último bloque al emplear un algoritmo de cifrado por bloques.

Es interesante que un algoritmo criptográfico por bloque carezca de *estructura de grupo*, ya que si ciframos un mensaje primero con la clave k_1 y el resultado con la clave k_2, es como si hubiéramos empleado una clave de longitud doble, aumentando la seguridad del sistema. Si, por el contrario, la transformación criptográfica presentara estructura de grupo, esto hubiera sido equivalente a cifrar el mensaje una única vez con una tercera clave, con lo que no habríamos ganado nada.

Los Modos principales de operación para algoritmos de cifrado por bloques son: ECB, CBC, CFB.

4.1.1 Modo ECB

El modo ECB (Bloque de Código Electrónico)simplemente subdivide la cadena que se quiere codificar en bloques de tamaño adecuado y se cifran todos ellos empleando la misma clave.

A favor de este método podemos decir que permite codificar los bloques independientemente de su orden, lo cual es adecuado para codificar bases de datos o ficheros en los que se requiera un acceso aleatorio. También es resistente a errores, pues si uno de los bloques sufriera una alteración, el resto quedaría intacto. Por contra, si el mensaje presenta patrones repetitivos, el texto cifrado también los presentará, y eso es peligroso, sobre todo cuando se codifica información muy redundante (como ficheros de texto), o con patrones comunes al inicio y final (como el correo electrónico). Un atacante puede en estos casos efectuar un ataque estadístico y extraer bastante información.

Otro riesgo bastante importante que presenta el modo ECB es el de la sustitución de bloques. El atacante puede cambiar un bloque sin mayores

problemas, y alterar los mensajes incluso desconociendo la clave y el algoritmo empleados. Simplemente se escucha una comunicación de la que se conozca el contenido, como por ejemplo una transacción bancaria a nuestra cuenta corriente. Luego se escuchan otras comunicaciones y se sustituyen los bloques correspondientes al número de cuenta del beneficiario de la transacción por la versión codificada de nuestro número (que ni siquiera nos habremos molestado en descifrar). En cuestión de horas nos habremos hecho ricos [Cripseg], [inco].

4.1.2 Modo CBC

El modo CBC (Cipher Book Chiang - Concatenación de Bloques Cifrados) incorpora un mecanismo de retroalimentación en el cifrado por bloques. Esto significa que la codificación de bloques anteriores condiciona la codificación del bloque actual, por lo que será imposible sustituir un bloque individual en el mensaje cifrado. Esto se consigue efectuando una operación XOR entre el bloque del mensaje que queremos codificar y el último criptograma obtenido.

En cualquier caso, dos mensajes idénticos se codificarán de la misma forma usando el modo CBC. Más aún, dos mensajes que empiecen igual se codificarán igual hasta llegar a la primera diferencia entre ellos. Para evitar esto se emplea un *Vector de Inicialización* (IV), que puede ser un bloque aleatorio, como bloque inicial de la transmisión. Este vector será descartado en destino, pero garantiza que siempre los mensajes se codifiquen de manera diferente, aunque tengan partes comunes [Cripseg] [inco].

4.1.3 Modo CFB

El modo CBC no empieza a codificar (o decodificar) hasta que no se tiene que transmitir (o se ha recibido) un bloque completo de información. Esta circunstancia puede convertirse en un serio inconveniente, por ejemplo en el caso de terminales, que deberían poder transmitir cada carácter que pulsa el usuario de manera individual. Una posible solución sería emplear un bloque completo para transmitir cada byte y rellenar el resto con ceros, pero esto hará que tengamos únicamente 256 mensajes diferentes en nuestra transmisión y que un atacante pueda efectuar un sencillo análisis estadístico para comprometerla. Otra opción sería rellenar el bloque con información aleatoria, aunque seguiríamos desperdiciando gran parte del ancho de banda de la transmisión. El modo de operación CFB (Cipher-Feedback Mode) permitirá codificar la información en unidades inferiores al tamaño del bloque, lo cual permite aprovechar totalmente la capacidad de transmisión del canal de comunicaciones, manteniendo además un nivel de seguridad adecuado [Cripseg].

4.1.4 Algoritmos de Cifrado en Bloque

4.1.4.1 DES

Este algoritmo simétrico encripta bloques de 64 bits de longitud con una clave de 64 bits de longitud. Dentro de la clave el último bit de cada byte es de paridad, con lo cual tenemos que la clave en realidad es de 56 bits, esto hace que haya 2^56 posibles claves para este algoritmo. Dependiendo de la naturaleza de la aplicación DES puede operar en modo CBC, ECB, CFB y otros modos más no vistos aquí como el modo OFB. Debido a que es uno de los algoritmos más difundidos describiré su funcionamiento [hispa], [Cripseg], [Principi], [inco]:

Este algoritmo utiliza un "dispositivo" denominado SBB (Standard Building

Block o Constructor Estándar de Bloques), el cual requiere como entrada un bloque de 64 bits y una clave de 48 bits, produciendo una salida de 64 bits. El DES requiere 16 dispositivos SBB. Tenemos una clave original, k, de 64 bits, 56 en realidad, de ella se extraen 16 subclaves k_i de 48 bits de longitud. El algoritmo es el siguiente:

1. Se aplica una Permutación Original (PO) a cada bloque de 64 bits. Produciendo una salida de 64 bits (dos de 32 bits).

2. Pasamos la salida del PO y la subclave k_1 por el primer SBB, la salida la pasamos por el segundo SBB con la subclave k_2 y así con los 16 SBB.

3. A la salida del último SBB le aplicamos la permutación PO^{-1}. De donde obtenemos el texto encriptado.

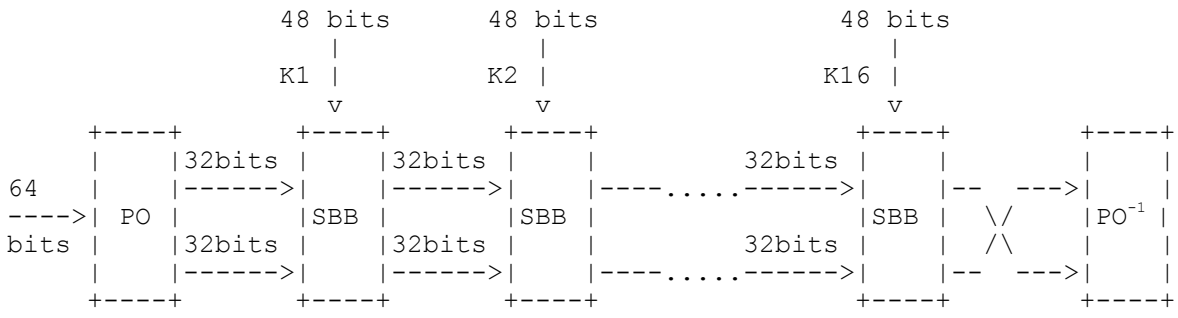


Figura 3: Encriptación con DES.

Para desencriptar tomamos como entrada el texto encriptado y aplicamos las subclaves k_i en orden inverso, es decir en el primer SBB.

Ésta es una lista de claves DES débiles y semi-débiles, las cuales deben evitar usarse. Las claves provienen de [Sch96]. Todos los claves se listan en hexadecimal.

Claves DES débiles
0101 0101 0101 0101
1F1F 1F1F E0E0 E0E0
E0E0 E0E0 1F1F 1F1F
FEFE FEFE FEFE FEFE

Claves DES semi-débiles
01FE 01FE 01FE 01FE
1FE0 1FE0 0EF1 0EF1
01E0 01E0 01F1 01F1
1FFE 1FFE 0EFE 0EFE
011F 011F 010E 010E
E0FE E0FE F1FE F1FE

FE01 FE01 FE01 FE01
E01F E01F F10E F10E
E001 E001 F101 F101
FE1F FE1F FE0E FE0E
1F01 1F01 0E01 0E01
FEE0 FEE0 FEF1 FEF1

El uso de estas claves débiles y semi-debiles debe ser rechazado, seguido de una solicitud de reemplazo de claves o de la negociación de una nueva SA.

4.1.4.2 Triple DES (3DES)

A mediados de 1988 se demostró que un ataque por fuerza bruta contra el algoritmo DES ya era posible, gracias al avance de la informática entre otras cosas. Pero la debilidad no la tiene el algoritmo, sino que la tiene la clave debido a que no posee suficiente longitud. Si aumentamos la clave este algoritmo sigue siendo seguro. Por ende este algoritmo realiza tres veces el DES, aumentando la longitud de clave a 192 bits (64 x 3) [hispa].

4.1.4.3 IDEA

El algoritmo IDEA (International Data Encryption Algorithm) data de 1992 y para muchos constituye el mejor y más seguro algoritmo simétrico disponible en la actualidad. Trabaja con bloques de 64 bits de longitud y emplea una clave de 128 bits reales (no hay bits de paridad como en el DES). Como en el caso de DES, se usa el mismo algoritmo tanto para cifrar como para descifrar. IDEA es un algoritmo bastante seguro, y hasta ahora se ha mostrado resistente a multitud de ataques, entre ellos el criptoanálisis diferencial. No presenta claves débiles, y su longitud de clave hace imposible en la práctica un ataque por fuerza bruta [Cripseg].

4.1.4.4 El Algoritmo AES

Su interés radica en que todo el proceso de selección, revisión y estudio, se ha efectuado de forma pública y abierta, lo cual convierte a Rijndael en un algoritmo perfectamente digno de la confianza de todos. Este es un sistema de cifrado por bloques, diseñado para manejar longitudes de clave y de bloque variables, ambas comprendidas entre los 128 y los 256 bits. AES, a diferencia de algoritmos como el DES, no posee estructura de red de Feistel (para descifrar basta con aplicar el mismo algoritmo, pero con las k_i en orden inverso). En su lugar se ha definido cada ronda como una composición de cuatro funciones invertibles [Cripseg].

4.1.4.5 RC2

Es un algoritmo propietario de la empresa RSA que tiene un tamaño de bloque de 64 bits. Permite utilizar los modos ECB y CBC. Fue desarrollado como alternativa del DES y tiene una longitud de clave variable que va de 64 a 256 bits [inco].

4.1.4.6 RC5

También pertenece a la empresa RSA. Se caracteriza por permitir bloques de 32, 64 o 128 bits. Su tamaño de clave varía de 0 a 2040 bits (255 bytes) [inco].

4.2 El Cifrado en Flujo

Supongamos que disponemos de un generador pseudoaleatorio capaz de generar secuencias criptográficamente aleatorias, de forma que la longitud de los posibles ciclos sea extremadamente grande. En tal caso podríamos, empleando semillas del generador como clave, podemos obtener cadenas de bits de usar y tirar, y emplearlas para cifrar mensajes simplemente aplicando la función XOR entre el texto en claro y la secuencia generada. Todo aquel que conozca

la semilla podría reconstruir la secuencia pseudoaleatoria y de esta forma descifrar el mensaje [hispa], [Principi], [popu], [inco], [Cripseg].

Dichos algoritmos no son más que la especificación de un generador pseudoaleatorio, y permiten cifrar mensajes de longitud arbitraria, sin necesidad de dividirlos en bloques para codificarlos por separado. Como cabría esperar, estos criptosistemas no proporcionan seguridad perfecta, ya que como empleamos un generador tenemos como máximo tantas secuencias distintas como posibles valores iniciales de la semilla.

4.2.1 Algoritmos de Cifrado en Flujo

4.2.1.1 RC4

Se caracteriza por utilizar la misma información de entrada que ha de cifrar para la generación de un número pseudoaleatorio que utilizará como clave, realizando un XOR entre la entrada y la clave. Esto significa que tanto el cifrado como el descifrado son operaciones idénticas. No se debe utilizar la misma clave más de una vez, ya que al utilizar un XOR como operación básica un atacante podría fácilmente descubrirla (XOR(XOR(X)) = X). La clave varía de 8 a 2048 bits, [inco], [Cripseg].

4.2.1.2 RC4 con MAC

Es una extensión del RC4 que busca asegurar la integridad en los datos mediante el uso de una función MAC (es una función que asegura la integridad de los datos, a partir del mensaje genera una secuencia de bits de tal forma que si es modificado, el receptor puede saberlo, ver Sección 6.3.3) [inco].

5. **Criptografía Asimétrica o Pública**

Los algoritmos asimétricos emplean generalmente longitudes de clave mucho mayores que los simétricos y difieren fundamentalmente de los algoritmos simétricos en que la claves no son únicas sino que forman pares. Por ejemplo, mientras que para algoritmos simétricos se considera segura una clave de 128 bits, para algoritmos asimétricos (si exceptuamos aquellos basados en curvas elípticas) se recomiendan claves de al menos 1024 bits. Además, la complejidad del cálculo de algoritmos asimétricos los hace considerablemente más lentos que los algoritmos de cifrado simétricos. En la práctica los métodos asimétricos se emplean únicamente para codificar la clave de sesión (simétrica) de cada mensaje o transacción particular, [www1], [www5], [hispa], [Principi], [popu], [inco], [Cripseg].
Por otra parte algoritmos asimétricos ofrecen un abanico superior de posibilidades, pudiendo emplearse para establecer comunicaciones seguras por canales inseguros (puesto que únicamente viaja por el canal la clave pública, que sólo sirve para cifrar), o para llevar a cabo autentificaciones.

En la práctica se emplea una combinación de criptosistemas simétricos y asimétricos, puesto que los segundos presentan el inconveniente de ser computacionalmente mucho más costosos que los primeros. En el mundo real se codifican los mensajes (largos) mediante algoritmos simétricos, que suelen ser muy eficientes, y luego se hace uso de la criptografía asimétrica para codificar las claves simétricas (cortas).

5.1 Aplicaciones de los Algoritmos Asimétricos

Los algoritmos asimétricos poseen dos claves diferentes en lugar de una, la

clave privada K_{Pr} y la clave pública K_{Pu}, eliminando el mayor problema de los sistemas de clave privada, dar a conocer únicamente al receptor autorizado la clave usada en el sistema de cifrado/descifrado. Una de ellas se emplea para codificar, mientras que la otra se usa para decodificar. Dependiendo de la aplicación que le demos al algoritmo, la clave pública será la de cifrado o viceversa. Como se ve se introduce un nuevo problema, la autenticación del origen de los datos. Puesto que todo el mundo conoce la clave pública, se puede enviar un mensaje falseando la procedencia. En los sistemas de clave privada esto no pasaba, ya que la clave la compartían únicamente el emisor y el receptor de la información, asegurando la confidencialidad y la procedencia de la información. Para que estos criptosistemas sean seguros también ha de cumplirse que a partir de una de las claves resulte extremadamente difícil calcular la otra. Para poder dar a conocer las claves públicas de los usuarios sin ningún riesgo, debemos asegurarnos que estas no pueden ser ni modificadas ni alteradas en ninguna forma. Con esta función se crearon las Autoridades de Certificación (Certification Authorities, CA), que son organismos encargados de distribuir las claves públicas y velar por ellas [Cripseg], [inco].

5.1.1 Protección de la Información

Una de las aplicaciones inmediatas de los algoritmos asimétricos es el cifrado de la información sin tener que transmitir la clave de decodificación, lo cual permite su uso en canales inseguros. Supongamos que **A** quiere enviar un mensaje a **B** (Figura 4). Para ello solicita a **B** su clave pública K_{Pu} (o la obtiene de una Autoridad de Certificación). **A** genera entonces el mensaje cifrado $E_{K_{Pu}}(m)$. Una vez hecho esto únicamente quien posea la clave K_{Pr} (en nuestro ejemplo, **B**) podrá recuperar el mensaje original m [Cripseg].

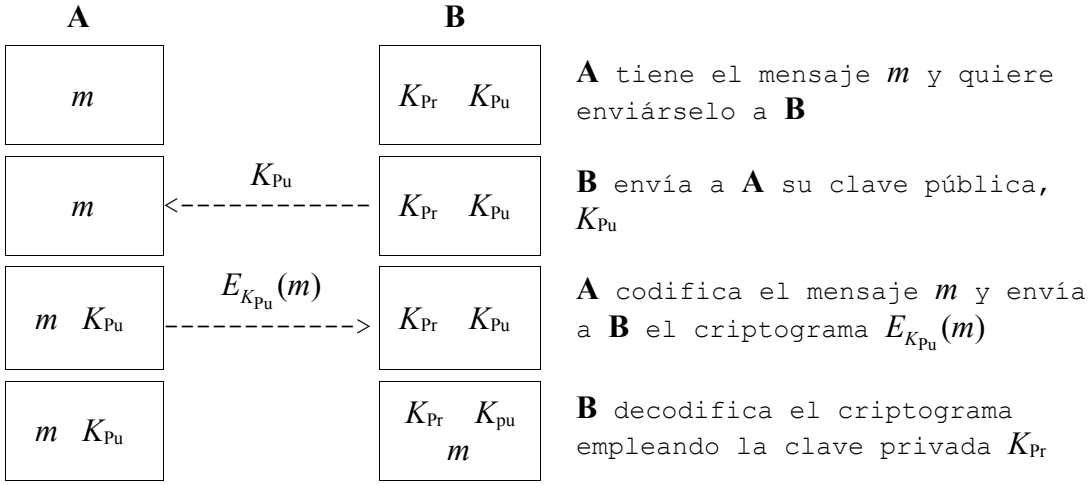


Figura 4: Transmisión de información empleando algoritmos asimétricos

Nótese que para este tipo de aplicación, la llave que se hace pública es aquella que permite codificar los mensajes, mientras que la llave privada es aquella que permite descifrarlos.

5.1.2 Autentificación

La segunda aplicación de los algoritmos asimétricos es la autentificación de mensajes, con ayuda de funciones resumen (ver Sección 6.2), que nos permiten obtener una firma digital a partir de un mensaje. Dicha firma es mucho más pequeña que el mensaje original. Supongamos que **A** recibe un mensaje m de **B** y quiere comprobar su autenticidad. Para ello **B** genera un resumen del mensaje $r(m)$ (ver Figura 5) y lo codifica empleando la clave de cifrado, que en este caso será la privada [Cripseg].

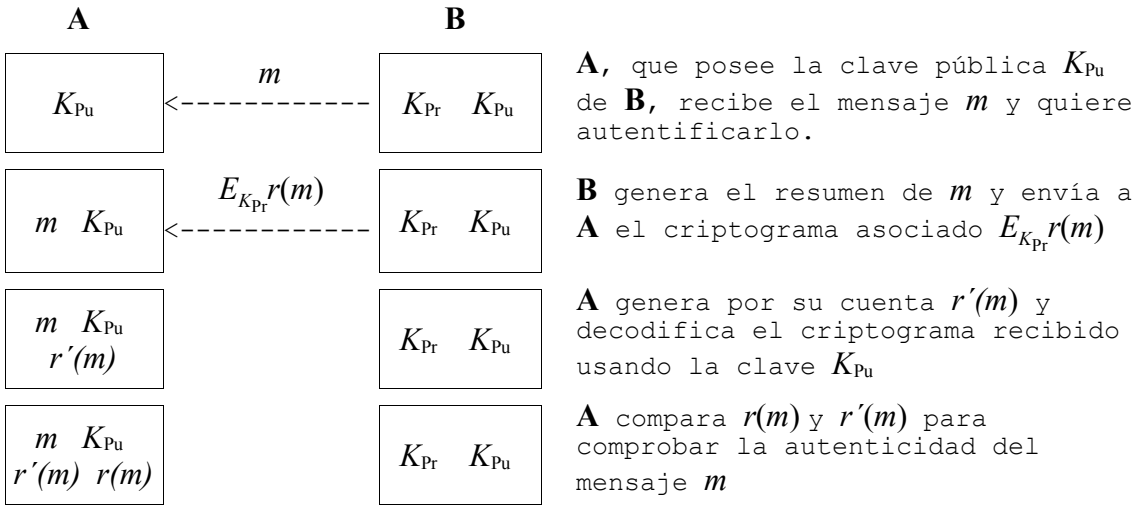


Figura 5: Autentificación de la información empleando algoritmos asimétricos.

La clave de descifrado se habrá hecho pública previamente, y debe estar en poder de **A**. **B** envía entonces a **A** el criptograma correspondiente a $r(m)$. **A** puede ahora generar su propio $r'(m)$ y compararla con $r(m)$ el cual es el valor obtenido del criptograma enviado por **B**. Si coinciden, el mensaje será auténtico, puesto que el único que posee la clave para codificar es precisamente **B**.

Nótese que en este caso la clave que se emplea para cifrar es la clave privada, justo al revés que para la simple codificación de mensajes.

En muchos de los algoritmos asimétricos ambas claves sirven tanto para cifrar como para descifrar, de manera que si empleamos una para codificar, la otra permitirá decodificar y viceversa. Esto es lo que ocurre con el algoritmo RSA, en el que un único par de claves es suficiente para codificar y autentificar.

5.2 Algoritmos Asimétricos

5.2.1 Algoritmo de Diffie-Hellman

Es un algoritmo asimétrico, basado en el problema de Diffie-Hellman (ver Sección 3.4), que se emplea fundamentalmente para acordar una clave secreta común entre dos interlocutores, sin necesidad de ningún secreto preestablecido entre ellos, a través de un canal de comunicación inseguro. La ventaja de este sistema es que no son necesarias llaves públicas en el

sentido estricto, sino información compartida por los dos comunicantes. Este algoritmo no proporciona ni autenticación ni cifrado, por ende no suele utilizarse para la protección de datos. Debido a que es uno de los algoritmos más utilizados con IPsec, describiré su funcionamiento [www1], [www3], [hispa], [inco], [CICESE], [Cripseg]:

Sean **A** y **B** los interlocutores en cuestión. En primer lugar, se calcula un número primo p y un generador α de $\mathbb{Z}_p^*$, con $2 \leq \alpha \leq p-1$. Esta información es pública y conocida por ambos. El algoritmo queda como sigue:

1. **A** escoge un número aleatorio x, tal que $1 \leq x \leq p-2$ y envía a **B** el valor: $\alpha^x \ (mod\ p)$

2. **B** escoge un número aleatorio y, tal que $1 \leq y \leq p-2$ y envía a **A** el valor: $\alpha^y \ (mod\ p)$

3. **B** recoge α^x y calcula $K = (\alpha^x)^y \ (mod\ p)$

4. **A** recoge α^y y calcula $K = (\alpha^y)^x \ (mod\ p)$

Puesto que x e y no viajan por la red, al final **A** y **B** terminan compartiendo el valor de K, sin que nadie que capture los mensajes transmitidos pueda repetir el cálculo.

5.2.1.1 Grupos de Diffie-Hellman

El material clave de Diffie-Hellman que ambos interlocutores intercambian puede constar de 768, 1.024 ó 2.048 bits y se conoce como grupos de Diffie-Hellman 1, 2 y 2048 respectivamente. Para una completa descripcion de los grupos (como el grupo exponencial modular clásico MODP, o los grupos de curvas elípticas como EC2N y ECP), como crear nuevos grupos (grupos privados) de D-H y consideraciones de seguridad sobre D-H vease el capítulo 9. La eficacia del grupo de Diffie-Hellman es proporcional a la de la clave calculada a partir del intercambio Diffie-Hellman. Los grupos Diffie-Hellman más eficaces pueden combinarse con mayores longitudes de claves para aumentar la dificultad del cálculo de una clave secreta [www3], [win2003].

Los grupos Diffie-Hellman se utilizan para determinar la longitud de los números primos base (del material clave) utilizados durante el proceso de intercambio de claves de Diffie-Hellman. La eficacia criptográfica de cualquier clave derivada de un intercambio Diffie-Hellman depende en parte de la eficacia del grupo de Diffie-Hellman en el que se basan los números primos.

El grupo 1 proporciona 768 bits de protección de clave, el grupo 2 proporciona 1024 bits y el grupo 2048 proporciona 2048 bits. Cuando se utiliza un grupo más seguro, la clave derivada del intercambio Diffie-Hellman es más segura y es más difícil que un intruso la averigüe. Si se especifican grupos no coincidentes en cada interlocutor, no será posible la negociación. El grupo no se puede cambiar durante la negociación.

5.2.1.2 Notas de Implementación

Si bien el grupo 2 es el más seguro puede ser que algunos sistemas operativos no lo soporten como es el caso de Windows 2000 y Windows XP que

solo se le brinda soporte en Windows Server 2003. Por ende para asegurar la interoperatibilidad se suele usar el grupo bajo de Diffie-Hellman, a menos que se sepa de antemano que grupo soporta nuestro interlocutor [www3], [win2003].

IKE negocia el grupo específico que se utiliza, lo que asegura que no haya errores en la negociación por falta de correspondencia en un grupo Diffie-Hellman entre los dos interlocutores.

Si se habilita el PFS (vea el Capítulo 10) de clave de sesión, se negocia una nueva clave de Diffie-Hellman durante la primera negociación de SA de modo rápido. Esta nueva clave elimina la dependencia entre la clave de sesión y el intercambio Diffie-Hellman realizado para la clave principal [win2003].

Tanto el interlocutor inicial como el que responde deben tener habilitado el PFS de clave de sesión; de lo contrario, la negociación no se producirá.

El grupo Diffie-Hellman es el mismo para las SA negociaciones en modo principal y en modo rápido. Cuando se habilita el PFS de clave de sesión, aunque el grupo Diffie-Hellman forme parte de la negociación de SA de modo principal, sólo afecta a la regeneración de claves durante el establecimiento de la clave de sesión.

5.2.2 El Algoritmo RSA

Sus claves sirven indistintamente tanto para codificar como para autentificar. Debe su nombre a sus tres inventores: Ronald Rivest, Adi Shamir y Leonard Adleman, y estuvo bajo patente de los Laboratorios RSA hasta el 20 de septiembre de 2000, por lo que su uso comercial estuvo restringido hasta esa fecha. Nadie ha conseguido probar o rebatir su seguridad, y se lo tiene como uno de los algoritmos asimétricos más seguros. RSA se basa en la dificultad para factorizar números grandes. Las claves públicas y privadas se calculan a partir de un número que se obtiene como producto de dos primos grandes [www1], [Principi], [Cripseg].

5.2.3 Criptografía de Curvas Elípticas

Para curvas elípticas existe un problema análogo al de los logaritmos discretos en grupos finitos de enteros. Esto nos va a permitir trasladar cualquier algoritmo criptográfico definido sobre enteros, y que se apoye en este problema, al ámbito de las curvas elípticas. La ventaja que se obtiene es que, con claves más pequeñas, se obtiene un nivel de seguridad equiparable [Cripseg], [www2].

Debido a la relación existente entre ambos, muchos algoritmos que se basan en el problema de la factorización pueden ser replanteados para descansar sobre los logaritmos discretos. De hecho, existen versiones de curvas elípticas de muchos de los algoritmos asimétricos más populares (ejemplos de implementaciones de curvas elíptica se puede encontrar en el capítulo 9).

6. **Autentificación**

Un paso muy importante en el establecimiento de comunicaciones de red seguras es la autentificación de la entidad en el otro extremo de la comunicación [ISAKMP]. Por autentificación entenderemos cualquier método que nos permita comprobar de manera segura alguna característica sobre un objeto. Dicha característica puede ser su origen, su integridad, su

identidad, etc [Cripseg].

Los mecanismos de autentificación pueden clasificarse dentro de dos categorías [ISAKMP]:

Débiles: Enviar claves de texto sin encriptar (texto en claro o cleartext) o otra información de autentificación sin protección en una red es débil, debido a la amenaza de lecturas con sniffer de red. El envío unidireccional de claves hasheadas (a las cuales se les hizo un resumen criptográfico) deficientemente elegidas con baja entropía son también débiles, debido a la amenaza de ataques por fuerza bruta a los mensajes sniffer.

Fuertes: Las firmas digitales, tales como el Estándar de Firmas Digitales (DDS - Digital Signature Standard) y las firmas Rivest-Shamir-Adleman (RSA) son claves públicas basadas en fuertes mecanismos de autentificación.

Consideraremos tres grandes tipos dentro de los métodos de autentificación [Cripseg]:

- Autentificación de mensajes: Queremos garantizar la procedencia de un mensaje conocido, de forma que podamos asegurarnos de que no es una falsificación. Este mecanismo se conoce habitualmente como *firma digital*.
- Autentificación de usuario mediante contraseña: En este caso se trata de garantizar la presencia de un usuario legal en el sistema. El usuario deberá poseer una contraseña secreta que le permita identificarse.
- Autentificación de dispositivo: Se trata de garantizar la presencia de un dispositivo válido. Este dispositivo puede estar solo o tratarse de una llave electrónica que sustituye a la contraseña para identificar a un usuario.

En este capítulo solo trataremos la autentificación mediante Firma Digital o por medio de funciones MAC, la autentificación mediante passwords, no son considerados en este contexto, debido a resientes declaraciones del Consejo de Arquitectura de Internet [IAB].

6.1 Funciones de Resumen o Hash

Algoritmo que genera un valor de resumen (hash) de algún dato, como una clave de mensaje o de sesión. Con un buen algoritmo de hash, los cambios que se produzcan en los datos de entrada pueden cambiar todos los bits del valor hash resultante, por lo que estos valores son útiles para detectar cualquier modificación en un objeto de datos, como un mensaje. Además, un buen algoritmo de hash hace que sea computacionalmente imposible crear dos entradas que tengan el mismo valor hash. Los algoritmos de hash comunes son MD2, MD4, MD5 y SHA-1. Estos algoritmos también se llaman funciones hash o MDC (Código Detector de Modificaciones), [www4], [www5], [Cripseg].

En la Sección 5 vimos que la criptografía asimétrica permitía autentificar información, es decir, poder asegurar que un mensaje m proviene de un emisor $\mathbf{A}$ y no de cualquier otro. Asimismo vimos que la autentificación debía hacerse empleando una función resumen y no codificando el mensaje completo.

Sabemos que un mensaje m puede ser autentificado codificando con la llave privada K_{Pr} el resultado de aplicarle una función resumen, $E_{K_{\mathrm{Pr}}}(r(m))$. Esa información adicional (que denominaremos firma o signatura del mensaje m) sólo puede ser generada por el poseedor de la clave privada K_{Pr}. Cualquiera que tenga la llave pública correspondiente estará en condiciones de decodificar y verificar la firma. Para que sea segura, la función resumen $r(x)$ debe cumplir ciertas características:

- El resumen del mensaje $r(m)$ debe ser de longitud fija, independientemente de la longitud de m.
- Dado m, es fácil calcular $r(m)$.
- Dado $r(m)$, es computacionalmente imposible recuperar m.
- Dado m, es computacionalmente imposible obtener un m' tal que $r(m)=r(m')$.
- Emplear signaturas de al menos 128 bits, siendo 160 bits el valor más usado.

Nota: Si solo aplicamos una función MDC (como MD5 o SHA-1) sin encriptar $r(m)$, solo estaremos protegiendo el mensaje de posibles modificaciones, sin autentificar su procedencia.

6.1.1 Estructura de una Función Resumen

En general, las funciones resumen se basan en la idea de funciones de compresión, que dan como resultado bloques de longitud n a partir de bloques de longitud m. Estas funciones se encadenan de forma iterativa, haciendo que la entrada en el paso i sea función del i-ésimo bloque del mensaje y de la salida del paso i-1 (ver Figura 6). En general, se suele incluir en alguno de los bloques del mensaje m (al principio o al final), información sobre la longitud total del mensaje. De esta forma se reducen las probabilidades de que dos mensajes con diferentes longitudes den el mismo valor en su resumen. En esta sección veremos dos algoritmos de generación de firmas: MD5 y SHA-1.

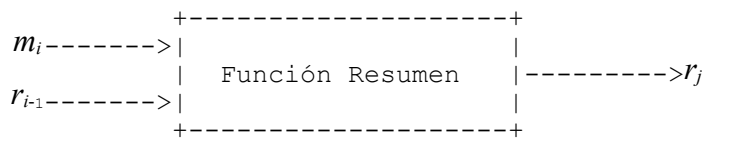


Figura 6: Estructura iterativa de una función resumen.

6.1.2 Algoritmos Generadores de Resúmenes

6.1.2.1 Algoritmo MD5

Resultado de una serie de mejoras sobre el algoritmo MD4 (el cual es una función hash que produce una secuencia de 128 bits asociados al mensaje original), diseñado por Ron Rivest, procesa los mensajes de entrada en bloques de 512 bits, y produce una salida de 128 bits. Siendo m un mensaje

de b bits de longitud, en primer lugar se alarga m hasta que su longitud sea exactamente 64 bits inferior a un múltiplo de 512. El alargamiento se lleva a cabo añadiendo un uno seguido de tantos ceros como sea necesario. En segundo lugar, se añaden 64 bits con el valor de b, empezando por el byte menos significativo. De esta forma tenemos el mensaje como un número entero de bloques de 512 bits, y además le hemos añadido información sobre su longitud. En los últimos tiempos el algoritmo MD5 ha mostrado ciertas debilidades, aunque sin implicaciones prácticas reales, por lo que se sigue considerando en la actualidad un algoritmo seguro, si bien su uso tiende a disminuir [MD5], [Cripseg].

6.1.2.2 El Algoritmo SHA-1

El algoritmo SHA-1 fue desarrollado por la NSA (Agencia Nacional de Seguridad de USA), para ser incluido en el estándar DSS (Digital Signature Standard). Produce firmas de 160 bits, a partir de bloques de 512 bits del mensaje original [Cripseg].

El algoritmo es similar a MD5, y se inicializa igual que éste, añadiendo al final del mensaje un uno seguido de tantos ceros como sea necesario hasta completar 448 bits en el último bloque, para luego yuxtaponer la longitud en bytes del propio mensaje. A diferencia de MD5, SHA-1 emplea cinco registros de 32 bits en lugar de cuatro.

6.2 Mecanismos de Autentificación Fuertes

6.2.1 Funciones de Autentificación de Mensaje (MAC)

Frente a los MDC (Códigos Detectores de Modificaciones), como el MD5 y el SHA-1, existe otra clase de funciones resumen, llamada genéricamente MAC (Códigos de Autentificación de Mensajes). Los MAC se caracterizan fundamentalmente por el empleo de una clave secreta para poder calcular la integridad del mensaje. Puesto que dicha clave sólo es conocida por el emisor y el receptor, el efecto conseguido es que el receptor puede, mediante el cálculo de dicha función, comprobar tanto la integridad como la procedencia del mensaje [HMACMD5], [CICESE], [Principi].

Existen multitud de MAC diferentes, pero lo más común es cifrar el mensaje mediante un algoritmo simétrico en modo CBC (ver Sección 4.2.2), y emplear la salida correspondiente al cifrado del último bloque.

Una de las funciones MAC es HMAC, Si bien el RFC 2104 no define estrictamente que significa el acrónimo HMAC pero da a suponer por el titulo del RFC "HMAC: Keyed-Hashing for Message Authentication" que es "Hash con Claves para la Autenticación de Mensajes" si bien algunos autores han supuesto que es HMAC es el acrónimo de "Hash Message Authentication Codes" [itu], [ECMWF], Código de Autenticación de Mensaje Troceado (o generado mediante la función hash). Debido a que MAC significa "Código de Autenticación de Mensaje".

Amén de cual sea el significado del acrónimo. Según lo descrito en [HMAC], HMAC es un algoritmo que consiste en aplicar una función resumen (hash) a la combinación de unos datos de entrada y una clave (que sólo la conocen el emisor y el receptor), siendo la salida una pequeña cadena de caracteres que denominamos extracto. Se puede utilizar HMAC con cualquier función de resumen, como por ejemplo MD5, SHA-1, junto con una clave secreta compartida. La solidez criptográfica de HMAC depende de las propiedades de la función de hash subyacente. El funcionamiento de AH se basa en un

algoritmo HMAC [HMACMD5], [CICESE], [win2003].

6.2.2 Firmas Digitales

Al usar claves públicas para firmas digitales, cada entidad requiere una clave pública y una privada. Los certificados son una parte esencial de los mecanismos de autentificación en una firma digital. Los certificados vinculan la identidad de una entidad específica (ya sea un host, un usuario o una aplicación) con sus claves públicas y posiblemente con otra información de seguridad relacionada, tales como privilegios, grupo, etc. La autentificación basada en firmas digitales requiere una tercera parte confiable o la creación de autoridades de certificación, la cuál firma y distribuye correctamente los certificados [ISAKMP].

6.2.2.1 Autoridades de Certificación

Los certificados requieren una infraestructura para la generación, verificación, revocación, administración y distribución. La Autoridad de Registración de Políticas en Internet (IPRA) [RFC-1422] a sido establecida para dirigir esta infraestructura por la IETF. La IPRA certifica las Autoridades de Certificación de Políticas (PCA). Las PCAs controlan a las Autoridades de Certificación (CA) las cuales certifican usuarios y entidades dependientes. El trabajo relacionado con la certificación actual incluye a los Sistemas de Nombres de Dominio (DNS) y a las Extensiones de Seguridad [DNSSEC] las cuales proporcionan la clave firmada a la entidad en el DNS. El Grupo de Trabajo para la Infraestructura de Clave Pública (PKIX) especifica un perfil para la Internet para los certificados X.509. Existen también trabajos realizados para desarrollar Servicios de Directorios X.500 que podrían proporcionar certificados X.509 a los usuarios. La oficina de correo de USA esta desarrollando una jerarquía de Autoridades de Certificación (CA). El Grupo de Trabajo para la Infraestructura de Clave Pública NIST ha estado desarrollando investigaciones en esta área. La Iniciativa de Seguridad de Sistemas de Información Multinivel (MISSI) del Departamento de Defensa del gobierno de USA (DOD) ha comenzado a desarrollar una infraestructura de certificación para el gobierno de USA. Alternativamente si no existe ninguna infraestructura de clave pública, Los PGP certificados de Red de Confianza (Web of Trust certificates) pueden ser utilizados para proporcionar autentificación y privacidad para el usuario en una comunidad de usuarios que se conocen y confían mutuamente [ISAKMP].

6.2.2.2 Nombramiento de la Entidad

El nombre de la entidad es su identidad y está ligado a su clave pública en los certificados. Las CA DEBEN definir la semántica para el nombramiento de los certificados. Un ejemplo de como una CA define su política de nombramiento se puede encontrar en [Berge]. Cuando se verifica un certificado, el nombre es verificado y ese nombre tendrá significado dentro del dominio de esa CA. Un ejemplo de esto son las extensiones de seguridad de los DNS que hacen los servidores CAs del DNS para las zonas y los nodos a los cuales sirven. Los registros de recursos se proporcionan para las claves públicas y las firmas de esas claves. Los nombres asociados a esas claves son asociados con las direcciones IP y con los nombres de dominio los cuales tienen un significado para las entidades que tienen acceso al DNS para esa información. Una Red de Confianza es otro ejemplo. Cuando se implementan redes de confianza, los nombres están ligados a las claves públicas. En PGP usualmente el nombre de la entidad es usualmente la dirección de e-mail el cuál tiene significado solamente para aquellos que entienden el correo electrónico. Otra red de confianza podría utilizar un esquema de nombramiento totalmente diferente [ISAKMP].

6.2.2.3 Certificados X.509

Un certificado es esencialmente una clave pública y un identificador, firmados digitalmente por una autoridad de certificación, y su utilidad es demostrar que una clave pública pertenece a un usuario concreto. El formato de certificados X.509 (Recomendación X.509 de CCITT: "The Directory - Autentication Framework". 1988) es el más común y extendido en la actualidad [www5], [hispa], [inco], [Cripseg].

El estándar X.509 sólo define la sintaxis de los certificados, por lo que no está atado a ningún algoritmo en particular, y contempla los siguientes campos:

- Versión.
- Número de serie.
- Identificador del algoritmo empleado para la firma digital.
- Nombre del certificador.
- Periodo de validez.
- Nombre del sujeto.
- Clave pública del sujeto.
- Identificador único de certificador.
- Identificador único de sujeto.
- Extensiones.
- Firma digital de todo lo anterior generada por el certificador.

Estos certificados se estructuran de forma jerárquica, de tal forma que nosotros podemos verificar la autenticidad de un certificado comprobando la firma de la autoridad que lo emitió, que a su vez tendrá otro certificado expedido por otra autoridad de rango superior. De esta forma vamos subiendo en la jerarquía hasta llegar al nivel más alto, que deberá estar ocupado por un certificador que goce de la confianza de toda la comunidad. Normalmente las claves públicas de los certificadores de mayor nivel se suelen publicar incluso en papel para que cualquiera pueda verificarlas. A ésta estructura se la conoce como, Infraestructura de Clave Publica PKI (Infraestructura de Clave Pública).

El mecanismo que debe emplearse para conseguir un certificado X.509 es enviar nuestra clave pública (nunca la privada) a la Autoridad de Certificación, la cual es una entidad encargada de establecer y avalar la autenticidad de las claves públicas pertenecientes a sujetos (normalmente usuarios o equipos) u otras entidades emisoras de certificados. Entre las actividades de una entidad emisora de certificados se encuentran enlazar claves públicas a nombres completos mediante certificados firmados, administrar los números de serie de los certificados y revocar certificados. Existen autoridades de certificación que, frente a una solicitud, generan un par llave pública-privada y lo envían al usuario. Hemos de hacer notar que en este caso, si bien tendremos un certificado válido, nuestro certificador podrá descifrar todos nuestros mensajes.

7. **Privacidad Bastante Buena (PGP**)

El nombre PGP responde a las siglas Pretty Good Privacy (Privacidad Bastante Buena). Actualmente PGP se ha convertido en un estándar internacional (RFC 2440), lo cual está dando lugar a la aparición de múltiples productos PGP, que permiten desde cifrar correo electrónico hasta codificar particiones enteras del disco duro (PGPDisk), pasando por la codificación automática y transparente de todo el tráfico TCP/IP (PGPnet) [Cripseg], [www1].

7.1 Fundamentos del PGP

PGP trabaja con criptografía asimétrica, y por ello tal vez su punto más fuerte sea precisamente la gran facilidad que ofrece al usuario a la hora de gestionar sus claves públicas y privadas. Si uno emplea algoritmos asimétricos, debe poseer las claves públicas de todos sus interlocutores, además de la clave privada propia. Con PGP surge el concepto de anillo de claves, que no es ni más ni menos que el lugar que este programa proporciona para que el usuario guarde todas las claves que posee. El anillo de claves es un único fichero en el que se pueden efectuar operaciones de extracción e inserción de claves de manera sencilla, y que además proporciona un mecanismo de identificación y autentificación de llaves completo y simple de utilizar. Esta facilidad en la gestión de claves es una de las causas fundamentales que han hecho a PGP tan popular. Los PGP 2.x.x emplean únicamente los algoritmos IDEA, RSA y MD5 [Cripseg].

7.2 Estructura de PGP

7.2.1 Codificación de Mensajes

Como el lector ya sabe, los algoritmos simétricos de cifrado son considerablemente más rápidos que los asimétricos. Por esta razón PGP cifra primero el mensaje empleando un algoritmo simétrico (ver Figura 7) con una clave generada aleatoriamente (clave de sesión) y posteriormente codifica la clave haciendo uso de la llave pública del destinatario. Dicha clave es extraída convenientemente del anillo de claves públicas a partir del identificador suministrado por el usuario, todo ello de forma transparente, por lo que únicamente debemos preocuparnos de indicar el mensaje a codificar y la lista de identificadores de los destinatarios. Nótese que para que el mensaje pueda ser leído por múltiples destinatarios basta con que se incluya en la cabecera la clave de sesión codificada con cada una de las claves públicas correspondientes [Cripseg].

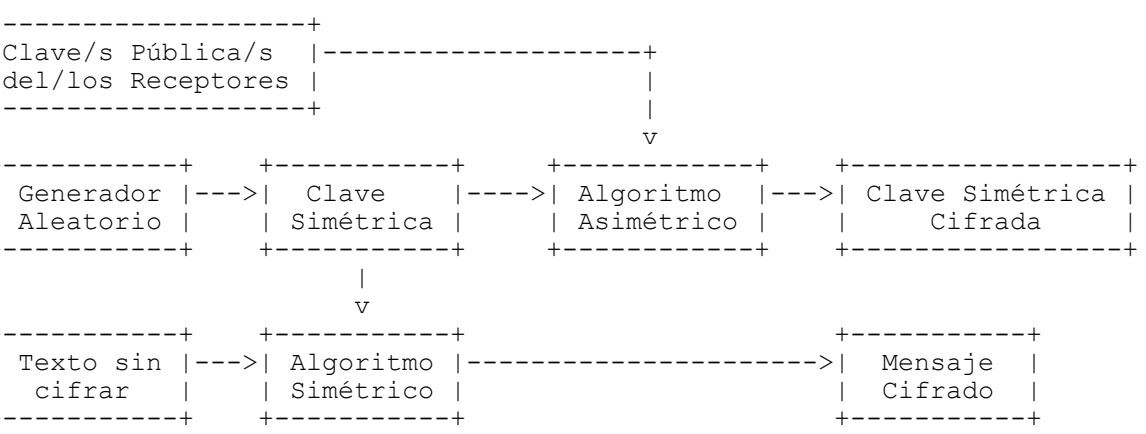


Figura 7: Codificación de un mensaje PGP

Cuando se trata de decodificar el mensaje, PGP simplemente busca en la cabecera las claves públicas con las que está codificado y nos pide una contraseña. La contraseña servirá para que PGP abra nuestro anillo de claves privadas y compruebe si tenemos una clave que permita decodificar el mensaje. En caso afirmativo, PGP descifrará el mensaje. Nótese que siempre que queramos hacer uso de una clave privada, habremos de suministrar a PGP

la contraseña correspondiente, por lo que si el anillo de claves privadas quedara comprometido, un atacante aún tendría que averiguar nuestra contraseña para descifrar nuestros mensajes. No obstante, si nuestro archivo de claves privadas cayera en malas manos, lo mejor será revocar todas las claves que tuviera almacenadas y generar otras nuevas.

Como puede comprenderse, gran parte de la seguridad de PGP reside en la calidad del generador aleatorio que se emplea para calcular las claves de sesión, puesto que si alguien logra predecir la secuencia de claves que estamos usando, podrá descifrar todos nuestros mensajes independientemente de los destinatarios a los que vayan dirigidos. Afortunadamente, PGP utiliza un método de generación de números pseudoaleatorios muy seguro (una secuencia aleatoria pura es imposible de conseguir), y protege criptográficamente la semilla que necesita. No obstante, consideraremos sensible al fichero que contiene dicha información y por lo tanto habremos de evitar que quede expuesto.

7.2.2 Firma Digital PGP

En lo que se refiere a la firma digital, las versiones actuales de PGP implementan el algoritmo DSS, que emplea la función resumen SHA-1 y el algoritmo asimétrico DSA. La firma digital o signatura puede ser añadida al fichero u obtenida en otro fichero aparte. Esta opción es muy útil si queremos firmar un fichero ejecutable.

8. **Amenazas y Ataques**

Las amenazas pueden ser clasificadas principalmente en cuatro grupos [Cripseg], [hispa], [inco]:

- Interrupción. El acceso a un recurso/comunicación se ve interrumpido ya sea físicamente (destrucción de la red...) o lógicamente (se modifica la localización, los derechos de acceso...).

- Intercepción. Alguien no autorizado consigue tener acceso al recurso/comunicación (pinchar la línea de red, sniffing...).

- Modificación. Obtención no sólo de acceso no autorizado al recurso/comunicación, sino también de la capacidad de modificarlo (modificación de los datos enviados/recibidos entre dos ordenadores...).

- Fabricación. Además de conseguir acceso al recurso/comunicación, se tiene la posibilidad de insertar información falsa.

Vistos los diferentes tipos de amenazas que pueden existir en una comunicación, podemos clasificar los posibles ataques en pasivos y activos.

- Ataques Pasivos: el atacante no altera la comunicación, tan sólo tiene acceso a ella. De esta forma puede saber que información circula por el canal, a que hora, la frecuencia y entre que personas. Este tipo de ataque es muy difícil de detectar ya que no aparece ningún signo que nos pueda advertir de que estamos siendo atacados.

- Ataques Activos: el atacante modifica el flujo de datos transmitidos o incluso crea uno falso, permitiendo incluso la suplantación de un usuario legítimo. Este tipo de ataques es mucho más grave ya que además de conseguir interceptar la comunicación, puede modificar su contenido falseándola, lo que implica que en caso de usar algún

sistema de seguridad este ha sido violado y se ha descubierto su clave de acceso o una de ellas (si es que hay) y el método utilizado para cifrar y descifrar la información. Estos ataques suelen acabar detectándose tras un cierto tiempo, su principal problema es si se detectan demasiado tarde.

Una vez visto los conceptos básicos veamos los tipos de ataques realizados a IPsec más frecuente:

Ataque por usuario interpuesto o Ataque en la trayectoria (o Man in de the middle Attacks)

Ataque a la seguridad en el que un intruso intercepta, y posiblemente modifica, datos que se transmiten entre dos usuarios. El intruso pretende hacerse pasar por la otra persona para cada uno de los usuarios. En un ataque por usuario interpuesto con éxito, los usuarios desconocen que hay un intruso entre ellos, que intercepta y modifica sus datos.

Ataque de Denegación de Servicio (o Saturación)

Ataque en el que un intruso aprovecha un defecto o una limitación de diseño de un servicio de red para sobrecargar o detener el servicio, de forma que éste no está disponible para su uso. Generalmente, este tipo de ataque se inicia para impedir que otros usuarios utilicen un servicio de red, como un servidor Web o un servidor de archivos.

Ataque por fuerza bruta

Si se tiene un criptograma mediante este método se probarían todas las claves posibles para obtener el texto plano. Si el conjunto de posibles claves es alto este sistema es inviable. Normalmente a este tipo de ataques no se les suele considerar como una forma de criptoanálisis ya que no busca puntos débiles, únicamente utiliza todas las claves posibles.

Ataque por Análisis de frecuencias

Este tipo de ataque es utilizado para romper sistemas criptográficos simétricos y se basa en estudiar la frecuencia con la que aparecen los distintos símbolos en un lenguaje determinado y luego estudiar la frecuencia con la que aparecen en los criptogramas, y de esta manera establecer una relación y obtener el texto plano.

Ataque por Tampering

Violación de seguridad en la comunicación, en la cual la información en transito es cambiada o reemplazada y es enviada hacia el receptor. (Definición extraída del Diccionario de IBM Corp.)

Ataque del Día de Cumpleaños (Birthday Attack)

El nombre deriva de la probabilidad de que dos o más personas en un grupo de 23 personas, compartan la misma fecha de cumpleaños es menor que 0.5, (conocida como paradoja del cumpleaños). El birthday attack es un nombre usado para referirse a una clase de ataque por fuerza bruta. Para una función hash que tiene como salida una cadena de 160 bits, es necesario recorrer entonces 2^80 mensajes para poder encontrar dos con el mismo hash, o sea encontrar una colisión.

Capítulo 6

Algoritmos de AH y ESP

1. **Introducción**

Este documento describe el uso del algoritmo HAMC (Sección 6.2.1 del Capítulo 5) en conjunto con el algoritmo MD5 (Sección 6.1.2.1 del Capítulo 5) y en conjunto con el algoritmo SHA-1 (Sección 6.1.2.2 del Capítulo 5), como mecanismo de autentificación, para la Carga de Seguridad Encapsulada (Capítulo 4) y para la Cabecera de Autentificación (Capítulo 3) en IPsec. HMAC con MD5 y HMAC con SHA-1 proporcionan autentificación del origen de datos y protección de integridad. Asimismo este documento describe el uso del algoritmo de cifrado DES (Sección 4.1.4.1 del Capítulo 5) en Modo Encadenamiento de Bloque Cifrado (CBC), con un Vector de Inicialización (IV) explícito (vea la Sección 4.1.2 del Capítulo 5 para una descripción detallada del modo CBC con IV), como mecanismo de confidencialidad dentro del contexto de ESP. Como así también define el algoritmo de encriptación NULL y su uso con la Carga de Seguridad Encapsulada [ESP] en IPsec. NULL no altera los datos del texto plano. En realidad, NULL, por si mismo, no hace nada. NULL proporciona los medios para que ESP proporcione autentificación y integridad sin confidencialidad.

Actualmente toda implementación de AH y de ESP deben soportar el uso de los algoritmos HMAC-MD5-96 [HMACMD5] y de HMAC-SHA-1-96 [HMACSHA] como mecanicismos de autentificación. Toda implementación de ESP debe soportar el uso de los algoritmos de cifrado DES-CBC [DES] y NULL [ESPNULL] como mínimo. Si bien es cierto que el algoritmo DES ya esta prácticamente en desuso, debido a su vulnerabilidad, todo ello es en post de la interoperativilidad, y cualquier implementación AH y/o ESP puede elegir (y es muy recomendable que lo haga) implementar otros tipos de algoritmos más fuertes, como por ejemplo, 3DES, IDEA, o AES, entre otros.

2. **Mecanismos de Autentificación Requeridos por AH y ESP**

Como [HMAC] proporciona un marco para incorporar varios algoritmos de hash con HMAC, es posible usarlo con algoritmos tales como MD5, o como SHA-1, para formar los algoritmos HMAC-MD5-96 y el HMAC-SHA-1-96, los cuales son en esencia el algoritmo MD5 [MD5] combinado con HMAC [HMAC] y el algoritmo SHA-1 [FIPS-180-1] combinado con el algoritmo HMAC [HMAC] respectivamente, como mecanismo de autentificación de claves, para la Carga de Seguridad Encapsulada [ESP] y para la Cabecera de Autentificación [AH] dentro del contexto de IPsec. EL propósito de HMAC-MD5-96 y de HMAC-SHA-1-96 es asegurar que el paquete es auténtico y que no puede ser modificado en tránsito.

HMAC es un algoritmo de autentificación de clave secreta, donde la integridad de los datos y la autentificación del origen de los datos como es proporcionada por HMAC es dependiente del alcance de la distribución de la clave secreta. Si solamente el origen y el destino conocen la clave HMAC, esto proporciona autentificación del origen de los datos e integridad de los datos para los paquetes enviados entre las partes; si el HMAC es correcto, esto prueba que el HMAC a sido agregado por el origen.

2.1 Modo y Algoritmo

HMAC-MD5-96 y HMAC-SHA-1-96 operan sobre bloques de datos de 64 bits. Los requerimientos para el relleno de HMAC-MD5-96 están especificados en [MD5] y son parte del algoritmo MD5. Si usted construye el MD5 de acuerdo con [MD5] no necesitará agregar rellenos adicionales para HMAC-MD5-96. Los

requerimientos para el relleno de HMAC-SHA-1-96 están especificados en [FIPS-180-1] y son parte del algoritmo SHA-1. Si usted construye SHA-1 de acuerdo con [FIPS-180-1] no necesitará agregar rellenos adicionales para HMAC-SHA-1-96. Con respecto al "relleno del paquete implícito" como lo define [AH] no es requerido.

HMAC-MD5-1-96 produce un valor de autentificación de 128 bits. HMAC-SHA-1-96 produce un valor de autentificación de 160 bits. Este valor de 128 bits (160 bits para el caso de HMAC-SHA-1-96) puede ser acortado como se describe en [HMAC]. Para usarse con ESP o AH, un valor truncado usando los primeros 96 bits BEBE ser soportado. En el envío, el valor truncado es almacenado dentro del campo de autentificación. En la recepción el valor completo de 128 bits (160 bits para el caso de HMAC-SHA-1-96) es calculado y los primeros 96 bits son comparados con el valor almacenado en el campo autentificación. Ninguna otra longitud del valor de autentificación es soportado por HMAC-MD5-96 ni por HMAC-SHA-1-96.

La longitud de 96 bits fue seleccionada porque es la longitud de autentificación por defecto como está especificado en [AH] y soluciona los requisitos de seguridad descriptos en [HMAC].

[Bellare96a] indica que "el funcionamiento (de HMAC) es esencialmente el de la función hash principal". El [RFC-1810] proporciona un análisis del funcionamiento y recomendaciones del uso de MD5 con los protocolos de Internet.

2.2 Material Clave

HMAC-MD5-96 y HMAC-SHA-1-96 son algoritmos de clave secreta. A pesar de que ninguna longitud de clave fija es especificada en [HMAC], cuando se usa en ESP o AH una longitud de clave fija de 128 bits (160 bits para el caso de HMAC-SHA-1-96) DEBE ser soportada. Solamente claves de 128 bits y de 160 bits DEBEN ser usadas por HMAC-MD5-96 y HMAC-SHA-1-96 respectivamente. Una longitud de clave de 128 bits (160 bits para el caso de HMAC-SHA-1-96) fue elegida basándose en las recomendaciones de [HMAC] (es decir longitudes de claves menores a la longitud de autentificación debilitan la seguridad y claves más largas que la longitud de autentificación no incrementan la seguridad).

El [HMAC] discute los requerimientos sobre el material clave, incluyendo una discusión de requerimientos de aleatoriedad fuerte. Una función seudo-aleatoria fuerte DEBE ser usada para generar la clave de 128 bits (160 bits para el caso de HMAC-SHA-1-96) requerida.

Si, en algún momento, un conjunto de claves débiles para HMAC es identificado, el uso de estas claves débiles debe ser rechazado, seguido de una solicitud de reemplazo de la/s clave/s o de la negociación de una nueva/s SA.

En el Capítulo 2 se describió el mecanismo general para obtener el material clave cuando claves múltiples son requeridas para una única SA (por ejemplo cuando una SA ESP solicita una clave para confidencialidad y una clave para autentificación).

Para proporcionar la autentificación del origen de los datos, los mecanismos de distribución de claves deben asegurar que claves únicas sean asignadas y que estén distribuidas solamente a las partes participantes en la comunicación.

[HMAC] hace la siguiente recomendación con relación al recambio de claves:

> Los ataques actuales no indican una frecuencia específica para el cambio de claves ya que estos ataques son prácticamente impracticables. Sin embargo, la renovación periódica de las claves es una práctica de seguridad fundamental que ayuda contra debilidades potenciales de la función y claves, reduce la información disponible a un criptoanálisis y limita el daño de una clave expuesta.

2.3 Consideraciones de Seguridad del Algoritmo HMAC-MD5-96 y HMAC-SHA-1-96

La seguridad proporcionada por HMAC-MD5-96 o por HMAC-SHA-1-96 se basa en la fuerza de HMAC y en menor grado, en la fuerza de MD5 (o en el caso de HMAC-SHA-1-96 de fuerza de SHA-1). [HMAC] requiere que HMAC no dependa de la propiedad de la resistencia fuerte a colisiones, que es importante de considerar cuándo se evalúa el uso de MD5, aunque, bajo pruebas recientes, a mostrado ser menos resistente a colisiones que en un primer momento.

[HMAC] indica que para que las "funciones hash sean minímamente coherentes" el birthday attack (vea la Sección 8 del Capítulo 5), el ataque más eficiente conocido contra HMAC, sea impracticable. Para un bloque de hash de 64 bits tal como HMAC-MD5-96 (o como HMAC-SHA-1-96) un ataque incluyendo el procesamiento exitoso de bloques de 2^64 (o de 2^80 para el caso de HMAC-SHA-1-96) no sería práctico a menos que se hubiera descubierto que el hash principal tuvo colisiones después de procesar bloques de 2^30. Un hash con tales características de resistencia débil a colisiones sería generalmente considerado inservible.

También es importante considerar que mientras que MD5 como así también SHA-1 nunca fueron desarrollados para ser usados como algoritmos de claves hash, HMAC tuvo ese criterio desde el principio. Mientras que el uso de MD5 en el contexto de seguridad de datos esta experimentando la reevaluación, la combinación de HMAC con el algoritmo MD5 ha estado sujeto a examen criptográfico.

Así como es cierto que para cualquier algoritmo criptográfico, parte de su fuerza recae en la correcta aplicación del algoritmo, la seguridad del mecanismo de administración de clave y su implementación, la fuerza de la clave secreta asociada y la correcta implementación de todos los sistemas participantes. El [RFC-2202] contiene vectores de prueba y ejemplos de código para asistir en la verificación de la exactitud del código de HMAC-MD5-96 y de HMAC-SHA-1-96.

3. **Mecanismos de Cifrado Requerido por ESP**

3.1 El Algoritmo de Cifrado DES-CBC con IV explícito en ESP

Esta sección describe el uso del algoritmo de cifrado DES en Modo Encadenamiento de Bloque Cifrado (CBC), con un Vector de Inicialización (IV) explícito, como mecanismo de confidencialidad dentro del contexto de la Carga de Seguridad Encapsulada [ESP] en IPsec.

El DES es un algoritmo de cifrado de bloque simétrico. El algoritmo se describió en el Capítulo 5 y se describe con más detalle en [FIPS-46-2] [FIPS-74] [FIPS-81]. El Capítulo 5 proporciona una descripción general del Modo Encadenamiento de Bloque Cifrado como así también [Schneier], un modo que es aplicable a varios algoritmos de encriptación.

DES-CBC es un algoritmo de bloque de clave secreta simétrica y tiene un tamaño de bloque de 64 bits.

3.1.1 Carga ESP

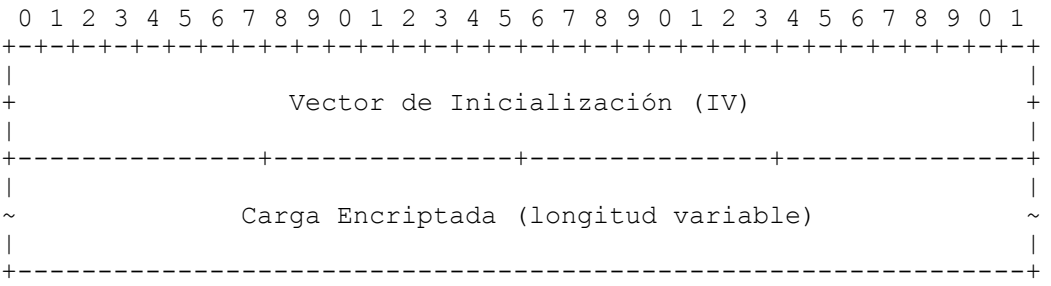
DES-CBC requiere de un Vector de Inicialización (IV) explícito de 8 octetos (64 bits). Este IV precede a la carga protegida (encriptada). El IV DEBE ser un valor aleatorio.

Incluir el IV en cada datagrama asegura de que la desencriptación de cada datagrama recibido pueda ser realizada, incluso cuando algunos datagramas se pierden, o lleguen en desorden.

Nota de implementación:

> La práctica común es utilizar datos aleatorios para el primer IV, para los restantes IV se usan datos aleatorios y los últimos 8 octetos de datos encriptados del proceso de encriptación del paquete anterior; esto extiende la lógica CBC a través de los paquetes. También tiene la ventaja de limitar la filtración de información del número generado aleatoriamente. No importa qué mecanismos se utilicen, el receptor NO DEBE asumir ningún significado de este valor, con excepción de que ése es un IV.
>
> Para evitar encriptación CBC de bloques de texto plano de la misma manera en diferentes paquetes, las implementaciones NO DEBEN utilizar un contador o otro origen de distancia de Hamming bajo para IVs.

El campo carga, según lo definido en [ESP], se analiza según el siguiente diagrama:



3.1.1.1 Tamaño del Bloque y Relleno

El algoritmo DES-CBC descripto en este capítulo DEBE utilizar un tamaño de bloque de 8 octetos (64 bits).

Cuando se requiera relleno, este DEBE ser realizado según las convenciones especificadas en el Capítulo 4.

3.1.2 Material Clave

DES-CBC es un algoritmo de clave secreta simétrico. El tamaño de la clave es de 64 bits. (Se conoce comúnmente como clave de 56 bits puesto que la clave tiene 56 bits significativos; el bit menos significativo en cada byte es el bit de paridad.)

El Capítulo 2 describe el mecanismo general para derivar material clave de

la transformación ESP. La derivación de la clave de una cierta cantidad de material clave no se diferencia entre asociaciones de seguridad (SA) de clave manual y de clave automática.

Este mecanismo DEBE derivar un valor de clave de 64 bit para utilizarse según ese cifrado. El mecanismo derivará valores de claves en bruto, el proceso de derivación en si mismo no es responsable de manejar paridad o verificaciones de claves débiles.

Las verificaciones de claves débiles DEBERÍA ser realizada. Si se encuentra tal clave, la clave DEBERÍA ser rechazada y una nueva SA se requerirá.

Una función seudo aleatoria fuerte DEBE ser utilizada para generar la clave requerida. Para una discusión sobre este asunto, referirse a [RFC1750].

3.1.2.1 Claves Débiles

El DES tiene 16 claves débiles conocidas, incluyendo también las claves llamadas semi-débiles. La lista de claves débiles puede ser encontrada en la Sección 4.1.4.1 del Capítulo 5.

3.1.2.2 Tiempo de Vida de las Claves

[Blaze] discute los costos y el tiempo de recuperación de claves de ataques por fuerza bruta. Presenta varias combinaciones de costo total/tiempo para recuperar una clave/costo por clave recuperada de 40 bit y claves DES de 56 bits, basado en 1995 estimaciones.

Mientras que una búsqueda por fuerza bruta de un espacio de clave DES de 56 bits puede ser considerado impracticable para el hacker, que está utilizando simples ciclos de CPU o otros recursos menos costosos, esto está dentro del alcance de alguien que quiere gastar un poco más de dinero.

Por ejemplo, en el año 1998, con un costo de $300.000, una clave DES de 56 bits se puede recuperar aproximadamente en 19 días usando tecnología comercial disponible y solamente 3 horas usando un chip desarrollado a pedido.

Se debe observar que hay otros ataques que pueden recuperar claves más rápido, los ataques por fuerza bruta están considerados como los de la "peor clase", aunque son los más fácil de implementar.

Se debe observar que con el correr del tiempo los costos de búsqueda total y/o parcial, así como también el tipo de recuperación parcial de clave seguirán disminuyendo.

Se debería observar que dado una cierta cantidad tráfico habrá mayor probabilidad de que el texto plano conocido pueda ser acumulado.

3.1.3 Consideraciones de Seguridad para el Algoritmo DES-CBC con IV

Los usuarios necesitan entender que el grado de seguridad proporcionada por esta especificación depende completamente de la fuerza del algoritmo DES, la exactitud de la implementación de esos algoritmos, la seguridad del mecanismo de administración de Asociación de Seguridad y de su implementación, de la fuerza de la clave [CN94], y la exactitud de las implementaciones en todos los nodos que participan.

[Bell95] y [Bell96] describen un ataque empleando cortar y pegar que se

aplica a todos los algoritmos de Encadenamiento de Bloque Cifrado. Este ataque se puede solucionar con el uso de mecanismos de autentificación.

El uso de mecanismos de encriptación sin ningún mecanismo de autentificación no se recomienda. Este cifrado puede ser utilizado en una transformación ESP que también incluya autentificación; esto también puede ser utilizado en una transformación ESP que no proporcione autentificación incluida (en ESP) pero hay una cabecera AH (proporcionando autentificación).

Cuando el relleno de ESP es utilizado, los bytes de relleno tienen un valor previsible. Proporcionando una pequeña cantidad de detección de sabotaje (tamper) sobre su propio bloque y sobre el bloque anterior en modo CBC. Esto hace que sea un poco más difícil realizar ataques uniendo (splicing) y evitando un posible canal secreto. Esta pequeña cantidad de texto plano conocido no crea ningún problema para los cifrados modernos.

[BS93] demostró un criptoanálisis diferencial basado en la elección del texto plano, el ataque requiere 2^47 pares de texto plano-texto cifrado, donde el tamaño de un par es el tamaño de un bloque DES (64 bits). [Matsui94] demostró un criptoanálisis lineal basado en la elección del texto plano conocido, el ataque solamente requería 2^43 pares de; texto plano, texto cifrado. Aunque estos ataques no son considerados prácticos, se deben tener en cuenta. [Wiener94] muestra el diseño de una máquina de cracking para DES que puede craquear una clave cada 3,5 horas. Esto es un ataque extremadamente práctico.

Uno o dos bloques de texto plano conocido son suficientes para recuperar una clave DES. Debido a que los datagramas IP comienzan típicamente con un bloque de texto conocido y/o predecible de la cabecera, los cambios frecuentes de clave no protegerán contra este ataque.

Se sugiere que el DES no es un buen algoritmo de encriptación para la protección de información de valor moderado. El triple DES es probablemente una mejor opción para tales propósitos.

Sin embargo, a pesar de estos riesgos potenciales, el nivel de privacidad proporcionado por ESP con DES-CBC es mayor que enviar el datagrama en texto plano a través de Internet.

En el caso de usar los valores aleatorios para los IV se ha refinado el siguiente resumen proporcionado por Steve Bellovin. Referirse a [Bell97] para mayor información.

> "El problema se presenta si usted utiliza un contador como IV, o otra fuente con una distancia de Hamming baja entre sucesivos IV, para la encriptación en modo CBC. En modo CBC, el "texto plano efectivo" para una encriptación es el XOR del texto plano actual y del texto cifrado del bloque precedente. Normalmente, ese es un valor aleatorio, que significa que el texto plano efectivo es algo aleatorio. Eso es favorable, debido a que muchos de los bloques de texto plano actual no cambian mucho entre paquetes.
>
> Para el primer bloque del texto plano el IV toma el lugar del bloque anterior del texto cifrado. Si el IV no se diferencia mucho del IV anterior, y el bloque actual del texto plano no se diferencia mucho del paquete anterior, entonces el texto plano efectivo tampoco se diferenciará mucho. Esto significa que usted tiene pares de bloques de texto cifrado combinados con los bloques de texto plano que se

diferencian en apenas algunas posiciones de bits. Esto puede ser una ventaja para los diversos ataques criptográficos."

Se debe hacer notar que una implementación no debe usar una fuente de distancia de Hamming baja para los IV.

3.2 Algoritmo de Encriptación NULL

Esta sección define el algoritmo de encriptación NULL y su uso con la Carga de Seguridad Encapsulada [ESP] en IPsec. NULL no altera los datos del texto plano. En realidad, NULL, por si mismo, no hace nada. NULL proporciona los medios para que ESP proporcione autentificación y integridad sin confidencialidad.

El Capítulo 4 especifica el uso de un algoritmo de encriptación opcional para proporcionar confidencialidad y el uso de un algoritmo de autentificación opcional para proporcionar autentificación y integridad. El algoritmo de encriptación NULL es un modo conveniente de representar la opción de no aplicar encriptación. Esto es referido como ESP_NULL en el Capítulo 8.

La especificación de la cabecera de Autentificación (Capítulo 3) proporciona un servicio similar, el cálculo de los datos de autentificación cubre, la parte de datos de un paquete, como así también las partes que no se modifican durante el trasporte en la cabecera IP. ESP_NULL no incluye la cabecera IP en el cálculo de los datos de autentificación. Esto puede ser útil para proporcionar servicios IP a través de dispositivos de redes no IP. La discusión de como ESP_NULL debería ser usado con dispositivos de redes no IP esta fuera del alcance de este libro.

3.2.1 Definición del Algoritmo

NULL esta matemáticamente definido por el uso de la función Identidad I aplicada a un bloque de datos b tal que:

$$NULL(b) = I(b) = b$$

3.2.1.1 Material Clave

Así como otros cifrados modernos, por ejemplo RC5 [RFC-2040], el algoritmo de encriptación NULL puede hacer uso de claves de longitud variable. Sin embargo no existe un incremento cuantificable de seguridad mediante el uso de longitudes de claves más largas.

3.2.1.2 Sincronización Criptográfica

Debido a la naturaleza desnacionalizada del algoritmo de encriptación NULL no es necesario trasmitir un IV o datos de sincronización criptográficos similares en cada paquete básico (o por cada SA). El algoritmo de encriptación NULL combina muchas de las mejores características del encadenamiento de bloque cifrado y del encadenamiento de flujo cifrado mientras que todavía no requiere la transmisión de un IV o datos de sincronización criptográficos análogos.

3.2.1.3 Relleno

NULL tiene un tamaño de bloque de 1 byte, de esa manera el relleno no es necesario.

3.2.1.4 Funcionamiento

El algoritmo de encriptación NULL es significativamente más rápido que otros algoritmos de encriptación simétricos y las implementaciones del algoritmo están disponibles para todo hardware y Sistema Operativo.

3.2.1.5 Vectores de Prueba

Los siguiente es un conjunto de vectores de prueba para facilitar el desarrollo de interoperatibilidad de implementaciones NULL.

```
test_case = 1
data =      0x123456789abcdef
data_len =  8
NULL_data = 0x123456789abcdef

test_case = 2
data =     "Network Security People Have A Strange Sense Of Humor"
data_len = 53
NULL_data ="Network Security People Have A Strange Sense Of Humor"
```

3.2.2 Requisitos operacionales de ESP_NULL

ESP_NULL esta definido por el uso de NULL dentro del contexto de ESP. Esta sección define ESP_NULL explicando los requisitos particulares de parámetros operacionales.

Para los propósitos de extracción de clave de IKE (vea el Capítulo 10), el tamaño de la clave para este algoritmo DEBE ser de cero bits (0), para facilitar la interoperatibilidad y para evitar problemas potenciales de control de exportación.

Para facilitar la interoperatibilidad, el tamaño del IV para este algoritmo debe ser de cero (0) bits.

El relleno PUEDE ser incluido en paquetes salientes como esta especificado en el Capítulo 4.

3.2.3 Consideraciones de seguridad para el Algoritmo NULL

El algoritmo de encriptación NULL no ofrece confidencialidad ni cualquier otro servicio de seguridad. Es simplemente un modo conveniente de representar el uso opcional de aplicar encriptación dentro de ESP. Por lo tanto ESP puede ser usado para proporcionar autentificación y integridad sin confidencialidad. Diferente a AH, estos servicios no son aplicados a ninguna parte de la cabecera IP. Al momento de la creación de este documento no hay evidencia para decir que ESP_NULL es menos seguro que AH cuando se usa el mismo algoritmo de autentificación (es decir un paquete asegurado usando ESP_NULL con algún algoritmo de autentificación es tan seguro criptográficamente hablando como un paquete asegurado usado AH con el mismo algoritmo de autentificación).

El uso de algoritmos de encriptación y de autentificación son opcionales en ESP, pero es imperativo que una SA ESP especifique, el uso de al menos un algoritmo de encriptación criptográficamente fuerte o un algoritmo de autentificación criptográficamente fuerte o uno de cada uno.

Al momento de la redacción de este documento no existen leyes conocidas que impidan la exportación de NULL con una longitud de clave de cero (0) bits.

Capítulo 7

El Protocolo de Gestión de Claves y Asociaciones de Seguridad en Internet (ISAKMP)

1. **Introducción**

Este capítulo describe el Protocolo de Gestión de Claves y Asociaciones de Seguridad en Internet (ISAKMP), que utiliza conceptos de seguridad necesarios para el establecimiento de Asociaciones de Seguridad (SA) y claves criptográficas en un entorno de Internet. Un protocolo que negocia, establece, modifica y cancela SAs y sus atributos es requerido para la Internet en desarrollo, donde existirán numerosos mecanismos de seguridad y varias opciones para cada mecanismo. El protocolo de seguridad de manejo de claves debe ser robusto para manejar la generación de claves públicas para la comunidad de Internet y los requerimientos de claves privadas para esas redes privadas que lo requieran. El Protocolo para el manejo de claves y Asociaciones de Seguridad (ISAKMP) define los procedimientos para autentificar comunicaciones entre usuarios, la creación y administración de Asociaciones de Seguridad, las técnicas de generación de claves y atenuación de amenazas (como por ejemplo, denegación de servicio y ataques de reenvío). Todo esto es necesario para entablar y mantener comunicaciones seguras (A través de los servicios de seguridad IP o cualquier otro protocolo de seguridad) en un entorno de Internet.

ISAKMP combina los conceptos de seguridad de autentificación, administración de claves, y de asociaciones de seguridad para establecer la seguridad requerida por el gobierno, comercio, y comunicaciones privadas en Internet.

ISAKMP define los procedimientos y los formatos de los paquetes para establecer, negociar, modificar y eliminar las Asociaciones de Seguridad. Las SAs contienen toda la información requerida para la ejecución de diversos servicios de seguridad de red, tales como los servicios de la capa IP (tales como la cabecera de autentificación y la cabecera de carga de seguridad encapsulada), trasporte o servicios de la capa aplicación o autoprotección del tráfico de negociación. ISAKMP define las cargas para el intercambio de generación de claves y autentificación de datos. Estos formatos proporcionan un marco consistente para la transferencia de claves y autentificación de datos que es independiente de la técnica de generación de claves, algoritmo de encriptación, y mecanismo de autentificación.

ISAKMP es diferente al protocolo de intercambio de claves para separar claramente los detalles de la administración de SA (y administración de claves) de los detalles de intercambio de claves. Puede haber diferentes protocolos de intercambio de claves, cada uno con propiedades de seguridad diferente. Sin embargo, un marco común es requerido para acordar el formato de los atributos de la SA, y para la negociación, modificación, y cancelación de SAs. ISAKMP proporciona este marco común.

La separación de la funcionalidad en tres partes agrega complejidad al análisis de seguridad a una implementación de ISAKMP. No obstante, la separación es importante para la interoperatibilidad entre sistemas con requerimientos de seguridad diferentes, y debería también simplificar el análisis de una futura evolución de un servidor ISAKMP.

ISAKMP está diseñado para soportar la negociación de SAs de los protocolos de seguridad de todas las capas de la pila de protocolos de red (IPsec, TLS, TLSP, OSPF, etc.). Centralizando la administración de SAs, ISAKMP reduce el costo de la funcionalidad duplicada dentro de cada protocolo de seguridad. ISAKMP también puede reducir el tiempo de inicio de conexión, negociando un conjunto de servicios al mismo tiempo.

El resto de la sección 1 establece la motivación para proporcionar

negociaciones seguras y detalla los componentes principales de ISAKMP, como por ejemplo,la administración, la autentificación, las SAs, la criptografía de clave pública o otros temas relacionados. La Sección 2 presenta la terminología y los conceptos relacionados con ISAKMP. La Sección 3 describe los diferentes formatos de carga ISAKMP. La Sección 4 describe como están compuestas las cargas ISAKMP y los tipos de intercambios para establecer SAs y realizar intercambios de claves en un modo autentificado. La modificación, cancelación, y notificación de error de una SA, son también analizados en esa sección. La Sección 5 describe el procesamiento de cada carga dentro del contexto de intercambios de ISAKMP, incluyendo el manejo de errores y temas relacionados. La Sección 6 describe los valores de los atributos necesarios para una SA ISAKMP. La Sección 7 describe los requisitos necesarios para definir un nuevo Dominio de Interpretación (DOI) dentro de ISAKMP.

1.1 La Necesidad de Negociación

Los servicios de seguridad requeridos para las comunicaciones dependen de la configuración de las redes individuales y del ambiente en el que estan inmersos. Las organizaciones que forman Redes Privadas Virtuales (VPN), también conocidas como Intranet, necesitarán un conjunto de funciones de seguridad para las comunicaciones dentro de las VPN y posiblemente muchas funciones de seguridad diferentes para comunicaciones fuera de las VPN, para soportar geográficamente componentes organizativos separados, clientes, proveedores, subcontratistas (con sus propias VPNs), gobiernos, y otros. Los departamentos dentro de las grandes organizaciones requerirán un número de SA para separar y proteger los datos (por ejemplo, los datos personales, de los datos de la compañía, etc.) en redes internas y en otras SAs para comunicarse dentro del mismo departamento. Usuarios móviles que quieren acceder a sus empresas (o a sus puestos de trabajo representan otro conjunto de requerimientos de seguridad. Entidades más pequeñas pueden resolver sus requisitos de seguridad estableciendo "redes de confianza". Los intercambios de ISAKMP proporcionan a estas comunidades de red la capacidad de presentar usuarios con funciones de seguridad que el usuario soporta en un modo acordado, autentificado y protegido sobre un conjunto de atributos de seguridad en común, es decir una, SA ínter operable.

1.2 Qué Puede ser Negociado

Las Asociaciones de Seguridad deben soportar diversos algoritmos de encriptación, autentificación y mecanismos de establecimiento de claves para IPsec, como así también para otros protocolos de seguridad. Las Asociaciones de seguridad también deben soportar certificados orientados a host para los protocolos de capas inferiores y certificados orientados a usuarios para protocolos de capas superiores. Algoritmos y mecanismos independientes se requieren en aplicaciones tales como, e-mail, conexión remota, transferencia de archivos, como así también sesiones orientadas a protocolos, protocolos de ruteo, y protocolos de capas de enlace. ISAKMP proporciona una SA común y protocolos de establecimiento de claves para esta gran variedad de protocolos de seguridad, aplicaciones, requerimientos de seguridad y ambientes de redes.

ISAKMP no está sujeto a ningún algoritmo criptográfico especifico, técnica de generación de claves o mecanismos de seguridad. Esta flexibilidad es beneficiosa por numerosas razones. Primero porque soporta ambientes de comunicaciones dinámicos. Segundo independencia de los mecanismos de seguridad específicos y suministra a los algoritmos un mejor camino migratorio progresivo para mecanismos y algoritmos. Cuando mejores mecanismos de seguridad son desarrollados o nuevos ataques a algoritmos de

encriptación actuales, mecanismos de autentificación o intercambios de generación de claves son descubiertos ISAKMP permitirá la actualización de algoritmos y mecanismos sin tener que desarrollar un nuevo protocolo o mejorar el actual.

ISAKMP tiene requisitos básicos para su autentificación y componentes de intercambio de claves. Estos requerimientos protegen contra la denegación de servicio, el reenvío/reflexión, ataques en la trayectoria, y ataques contra secuestro de la conexión. Esto es importante porque estos son los tipos de ataques que están dirigidos hacia los protocolos. El completo soporte de SA, el cual proporciona, mecanismos y algoritmos independientes y protección de los protocolos contra amenazas son las fortalezas de ISAKMP.

1.3 Asociaciones de Seguridad y Administración

Una SA es una relación entre dos o más entidades que describe como las entidades utilizan los servicios de seguridad para comunicarse en forma segura. Esta relación esta representada por un conjunto de información que puede ser considerada como un contrato entre las entidades. La información debe ser acordada y compartida por todas las entidades. Algunas veces solo la información es referida como una SA, pero esto es una ejemplificación física de la relación existente. La existencia de esta relación, representada por la información, es lo que proporciona la información de seguridad necesaria para que ínter-operen de forma segura. Todas las entidades se deben adherir a la SA para que sean posibles las comunicaciones seguras. Cuando se accede a los atributos de las SAs, las entidades usan un puntero o un identificador que hace referencia a un SPI. El Capítulo 2 proporcionó los detalles referentes a las definiciones de SA y de SPI.

1.3.1 Asociaciones de Seguridad y Registraciones

Los atributos requeridos y recomendados para una SA IPsec (AH, ESP) se definieron en el Capítulo 2. Los atributos específicos para una SA IPsec incluyen, pero no están limitados a, mecanismos de autentificación, algoritmos criptográficos, modos de algoritmos, longitud de las claves, y el Vector de Inicialización (IV). Otros protocolos que proporcionen algoritmos y mecanismos independientes de seguridad DEBEN definir sus requerimientos para los atributos SA. ISAKMP tiene su propia SA (la SA ISAKMP, la cual es diferente a la SA IPsec), la separación de una definición específica es importante para asegurar que ISAKMP pueda establecer SAs para todos los posibles protocolos de seguridad y aplicaciones.

NOTA: Vea el Capítulo 8 para un debate de los atributos de las SA que deberían ser considerados para las definiciones de un protocolo de seguridad o aplicaciones.

Para facilitar la rápida identificación de atributos específicos (por ejemplo, un algoritmo de encriptación específico) entre varias entidades se DEBEN designar identificadores de atributos y estos identificadores deben ser registrados por una autoridad central. La Autoridad de Asignación de Números en Internet (IANA) proporciona esta función para Internet.

1.3.2 Requisitos de ISAKMP

El establecimiento de SA DEBE ser parte del protocolo de manejo de claves definidos para las redes basadas en IP. El concepto de SA es requerido para

soportar protocolos de seguridad en ambientes diversos y dinámicos de red. La autentificación y el intercambio de claves deben estar vinculados para asegurar que la clave este establecida con la parte autentificada [DOW92], el establecimiento de una SA debe estar vinculado con la autentificación y el protocolo de intercambio de claves.

ISAKMP proporciona el protocolo de intercambio para establecer una SA entre entidades negociantes después del establecimiento de una SA para estas entidades negociantes en representación de algún protocolo (por ejemplo ESP/AH). Primero, un intercambio inicial de protocolo permite un conjunto de atributos de seguridad acordados. Este conjunto básico proporciona protección para los intercambios subsiguientes de ISAKMP. También indica el método de autentificación y el intercambio de claves que serán realizados como parte del protocolo ISAKMP. Si un conjunto básico de atributos de seguridad ya esta en su sitio entre las entidades de negociación del servidor, el intercambio ISAKMP inicial puede ser omitido y el establecimiento de la SA puede ser realizado directamente. Después de que el conjunto básico de atributos de seguridad haya sido acordado, la autenticidad de identidad inicial, y las claves requeridas generadas, la SA establecida puede ser usada para comunicaciones subsiguientes por la entidad que invocó a ISAKMP. El conjunto básico de atributos de SA que DEBE ser implementado para proporcionar interoperatibilidad entre ISAKMPs están definidos en la Sección 6.

1.4 Requerimientos de Autentificación para ISAKMP

La autentificación fuerte DEBE ser proporcionada en los intercambios ISAKMP. Si no se puede autentificar a la entidad del otro extremo, la SA y el establecimiento de claves de sesión serán dudosos. Sin autentificación no se puede confiar en la identificación de la entidad, lo que hace al control de acceso cuestionable. Mientras que la encriptación (por ejemplo ESP) y la integridad (por ejemplo AH) protegerán comunicaciones subsiguientes de mirones (sniffer) pasivos, sin autentificación es posible que la SA y las claves hayan sido establecidas por otras personas, las cuales realizaron un ataque activo modificando el flujo de datos trasmitidos interfiriendo la comunicación y ahora se está robando toda su información personal.

Un algoritmo de firma digital DEBE ser usado dentro del componente de autentificación de ISAKMP. Sin embargo, ISAKMP no exige un algoritmo para las firmas digitales o Autoridad de Certificación (CA) específico. ISAKMP permite a una entidad iniciar una comunicación indicando que CAs esta utilizando. Después de la selección de una CA, el protocolo proporciona la infraestructura para utilizar el intercambio de autentificación actual. El protocolo proporciona facilidades para la identificación de diferentes autoridades de certificación, tipos de certificados (por ejemplo X.509, PKCS #7, PGP, DNS SIG y registro de claves) y intercambios de certificados determinados.

ISAKMP utiliza firmas digitales, basadas en criptografía de clave pública, para la autentificación. Existen otros sistemas fuertes de autentificación disponible, que se podrían especificar como mecanismos opcionales de autentificación para ISAKMP. Algunos de estos sistemas de autentificación confían en una tercera parte llamada Centro de Distribución de Claves (KDC), para distribuir claves de sesiones secretas. Un ejemplo es Kerberos, donde la tercera parte confiable es el servidor de Kerberos, que guarda las claves secretas de todos sus clientes y servidores dentro de su dominio de red. Un cliente que tiene una clave secreta proporciona autentificación ante servidores.

Las especificaciones de ISAKMP no especifican el protocolo para la comunicación con las Terceras Partes de Confianza (TTP) o los servicios de directorios de certificados. Estos protocolos están definidos por las TTP y los servicios de directorios y están fuera del alcance de este libro.

1.5 Criptografía de Clave Pública

La criptografía de clave pública es un modo más flexible, escalable y eficiente para que los usuarios obtengan secretos y claves compartidas para soportar un gran número de formas para ínter-operar con los usuarios de Internet. Diversos algoritmos de generación de claves, están disponibles para los usuarios (ver el Capítulo 9 [OAKLEY], el Capítulo 10 [IKE], [DOW92] y [ANSI]). Las propiedades de los protocolos de intercambio de claves incluyen: autentificación, simetría, perfect forward secrecy, un método para el establecimiento de claves y la protección posterior del tráfico.

NOTA: Las claves criptográficas pueden proteger información por largos periodos de tiempo. Sin embargo esto se basa en la presunción de que las claves son usadas para la protección de comunicaciones y son destruidas después de haber sido usadas y no son almacenadas por ninguna razón.

1.5.1 Propiedades del Intercambio de Claves

Los dos métodos más comunes en la criptografía de clave pública para el establecimiento de claves son:

El métodos de Trasporte de Claves:
Un ejemplo de trasporte de claves es el uso del algoritmo RSA para encriptar las claves de sesión generadas aleatoriamente (para encriptar comunicaciones subsiguientes) con los receptores de las claves públicas. La clave aleatoriamente encriptada es luego enviada al receptor, que la desencripta utilizando su clave privada. En este punto ambos extremos de la comunicación, tienen la misma clave de sesión, sin embargo esta fue creada a partir de la entrada de información unidireccional. La ventaja del método de trasporte de claves es que tiene menos gasto computacional que método generación de claves.

El métodos de Generación de Claves:
El algoritmo de Diffie-Hellman (vea el Capítulo 5) ilustra la generación de claves utilizando criptografía de clave pública. El algoritmo D-H se inicia con dos usuarios que intercambian información pública. Cada usuario después combina matemáticamente la información pública del otro usuario con su propia información secreta para calcular un valor secreto compartido. Este valor secreto puede ser utilizado como una clave de sesión o como una clave de encriptación de clave para encriptar la clave de sesión generada aleatoriamente. Este método genera una clave de sesión basada en información pública y secreta, compartida por ambos usuarios. La ventaja del algoritmo de Diffie-Hellman es que la clave usada para encriptar mensajes se obtiene de la información compartida por ambos usuarios y la independencia de las claves entre un intercambio de claves y el otro proporciona perfect forward secrecy.

Los intercambios de claves pueden ser autentificados durante el protocolo o después del protocolo ISAKMP. La autentificación del intercambio de claves durante el protocolo se lleva a cabo cuando cada parte proporciona prueba

de que tiene la clave de sesión secreta antes de finalizar el protocolo. La prueba puede ser proporcionada encriptando datos conocidos en la sesión de claves secretas durante el intercambio del protocolo. La autentificación después del protocolo debe ocurrir en comunicaciones subsiguientes. La autentificación durante el protocolo es la más óptima. Por lo tanto las comunicaciones subsiguientes no son iniciadas si la clave de sesión secreta no esta establecida con la parte deseada.

1.5.2 Requisitos para ISAKMP

Un intercambio de claves autentificado debe ser utilizado por ISAKMP. Los usuarios deberían elegir los algoritmos de establecimiento de claves basándose en sus propios requerimientos. ISAKMP no especifica un protocolo de intercambio de claves determinado. Sin embargo, el Capítulo 10 [IKE] describe una propuesta para el uso de [OAKLEY] (ver el Capítulo 9) en conjunto con ISAKMP. Los requerimientos que deben ser evaluados al elegir un algoritmo de establecimiento de claves deben incluir: el método de establecimiento de clave (generación o trasporte), perfect forward secrecy, el costo computacional, depósito de claves y la fuerza de las claves. De acuerdo con los requerimientos del usuario, ISAKMP permite a una entidad iniciar comunicaciones para que indique que intercambio de claves soporta. Después de la selección de un intercambio de claves el protocolo proporciona los mensajes requeridos para el establecimiento de la clave verdadera (también denominada clave de sesión).

1.6 Protección Proporcionada por ISAKMP

1.6.1 Anti-Saturación (Denegación de Servicio)

De los numerosos servicios de seguridad disponibles, la protección contra la denegación de servicio siempre va ha ser uno de los más difíciles de solucionar. Un "cookie" o token anti-saturación (ACT) está destinado a proteger los recursos computacionales para evitar ataques sin malgastar recursos excesivos de CPU, para determinar sus autenticidades. Un intercambio previo con operaciones de claves públicas que consuman mucha CPU pueden frustrar ciertas tentativas de denegación de servicios (por ejemplo, con inundaciones de falsas direcciones IP de origen). La protección absoluta contra la denegación de servicio es imposible pero este token anti-saturación proporciona una técnica para hacerlo más fácil de manejar.

Como se observará en los intercambios mostrados en la Sección 4, los mecanismos de anti-saturación deberían ser usados en conjunto con mecanismos de recolección de información de estado no válida. Un atacante silencioso puede inundar un servidor usando paquetes con direcciones IP falsas y se saturará el servidor. Tales técnicas de administración de memoria agresiva DEBERÍAN ser empleados por los protocolos que utilizan ISAKMP, que no realizan un minucioso examen inicial (solo en la etapa de anti-saturación) como se describe en [Karn].

1.6.2 Secuestro de la Conexión

ISAKMP previene el secuestro de la conexión vinculando la autentificación, el intercambio de claves y el intercambio de SAs. Esta vinculación impide a un atacante completar la autentificación y que luego intervenga y tome la personalidad de una entidad durante los intercambios de claves y SA.

1.6.3 Ataques en la Trayectoria (Man-in-the-Middle Attacks)

La intercepción de atacantes en la trayectoria incluyen: la intercepción, la inserción, la cancelación y la modificación de mensajes, mensajes reflejados por atrás del emisor reeditando mensajes viejos y redireccionando mensajes. Las características de ISAKMP evitan que estos tipos de ataques sean exitosos. La vinculación de intercambios ISAKMP previene la inserción de mensajes en el intercambio de los protocolos. La máquina de estado del protocolo ISAKMP se define como un eliminador de mensajes, que no permite que SA parciales sean creadas, la máquina de estado eliminará todos los estados y volverá a la inactividad. La máquina de estado también evita que la reflexión de un mensaje cause daños. Los requerimientos para una nueva cookie con contenido dinámico para cada nueva SA establecida previene de ataques que involucren el reenvío de mensajes viejos. El requerimiento de autentificación fuerte de ISAKMP previene que una SA sea establecida con cualquier otra parte que no sea la deseada. Los mensajes pueden ser redireccionados a un destino diferente o modificados pero esto será detectado y una SA no será establecida. La especificación de ISAKMP define donde a ocurrido un procesamiento anormal y recomienda notificar a la parte apropiada de esta anormalidad.

1.7 Comunicaciones Multicast

Se espera que las comunicaciones multicast requieran de los mismos servicios de seguridad que las comunicaciones unicast y pueden introducir la necesidad de servicios de seguridad adicionales. Esta versión de ISAKMP no soporta la distribución de claves multicast. Para una introducción relacionada con la seguridad multicast, consulte [HM97] y [RFC-2093]. Los temas de multicast también son debatidos en [RFC-1949] y en [BC].

2. **Conceptos y Terminología**

2.1 Terminología de ISAKMP

Asociación de seguridad (SA)
: Una Asociación de seguridad es un conjunto de parámetros específicos del protocolo de seguridad que definen completamente los servicios y mecanismos necesarios para proteger el tráfico en ese lugar del protocolo de seguridad. Estos parámetros pueden incluir identificadores de algoritmo, modos, claves criptográficas, etc. La SA hace referencia a su protocolo de seguridad asociado (por ejemplo "SA ISAKMP", "SA ESP", "SA TLS").

SA ISAKMP
: Una SA usada por los servidores ISAKMP para proteger su propio tráfico. Las Secciones 2.3 y 2.4 proporcionan más detalles acerca de las SAs ISAKMP.

Índice de Parámetros de Seguridad (SPI)
: Un identificador para una Asociación de Seguridad, relativo a algún protocolo de seguridad. Cada protocolo de seguridad tiene su propio "espacio-SPI". Un par (protocolo de seguridad, SPI) pueden identificar unívocamente a una SA. La unívocariedad (exclusividad) de la SPI es dependiente de la implementación, pero puede estar basada en sistemas, en protocolos, u otras opciones. Dependiendo del DOI, información adicional (por ejemplo, las direcciones de los host) puede ser necesarias para identificar a una SA. El DOI también determinará cuales SPIs (es decir, los SPIs del iniciador o del respondedor) son enviados durante la comunicación.

Perfect Forward Secrecy
Según lo descripto en [DOW92], un protocolo de intercambio de claves autentificado, proporciona perfect forward secrecy si la divulgación del material clave por largos periodos de tiempo no compromete la confidencialidad del intercambio de claves de las comunicaciones previas. Las características del perfect forward secrecy no se aplican al intercambio de claves cuando este está desprovisto de autentificación.

Dominio de Interpretación (DOI)
Define los formatos de cargas, tipos de intercambio y convenciones para nombrar información relevante a la seguridad, tales como políticas de seguridad, algoritmos criptográficos y modos. Un identificador de DOI es usado para interpretar las cargas de ISAKMP. Un sistema DEBERÍA soportar múltiples Dominios de Interpretación simultáneamente. El concepto de DOI se basa en trabajos previos del Grupo de Trabajo de TSIG CIPSO, pero se extiende más allá de la interpretación de etiquetas de seguridad para incluir el nombramiento y la interpretación de los servicios de seguridad. Un DOI define:

- Una "situación": el conjunto de información que será usado para determinar los servicios de seguridad requeridos.
- El conjunto de políticas de seguridad que deben o podrían ser soportados.
- La sintaxis para la especificación de los propósitos de los servicios de seguridad sugeridos.
- Un esquema para nombrar información relativa a la seguridad, incluyendo algoritmos de encriptación, algoritmos de intercambio de claves, atributos de política de seguridad y autoridades de certificación.
- Los formatos específicos de los contenidos de las diversas cargas.
- Tipos de intercambio adicionales, si son requeridos.

Las reglas de Seguridad del DOI de IP de la IETF se presentan en el Capítulo 8. Las especificaciones de las reglas para DOI personalizados se comentan en la Sección 7.

A parte de los términos aquí presentados se recomienda que el lector lea el Glosario suministrado como Apéndice en este libro, sobretodo los términos: Protocolo de Seguridad, Conjunto de Protección, Situación, Proposición, Tipos de Intercambios; entre otros para lograr una adecuada comprensión del presente capítulo.

2.2 Ubicación de ISAKMP

La Figura 1 muestra la ubicación de ISAKMP dentro de un contexto de un sistema en la arquitectura de red. Una parte importante de la negociación de los servicios de red es considerar a la "PILA" entera de las SAs como una unidad. Esto es denominado comúnmente como "conjunto de protección".

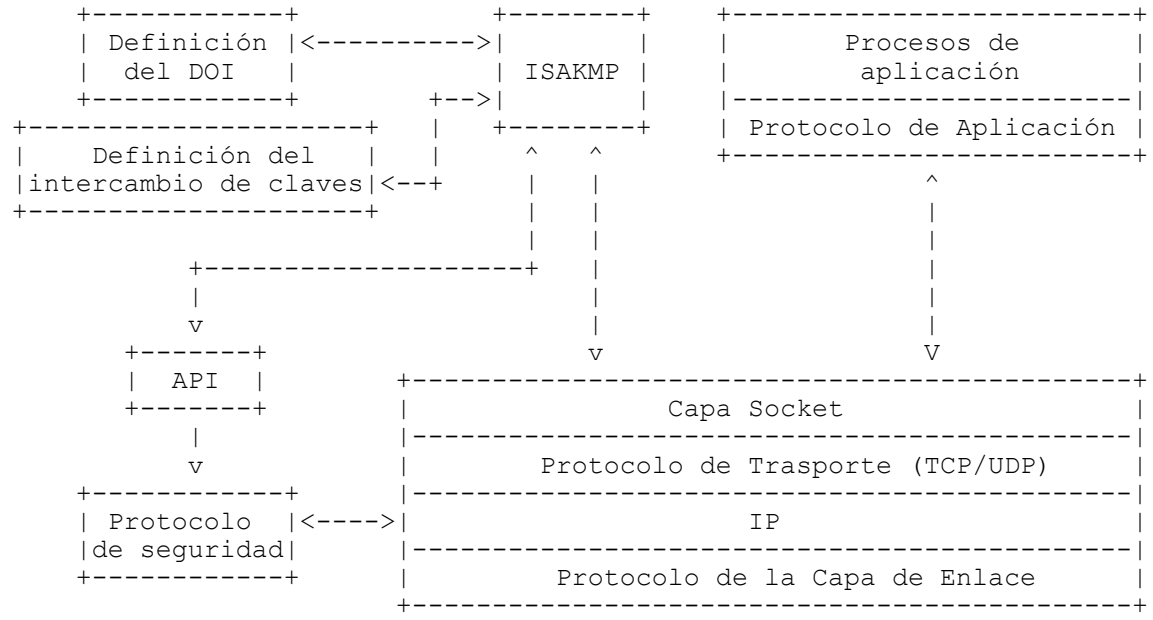


Figura 1: Ubicación de ISAKMP

2.3 Fases de la Negociación

ISAKMP ofrece dos "fases" para la negociación. En la Fase 1, dos entidades concuerdan en como proteger futuras negociaciones del tráfico entre ellas mismas, estableciendo una SA ISAKMP. Esta SA ISAKMP es luego usada para proteger las negociaciones requeridas por las SA de los Protocolos. Dos entidades pueden negociar (si están activos) múltiples SA ISAKMP.

La Fase 2 de la negociación es usada para establecer la SA para otros protocolos de seguridad. Esta segunda fase puede ser usada para establecer múltiples SA. Las SA establecidas por ISAKMP durante esta fase pueden ser usadas por un protocolo de seguridad para proteger los intercambios de datos/mensajes.

Mientras que este método de dos fases tiene un costo elevado para la inicialización en la mayoría de los escenarios simples, hay varias razones por la que este método es beneficioso en la mayoría de los casos.

Primero, las entidades (por ejemplo servidores ISAKMP) pueden amortizar el costo de la Fase 1 a través de varias negociaciones en la Fase 2. Esto permite que múltiples SAs estén relacionadas entre usuarios por un cierto lapso de tiempo sin tener que iniciar cada una de las comunicaciones.

Segundo, los servicios de seguridad negociados durante la Fase 1 proporcionan propiedades de seguridad para la Fase 2. Por ejemplo, después de la negociación de la Fase 1, la encriptación proporcionada por la SA ISAKMP puede proporcionar protección de identidad (potencialmente permitiendo el uso de intercambios más simples) en la Fase 2. Por otra parte, si un canal establecido durante la Fase 1 no es adecuado para proteger las identidades, la Fase 2 luego deberá negociar los mecanismos de seguridad adecuados.

Tercero, tener una SA ISAKMP reduce considerablemente el costo de actividad de administración externo a ISAKMP brindando una "trayectoria confiable" para una SA ISAKMP, las entidades (por ejemplo servidores de ISAKMP) tendrían que pasar por una reautorización completa de cada error de notificación o la cancelación de una SA.

La negociación llevada a cabo en cada fase es realizada usando intercambios ISAKMP definidos (ver Sección 4) o intercambios definíos en un intercambio de claves dentro de un DOI.

Note que los servicios de seguridad se pueden aplicar de manera diferente en cada una de las fases de la negociación. Por ejemplo, diferentes partes son autentificadas durante cada una de las fases de la negociación. Durante la Fase 1, las partes que son autentificadas pueden ser servidores ISAKMP o host, mientras que en la Fase 2 usuarios o programas de nivel de aplicación son autentificados.

2.4 Identificar SA

A pesar de que existen canales seguros de bootstrapping entre sistemas, ISAKMP no puede asumir la existencia de servicios de seguridad y debe proporcionar algunas protecciones para sí mismo. Por lo tanto, ISAKMP considera una SA ISAKMP diferente a las de otros tipos y administra las SA ISAKMP para sí mismo, con su propio espacio de nombres. ISAKMP usa dos campos de cookies en la cabecera de ISAKMP para identificar SA ISAKMP. El campo Identificador (ID) del Mensaje en la Cabecera de ISAKMP y el campo SPI en la carga de la Propuesta son usados durante el establecimiento de la SA para identificar la SA de otros protocolos de seguridad. La interpretación de estos 4 campos es dependiente de la operación que se lleve a cabo.

La tabla siguiente muestra la presencia o ausencia de diversos campos durante el establecimiento de la SA. Los siguientes campos son necesarios para las diversas operaciones asociadas con el establecimiento de la SA: cookies en la cabecera de ISAKMP, el campo Identificador (ID) del Mensaje en la cabecera de ISAKMP, y el campo SPI en la carga de la Propuesta. Una 'X' en la columna significa que el valor debe estar presente. Un cero (0) significa que el valor no esta presente. Una 'NA' en la columna significa que el valor en la columna no es aplicable en esa operación.

N°	Operación	Cookie del Iniciador	Cookie del Respondedor	ID del Mensaje	SPI
(1)	Inicio de la negociación SA ISAKMP	X	0	0	0
(2)	El respondedor de la negociación SA ISAKMP	X	X	0	0
(3)	Iniciador, negociación de la otra SA	X	X	X	X
(4)	El respondedor, negociación de otra SA	X	X	X	X
(5)	Otros (KE, ID, etc.)	X	X	X/0	NA
(6)	Protocolo de Seguridad (AH, ESP)	NA	NA	NA	X

En la primera línea (1) de la tabla, el iniciador incluye el campo del Cookie del Iniciador en la cabecera de ISAKMP usando los procedimientos descriptos en las Secciones 2.5.3 y 3.1.

En la segunda línea (2) de la tabla, el respondedor incluye los campos de las cookies del iniciador y del respondedor en la cabecera de ISAKMP usando los procedimientos descriptos en las Secciones 2.5.3 y 3.1. Mensajes adicionales pueden ser intercambiados entre usuarios ISAKMP, dependiendo

del primer tipo de intercambio ISAKMP utilizado durante la Fase 1 de la negociación. Una vez que la Fase 1 del intercambio a finalizado, las cookies del iniciador y del respondedor son incluidos en la cabecera de ISAKMP para todas las comunicaciones subsiguientes entre los usuarios de ISAKMP.

Durante la Fase 1 de la negociación, la cookie del iniciador y del respondedor determinan la SA ISAKMP. Por lo tanto el campo SPI en la carga de la Propuesta es redundante y PUEDE ser cero (0) o PUEDE contener la identidad del cookie del transmisor.

En la tercera línea (3) de la tabla, el iniciador asocia el ID del Mensaje con los Protocolos contenidos en la Propuesta SA. Este ID de Mensaje y los SPI del iniciador asociados con cada protocolo en la Propuesta son enviados al respondedor. Los SPIs serán utilizados por los protocolos de seguridad una vez que la Fase 2 de la negociación este terminada.

En la cuarta línea (4) de la tabla, el que responde incluye el mismo ID de Mensaje y los mismos SPIs que están asociados con cada protocolo en la Propuesta aceptada. Esta información se devuelve al iniciador.

En la quinta línea (5) de la tabla, el iniciador y el que responde usan el campo de ID del Mensaje en la cabecera de ISAKMP para mantener el camino de la negociación del protocolo en proceso. Esto es solo aplicable para el intercambio de la Fase 2 y el valor DEBE ser cero para el intercambio de la Fase 1, por que las cookies combinadas identifican la SA ISAKMP. El campo SPI en la carga de la Propuesta no es aplicable por que la carga de la Propuesta es solamente usada durante los intercambios de negociación de mensajes de SA (pasos 3 y 4).

En la secta línea (6) de la tabla, la Fase 2 de la negociación es terminada. Los protocolos de seguridad usan los SPIs para determinar que mecanismos y servicios de seguridad aplicar a la comunicación entre ellos. El valor del SPI mostrado en la secta línea no es el campo SPI de la carga de la Propuesta sino que es el campo del SPI contenido dentro de la cabecera del protocolo de seguridad.

Durante el establecimiento de la SA, una SPI debe ser generada. ISAKMP esta diseñado para tratar con SPIs de diferentes tamaños, esto se logra usando campos con tamaños de SPIs dentro de la carga de la Propuesta durante el establecimiento de la SA. El manejo de los SPIs será detallado por la especificación de DOI (por ejemplo el Capítulo 8 describe de DOI de IPsec).

Cuando una SA es inicialmente establecida, uno de los extremos asume el rol de iniciador y el otro el rol de respondedor. Una vez que la SA esta establecida, ambos (el iniciador original y el respondedor original) pueden iniciar la Fase 2 de la negociación como entidades pares. Por ende, las SA ISAKMP son bidireccionales por naturaleza.

Además, ISAKMP permite al iniciador y al respondedor tener el mismo control durante el proceso de negociación. Mientras que ISAKMP es configurada para permitir la negociación de una SA que incluye múltiples propuestas, el iniciador puede mantener cierto control haciendo solamente una propuesta de acuerdo con la política de seguridad local del iniciador. Una vez que el iniciador envía una propuesta que contiene más de una opción (que son enviadas en orden decreciente de preferencia), el iniciador le pasa el control al respondedor. Una vez que el respondedor a tomado el control del establecimiento de la SA, este puede hacer que sus políticas tomen precedencia sobre las del iniciador dentro de un contexto de opciones

múltiples ofrecidas por el iniciador. Esto se logra seleccionando la mejor opción para la política de seguridad local del respondedor y devolviendo esta selección al iniciador.

2.5 Temas Diversos

2.5.1 Protocolo de trasporte

ISAKMP puede ser implementado sobre cualquier protocolo de trasporte o sobre el mismo protocolo IP. Las implementaciones DEBEN incluir la capacidad de enviar y recibir tráfico ISAKMP utilizando el Protocolo de Datagrama de Usuario (UDP) sobre el puerto 500. El puerto UDP 500 ha sido asignado para el tráfico de ISAKMP por la Autoridad de Asignación de Números en Internet (IANA). Implementaciones adicionales PUEDEN soportar otros protocolos de trasporte o enviar y recibir trafico ISAKMP sobre el mismo protocolo IP.

2.5.2 Campos Reservados

La existencia de campos RESERVADOS dentro de la carga ISAKMP son usados estrictamente para preservar el alineamiento de los bytes. Todos los campos RESERVADOS en el protocolo ISAKMP DEBEN contener el valor cero (0) cuando un paquete es enviado. El receptor DEBERÍA corroborar que los campos RESERVADOS contengan el valor cero (0) y descartar el paquete si otros valores son encontrados

2.5.3 Creación de Token ("Cookies") Anti-Saturación

Los detalles de la generación de cookies dependen de la implementación, pero DEBEN satisfacer estos requerimientos básicos [Karn]:

1. La cookie debe depender de las partes específicas. Esto evita que un atacante obtenga una cookie usando una dirección IP real y un puerto UDP, y luego use esto para saturar a la victima con peticiones de Diffie-Hellman a partir de direcciones IP o puertos elegidos aleatoriamente.

2. No debe ser posible que cualquier persona, con excepción de la entidad, emita la generación de cookies que serán aceptadas por esa entidad. Esto implica que la entidad emisora debe utilizar información local secreta en la generación y en las subsiguientes verificaciones de cookies. No debe ser posible deducir esta información secreta de ninguna cookie en particular.

3. La función de generación de cookies debe ser rápida para impedir ataques que intentan sabotear los recursos de la CPU.

El método sugerido por Karn's para la creación de cookies es realizar un hash (por ejemplo MD5) sobre la dirección de origen y destino IP, la dirección de los puertos de origen y de destino UDP y un valor secreto aleatorio generado localmente. ISAKMP requiere que la cookie sea única para cada SA establecida con el propósito de ayudar a prevenir ataques de reenvío, por lo tanto, la fecha y el tiempo DEBEN ser agregados a la información condensada (hashed). Las cookies generadas son colocadas en los campos de las cookies del Iniciador y del Respondedor de la Cabecera ISAKMP (como se describe en la Sección 3.1). Estos campos tienen una longitud de 8 octetos, por ende se requiere que la cookie generada tenga 8 octetos. Los mensajes de Notificación y Cancelación (ver las Secciones 3.14, 3.15 y 4.8) unidireccionalmente trasmitidos y que están bajo la protección de una SA

ISAKMP existente, no requerirán la generación de una nueva cookie. Una excepción a esto es la transmisión de un Mensaje de Notificación durante el intercambio de la Fase 1, antes de terminar el establecimiento de una SA. Las Secciones 3.14 y 4.8 proporcionan detalles adicionales.

3. **Cargas de ISAKMP**

ISAKMP define varios tipos de cargas, que son usadas para transmitir información según los datos de la SA, o los datos del intercambio de claves, dentro de las formas definidas en el DOI. Una carga consiste en una cabecera de carga genérica y en octetos encadenados que están ocultos para ISAKMP. ISAKMP usa funcionalidades específicas del DOI para sintetizar e interpretar esas cargas. Múltiples cargas pueden ser enviadas en un único mensaje de ISAKMP.

Las cargas de ISAKMP proporcionan bloques modulares para la construcción de mensajes ISAKMP. La presencia y el ordenamiento de las cargas ISAKMP se define y depende del Campo Tipo de Intercambio ubicado en la Cabecera de ISAKMP (ver figura 2). Los tipos de carga de ISAKMP son analizados desde la Sección 3.4 hasta la Sección 3.15. Las descripciones de las cargas de ISAKMP, y los mensajes e intercambios se muestran usando el ordenamiento de octetos de red.

3.1 Formato de la Cabecera de ISAKMP

El mensaje de ISAKMP tiene un formato de cabecera fijo, como muestra la Figura 2, seguido por un número de cargas variables. Una cabecera fija simplifica el procesamiento, proporcionando beneficios al procesamiento del análisis de software del Protocolo que es menos complejo y mas fácil de implementar. La cabecera fija contiene la información requerida por el protocolo para mantener el estado, procesar las cargas y posiblemente prevenir la denegación de servicio o ataques de reenvío. La Figura 2 muestra el formato de la Cabecera ISAKMP (donde MjVer = Versión mayor; MnVer = Versión menor).

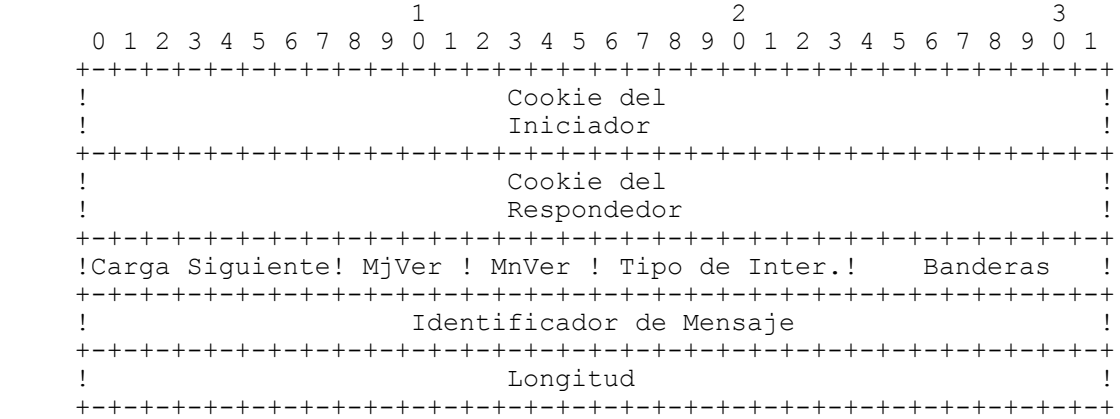


Figura 2: Formato de la Cabecera ISAKMP

Los campos de la Cabecera ISAKMP se definen de la siguiente forma:

- Cookie del Iniciador (8 octetos): Cookie de la entidad que inicia el establecimiento, modifica, o cancela la SA.

- Cookie del Respondedor (8 octetos): Cookie de la entidad que responde al requerimiento del establecimiento de una SA, o cancelación de la SA.

- Carga siguiente (1 octeto): indica el tipo de carga en el primer mensaje el formato de cada carga es definido desde la Sección 3.4 hasta la Sección 3.16. El procesamiento para las cargas se define en la Sección 5.

Tipos de Carga Siguiente	Notación	Valor
Ninguna		0
Carga de Asociación de Seguridad	SA	1
Carga de la Propuesta	P	2
Carga de Transformación	T	3
Carga de Intercambio de Claves	KE	4
Carga de Identificación	ID	5
Carga de Certificado	CERT	6
Carga de Solicitud de Certificado	CR	7
Carga de Hash	HASH	8
Carga de Firma	SIG	9
Carga de Nonce	NONCE	10
Carga de Notificación	N	11
Carga de Cancelación	D	12
Carga de Identificación del Vendedor	VID	13
RESERVADO		14-127
Uso Privado		128-255

- Versión Mayor (4 bits): indica la versión mayor del protocolo ISAKMP en uso. Las implementaciones basadas en esta versión de Draft-Internet de ISAKMP DEBEN fijar la versión mayor en uno. Las implementación basadas en versiones previas al Draft-Internet de ISAKMP deben fijar la versión mayor en 0. Las implementaciones nunca DEBERÍAN aceptar paquetes con un número de versión mayor que estos.

- Versión Menor (4 bits): indica la versión menor del protocolo ISAKMP en uso. Las implementaciones basadas en los Draft-Internet de ISAKMP DEBEN fijar la versión menor en cero. Las implementaciones basadas en versiones previas a los Draft-Internet de ISAKMP deben fijar la versión menor en 1. Las implementaciones nunca DEBERÍAN aceptar paquetes con un número de versión superior a estos, dado que los números de la versión mayor son idénticos.

- Tipo de intercambio (1 octeto): Indica el tipo de intercambio usado. Esto indica los ordenamientos de los mensajes y la carga en los intercambios de ISAKMP.

Tipos de Intercambios	Valor
Ninguno	0
Base	1
Protección de Identidad	2
Solamente Autentificación	3
Agresivo	4
Informativo	5
Uso futuro de ISAKMP	6-31
Uso especifico del DOI	32-239
Uso privado	240-255

- Banderas (Flags) (1 octeto): Indica las opciones específicas que se fijan para los intercambios ISAKMP. Las banderas enumeradas debajo son específicas del campo de Banderas comenzando con el bit menos significativo, es decir el bit de Encriptación es el que se encuentra en la posición cero en el campo de Banderas, el bit de Commit está en la posición 1 en el campo de Banderas, y el bit de Solo Autentificación en la posición 2 del campo de Banderas. Los bits restantes del campo de Banderas se BEBEN fijar en cero antes de la transmisión.

  - Bit de Encriptación (1 bit): si está en 1, todas las cargas que siguen a la cabecera son encriptadas usando algoritmos de encriptación, identificados por la SA ISAKMP. El identificador de la SA ISAKMP es la combinación de la cookie del iniciador y del respondedor. Se RECOMIENDA que la encriptación de las comunicaciones se realicen lo antes posible entre los usuarios. Para todos los intercambios ISAKMP descriptos en la Sección 4.1, la encriptación DEBERÍA comenzar después de que ambas partes hallan intercambiado las cargas de Intercambio de Claves. Si el Bit de encriptación está en cero (0) las cargas no son encriptadas.

  - Bit de Commit (1 bit): Este bit es usado para señalar la sincronización del intercambio de claves. Es usado para asegurar que el material encriptado no se reciba antes del término del establecimiento de la SA. EL bit de Commit puede ser fijado (en cualquier momento) por cualquiera de las partes que participan en el establecimiento de la SA, y puede ser usado durante las dos fases del establecimiento de la SA ISAKMP. Sin embargo, el valor DEBE ser puesto en cero (resetiado) después de la Fase 1 de la negociación. Si está en (1), la entidad que no lo haya fijado DEBE esperar un Intercambio Informativo conteniendo una carga de Notificación (con el Mensaje de Notificación CONECTADO) de la entidad que fijó el Bit de Commit. En este caso, el campo Identificador de Mensaje del Intercambio Informativo DEBE contener el Identificador de Mensaje de la ISAKMP original de la Fase 2 de negociación de la SA. Esto se hace para asegurar que el Intercambio Informativo con el Mensaje de Notificación CONECTADO pueda ser asociado con la correcta Fase 2 de la SA. La recepción y procesamiento del Intercambio Informativo indica que el establecimiento de la SA fue exitoso y que cualquier entidad puede ahora proceder con la comunicación del tráfico encriptado. En sincronizaciones adicionales del intercambio de claves, el Bit de Commit puede ser usado para proteger contra la pérdida de trasmisiones en redes no confiables y para la defensa de múltiples retrasmisiones.

    NOTA: Es siempre posible que el mensaje final de un intercambio se pueda perder. En este caso, la entidad que se prepara para recibir el mensaje final de un intercambio recibiría el mensaje de la negociación de la Fase 2 de la SA seguido de un intercambio de la Fase 1 o el tráfico encriptado seguido de un intercambio de la Fase 2. El manejo de esta situación no está estandarizado, pero se propone las siguientes posibilidades. Si la entidad que espera el Intercambio Informativo puede verificar el mensaje recibido (es decir, el mensaje de negociación de la Fase 2 de la SA o el tráfico encriptado), entonces PUEDE considerarse que la SA fue establecida y continuar con el procesamiento. Otra opción es

retrasmitir el último mensaje ISAKMP para forzar a la otra entidad a retrasmitir el mensaje final. Esto sugiere que las implementaciones pueden considerar la retención del último mensaje (localmente) hasta que estén seguras de que la SA está establecida.

- Bit de Solo Autentificación (1 bit): este bit esta diseñado para ser usado con el Intercambio Informativo de una carga de Notificación y permitirá la transmisión de información con comprobación de integridad, pero no de encriptación (por ejemplo en modo de emergencia). La Sección 4.8 indica que un Intercambio Informativo de la Fase 2 DEBE ser enviado bajo la protección de una SA ISAKMP. Esto es solo una excepción a esa política. Si el bit de Solo Autentificación está en (1), solamente los servicios de autentificación de seguridad serán aplicados a toda la carga de Notificación de Intercambio Informativo y la carga no será encriptada.

- Identificador (ID) de Mensaje (4 octetos): El Identificador de Mensaje solamente se usa para identificar el protocolo durante las negociaciones de la Fase 2. Este valor es generado aleatoriamente por el iniciador de la Fase 2. En el caso de establecimientos simultáneos de SA (es decir colisiones), el valor de este campo será probablemente diferente porque son generados independientemente y, así, dos SAs seguirán con el establecimiento. Sin embargo es improbable de que existan establecimientos simultáneos. Durante las negociaciones de la Fase 1, el valor DEBE ser cero.

- Longitud (4 octetos): Longitud total del mensaje (cabecera más cargas) en octetos. La encriptación puede expandir el tamaño de un mensaje ISAKMP.

3.2 Cabecera de Carga Genérica

Cada carga de ISAKMP definidas desde la Sección 3.4 hasta la Sección 3.16 comienzan con una cabecera de carga genérica, como la que se muestra en la Figura 3, la cual proporciona una capacidad de encadenamiento de cargas y claramente define los límites de una carga.

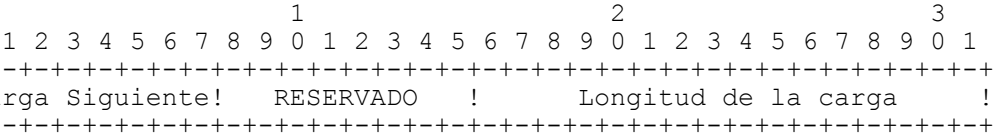


Figura 3: Formato de la Cabecera de Carga Genérica

Los campos de la cabecera de la carga genérica son definidos de la siguiente forma:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero. Este campo proporciona la capacidad de "encadenamiento".

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.

3.3 Atributos de los Datos

Hay varios casos dentro de ISAKMP donde es necesario representar los Atributos de los Datos. Un ejemplo de esto es los Atributos de la SA contenidas en la carga de Transformación (descriptos en la Sección 3.6). Estos Atributos de los Datos no son cargas de ISAKMP, pero están contenidas dentro de las cargas de ISAKMP. El formato de los Atributos de los Datos proporciona flexibilidad para la representación de diferentes tipos de información. Pueden existir múltiples Atributos de los Datos dentro de una carga. La longitud de los atributos de los datos será de 4 octetos o estará definida por el campo Longitud de los Atributos. Esto es realizado usando el bit de Formato de los Atributo descriptos debajo. La información específica acerca de los atributos para cada dominio será descripta en un documento de DOI, por ejemplo, el Capítulo 8 define el DOI IPsec.

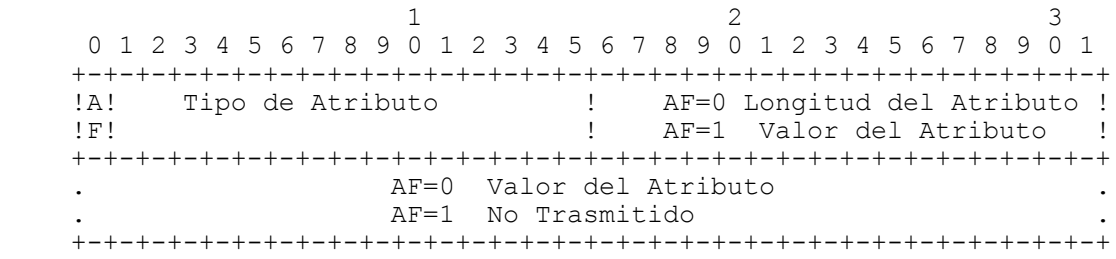


Figura 4: Formato de los Atributo de los Datos

Los campos de los Atributos de los Datos se definen de la siguiente forma:

- Formato de Atributo (AF): es el bits más significativo, indica si los atributos de los datos siguen el formato de Tipo/Longitud/valor (TLV) o uno más corto que sería, Tipo/valor (TV).

Si el AF es cero (0), entonces los atributos de los datos tienen el formato Tipo/Longitud/valor (Variable).

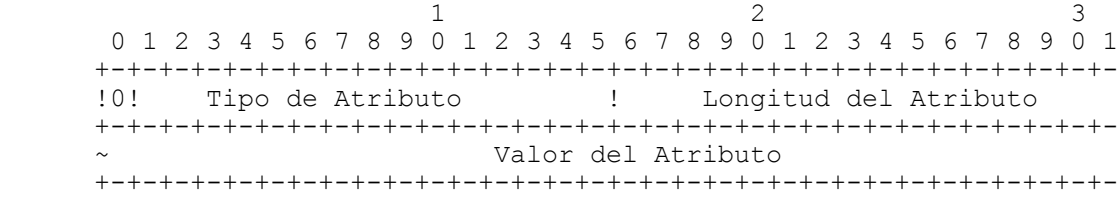


Figura 4a: Formato de los Atributo de los Datos TLV

Si el bit AF es uno (1), entonces los Atributos de los Datos tienen el formato Tipo/Valor (Basico).

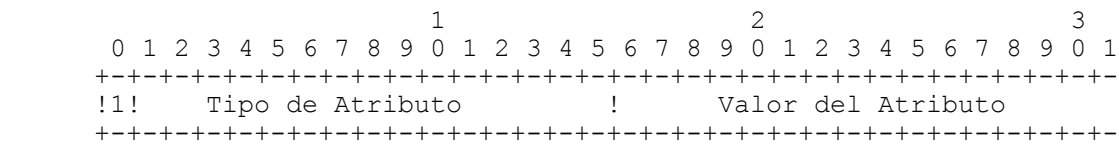


Figura 4b: Formato de los Atributo de los Datos TV

- Tipo de Atributo (2 octetos): identificador único para cada tipo de

Atributo. Estos atributos se definen como parte de la información especifica del DOI.

- Longitud del Atributo (2 octetos): La longitud en octetos del Valor del Atributo. Cuando el bit AF está en uno (Figura 4b), el Valor del Atributo es de solamente de 2 octetos y el campo Longitud del Atributo no está presente.

- Valor del Atributo (longitud variable): Valor del Atributo asociado con el Tipo de Atributo especifico del DOI. Si el bit AF esta en cero (Figura 4a), este campo tiene una longitud variable determinada por el campo de Longitud del Atributo. Si el bit AF esta en uno (Figura 4b), el Valor del Atributo tiene una longitud de 2 octetos.

3.4 Carga SA

La carga SA es usada para negociar los atributos de seguridad, para indicar el Dominio de Interpretación (DOI) y la situación (el conjunto de información que será utilizado para determinar los servicios de seguridad requeridos) bajo la cual se esta llevando a cabo. La Figura 5 muestra el formato de la carga SA.

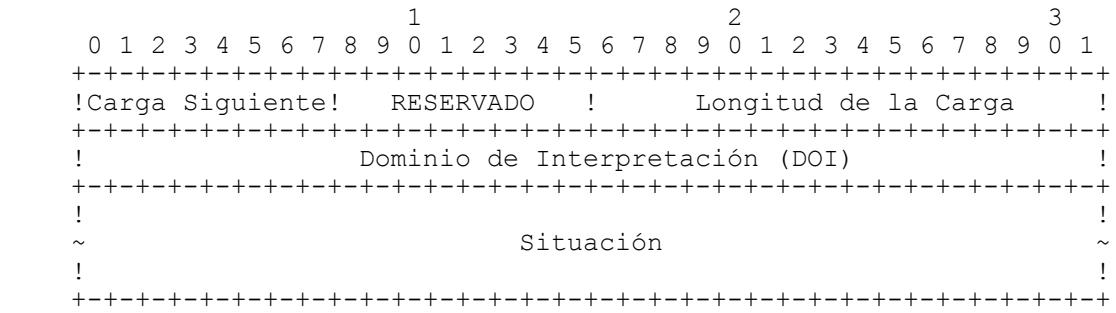


Figura 5: Formato de la carga SA

Los campos de la carga SA se definen de la siguiente forma:

- Carga Siguiente (1 octeto): identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje este campo contendrá el valor cero (0). Este campo NO DEBE contener los valores de las cargas de la Propuesta o Transformación ya que estas son consideradas parte de la negociación de la SA. Por ejemplo, este campo contendría el valor "10" (carga Nonce), en el primer mensaje de un Intercambio Base (ver Sección 4.4) y el valor "0" en el primer mensaje en un Intercambio de Protección de Identidad (ver Sección 4.5).

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): Longitud en octetos de toda la carga SA, incluyendo la carga SA, todas las cargas de la Propuesta y todas las cargas de Trasformación asociadas con la SA propuesta.

- Dominio de Interpretación (4 octetos): Identifica el DOI (como se describe en la Sección 2.1) bajo el cual la negociación se esta llevando a cabo. El DOI es un número entero sin signo de 32 bits. Un valor de DOI de cero (0) durante el intercambio de la Fase 1 específica una SA ISAKMP genérica la cual puede ser usada por

cualquier protocolo durante el intercambio de la Fase 2. Los Atributos SA necesarios están definidos en la Sección 6.4. Un valor de DOI de 1 es asignado al DOI IPsec (vea el Capítulo 8). Todos los otros valores de DOI están reservados por la IANA para usos futuros. Otros DOI pueden ser definidos usando la descripción de la Sección 7. Este campo DEBE estar presente en la carga SA.

- Situación (longitud variable): Un campo específico del DOI que identifica la situación bajo la cual la negociación es llevada a cabo. La Situación es usada para tomar las decisiones de política concerniente a los atributos de seguridad que están siendo negociadas. Las especificaciones para la Situación del DOI de Seguridad IP de la IETF se definen en el Capítulo 8. Este campo DEBE estar presente en la carga SA.

El tipo de carga para la Carga SA es uno (1)

3.5 Carga de la Propuesta

La Carga de la Propuesta contiene información usada durante la negociación de la SA. La propuesta consiste en mecanismos de seguridad, o trasformaciones, que serán usados para asegurar el canal de comunicaciones. La Figura 6 muestra el formato de la Carga de la Propuesta. Una descripción de su uso puede encontrarse en la Sección 4.2.

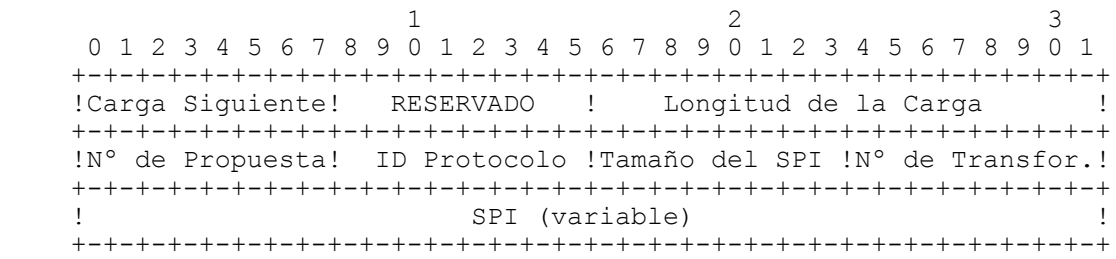


Figura 6: Formato de la Carga de la Propuesta

Los campos de la Carga de la Propuesta se definen de la siguiente forma:

- Carga siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Este campo DEBE contener solamente el valor 2 o cero. Si hay Cargas de la Propuesta adicionales en el mensaje, este campo contendrá el valor 2. Si la carga de la Propuesta actual es la última dentro de la propuesta de la SA, este campo tendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): Longitud en octetos de toda la carga de la Propuesta, incluyendo la cabecera de carga genérica, la carga de la Propuesta, y todas las cargas de Transformación asociadas con esta propuesta. En el caso de que existan múltiples propuestas con el mismo número de propuesta (ver la Sección 4.2), el campo Longitud de la Carga solamente se aplica a la carga de la Propuesta actual y no a todas.

- Número de Propuesta (1 octeto): Identificador del número de Propuesta para la carga actual. Una descripción del uso de este campo se encuentra en la Sección 4.2.

- Identificador de Protocolo (1 octeto): Especifica el Identificador de Protocolo para la negociación actual. Ejemplos de esto incluyen ESP IPSEC, AH IPSEC, OSPF, TLS, etc.

- Tamaño del SPI (1 octeto): La longitud en octetos del SPI como es definido por el Identificador de Protocolo. En el caso de ISAKMP, el par de cookies del Iniciador y del Respondedor de la Cabecera de ISAKMP es el SPI de ISAKMP, por lo tanto, el Tamaño del SPI es irrelevante y PUEDE variar desde 0 a 16 octetos. Si el Tamaño del SPI no es cero, el contenido del campo de SPI DEBE ser ignorado. Si el Tamaño del SPI no es múltiplo de 4 octetos tendrá algún tipo de incidencia en el campo del SPI y de la alineación de todas las cargas en el mensaje. El DOI establecerá el tamaño del SPI para otros protocolos.

- Número de Trasformaciones (1 octeto): Especifica el número de trasformaciones de la Propuesta. Cada uno de estos está contenido en una carga de Transformación.

- SPI (variable): El SPI de la entidad emisora. En el caso de que el Tamaño del SPI no sea múltiplo de 4 octetos, no habrá relleno aplicable a la carga, sin embargo, este puede ser aplicado al final del mensaje.

El tipo de carga para la Carga de la Propuesta es dos (2)

3.6 Carga de Transformación

La Carga de Transformación contiene información usada durante la negociación de la SA. La Carga de Transformación consiste en un mecanismo de seguridad específico, o trasformaciones, con el objetivo de asegurar el canal de comunicación. La carga de Transformación también contiene los atributos SA asociados con la transformación específica. Estos atributos SA están especificados en el DOI. La Figura 7 muestra el formato de la Carga de Transformación. Una descripción de su uso puede ser encontrado en la Sección 4.2.

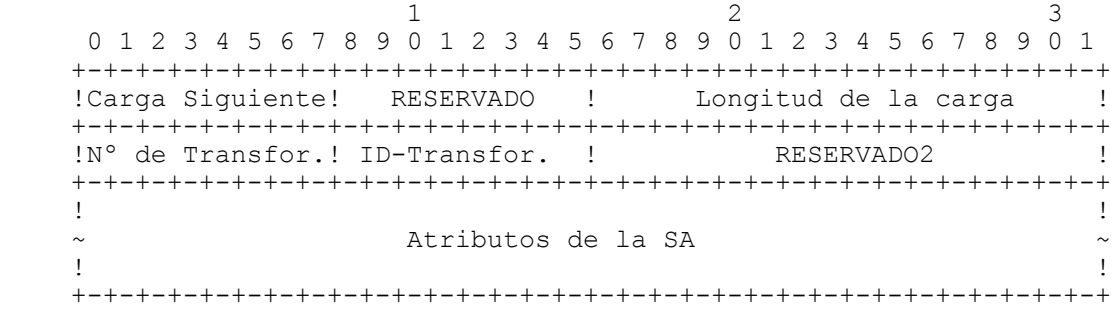


Figura 7: Formato de la Carga de Transformación

Los campos de la Carga de Transformación se definen de la siguiente forma:

- Carga siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Este campo solamente DEBE contener el valor 3 o cero. Si hay cargas de Transformación adicionales en la propuesta, este campo contendrá el valor 3. Si la carga de Transformación actual es la última dentro de la propuesta, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la carga (2 octetos): La longitud en octetos de la presente carga, incluyendo la cabecera de carga genérica, valores de Transformación, y todos los Atributos SA.

- Número de transformación (1 octeto): Identifica el número de Transformación de la presente carga. Si hay más de una transformación propuesta para un protocolo específico, dentro de la carga de transformación, cada carga de Transformación tiene un único Número de Transformación. Una descripción del uso de este campo se encuentra en la Sección 4.2.

- Identificador de Transformación (1 octeto): Especifica el identificador de Transformación para el protocolo dentro de la propuesta actual. Estas trasformaciones están definidas por el DOI y dependen del protocolo que se está negociando.

- RESERVADO2 (1 octeto): No utilizado, debe contener ceros.

- Atributos SA (longitud variable): Este campo contiene los atributos de la SA como están definidos para la trasformación dada en el campo Identificador de Transformación. Los Atributos SA se BEBERÍAN representar usando el formato de los Atributos de los Datos descriptos en la Sección 3.3. Si los atributos de la SA no están alineados en límites de 4 bytes, las cargas subsiguientes no estarán alineadas y se necesitará agregar relleno al final del mensaje para crear un mensaje alineado a 4 octetos.

El tipo de carga para la Carga de Transformación es tres (3)

3.7 Carga de Intercambio de Claves

La Carga de Intercambio de Claves soporta una variedad de técnicas de intercambio de claves. Ejemplos de intercambio de claves son Oakley (Capítulo 9), Diffie-Hellman (Capítulo 5), el intercambio de claves mejorado de Diffie-Hellman descripto en x9.42 [ANSI], y el intercambio de claves basado en RSA usado por PGP. La Figura 8 muestra la Carga de Intercambio de Claves.

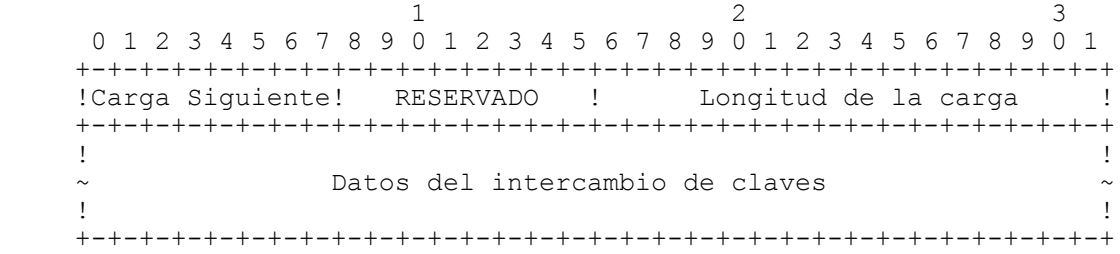


Figura 8: Formato de la Carga de Intercambio de Claves

Los campos de la Carga de Intercambio de Claves se definen de la siguiente forma:

- Carga siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): Longitud en octetos de la carga actual, incluyendo la cabecera de carga genérica.

- Datos del Intercambio de Claves (longitud variable): Datos requeridos para generar una clave de sesión. La interpretación de este dato es especificado por el DOI y por el algoritmo de Intercambio de Claves asociado. Este campo también puede contener indicadores de claves pre-establecidas (o pre-compartidas).

El tipo de carga para la Carga de Intercambio de Claves es cuatro (4)

3.8 Carga de Identificación

La Carga de Identificación contiene datos específicos del DOI usados para intercambiar información de identificación. Está información es usada para determinar las identidades de los usuarios de la comunicación y puede ser usada para determinar la autentificación de la información. La Figura 9 muestra el formato de la Carga de Identificación.

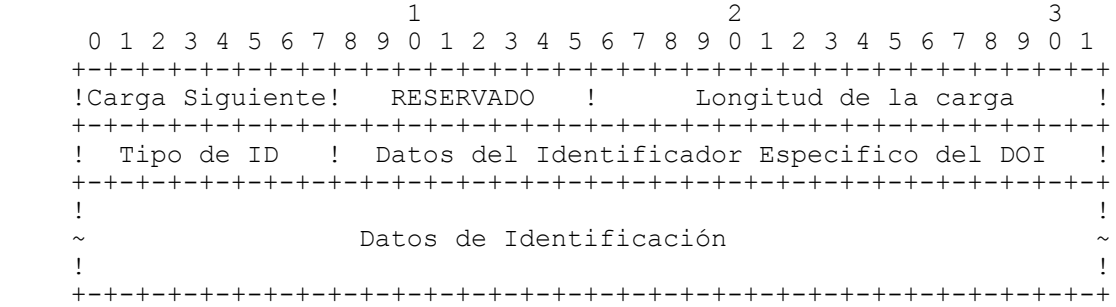


Figura 9: Formato de la Carga de Identificación

Los campos de la Carga de identificación se definen de la siguiente forma:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): La longitud en octetos de la carga actual, incluyendo la cabecera de carga genérica.

- Tipo de Identificador (1 octeto): especifica el tipo de Identificación que se está usando.

- Datos del Identificador Especifico del DOI (3 octetos): Contiene los datos de Identificación específicos del DOI. Si este campo no es usado, DEBE contener ceros. Este campo es dependiente del DOI.

- Datos de Identificación (Longitud variable): contiene información de identificación. Los valores para este campo son específicos del DOI y el formato es especificado por el campo Tipo de Identificador. Los detalles específicos para los Datos de Identificación del DOI de Seguridad IP de la IETF se detallan en el Capítulo 8.

El tipo de carga para la Carga de Identificación es cinco (5).

3.9 Carga de Certificado

La Carga de Certificado proporciona un medio para trasportar certificados o otra certificación relacionada con la información vía ISAKMP y puede aparecer en cualquier mensaje ISAKMP. Las cargas de Certificado DEBERÍAN estar incluidas en un intercambio siempre que un apropiado servicio de directorio (por ejemplo DNS seguros [DNSSEC]) no esté disponible para distribuir los certificados. La carga de Certificado DEBE ser aceptada en cualquier momento durante el intercambio. La Figura 10 muestra el formato de la Carga de Certificado.

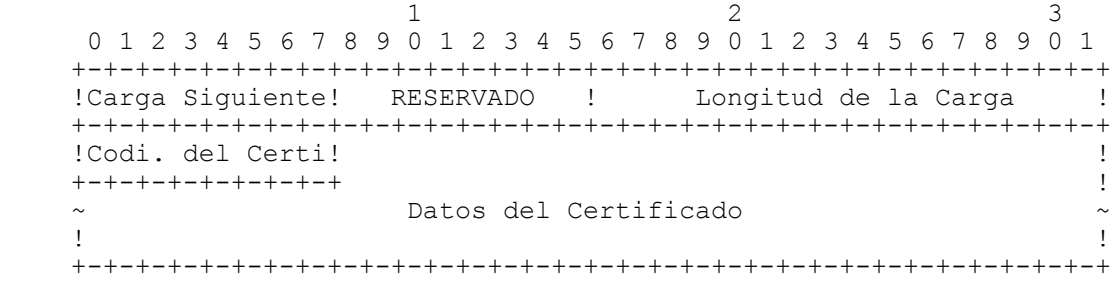


Figura 10: Formato de la Carga de Certificado.

NOTA: Los tipos de Certificados y formatos, generalmente no están ligados a un DOI. Se espera que existan solamente pocos tipos de certificaciones, y que la mayoría de los DOIs acepten estos tipos.

Los campos de la Carga de Certificado están definidos de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.
- RESERVADO (1 octeto): No utilizado, debe contener ceros.
- Longitud de la Carga (2 octetos): Longitud en octetos de la carga actual, incluyendo la cabecera de carga genérica.
- Codificación del Certificado (1 octeto): Este campo indica el tipo de certificado o certificados relacionados a la información contenida en el campo de Datos del Certificado.

Tipos de Certificados	Valor
Ninguno	0
PKCS N°7 encapsulado en certificados X.509	1
Certificados PGP	2
Clave designada por DNS	3
Certificados X.509-firma	4
Certificados X.509-intercambio de claves	5
Tokens Kerberos	6
Lista de Revocación de Certificados (CRL)	7
Lista de Revocación de Autoridad (ARL)	8
Certificados SPKI	9
Certificados X.509- Atributos	10
RESERVADO	11-255

- Datos del certificado (longitud variable): Codificación actual de los datos del certificado. El tipo de certificado está indicado por el campo Codificación del Certificado.

El tipo de carga para la Carga de Certificado es seis (6).

3.10 Carga de Solicitud de Certificado

La Carga de solicitud de Certificado proporciona un medio para solicitar certificados vía ISAKMP y puede aparecer en cualquier mensaje. Las cargas de solicitud de Certificado DEBERÍAN estar incluidas en un intercambio siempre que un apropiado servicio de directorio (por ejemplo DNS seguros [DNSSEC]) no este disponible para distribuir los certificados. La carga de Solicitud de Certificados DEBE ser aceptada en cualquier momento del intercambio. El respondedor de la carga de Solicitud de Certificados DEBE enviar su certificado, en el caso de que los certificados sean soportados, basados en los valores contenidos en la carga. Si múltiples certificados son requeridos, múltiples cargas de Solicitud de Certificados DEBERÍAN ser enviadas. La Figura 11 muestra el formato de la Carga de Solicitud de Certificado.

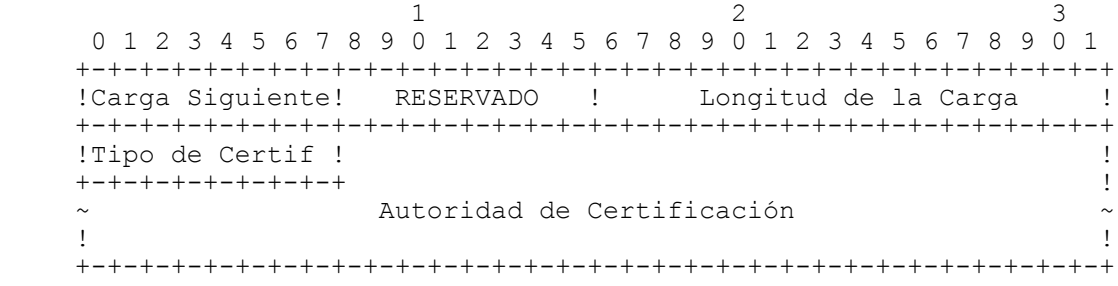


Figura 11: Formato de la Carga de solicitud de Certificado

Los campos de carga de Solicitud de Certificado son los siguientes:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): la longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.

- Tipo de Certificado (1 octeto): contiene una codificación del tipo de certificado requerido. Valores aceptables se encuentran en la tabla de la Sección 3.9.

- Autoridad de Certificación (longitud variable): Contiene una codificación de una autoridad de certificados aceptables para el tipo de certificado solicitado. Por ejemplo, para un certificado X.509 este campo contendrá la codificación del Nombre Distintivo del Nombre de la Entidad Emisora de una autoridad de certificación X.509 aceptable por el emisor de esta carga. Esta sería incluida para asistir al respondedor en la determinación de cuánto de esa cadena de certificación necesitaría ser enviada en respuesta a esta solicitud. Si no hay una autoridad de certificación específica requerida, este campo no DEBERÍA ser incluido.

El tipo de carga para la Carga de solicitud de Certificado es siete (7).

3.11 Carga Hash

La Carga Hash contiene los datos generados por la función hash (seleccionada durante el intercambio del establecimiento de la SA), sobre una cierta parte del mensaje y/o del estado de ISAKMP. Esta carga puede ser usada para verificar la integridad de los datos en un mensaje ISAKMP, o para la autentificación de las entidades de la negociación. La Figura 12 muestra el formato de la Carga Hash.

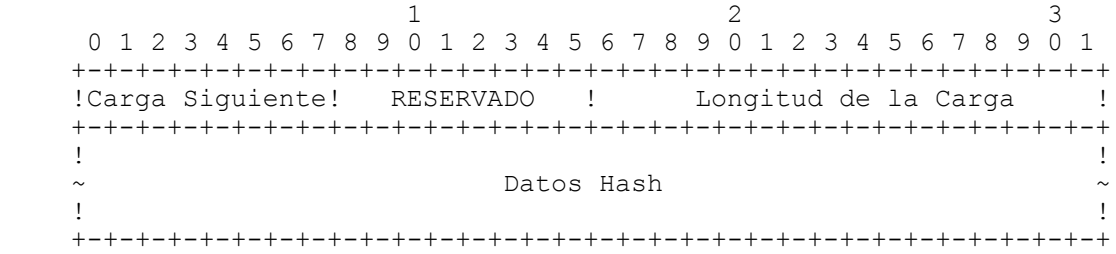


Figura 12: Formato de la carga Hash.

Los campos de la Carga de Hash se definen de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): la longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.

- Datos Hash (longitud variable): Datos que resultad de aplicar la función hash al mensaje ISAKMP y/o a su estado.

El tipo de carga para la Carga Hash es ocho (8).

3.12 Carga de la Firma

La Carga de la Firma contiene generalmente datos para la función de la firma digital (seleccionadas durante el intercambio del establecimiento de

la SA), sobre cierta parte del mensaje y/o del estado de ISAKMP. Esta carga es usada para verificar la integridad de los datos en un mensaje ISAKMP y puede ser usada para servicios de no repudio. La Figura 13 muestra el formato de la Carga de la Firma.

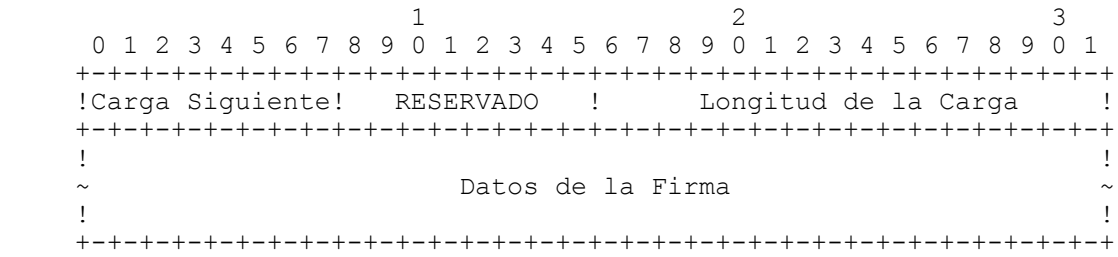


Figura 13: Formato de la Carga de la Firma

Los campos de la Carga de la Firma se definen de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.

- Datos de la Firma (longitud variable): Los datos que resultan de aplicar la función de una firma digital al mensaje y/o estado de ISAKMP.

El tipo de carga para la Carga de la Firma es nueve (9).

3.13 Carga Nonce

La Carga Nonce contiene información aleatoria para garantizar la vida de la conexión durante un intercambio y para proteger contra ataques de reenvío. Si el nonce es usado para un intercambio de clave particular, el uso de la carga nonce será dictaminado por el intercambio de claves. El nonce puede ser trasmitido como parte de los datos del intercambio de claves, o como una carga separada. Sin embargo, esta es definida por el intercambio de claves, y no por ISAKMP. La Figura 14 muestra el formato de la Carga Nonce.

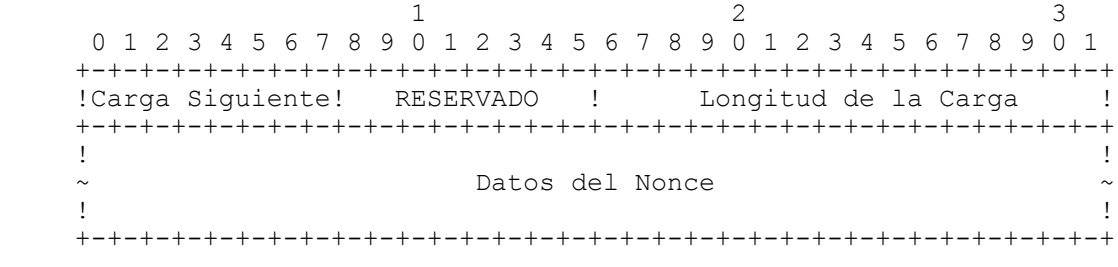


Figura 14: Formato de la Carga Nonce

Los campos de la Carga Nonce son definidos de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la

siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.
- Longitud de la Carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.
- Datos del Nonce (longitud variable): contiene información aleatoria generada por la entidad transmisora.

El tipo de carga para la Carga Nonce es diez (10).

3.14 Carga de Notificación

La Carga de Notificación puede incluir datos de ISAKMP y datos específicos del DOI, y se utiliza para trasmitir datos informativos, tales como condiciones de error, en entidades de ISAKMP. Es posible enviar múltiples cargas de Notificación en un único mensaje ISAKMP. La Figura 15 muestra el formato de la Carga de Notificación.

La notificación que ocurre durante (o que se refiere a) la negociación de la Fase 1 es identificada por el par de cookies del Iniciador y del Respondedor en la Cabecera de ISAKMP. El Identificador de Protocolo, en este caso, es ISAKMP y el valor del SPI es cero porque el par de cookies en la Cabecera ISAKMP identifican a la SA ISAKMP. Si la notificación ocurre antes de que se haya completado el intercambio de información de las claves, entonces la Notificación estará desprotegida.

La notificación que ocurre durante (o se refiere a) la Fase 2 de la negociación es identificada por el par de cookies del Iniciador y del Respondedor en la Cabecera de ISAKMP, el Identificador de Mensajes y el SPI asociados con la negociación actual. Un ejemplo de este tipo de notificación es usado para indicar por qué una propuesta fue rechazada.

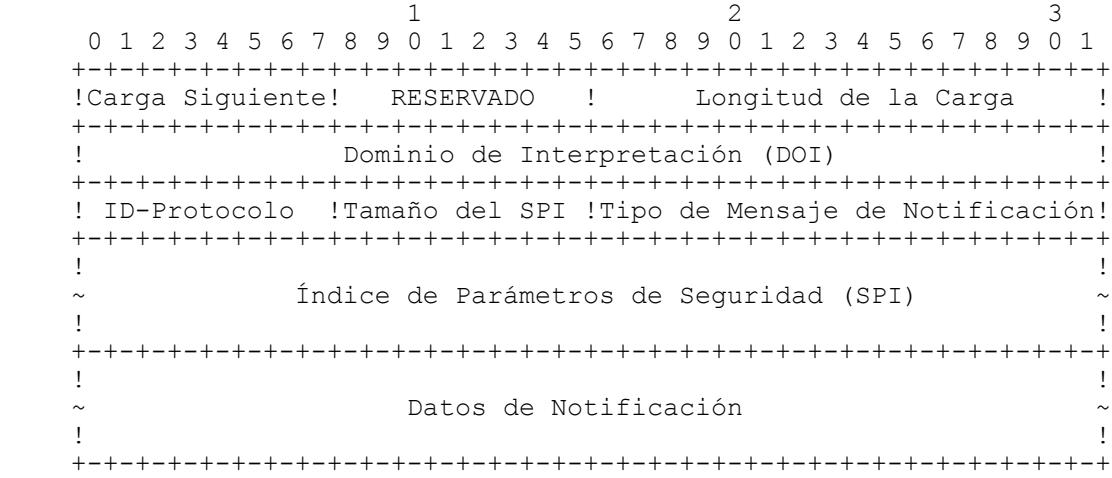


Figura 15: Formato de la Carga de Notificación

Los campos de la Carga de Notificación se definen de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el

mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.
- Longitud de la Carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.
- Dominio de Interpretación (4 octetos): Identifica el DOI (como está descripto en la Sección 2.1) bajo el cual la notificación se está llevando a cabo. Para ISAKMP este valor es cero (0) y para el DOI IPsec es uno (1). Otros DOIs pueden ser definidos usando la descripción de la Sección 7.
- Identificador de Protocolo (1 octeto): Especifica el identificador del protocolo para la notificación actual. Ejemplos de esto incluyen ESP IPSEC, AH IPSEC, OSPF, TLS, etc.
- Tamaño del SPI (1 octeto): La longitud en octetos del SPI como es definido por el Identificador de Protocolo. En el caso de ISAKMP, el par de cookies del Iniciador y del Respondedor de la Cabecera ISAKMP es el SPI de ISAKMP, por lo tanto, el Tamaño del SPI es irrelevante y PUEDE variar desde cero (0) a dieciséis (16) octetos. Si el Tamaño del SPI no es cero, el contenido del campo del SPI DEBE ser ignorado. El Dominio de Interpretación (DOI) determinará el tamaño del SPI para otros protocolos.
- Tipo de Mensaje de Notificación (2 octetos): Especifica el tipo de mensaje de notificación (ver Sección 3.14.1). Contenidos adicionales, si es especificado por el DOI, son colocados en el campo de Datos de Notificación.
- SPI (longitud variable): El SPI de la entidad receptora. El uso del campo SPI se describió en la Sección 2.4. La longitud de este campo es determinada por el campo Tamaño del SPI y no necesariamente se debe alinear a límites de 4 octetos.
- Datos de Notificación (longitud variable): Datos informativos o de error trasmitidos, además del Tipo de Mensaje de Notificación. Los valores para este campo son específicos del DOI

El tipo de carga para la Carga de Notificación es once (11).

3.14.1 Tipos de Mensaje de Notificación

La información de notificación puede tener mensajes de error especificando por qué una SA no pudo ser establecida. También puede tener datos de estado para que un manejador de procesos en una base de datos de SA pueda comunicarse con los procesos pares. Por ejemplo, una interfaz de usuario segura o un security gateway pueden usar el Mensaje de Notificación para sincronizar comunicaciones SA. La tabla siguiente enumera los tipos de Mensajes de Notificación y sus valores correspondientes. Los valores en el rango de Uso Privado son valores específicos del DOI.

MENSAJES DE NOTIFICACIÓN - TIPOS DE ERRORES	
Errores	Valor
TIPO DE CARGA NO VÁLIDA	1
DOI NO SOPORTADO	2
SITUACIÓN NO SOPORTADA	3
COOKIE NO VÁLIDO	4
VERSIÓN MAYOR NO VÁLIDA	5
VERSIÓN MENOR NO VÁLIDA	6
TIPO DE INTERCAMBIO NO BALIDO	7
BANDERAS NO VALIDAS	8
IDENTIFICADOR DE MENSAJE NO VÁLIDO	9
IDENTIFICADOR DE PROTOCOLO NO VÁLIDO	10
SPI NO VÁLIDO	11
IDENTIFICADOR DE TRANSFORMACIÓN NO VÁLIDO	12
ATRIBUTOS NO SOPORTADOS	13
ELECCIÓN DE LA PROPUESTA NO VÁLIDA	14
SINTAXIS DE LA PROPUESTA DEFICIENTE	15
CARGA MAL FORMADA	16
INFORMACIÓN DE CLAVE NO VÁLIDA	17
INFORMACIÓN DEL IDENTIFICADOR NO VÁLIDO	18
CODIFICACIÓN DE CERTIFICADO NO VÁLIDO	19
CERTIFICADO NO VÁLIDO	20
TIPO DE CERTIFICADO NO SOPORTADO	21
AUTORIDAD DE CERTIFICACIÓN NO VÁLIDA	22
INFORMACIÓN DE HASH NO VÁLIDA	23
ERROR EN LA AUTENTIFICACIÓN	24
FIRMA NO VÁLIDA	25
NOTIFICACIÓN DE DIRECCIÓN	26
NOTIFICACIÓN DE TIEMPO DE VIDA DE LA SA	27
CERTIFICADO NO DISPONIBLE	28
TIPO DE INTERCAMBIO NO SOPORTADO	29
RESERVAD (Uso Futuro)	31-8191
USO PRIVADO	8192-16383

MENSAJE DE NOTIFICACIÓN - TIPOS DE STATUS	
Estado	Valor
CONECTADO	16384
RESERVADO (Uso Futuro)	16385 - 24575
Códigos específicos de DOI	24576 - 32767
USO PRIVADO	32768 - 40959
RESERVADO (Uso futuro)	40960 – 65535

3.15 Carga de Cancelación

La Carga de Cancelación contiene un identificador de SA específico de un protocolo que el emisor ha revocado para esta base de datos de SA y por consiguiente ya no es válida. Es posible enviar múltiples SPIs en una carga

de Cancelación, sin embargo, cada SPI DEBE ser del mismo protocolo. La mezcla de identificadores de protocolo NO DEBE ser realizada en la carga de Cancelación. La Figura 16 muestra el formato de la Carga de Cancelación.

La cancelación concerniente a una SA ISAKMP contendrá un Identificador de Protocolo de ISAKMP y los SPIs son las cookies del Iniciador y Respondedor de la Cabecera de ISAKMP. La cancelación concerniente a una SA de Protocolo, tales como ESP o AH, contendrán el Identificador de Protocolo de ese protocolo (por ejemplo ESP, AH) y la SPI son las SPIs de la entidades emisoras.

Nota: La Carga de Cancelación no es una solicitud del respondedor para cancelar una SA, sino que es una notificación del iniciador al respondedor. Si el respondedor elige ignorar el mensaje, la siguiente comunicación del respondedor al iniciador, que use esa SA, fallará. Se espera que un respondedor reconozca el acuse de recibo de la carga de Cancelación.

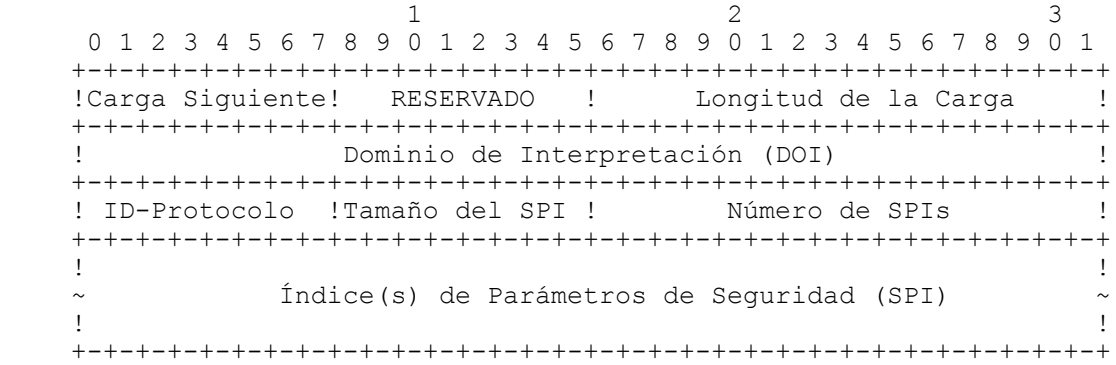


Figura 16: Formato de la Carga de Cancelación

Los campos de la Carga de Cancelación se definen de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.

- RESERVADO (1 octeto): No utilizado, debe contener ceros.

- Longitud de la Carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.

- Dominio de Interpretación (4 octetos): Identifica el DOI (como se describe en la Sección 2.1) bajo el cual esta cancelación esta tomando lugar. Para ISAKMP este valor es cero (0) y para el DOI IPsec es uno (1). Otros DOIs pueden ser definidos usando la descripción de la Sección 7.

- Identificador de Protocolo (1 octeto): ISAKMP puede establecer SA para varios protocolos, incluyendo ISAKMP y IPsec. Este campo identifica a qué Base de Datos de Asociaciones de Seguridad (SAD) se aplicará la solicitud de cancelación.

- Tamaño del SPI (1 octeto): Longitud en octetos del SPI como esta definido por el Identificador de Protocolo. En el caso de ISAKMP, el par de cookies del Iniciador y del Respondedor es el SPI de ISAKMP. En este caso el Tamaño del SPI sería de 16 octetos para cada uno de los

SPI que están siendo cancelados.

- Número de SPIs (2 octetos): El número de SPIs contenidos en la Carga de Cancelación. El tamaño de cada SPI es definido por el campo Tamaño del SPI.

- Índice(s) de parámetros de seguridad (longitud variable): identifica la SA(s) que serán canceladas. Los valores para este campo están en el DOI y protocolo especifico. La longitud de este campo es determinada por los campos Tamaño del SPI y Número de SPIs.

El tipo de carga para la carga de Cancelación es doce (12)

3.16 Carga de Identificador del Vendedor

La Carga de Identificador del Vendedor contiene una constante definida por el vendedor. La constante es usada por los vendedores para identificar y reconocer instancias remotas de sus aplicaciones. Este mecanismo permite a un vendedor experimentar nuevas características, manteniendo la compatibilidad. La Figura 17 muestra la Carga de Identificación de Vendedor.

La carga de Identificación del Vendedor no es un anuncio de que el emisor enviará tipos de cargas privadas. Un vendedor que envía un Identificador de vendedor no DEBE hacer ninguna conjetura sobre cargas privadas que podrían ser enviadas a menos que un Identificador de Vendedor sea también recibido. Múltiples cargas de Identificador de Vendedor PUEDEN ser enviadas. Una implementación NO REQUIERE comprender las cargas de Identificación de Vendedor, como así también NO REQUIERE enviar todas las cargas de Identificación de Vendedor. Si una carga privada fue enviada, sin acuerdo previo, una implementación puede rechazar la propuesta por medio de un mensaje de notificación TIPO DE CARGA NO VÁLIDA.

Si una Carga de Identificador de Vendedor es enviada, esta DEBE ser enviada durante la Fase 1 de la negociación. La recepción de una carga de Identificador de Vendedor familiar en la Fase 1 de la negociación permite que una implementación haga uso de los números de carga de Uso Privado (128 a 255), descriptos en la Sección 3.1 para extensiones especificas del vendedor durante la Fase 2 de la negociación. La definición de "familiar" se usa para determinar implementaciones. Algunos vendedores pueden desear implementar otras extensiones de vendedor antes de la estandarización. No obstante, esta práctica no DEBERÍA difundirse y los vendedores deben trabajar hacia una estandarización.

La constante definida por el vendedor DEBE ser única. La elección del hash y el texto a hashiar la decide el vendedor. Como ejemplo, los vendedores pueden generar su identificador de vendedor tomando un simple hash de la cadena de caracteres que contiene el nombre del producto, y la versión del producto.

Un hash es usado en lugar de un registro de vendedor para evitar problemas de políticas criptográficas locales con listas de productos "aprobados", para evitar tener una lista de vendedores, y evitando que productos clasificados aparezcan en alguna lista. Por ejemplo:

"Compañía IPsec. Versión 97.1"

(no incluido textualmente) Tiene un hash MD5 igual a: 48544f9b1fe662af98b9b39e50c01a5a, cuando se usa MD5FILE. Los vendedores

pueden incluir todo el hash, o solo una parte, como parte de los datos de la carga. Hay implementaciones de seguridad de este hash por lo tanto su elección es arbitraria.

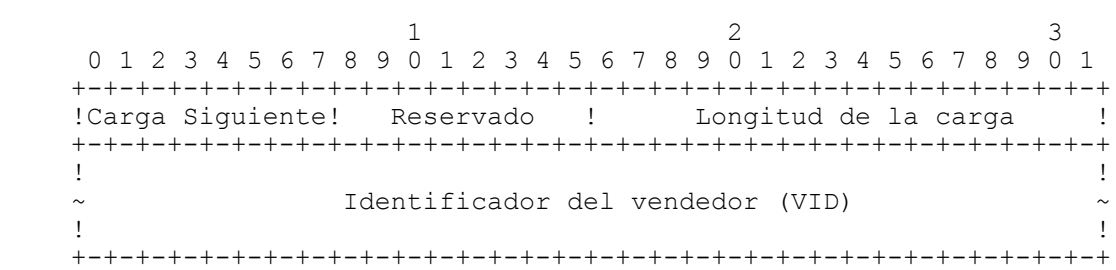


Figura 17: Formato de la Carga de Identificador del Vendedor

La Carga de Identificación del Vendedor está definida de la siguiente manera:

- Carga Siguiente (1 octeto): Identificador del tipo de carga de la siguiente carga en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero.
- RESERVADO (1 octeto): No utilizado, debe contener ceros.
- Longitud de la Carga (2 octetos): Longitud de la carga actual en octetos, incluyendo la cabecera de carga genérica.
- Identificador del vendedor (longitud variable): Hash de la cadena de caracteres del vendedor más la versión (como se describió anteriormente)

El tipo de carga para la Carga de Identificador del Vendedor es trece (13).

4. **Intercambios ISAKMP**

ISAKMP proporciona la sintaxis básica para el intercambio de mensajes. Los bloques básicos de construcción para los mensajes ISAKMP son los tipos de carga descriptos en la Sección 3. Esta sección describe los procedimientos para el establecimiento y modificación de SAs, y el conjunto de intercambios que por defecto PUEDEN ser usados para una interoperatibilidad inicial. Otros intercambios serán definidos teniendo en cuenta el DOI y el intercambio de claves. El Capítulo 8 (El DOI de IPsec) y el Capítulo 10 (IKE) son ejemplos de cómo esto es logrado. La Sección 7 explica los procedimientos para lograr estas inclusiones.

4.1 Tipos de Intercambios ISAKMP

Un tipo de intercambio es una especificación de un número de mensajes en un intercambio ISAKMP, y los tipos de carga que están contenidos en cada uno de estos mensajes. Cada tipo de intercambio está diseñado para proporcionar, un conjunto específico de servicios de seguridad, tales como el anonimato de los participantes, perfect forward secrecy para el material clave, autentificación para los participantes, etc. En esta sección se define el conjunto por defecto de tipos de intercambio ISAKMP. Si se requiere, otros tipos de intercambio se pueden agregar para soportar intercambios adicionales de claves.

ISAKMP permite la creación de intercambios para el establecimiento de SA y material claves. Hay actualmente 5 Tipos de Intercambio por defecto definidos por ISAKMP. Desde la Sección 4.4 hasta la Sección 4.8 se describen estos intercambios. Los intercambios definen los contenidos y el ordenamiento de los mensajes ISAKMP entre usuarios. La mayoría de los intercambios incluirán todos los tipos de cargas básicas (vea la tabla "Tipos de Carga Siguiente" de la Sección 3.1 que describe la notación usada para cada tipo de carga): SA, KE, ID , SIG , aunque también se pueden incluir otros tipos de intercambio. La diferencia principal entre los tipos de intercambio es el ordenamiento de los mensajes y de las cargas dentro de cada mensaje. El ordenamiento de las cargas dentro de los mensajes no esta definido, pero para el procesamiento eficiente se RECOMIENDA que la carga de SA sea la primer carga dentro de un intercambio. El procesamiento de cada carga dentro de un intercambio se describe en la Sección 5.

Desde la Sección 4.4 hasta la Sección 4.8 se ofrece un conjunto de intercambios ISAKMP por defecto. Estos intercambios proporcionan diferentes protecciones de seguridad para el intercambio mismo y la información intercambiada. Los diagramas en cada una de las siguientes secciones muestran el ordenamiento del mensaje para cada tipo de intercambio, como así también las cargas incluidas en cada mensaje, y proporcionan notas básicas que describen que ha sucedido después de cada mensaje intercambiado. Ninguno de estos ejemplos incluyen "cargas opcionales", como por ejemplo la carga de Certificado o la carga de Solicitud de Certificado. Además, ninguno de estos ejemplos incluye un intercambio inicial de las cabeceras de ISAKMP (conteniendo las cookies del iniciador y del respondedor) que proporcionarían protección contra saturación (ver Sección 2.5.3).

Los intercambios definidos no pretenden satisfacer todos los requerimientos del DOI y de los protocolos de intercambio de claves. Si los intercambios definidos satisfacen los requerimientos del DOI, podrían ser usados como se explicó. Si los intercambios definidos no satisfacen los requerimientos de seguridad definidos por el DOI, el DOI DEBE especificar nuevos tipos de intercambio y las secuencias válidas de las cargas que hacen un intercambio exitoso, y como construir y interpretar estas cargas. Todas las implementaciones de ISAKMP DEBEN implementar Intercambios Informativos y DEBERÍAN implementar los otros 4 intercambios. Sin embargo, esto depende de la definición del DOI y de los protocolos de intercambio asociados.

Como se explicó arriba, estos tipos de intercambio pueden ser usados en cualquier Fase de la negociación, no obstante, deben proporcionar diferentes propiedades de seguridad en cada una de las fases. Con cada uno de estos intercambios, la combinación de las cookies y de los campos del SPI identifican si este intercambio esta siendo usado en la primera o en la Fase 2 de la negociación.

4.1.1 Notación

La siguiente notación se usa para describir los tipos de intercambio ISAKMP, como se muestra en la siguiente sección, con los formatos de los mensajes y las cargas asociadas:

Notación	Significado
HDR	Es una cabecera de ISAKMP cuyo tipo de intercambio define el ordenamiento de la carga.
SA	Es una carga de negociación de SA con una o más Propuestas y cargas de Transformación. Un iniciador PUEDE proporcionar múltiples propuestas para la negociación, un respondedor DEBE contestar solo una.
KE	Es la carga de intercambio de claves.
IDx	Es la carga de Identificación para "x". x puede ser "ii" o "ir" para el iniciador y respondedor de ISAKMP, respectivamente, o x puede ser "ui", "ur" (cuando un demonio de ISAKMP es un negociador proxy), para el usuario iniciador y respondedor respectivamente.
HASH	Es la carga hash.
SIG	Es la carga de la firma. Los datos a firmar son específicos del intercambio.
AUTH	Es un mecanismo de autentificación genérico, como HASH o SIG.
NONCE	Es la carga NONCE.
*	Significa encriptación de la carga después de la cabecera ISAKMP. Esta encriptación DEBE comenzar inmediatamente después de la cabecera ISAKMP y todas las cargas que siguen a la cabecera de ISAKMP DEBEN estar encriptadas.
=>	Comunicación desde el "iniciador al respondedor".
<=	Comunicación desde el "respondedor al iniciador".

4.2 Establecimiento de Asociaciones de Seguridad

Las cargas, SA, la de la Propuesta, y la de Transformación son utilizadas para construir los mensajes ISAKMP para la negociación y el establecimiento de SAs. Un mensaje de establecimiento de SA consiste en una única carga SA seguida de al menos una y posiblemente muchas, cargas de Propuesta, y al menos una y posiblemente muchas, cargas de Transformación asociadas con cada carga de la Propuesta. Debido a que estas cargas se consideran en conjunto, las cargas SA apuntarán a cualquiera de las cargas siguientes y no a la carga de la Propuesta incluida en la carga SA. La carga SA contiene, el DOI y la situación para la SA propuesta. Cada carga de la Propuesta contiene un SPI, esto garantiza que el SPI esta asociado con el Identificador de Protocolo en concordancia con el Capítulo 2. Las cargas de la Propuestas pueden o no tener el mismo SPI, ya que es una implementación dependiente. Cada carga de Transformación contiene mecanismos de seguridad específicos para ser usados por el protocolo designado. Se espera que la Propuesta y las cargas de Transformación sean usadas solamente durante la negociación del establecimiento de la SA. La creación de cargas para la negociación y establecimiento de SA descriptas en esta sección se aplican a todos los intercambios ISAKMP que se describen desde la Sección 4.4 hasta la Sección 4.8. Los ejemplos mostrados en el punto 4.2.1 contienen solamente las cargas, SA, la de la Propuesta y la de Transformación, y no contienen otras cargas que podrían existir en un intercambio ISAKMP determinado.

La carga de la Propuesta proporciona a la entidad iniciadora la capacidad de presentarle a la entidad que responde los protocolos de seguridad y mecanismos de seguridad asociados para el uso de la SA que se están negociando. Si la negociación del establecimiento de una SA es para un conjunto combinado de protección que consiste de múltiples protocolos, DEBERÁ existir múltiples cargas de Propuesta, cada una con el mismo número de Propuesta. Estas propuestas DEBEN considerarse como una unidad y NO DEBEN estar separadas por una propuesta con un número de propuesta

diferente. El uso del mismo número de Propuesta en múltiples cargas de Propuesta proporciona lógica de operación AND, es decir Protocolo 1 AND Protocolo 2. La Sección 4.2.1.1 muestra un ejemplo de un conjunto de protección ESP AND AH. Si la negociación del establecimiento de SA es para diferentes conjuntos de protección, DEBERÁN existir múltiples cargas de Propuesta cada una con un número de Propuesta incrementalmente único. Las diferentes propuestas DEBEN ser presentadas en el orden de preferencia del iniciador. El uso de diferentes números de Propuesta en múltiples cargas de Propuesta proporciona lógica de operación OR, es decir, Propuesta 1 OR Propuesta 2, donde cada propuesta puede tener más de un protocolo. La Sección 4.2.1.2 muestra un ejemplo de un conjunto de protección AH AND ESP, OR solamente un conjunto de protección ESP. Observe que el campo Carga Siguiente de la carga de la Propuesta apunta a otra carga de la Propuesta (si existiera). La existencia de una carga de Propuesta implica la existencia de una o mas cargas de Transformación.

La Carga de Trasformación proporciona a la entidad iniciadora la capacidad de presentar a la entidad que responde múltiples mecanismos, o trasformaciones, para un protocolo dado. La carga de la Propuesta identifica a un Protocolo para el cual los servicios y mecanismos se están negociado. La carga de Transformación permite a la entidad iniciadora presentar múltiples trasformaciones posibles soportadas para el protocolo propuesto. Pueden existir muchas trasformaciones asociadas con una carga de Propuesta específica, cada una identificará una carga de Transformación separada. Las múltiples transformaciones DEBEN ser presentadas con números crecientes únicos de acuerdo al orden de preferencia del iniciador. La entidad receptora DEBE seleccionar una única transformación para cada protocolo dentro de una propuesta o rechazar la propuesta entera. El uso del número de Transformación en las cargas de Trasformaciones múltiples proporciona un segundo nivel de operación OR, es decir Trasformación 1 OR Trasformación 2 OR Trasformación 3. La Sección 4.2.1.1 muestra 2 trasformaciones posibles para ESP y una única transformación para AH. La Sección 4.2.1.2 muestra una transformación para AH AND una transformación para ESP OR dos trasformaciones para ESP. Observe que el campo Carga Siguiente de la carga de Transformación puede apuntar hacia cero o más cargas de Transformación.

Cuando se responde a una carga SA, el respondedor DEBE enviar una carga SA con la propuesta seleccionada, la cual consistirá de múltiples cargas de Propuestas y sus cargas de Transformación asociadas. Cada una de las cargas de la Propuesta DEBE contener una única carga de Transformación asociada con el protocolo. El respondedor DEBERÍA retener el campo Número de Propuesta dentro de la carga de Propuesta y el campo Número de Transformación en cada carga de Transformación de la de la propuesta seleccionada. La retención de los números de la Propuesta y Transformación deberá acelerar el procesamiento del protocolo del iniciador por la anulación de la necesidad de comparar la selección del respondedor con cada una de las opciones ofrecidas. Estos valores permiten al iniciador realizar la comparación directa y rápidamente. El iniciador DEBE verificar que la carga SA recibida del respondedor concuerde con las propuestas enviadas inicialmente.

4.2.1 Ejemplos de Establecimientos de SA

4.2.1.1 Ejemplo N°1 – Conjunto de Protección ESP AND AH

Este ejemplo muestra una Propuesta para un conjunto de protección combinado con 2 protocolos diferentes. El primer protocolo esta presentado por dos trasformaciones soportadas por el oferente. El segundo protocolo esta

presentado por una única transformación. Un ejemplo de esta propuesta puede ser: Protocolo 1 es, ESP con Transformación 1 con 3DES y Transformación 2 con DES AND Protocolo 2 es, AH con transformación 1 con SHA. El respondedor DEBE elegir una de las 2 trasformaciones que el oferente propone para ESP. El conjunto de protección resultante será (1) 3DES AND SHA, OR (2) DES AND SHA, dependiendo de que transformación ESP fue seleccionada por el respondedor. Observe que este ejemplo es mostrado usando el Intercambio Base.

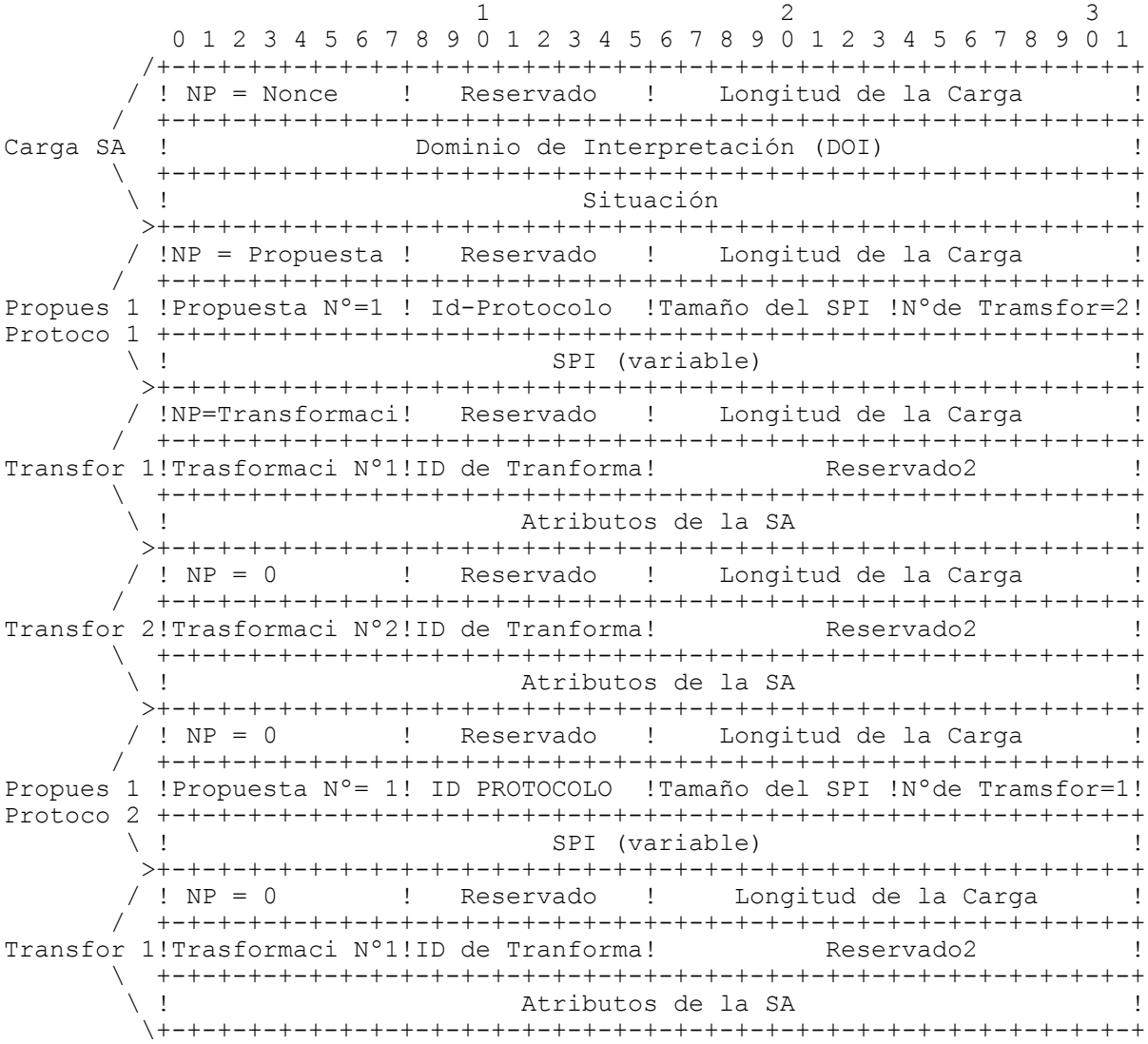


Figura 18: 2 Trasformaciones para ESP y 1 transformación para AH

4.2.1.2 Ejemplo N°2 – Conjunto de Protección ESP AND AH, OR Solamente ESP

Este segundo ejemplo muestra una Propuesta para 2 conjuntos de protección diferente. El primer conjunto de protección es presentado con una transformación para el primer protocolo y una transformación para el segundo. El segundo conjunto de protección es presentado con 2 transformaciones para un solo protocolo. Un ejemplo para esta propuesta

puede ser: Propuesta 1 con Protocolo 1 con AH con Transformación 1 con MD5 AND Protocolo 2 con ESP con Transformación 1 con 3DES. Esto es seguido por la Propuesta 2 con Protocolo 1 con ESP con Transformación 1 con DES y Transformación 2 con 3DES. El respondedor DEBE seleccionar una de las dos propuestas. Si la segunda Propuesta es seleccionada, el respondedor DEBE seleccionar una de las dos trasformaciones para ESP. El conjunto de protección resultante será (1) MD5 AND 3DES, OR la selección entre (2) DES, OR (3) 3DES.

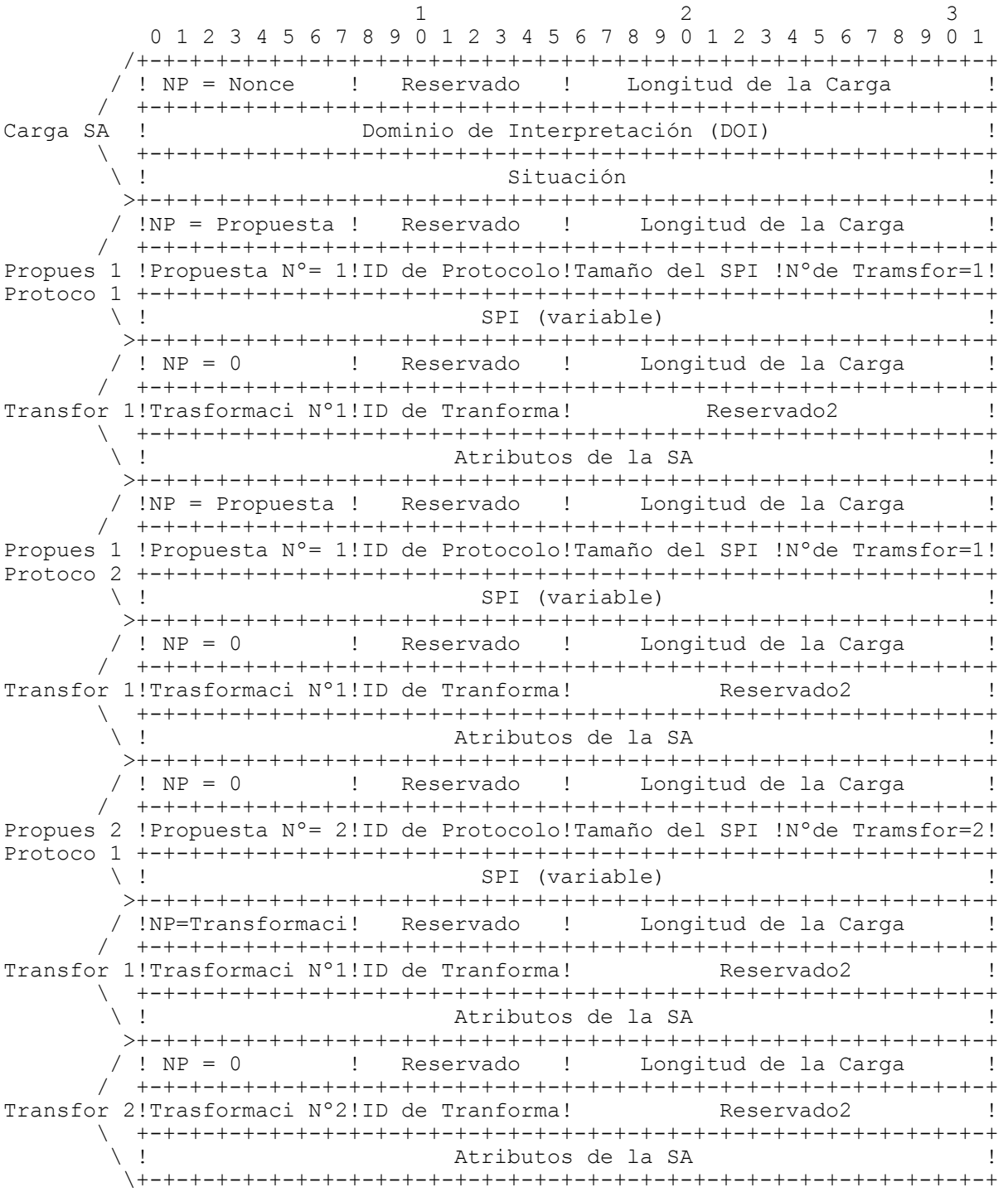


Figura 19: 1 transformación para AH AND 1 transformación para ESP OR 2 trasformaciones para ESP.

4.3 Modificación de Asociaciones de Seguridad

La modificación de una SA dentro de ISAKMP es llevada a cabo mediante la creación de una nueva SA y las comunicaciones que se inician usarán esa nueva SA. La cancelación de la antigua SA puede hacerse en cualquier momento después de que la nueva SA haya sido establecida. La cancelación de la antigua SA depende de la política de seguridad local. La modificación de SAs usa el método de "creación de una nueva SA seguida de la cancelación de la antigua SA" esto se hace para evitar vulnerabilidades potenciales dentro de la sincronización de la modificación de los atributos de la SA existentes. El procedimiento para la creación de nuevas SAs se definió en la Sección 4.2. El procedimiento para la cancelación de SAs esta definido en la Sección 5.15.

La modificación de una SA ISAKMP (Fase 1 de la negociación) sigue el mismo procedimiento que la creación de una SA ISAKMP. No existe relación entre las 2 SAs, y el par de cookies del iniciador y del respondedor DEBERÍAN ser diferentes, como se definió en la Sección 2.5.3.

La modificación de una SA de Protocolo (Fase 2 de la negociación) sigue el mismo procedimiento que la creación de una SA de Protocolo. La creación de una nueva SA está protegida por la SA ISAKMP existente. No hay relación entre las dos SA del Protocolo. La aplicación del protocolo no DEBERÍA comenzar a utilizar la nueva SA creada para el tráfico saliente y DEBERÍA continuar soportando el tráfico entrante en la antigua SA hasta que la SA este cancelada o hasta que el tráfico de la antigua SA este bajo la protección de la nueva SA creada. Según lo indicado anteriormente en esta sección, la cancelación de una SA antigua depende de la política de seguridad local.

4.4 Intercambio Base

El Intercambio Base está diseñado para permitir que el Intercambio de Claves y la Autentificación relacionen información trasmitida simultáneamente. La combinación del Intercambio de Claves y la información de Autentificación relacionada dentro de un mensaje reduce el número de viajes de ida y vuelta a expensas de no proporcionar protección de identidad. La protección de identidad no es proporcionada porque las identidades se intercambian antes de que un secreto común compartido haya sido establecido, por consiguiente, la encriptación de las identidades no es posible. El siguiente diagrama muestra los mensajes con las posibles cargas enviadas en cada mensaje y las notas de ejemplo de un Intercambio Base.

INTERCAMBIO BASE				
N°	Iniciador	Dirección	Respondedor	Notas
(1)	HDR; SA; NONCE	=>		Comienzo SA-ISAKMP o negociación Proxy
(2)		<=	HDR; SA; NONCE	SA básica acordada
(3)	HDR; KE; IDii; AUTH	=>		Clave generada (por el respondedor) identidad del iniciador verificada por el respondedor
(4)		<=	HDR; KE; IDir; AUTH	Identidad del respondedor verificada por el iniciador, clave generada (por el iniciador), SA establecida

En el primer mensaje (1), el iniciador genera una propuesta que considera adecuada para proteger el tráfico en una situación dada. Las cargas, SA, la de la Propuesta, y la de Transformación, son incluidas en la carga SA (para simplificar la notación). La información aleatoria usada para garantizar la vida de la conexión y protección contra ataques de reenvío también es trasmitida. La información aleatoria que proveen ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proveer prueba compartida de la participación en el intercambio.

En el segundo mensaje (2), el respondedor indica el conjunto de protección que ha aceptado con las cargas, SA, la de la Propuesta, y la de Transformación. Nuevamente, la información aleatoria que es usada para proteger contra ataques de reenvío y garantizar la vida de la conexión también es trasmitida. La información aleatoria proporcionada por ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio. La política de seguridad local dictaminará la acción del respondedor si no es aceptado el conjunto de protección propuesto. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

En el tercer (3) y cuarto (4) mensaje, el iniciador y el respondedor, respectivamente intercambian material clave usado para llegar a un secreto común compartido y a la identificación de la información. Esta información es trasmitida bajo la protección de una función de autentificación acordada. La política de seguridad local dictaminará la acción si un error llegara a ocurrir durante estos mensajes. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

4.5 Intercambio de Protección de Identidad

El Intercambio de Protección de Identidad está diseñado para separar la información de Intercambio de Claves de la de Identificación y de la información relacionada con la Autentificación. La separación del Intercambio de Claves de la de Identificación y de la información relacionada con la Autentificación proporcionan protección para las entidades de la comunicación, a costa de dos mensajes adicionales. Las identidades se intercambian bajo la protección de un secreto común compartido establecido anteriormente. El siguiente diagrama muestra los mensajes con las posibles cargas enviadas en cada mensaje y las notas de ejemplo de un Intercambio de Protección de Identidad.

INTERCAMBIO DE PROTECCIÓN DE IDENTIDAD				
N°	Iniciador	Dirección	Respondedor	Notas
(1)	HDR; SA	=>		Comienzo SA-ISAKMP o negociación Proxy
(2)		<=	HDR; SA	SA básica acordada
(3)	HDR; KE; NONCE	=>		
(4)		<=	HDR; KE; NONCE	Clave generada (por el Iniciador y Respondedor)
(5)	HDR*; KE; IDii; AUTH	=>		Identidad del Iniciador Verificada por el Respondedor
(6)		<=	HDR*; IDir; AUTH	Identidad del Respondedor Verificada por el Iniciador, SA establecida

En el primer mensaje (1), el iniciador genera una propuesta que considera adecuada para proteger el tráfico en una situación dada. Las cargas, SA, la de la Propuesta, y la de Transformación, son incluidas en la carga SA (para simplificar la notación).

En el segundo mensaje (2), el respondedor indica el conjunto de protección que ha aceptado con las cargas, SA, la de la Propuesta, y la de Transformación. La política de seguridad local determinará la acción del respondedor si no se acepta el conjunto de protección propuesto. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

En el tercer (3) y cuarto (4) mensaje, el iniciador y el respondedor, respectivamente intercambian material clave usado para llegar a un secreto común compartido y la información aleatoria que es usada para garantizar la vida de la conexión y proteger contra ataques de reenvío. La información aleatoria proporcionada por ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio. La política de seguridad local dictaminará la acción ha seguir si un error ocurre durante estos mensajes. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

En el quinto (5) y sexto (6) mensaje, el iniciador y el respondedor, respectivamente, intercambian información de identificación y los resultados de la función de autentificación acordada. Esta información es trasmitida bajo la protección de un secreto común compartido. La política de seguridad local dictaminará la acción a seguir si ocurre un error durante estos mensajes. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

4.6 Intercambio de Solamente Autentificación

El Intercambio de Solamente Autentificación está diseñado para permitir solamente la Autentificación relacionada con la información ha trasmitir. El beneficio de este intercambio es la capacidad de realizar solamente la autentificación sin otro costo computacional de claves. Usando este intercambio durante la negociación, ninguna información trasmitida será encriptada. Sin embargo, la información puede ser encriptada en otros lugares. Por ejemplo, si la encriptación es negociada durante la Fase 1 de una negociación y solamente el intercambio de autentificación es usado en

la Fase 2 de la negociación, solamente el intercambio de autentificación será encriptado por la SAs de ISAKMP negociadas en la Fase 1. El siguiente diagrama muestra los mensajes con las posibles cargas enviadas en cada mensaje y las notas de ejemplo de un Intercambio de Solamente Autentificación.

INTERCAMBIO DE SOLAMENTE AUTENTIFICACIÓN				
N°	Iniciador	Dirección	Respondedor	Notas
(1)	HDR; SA; NONCE	=>		Comienzo SA-ISAKMP o negociación Proxy
(2)		<=	HDR; SA; NONCE; IDir; AUTH	SA básica acordada, Identidad del Respondedor verificada por el Iniciador
(3)	HDR; IDii; AUTH	=>		Identidad del Iniciador verificada por el Respondedor, SA establecida

En el primer mensaje (1), el iniciador genera una propuesta que considera adecuada para proteger el tráfico en una situación dada. Las cargas, SA, la de la Propuesta, y la de Transformación, son incluidas en la carga SA (para simplificar la notación). La información aleatoria que es usada para garantizar la vida de la conexión y proteger, contra ataques de reenvío también es trasmitida. La información aleatoria proporcionada por ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio.

En el segundo (2) mensaje, el respondedor indica el conjunto de protección que ha aceptado con las cargas, SA, la de la Propuesta, y la de Transformación. Una vez más, la información aleatoria que es usada para garantizar la vida de la conexión y la protección contra ataques de reenvío también es trasmitida. La información aleatoria proporcionada por ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio. Además, el respondedor debe transmitir información de identificación. Toda esta información es trasmitida bajo la protección de la función de autentificación acordada. La política de seguridad local dictaminará la acción del respondedor si no se acepta el conjunto de protección propuesto. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

En el tercer (3) mensaje, el iniciador trasmite la información de identificación. Esta información es trasmitida bajo la protección de una función de autentificación acordada. La política de seguridad local dictaminará la acción ha seguir si un error ocurre durante estos mensajes. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

4.7 Intercambio Agresivo

El Intercambio Agresivo está diseñado para permitir que la SA, el Intercambio de Claves y las cargas relacionadas con la Autentificación sean trasmitidas en forma simultánea. Combinar la SA, el intercambio de claves, y la información relacionada con la Autentificación en un mensaje, reduce el número de viajes de ida y vuelta a expensas de no proporcionar la protección de identidad. La protección de identidad no es proporcionada porque las identidades se intercambian antes de que un secreto común

compartido haya sido establecido, por consiguiente, la encriptación de las identidades no es posible. Además, el Intercambio Agresivo es un intento para establecer toda la información relevante a la seguridad en un único intercambio. El siguiente diagrama muestra los mensajes con las posibles cargas enviadas en cada mensaje y las notas de ejemplo de un Intercambio Agresivo.

INTERCAMBIO AGRESIVO				
N°	Iniciador	Dirección	Respondedor	Notas
(1)	HDR; SA; KE; NONCE; IDii	=>		Comienzo SA-ISAKMP o negociación Proxy y Intercambio de Claves
(2)		<=	HDR; SA; KE; NONCE; IDir; AUTH	Identidad del Iniciador verificada por el Respondedor, Clave generada, SA básica acordada
(3)	HDR*; AUTH	=>		Identidad del Respondedor verificada por el Iniciador, SA establecida

En el primer mensaje (1), el iniciador genera una propuesta que considera adecuada para proteger el tráfico para la situación dada. Las cargas, SA, la de la Propuesta, y la de Transformación, son incluidas en la carga SA (para simplificar la notación). Solamente puede existir una Propuesta y una Transformación ofrecida (es decir no hay elección) acordada para el funcionamiento del intercambio agresivo. El material clave usado para llegar a un secreto común compartido y la información aleatoria que es usada para garantizar la vida de la conexión y protección contra ataques de reenvío también es trasmitida. La información aleatoria proporcionada por ambas partes DEBERÍA ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio. Además, el iniciador trasmite información de identificación.

En el segundo (2) mensaje, el respondedor indica el conjunto de protección que ha aceptado con las cargas, SA, la de la Propuesta, y la de Transformación. El material clave usado para llegar a un secreto común compartido y la información aleatoria que es usada para garantizar la vida de la conexión y proteger contra ataques de reenvío también es transmitida. La información aleatoria proporcionada por ambas partes debe ser usada por el mecanismo de autentificación para proporcionar prueba compartida de la participación en el intercambio. Además, el respondedor trasmite la información de identificación. Toda esta información es trasmitida bajo la protección de una función de autentificación acordada. La política de seguridad local dictaminará la acción del respondedor si el conjunto de protección propuesto no es aceptado. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

En el tercer (3) mensaje, el iniciador trasmite los resultados de la función de autentificación acordada. Esta información es trasmitida bajo la protección de un secreto común compartido. La política de seguridad local dictaminará la acción ha seguir si ocurre un error durante estos mensajes. Una posible acción es la transmisión de una carga de Notificación como parte de un Intercambio Informativo.

4.8 Intercambio Informativo

El Intercambio Informativo está diseñado como una transmisión unidireccional de información que puede ser usada para la administración de SA. El siguiente diagrama muestra los mensajes con las posibles cargas enviadas en cada mensaje y las notas de ejemplo de un Intercambio Informativo.

INTERCAMBIO INFORMATIVO				
N°	Iniciador	Dirección	Respondedor	Notas
(1)	HDR*; N/D	=>		Notificación de Error o Cancelación

En el primer mensaje (1), el iniciador o el respondedor transmite una Notificación ISAKMP o una carga de Cancelación.

Si el Intercambio Informativo ocurre antes que el Intercambio de material clave, durante la Fase 1 de la negociación de ISAKMP, no habrá protección para el Intercambio Informativo. Una vez que el material clave haya sido intercambiado o una SA ISAKMP haya sido establecida, el Intercambio Informativo DEBE ser trasmitido bajo la protección proporcionada por el material clave o la SA ISAKMP.

Todos los intercambios son similares, en que en el comienzo de cada intercambio, la sincronización criptográfica DEBE ocurrir. El Intercambio Informativo es un intercambio y no un mensaje ISAKMP. Por ende, la generación de un Identificador de Mensaje (MID) para un Intercambio Informativo DEBERÍA ser independiente de los IVs o de otras comunicaciones en curso. Esto asegura que la sincronización de la criptografía es mantenida para las comunicaciones existentes y el Intercambio Informativo será procesado correctamente. La única excepción a esto es cuando el Bit de Commit de la cabecera de ISAKMP está en uno. Cuando el Bit de Commit está en uno, el campo Identificador de Mensaje del Intercambio Informativo DEBE contener el Identificador de Mensaje de la negociación de la SA de la Fase 2 de ISAKMP primitiva, en vez de un nuevo Identificador de Mensaje (MID). Esto se realiza para asegurar que el Intercambio Informativo está vinculado con el Mensaje de Notificación pudiendo ser asociado con la correcta Fase 2 de la SA. Para una descripción del Bit de Commit véase la Sección 3.1.

5. **Procesamiento de la Carga ISAKMP**

La Sección 3 describe las cargas de ISAKMP. Estas cargas son usadas en los intercambios descriptos en la Sección 4 y pueden ser usados en los intercambios para DOI específicos. Esta sección describe el procesamiento para cada una de las cargas. Se sugiere que los eventos descriptos en esta sección sean registrados en un apropiado sistema de auditoría de archivos. El procesamiento de los mensajes es determinado por la política de seguridad local, por lo tanto esta sección solo "sugiere" acciones.

5.1 Procesamiento General del Mensaje

Cada mensaje ISAKMP tiene un procesamiento básico aplicado para asegurar la confiabilidad del protocolo, y para minimizar amenazas, tales como denegación de servicio y ataques de reenvío. Todos los procesamientos DEBERÍAN incluir chequeos de la longitud del paquete para asegurarse que el paquete recibido tiene la longitud dada en la cabecera de ISAKMP. Si la longitud del mensaje ISAKMP y el valor en el campo de longitud de la carga en la cabecera de ISAKMP no son los mismos, el mensaje ISAKMP DEBE ser

rechazado y la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. El evento, LONGITUD DE LAS CARGAS DESIGUALES, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

2. Un Intercambio Informativo con una carga de Notificación conteniendo el mensaje, LONGITUD DE CARGA DESIGUAL, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de política de seguridad.

Al trasmitir un mensaje ISAKMP, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Fijar el timer y inicializar un retry counter.

   NOTA: las implementaciones NO DEBEN usar un valor de timer prefijado. En lugar de esto los valores del timer de transmisión deben ser ajustados dinámicamente basados en el tiempo de ida y vuelta. Además, sucesivas retransmisiones del mismo paquete deben estar separadas por intervalos de tiempo cada vez más largos (por ejemplo cuando un host que ha experimentado una colisión en una red espera un tiempo exponencial para retransmitir).

2. Si el timer espira, el mensaje ISAKMP es reenviado y el retry counter es decrementado.

3. Si el retry counter llega a cero (0), el evento, ALCANZÓ EL LIMITE DE REINTENTOS, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

4. El mecanismo del protocolo ISAKMP borra todos los estados y retorna a al estado INACTIVO.

5.2 Procesamiento de la Cabecera de ISAKMP

Cuando se crea un mensaje ISAKMP, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Crear la cookie respectiva. Ver Sección 2.5.3 para más detalles.

2. Determinar las características de seguridad relevantes de la sesión (es decir el DOI y situación).

3. Construir una Cabecera ISAKMP con los campos descriptos en la Sección 3.1.

4. Construir otras cargas ISAKMP, dependiendo del tipo de intercambio.

5. Trasmitir el mensaje al host de destino como se describe en la Sección 5.1.

Cuando un mensaje ISAKMP es recibido, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Verificar la "cookies" del iniciador y del respondedor. Si la validación de la cookie falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, COOKIE NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, COOKIE NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Comprobar el Campo Carga Siguiente para confirmar si es válido. Si la validación del campo Carga Siguiente falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, CARGA SIGUIENTE NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, TIPO DE CARGA NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

3. Comprobar el campo Versión Mayor y el campo Versión Menor para confirmar si son los correctos (ver Sección 3.1). Si la validación del campo de la versión falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, VERSIÓN ISAKMP NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, VERSIÓN MAYOR NO VÁLIDA o VERSIÓN MENOR NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

4. Comprobar el campo Tipo de Intercambio para confirmar si es válido. Si la validación del campo Tipo de Intercambio falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, TIPO DE INTERCAMBIO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, TIPO DE INTERCAMBIO NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5. Comprobar el campo Banderas para asegurarse de que contienen los valores correctos. Si la validación del campo Banderas falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, BANDERAS NO VALIDAS, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, BANDERAS NO VALIDAS, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

6. Comprobar el campo Identificador de Mensaje para asegurarse que contiene los valores correctos. Si la validación del Identificador de

Mensaje falla, el mensaje es descartado y las siguientes acciones son tomadas:

(a) El evento, IDENTIFICADOR DE MENSAJE NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, IDENTIFICADOR DE MENSAJE NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

7. El procesamiento del mensaje ISAKMP continúa según lo definido por el campo Carga Siguiente.

5.3 Procesamiento de la Cabecera de Carga Genérica

Cuando se crea cualquiera de las Cargas ISAKMP descriptas desde la Sección 3.4 hasta la Sección 3.15 una Cabecera de Carga Genérica es colocada al comienzo de estas cargas. Al crear una Cabecera de Carga Genérica, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Colocar el valor de la Carga Siguiente en el campo de Carga Siguiente. Estos valores están descriptos en la Sección 3.1.

2. Colocar el valor cero (0), en el campo RESERVADO.

3. Colocar la longitud (en octetos) de la carga en el campo Longitud de la Carga.

4. Construir las cargas según lo definido en el resto de esta sección.

Cuando se reciben cualquiera de las cargas ISAKMP, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Comprobar el campo Carga Siguiente para confirmar si es válido. Si la validación del campo Carga Siguiente falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, CARGA SIGUIENTE NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, TIPO DE CARGA NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Comprobar que el campo RESERVADO contenga el valor cero. Si el valor en el campo RESERVADO no es cero, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, CAMPO RESERVADO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, SINTAXIS DE LA PROPUESTA DEFICIENTE o CARGA MAL FORMADA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

3. Procesar las cargas restantes según lo definido por el campo Carga Siguiente.

5.4 Procesamiento de la Carga SA

Cuando se crea una carga SA, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el Dominio de Interpretación para el cual se está realizando esta negociación.

2. Determinar la situación (Secciones 3.4) dentro del DOI determinado para el cual se está realizando esta negociación.

3. Determinar la/s propuesta/s (Secciones 3.5) y la/s transformación/es (Secciones 3.6) dentro de la situación.

4. Construir una carga SA.

5. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando una carga SA es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Dominio de Interpretación (DOI) es soportado. Si la determinación del DOI falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, DOI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, DOI NO SOPORTADO, DEBE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Determinar si la Situación dada puede ser protegida. Si la determinación de la Situación falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, SITUACIÓN NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, SITUACIÓN NO SOPORTADA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

3. Procesar las cargas restantes (es decir, la carga de la Propuesta y la de Transformación) a la de la carga SA. Si la Propuesta de la SA (como se describe en las Secciones 5.5 y 5.6) no es aceptada, la siguientes acciones son tomadas:

   (a) El evento, PROPUESTA NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, ELECCIÓN DE LA PROPUESTA NO

VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.5 Procesamiento de la Carga de la Propuesta

Cuando se crea una Carga de la Propuesta la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el Protocolo para esta propuesta.

2. Determinar el número de propuestas que serán ofrecidas para este protocolo y el número de trasformaciones para cada propuesta. Las trasformaciones están descritas en la Sección 3.6.

3. Generar un único SPI seudo aleatorio.

4. Construir una carga de la Propuesta.

Cuando una carga de Propuesta es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Protocolo es soportado. Si el campo Identificador de Protocolo no es válido, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, PROTOCOLO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, IDENTIFICADOR DE PROTOCOLO NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Determinar si el SPI es válido. Si el SPI no es válido, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, SPI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, SPI NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de política de seguridad.

3. Asegurar que la Propuestas estén presentadas conforme a los detalles dados en las Secciones 3.5 y 4.2. Si las propuestas no están formuladas correctamente las siguientes acciones son tomadas:

   (a) Los posibles eventos, SINTAXIS DE LA PROPUESTA DEFICIENTE, PROPUESTA NO VÁLIDA, PUEDEN ser registrados en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, SINTAXIS DE LA PROPUESTA DEFICIENTE o CARGA MAL FORMADA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

4. Procesar las cargas de la Propuesta y Transformación según lo definido

por el campo Carga Siguiente. Ejemplos del procesamiento de estas cargas están dados en la Sección 4.2.1.

5.6 Procesamiento de la Carga de Transformación

Cuando se crea una Carga de Transformación, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el número de Transformación para esta transformación.

2. Determinar el número de trasformaciones que serán ofrecidas para esta propuesta. Las Trasformaciones se describen en la Sección 3.6.

3. Construir una Carda de Transformación.

Cuando una carga de Transformación es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si la Transformación es soportada. Si el campo Identificador de Transformación contiene un valor no conocido o no soportado, la carga de Transformación DEBE ser ignorada y NO DEBE causar la generación de un evento de TRANSFORMACIÓN NO VÁLIDA. Si el campo Identificador de Transformación no es válido, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, TRASFORMACIÓN NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, IDENTIFICADOR DE TRANSFORMACIÓN NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Asegurar que las Trasformaciones estén presentadas conforme a los detalles dados en las Secciones 3.6 y 4.2. Si las transformaciones no están formuladas correctamente, las siguientes acciones son tomadas:

   (a) Los posibles eventos, SINTAXIS DE LA PROPUESTA DEFICIENTE, TRANSFORMACIÓN NO VÁLIDA, ATRIBUTOS NO VALIDOS, son registrados en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, SINTAXIS DE LA PROPUESTA DEFICIENTE, CARGA MAL FORMADA o ATRIBUTOS NO SOPORTADOS, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

3. Procesar las cargas de Transformación y Propuestas subsiguientes, según lo definido por el campo Carga Siguiente. Ejemplos del procesamiento de estas cargas están dados en la Sección 4.2.1.

5.7 Procesamiento de la Carga de Intercambio de Claves

Cuando se crea una Carga de Intercambio de Claves la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el Intercambio de Claves que será utilizado como lo define el DOI.

2. Determinar el uso del campo de Datos de Intercambio de Claves como lo define el DOI.

3. Construir una carga de Intercambio de Claves.

4. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando una carga de Intercambio de Claves es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Intercambio de Claves es soportado. Si la determinación del intercambio de claves falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, INFORMACIÓN DE CLAVE NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, INFORMACIÓN DE CLAVE NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.8 Procesamiento de la Carga de Identificación

Cuando se crea una Carga de Identificación, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar la información de Identificación que será usada según lo definido por el DOI (y posiblemente la situación).

2. Determinar el uso del campo de Datos de Identificación según lo definido por el DOI.

3. Construir una carga de Identificación.

4. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando una carga de Identificación es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Tipo de Identificación es soportado. Esto puede estar basado en el DOI y la Situación. Si la determinación de Identificación falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, INFORMACIÓN DEL IDENTIFICADOR NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, INFORMACIÓN DEL IDENTIFICADOR NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.9 Procesamiento de la Carga de Certificado

Cuando se crea una Carga de Certificado, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar la Codificación de Certificación que será usada. Esto puede estar especificado por el DOI.

2. Asegurar la existencia del certificado formateado según lo definido en la Codificación del Certificado.

3. Construir una carga de Certificado.

4. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando una Carga de Certificado es recibida, la entidad receptora (iniciador o respondedor) debe hacer lo siguiente:

1. Determinar si la Codificación de Certificación es soportada. Si la Codificación de Certificación no es soportada, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, TIPO CERTIFICACIÓN NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, CODIFICACIÓN DE CERTIFICACIÓN NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Procesar el campo Datos del Certificados. Si los Datos Certificados no son válidos o está formateado inapropiadamente, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, CERTIFICADO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, CERTIFICADO NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.10 Procesamiento de la Carga de Solicitud de Certificado

Cuando una Carga de Solicitud de Certificado se crea, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el tipo de Codificación de Certificación que será solicitado. Esto puede estar especificado por el DOI.

2. Determinar el nombre de una Autoridad de Certificación (CA) aceptable a la cual se le solicitará (si es aplicable).

3. Construir una carga de Solicitud de Certificado.

4. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando la Carga de Solicitud de Certificado es recibida, la entidad receptora (iniciador o respondedor) debe hacer lo siguiente:

1. Determinar si la Codificación de Certificación es soportada. Si la Codificación de Certificación no es válida, la carga es descartada y

las siguientes acciones son tomadas:

(a) El evento, TIPO DE CERTIFICACIÓN NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, CODIFICACIÓN DE CERTIFICADO NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

Si la Codificación de Certificación no es soportada, la carga es descartada y las siguientes acciones son tomadas:

(a) El evento, TIPO DE CERTIFICACIÓN NO SOPORTADO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

(b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, TIPO DE CERTIFICACIÓN NO SOPORTADO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Determinar si la Autoridad de Certificación es soportada por la Codificación de Certificación especificada. Si la Autoridad de Certificación no es válida o es formateada inapropiada mente, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, AUTORIDAD DE CERTIFICACIÓN NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, AUTORIDAD DE CERTIFICACIÓN NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

3. Procesar la Solicitud de Certificación. Si un Tipo de Certificado solicitado con una Autoridad de Certificación especifica no está disponible, la carga es descartada y las siguientes acciones son tomadas:

   (a) El evento, CERTIFICACIÓN NO DISPONIBLE, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, CERTIFICACIÓN NO DISPONIBLE, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.11 Procesamiento de la Carga Hash

Cuando una Carga Hash es creada la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar la función Hash que será usada según lo definido por la negociación de la SA.

2. Determinar el uso del campo Datos Hash según lo definido por el DOI.

3. Construir una carga Hash.

4. Trasmitir el mensaje a la entidad receptora según lo descripto en la Sección 5.1.

Cuando una carga Hash es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Hash es soportado. Si la determinación del hash falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, INFORMACIÓN DE HASH NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, INFORMACIÓN DE HASH NO VÁLIDO, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Realizar la función de Hash como se explica en el DOI y/o en los capítulos de los protocolos de Intercambio de Claves. Si la función Hash falla, el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, VALOR DE HASH NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, ERROR EN LA AUTENTIFICACIÓN, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.12 Procesamiento de la Carga de la Firma

Cuando se crea una Carga de Firma, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar la función de la Firma que será usada según lo definido por la negociación de la SA.

2. Determinar el uso del campo Datos de la Firma según lo definido por el DOI.

3. Construir la carga de la Firma.

4. Trasmitir el mensaje a la entidad receptora como se describió en la Sección 5.1.

Cuando una carga de Firma es recibida la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si la Firma es soportada. Si la determinación de la Firma falla el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, INFORMACIÓN DE LA FIRMA NO VÁLIDA, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, FIRMA NO VÁLIDA, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

2. Realizar la Función de la firma conforme al DOI y/o los capítulos de Intercambio de Claves. Si la función de la Firma falla el mensaje es descartado y las siguientes acciones son tomadas:

   (a) El evento, VALOR DE LA FIRMA NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

   (b) Un Intercambio Informativo con una carga de Notificación conteniendo el tipo de mensaje, ERROR EN LA AUTENTIFICACIÓN, PUEDE ser enviado a la entidad transmisora. Esta acción es dictaminada por un sistema de políticas de seguridad.

5.13 Procesamiento de la Carga Nonce

Cuando se crea una carga Nonce, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Crear un valor aleatorio único que será usado como un nonce.

2. Construir una carga Nonce.

3. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Cuando una carga Nonce es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. No hay procesamientos específicos para el procesamiento de cargas nonces. Los procedimientos están definidos por el tipo de intercambio (y posiblemente por el DOI y las descripciones del intercambio de claves).

5.14 Procesamiento de la Carga de Notificación

Durante las comunicaciones es posible que ocurran errores. El Intercambio Informativo con una Carga de Notificación proporciona un método controlado para informar a una entidad que errores se han producido durante el procesamiento. Se RECOMIENDA que las Cargas de Notificación sean enviadas en un Intercambio Informativo separado en lugar de anexarlas a una Carga de Notificación de un intercambio existente.

Cuando se crea una Carga de Notificación, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el DOI para esta Notificación.

2. Determinar el Identificador del Protocolo para esta Notificación.

3. Determinar el tamaño del SPI basándose en el campo Identificador de Protocolo. Este campo es necesario porque diferentes protocolos de seguridad tienen diferentes tamaños de SPI. Por ejemplo, ISAKMP combina el par de cookies del Iniciador y el Respondedor (16 octetos) como un SPI, mientras que ESP y AH tienen SPIs de 4 octetos.

4. Determinar el Tipo de Mensaje de Notificación basándose en el error o en el estado del mensaje deseado.

5. Determinar el SPI que se asocia con esta notificación.

6. Determinar si los Datos de Notificación adicional serán incluidos. Esto es información adicional especificada por el DOI.

7. Construir una carga de Notificación.

8. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Debido a que un Intercambio Informativo con una carga de Notificación es un mensaje unidireccional, no se realizarán retransmisiones. La política de seguridad local dictaminará los procedimientos a seguir. Sin embargo, se RECOMIENDA que un evento de, ERROR DE CARGA DE NOTIFICACIÓN, sea registrado en un apropiado sistema de auditoría de archivos por la entidad receptora.

Si un Intercambio Informativo ocurre antes del intercambio de material clave durante la Fase 1 de la negociación de ISAKMP, no se le proporcionará protección al Intercambio Informativo. Una vez que el material clave ha sido intercambiado o la SA ISAKMP ha sido establecida, el Intercambio Informativo DEBE ser trasmitido bajo la protección proporcionada por el material clave o la SA ISAKMP.

Cuando una carga de Notificación es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar si el Intercambio Informativo tiene alguna protección aplicada por medio de la comprobación del Bit de Encriptación y del Bit de Solo Autentificación en la cabecera de ISAKMP. Si el Bit de Encriptación está fijado, es decir el Intercambio Informativo es encriptado, el mensaje DEBE ser desencriptado usando la (en proceso o ya establecida) SA ISAKMP. Una vez que la desencriptación es completada el proceso puede continuar como se describe abajo. Si el Bit de Solo Autentificación esta fijado, el mensaje DEBE ser autentificado usando la (en proceso o ya establecida) SA ISAKMP. Una vez que la autentificación es completada, el proceso puede continuar como se describe debajo. Si el Intercambio Informativo no es encriptado o autentificado, el procesamiento de la carga puede continuar como se describe debajo.

2. Determinar si el DOI es soportado. Si la determinación del DOI falla, la carga es descartada y la siguiente acción es tomada:

   (a) El evento, DOI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

3. Determinar si el Identificador de Protocolo es soportado. Si la determinación del Identificador de Protocolo falla, la carga es descartada y la siguiente acción es tomada:

   (a) El evento, IDENTIFICADOR DE PROTOCOLO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

4. Determinar si el SPI es válido. Si el SPI no es válido, la carga es descartada y la siguiente acción es tomada:

   (a) El evento, SPI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

5. Determinar si el Tipo de Mensaje de Notificación es válido. Si el Tipo

de Mensaje de Notificación no es válido, la carga es descartada y la siguiente acción es tomada:

(a) El evento, TIPO DE MENSAJE NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

6. Procesar la carga de Notificación, incluyendo los Datos de Notificación adicionales y tomar la acción apropiada, de acuerdo con la política de seguridad local.

5.15 Procesamiento de la Carga de Cancelación

Durante las comunicaciones es posible que hosts puedan estar comprometidos o que la información pueda ser interceptada durante la transmisión. La determinación de que esto ha ocurrido no es una tarea fácil y esta fuera del alcance de este libro. Sin embargo si se descubre que las trasmisiones son comprometidas, es necesario establecer una nueva SA y cancelar la actual.

El Intercambio Informativo con una Carga de Cancelación proporciona un método controlado de informar a una entidad usuaria de que la entidad transmisora a cancelado la/s SA/s. La cancelación de SA siempre DEBE ser realizada bajo la protección de una SA ISAKMP. La entidad receptora DEBERÍA limpiar la base de datos de SA local. Sin embargo, bajo el recibo de un mensaje de Cancelación las SAs enumeradas en el campo SPI de la carga de Cancelación no pueden ser usados por la entidad transmisora. El procedimiento del establecimiento de SA debe ser invocado para el restablecimiento de comunicaciones seguras.

Cuando se crea una Carga de Cancelación, la entidad transmisora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Determinar el DOI para esta cancelación.

2. Determinar el Identificador de Protocolo para esta cancelación.

3. Determinar el tamaño del SPI basándose en el campo Identificador de Protocolo. Este campo es necesario porque, diferentes protocolos de seguridad tienen diferentes tamaños de SPI. Por ejemplo, ISAKMP combina el par de cookies del iniciador y del respondedor (16 octetos) como un SPI, mientras que ESP y AH tienen SPIs de 4 octetos.

4. Determinar el número de SPIs que serán cancelados para este protocolo.

5. Determinar el o los SPI(s) que serán asociados con esta cancelación.

6. Construir una carga de Cancelación.

7. Trasmitir el mensaje a la entidad receptora como se describe en la Sección 5.1.

Debido a que un Intercambio Informativo con una carga de cancelación es un mensaje unidireccional, no se realizan retransmisiones. La política de seguridad local determinará el procedimiento a seguir. Sin embargo, se RECOMIENDA que un evento de, ERROR DE CARGA DE CANCELACIÓN sea registrado en un apropiado sistema de auditoría de archivos por la entidad receptora.

Como se describió anteriormente, un Intercambio Informativo con una carga de Cancelación DEBE ser trasmitido bajo la protección proporcionada por una

SA ISAKMP.

Cuando una carga de Cancelación es recibida, la entidad receptora (iniciador o respondedor) DEBE hacer lo siguiente:

1. Debido a que el Intercambio Informativo está protegido por un cierto servicio de seguridad (por ejemplo, autentificación para una SA de Solo Autentificación, o encriptación para otros intercambios), el mensaje DEBE tener estos servicios de seguridad aplicados usando la SA ISAKMP. Una vez que el procesamiento del servicio de seguridad es completado el procesamiento puede continuar como se describe debajo. Cualquier error que ocurra durante el procesamiento del servicio de seguridad será evidente al controlar la información en la carga de Cancelación. La política de seguridad local DEBERÍA dictaminar cualquier acción a seguir como resultado de errores en el procesamiento del servicio de seguridad.

2. Determinar si el DOI es soportado. Si la determinación del DOI falla, la carga es descartada y las siguiente acción es tomada:

   (a) El evento, DOI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

3. Determinar si el Identificador de Protocolo es soportado. Si la determinación del Identificador de Protocolo falla, la carga es descartada y la siguiente acción es tomada:

   (a) El evento, IDENTIFICADOR DE PROTOCOLO NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

4. Determinar si el SPI es válido para cada SPI incluido en la carga de Cancelación. Para cada SPI que no sea válido, la siguiente acción es tomada:

   (a) El evento, SPI NO VÁLIDO, PUEDE ser registrado en un apropiado sistema de auditoría de archivos.

5. Procesar la carga de Cancelación y tomar una acción apropiada, de acuerdo con la política de seguridad local. Como se describió anteriormente, una acción apropiada DEBERÍA incluir la limpieza de la base de datos de la SA local.

6. **Atributos de una Asociación de Seguridad ISAKMP**

6.1 Antecedentes/Fundamentos

Como se explicó en secciones previas, ISAKMP está diseñado para proporcionar un marco flexible y extensible para el establecimiento y la administración de SAs y claves criptográficas. Este marco proporcionado por ISAKMP consiste de definiciones de carga y cabeceras, y de tipos de intercambio para guiar el mensaje y los intercambios de carga y las pautas generales de procesamiento. ISAKMP no define los mecanismos que serán usados para establecer y administrar las SAs y claves criptográficas en un modo autentificado y confidencial. La definición de estos mecanismos y sus aplicaciones están bajo la incumbencia de los Dominios de Interpretación (DOIs) individuales.

Esta sección describe los valores de ISAKMP para el DOI de Seguridad IP en Internet, Protocolos de seguridad soportados, y valores de identificación

para las negociaciones de la Fase 1 de ISAKMP. El DOI de Seguridad IP en Internet es OBLIGATORIO para las implementaciones de seguridad IP (descripto en el Capítulo 8). El Capítulo 9 y el Capítulo 10 describen, en detalle, los mecanismos y sus aplicaciones para el establecimiento y la administración de SAs y claves criptográficas para la seguridad IP.

6.2 Valor Asignado al DOI de IPsec

Como se describe en el Capítulo 8, el número asignado al DOI de Seguridad IP en Internet (DOI de IPsec) es uno (1).

6.3 Protocolos de Seguridad Soportados

Los valores para los protocolos de seguridad soportados son especificados en los más recientes "números asignados" [STD-2]. En la siguiente tabla están los valores para los protocolos de seguridad soportados por ISAKMP para el DOI de IPsec.

Protocolo	Valor Asignado
Reservado	0
ISAKMP	1

Todos los DOIs DEBEN reservar a ISAKMP el identificador de protocolo uno (1). Todos los otros protocolos de seguridad dentro del DOI serán enumerados consecuentemente.

Los valores del protocolo de seguridad de 2 hasta 15359 están reservados por la IANA para uso futuro. Los valores de 15360 hasta 16383 están permanentemente reservados para el uso privado entre implementaciones mutuamente acordadas. Tales valores de uso privado son poco probables de ser ínter operables a través de diferentes implementaciones.

6.4 Valores del Tipo de Identificación de ISAKMP

La tabla siguiente enumera los valores asignados al campo Tipo de Identificación encontrados en la carga de identificación durante un intercambio genérico de Fase 1.

Tipo de Identificador	Valor
Identificador de dirección IPv4	0
Identificador de dirección de subred IPv4	1
Identificador de dirección IPv6	2
Identificador de dirección de subred IPv6	3

6.4.1 Identificador de Dirección IPv4

El tipo de Identificador de dirección IPv4 especifica un valor de cuatro (4) octetos para la dirección de IPv4.

6.4.2 Identificador de Dirección de Subred IPv4

El tipo de Identificador de dirección de subred para IPv4, especifica una serie de direcciones para IPv4, representado por dos valores de cuatro (4) octetos. El primer valor es una dirección IPv4, el segundo valor es una máscara de red IPv4. Note que los unos en la máscara de red indican que el bit correspondiente en la dirección es fijo, mientras que los ceros indica un bit "comodín".

6.4.3 Identificador de Dirección IPv6

El tipo de identificador de dirección IPv6 especifica un valor de dieciséis (16) octetos para la dirección IPv6.

6.4.4 Identificador de Dirección de Subred IPv6

El tipo de Identificador de dirección de subred de IPv6, especifica una serie de direcciones para IPv6, representados por dos valores de dieciséis (16) octetos. El primer valor es una dirección IPv6 el segundo es una máscara de red IPv6. Note que los unos en la mascara de red indican que el bit correspondiente en la dirección es fijo, mientras que los ceros indican un bit "comodín".

7. **Definición de un Nuevo Dominio de Interpretación**

El Dominio de Interpretación (DOI) es un campo de 32-bits que identifica el dominio bajo el cual la negociación de la SA se está llevando a cabo. El DOI de Internet puede ser suficiente para resolver los requerimientos de seguridad de una gran parte de la comunidad de Internet. Sin embargo algunos grupos pueden tener la necesidad de reformar algunos aspectos del DOI, tal vez agregar un conjunto de algoritmos criptográficos diferentes, o tal vez por que quieran tomar decisiones relevantes a la seguridad basados en algo más que un identificador de host o un identificador de usuario. También, un grupo particular puede tener la necesidad de un nuevo tipo de intercambio, como por ejemplo para soportar la administración de claves para grupos multicast.

Esta sección discute los lineamientos para la definición de un nuevo DOI. Una completa especificación del DOI de Internet puede encontrarse en el Capítulo 8.

Definir un nuevo DOI es probable que sea un proceso que lleve mucho tiempo. En lo posible, se recomienda que el diseñador comience con un DOI existente y que modifique solamente las partes que son inaceptables.

Si un diseñador elige comenzar de cero, DEBE ser definido:

- Una "situación": El conjunto de información que será utilizado para determinar los servicios de seguridad requeridos.
- El conjunto de políticas de seguridad que debe ser soportado.
- Un esquema para nombrar información de seguridad relevante, incluyendo algoritmos de encriptación, algoritmos de intercambios de claves etc.
- Una sintaxis para la especificación de los servicios de seguridad propuestos, atributos, y autoridades de certificación.
- El formato específico para los contenidos de varias cargas.
- Tipos de intercambios adicionales, si son requeridos.

7.1 Situación

La situación es la base para decidir como proteger un canal de comunicaciones. Debe contener todos los datos que serán usados para

determinar los tipos y las fuerzas de protección aplicadas a una SA. Por ejemplo el departamento de defensa de USA probablemente use algoritmos no publicados y tendría atributos adicionales especiales que negociar. Estos atributos de seguridad adicionales estarían incluidos en la situación.

7.2 Políticas de Seguridad

Las políticas de seguridad definen la forma en que varios tipos de información deben estar clasificados y protegidos. El DOI debe definir el conjunto de políticas de seguridad soportado, por que ambas partes en una negociación deben confiar en que la otra parte comprende la situación, y protegerá la información apropiadamente, en tránsito y almacenada. En un ambiente corporativo, por ejemplo, ambas partes en una negociación deben acordar el significado del término "información privada" antes de que puedan negociar como protegerla.

Note que incluyendo las políticas de seguridad requeridas en el DOI esto solamente especifica que los hosts participantes entiendan e implementen aquellas políticas en un contexto de un sistema global.

7.3 Esquemas de Nombramiento

Todos los DOI deben definir un modo consistente de nombrar algoritmos criptográficos, autoridades de certificación, etc. Esto puede ser usualmente realizado utilizando las convenciones de nombramiento de la IANA, tal vez con algunas extensiones privadas.

7.4 Sintaxis para la Especificación de Servicios de Seguridad

Además de simplificar la especificación de cómo nombrar entidades, el DOI también debe especificar el formato de la propuesta para proteger el tráfico bajo una determinada situación.

7.5 Especificación de la Carga

El DOI debe especificar el formato para cada uno de los tipos de carga. Para varios tipos de carga, ISAKMP a incluido campos que tendrían que estar presentes a través de todos los DOI (tales como, autoridad de certificación en la carga de certificado, o un identificador de intercambio de claves en la carga de intercambio de claves).

7.6 Definición de Nuevos Tipos de Intercambio

Si los tipos básicos de intercambio son inadecuados para resolver los requisitos dentro de un DOI, un diseñador puede definir hasta 13 tipos de intercambio extras por DOI. El diseñador crea un nuevo tipo de intercambio eligiendo un valor no usado de tipo de intercambio, y definiendo una secuencia de mensajes compuesta de encadenamientos de tipos de carga de ISAKMP.

Note que cualquiera de los nuevos tipos de intercambio debe ser rigurosamente analizado para evitar vulnerabilidades. Puesto que esto es una tarea costosa e imprecisa, un nuevo tipo de intercambio debe ser creado si es absolutamente necesario.

8. **Consideraciones de Seguridad**

Las técnicas de análisis criptográficos están mejorando día a día. La mejora constante en el procesamiento hace que ataques criptográficos de

cálculo informático sean más realistas. Nuevos algoritmos criptográficos y técnicas de generación de claves públicas son desarrollados constantemente. Nuevos servicios de seguridad y mecanismos de seguridad son desarrollados a paso acelerado. Un método constante para elegir servicios de seguridad y mecanismos de seguridad y para intercambiar atributos requeridos por los mecanismos es importante para la seguridad de la compleja estructura de Internet. Sin embargo, un sistema que se sierra en si mismo en un único algoritmo criptográfico, técnica de intercambio de claves o mecanismos de seguridad será cada día más vulnerable.

UDP es un Protocolo no confiable y por lo tanto su uso en ISAKMP introduce un gran número de consideraciones de seguridad. Ya que UDP no es confiable, pero un protocolo de administración de claves debe ser confiable, la confiabilidad se construye dentro de ISAKMP. Mientras que ISAKMP utiliza UDP como su mecanismo de trasporte, ISAKMP no confía en la información de UDP (por ejemplo la suma de comprobación o longitud) para su procesamiento.

Otro tema que debe ser considerado en el desarrollo de ISAKMP es el efecto de firewalls en el protocolo. Muchos de los firewalls filtran los paquetes UDP salientes asiendo que la dependencia en UDP sea cuestionable en ciertos entornos.

Una vez que la clave de sesión privada es creada, debe ser almacenada en forma segura. No proteger adecuadamente las claves privadas para accesos internos o externos al sistema anula totalmente cualquier protección proporcionada por los servicios de seguridad IP.

9. **Conclusiones**

ISAKMP está diseñado para proporcionar gestión de claves y negociación de SA para varios protocolos de seguridad. ISAKMP es un protocolo bien diseñado que apunta a la Internet del Futuro. El crecimiento masivo de Internet conducirá a una gran diversidad de la utilización de la red, comunicaciones, requerimientos de seguridad, y mecanismos de seguridad. ISAKMP contiene todas las características que serán necesarias para ese ambiente de comunicaciones de red dinámico y amplio.

La característica de SA ISAKMP junto con la autentificación y el establecimiento de claves proporcionan la seguridad y flexibilidad que serán necesarias para la diversidad y crecimiento futuro. Esta diversidad de seguridad de múltiples técnicas de intercambio de claves, algoritmos de encriptación, mecanismos de autentificación, servicios de seguridad, y atributos de seguridad permitirán a los usuarios seleccionar la seguridad apropiada para sus redes, comunicaciones, y necesidades de seguridad. El uso de las características de SA permite especificar y negociar requerimientos de seguridad con otros usuarios. Un beneficio adicional de soportar múltiples técnicas en un único protocolo es que a medida que nuevas técnicas son desarrolladas, estas pueden ser fácilmente agregadas al protocolo, esto proporciona un camino para el crecimiento de los servicios de seguridad de Internet. ISAKMP soporta SAs definidas públicas y privadamente, haciéndolas ideal para el gobierno, comercio y comunicaciones privadas.

ISAKMP proporciona la capacidad de establecer SAs para múltiples protocolos de seguridad y aplicaciones. Estos protocolos o aplicaciones pueden estar o no orientados a sesiones. Teniendo un protocolo para el establecimiento de SAs que soporte múltiples protocolos de seguridad elimina la necesidad de múltiples, autentificaciones similares, intercambios de claves y protocolos

de establecimientos de SAs cuando más de un Protocolo de seguridad está en uso o es requerido. Así como IP proporciona la capa de red común para la Internet un protocolo de establecimiento de seguridad común es necesario para que la seguridad se convierta en una realidad en Internet. ISAKMP proporciona la base común que permite a todos los otros protocolos de seguridad ínter operar.

ISAKMP sigue buenos principios de diseño de seguridad. No está vinculado a otros protocolos de trasporte inseguros, por lo tanto no es vulnerable o debilitado por ataques a otros protocolos. También, cuando más protocolos de trasporte seguros son desarrollados, ISAKMP puede fácilmente emigrar con ellos. ISAKMP también proporciona protección contra ataques vinculados a protocolos. Esta protección proporciona seguridad de que las SAs y el establecimiento de claves están con la parte deseada y no con un atacante.

ISAKMP también sigue buenos principios de diseño de protocolo. La información específica del protocolo solo está en la cabecera del protocolo, siguiendo los principios de diseño de IPv6. Los datos trasportados por el protocolo están separados dentro de cargas funcionales. A medida que Internet crece y evoluciona, nuevas cargas para soportar nuevas funcionalidades de seguridad pueden ser agregadas sin modificar el protocolo entero.

Capítulo 8

El DOI de Seguridad IP en Internet para ISAKMP

1. **Introducción**

Dentro de ISAKMP, un Dominio de Interpretación es usado para relacionar un grupo de protocolos usando ISAKMP para negociar SAs. Los protocolos de seguridad comparten un DOI de protocolo de seguridad elegido y transformaciones criptográficas a partir de un espacio de nombramiento y de un identificador de protocolo de intercambio de claves común. También comparten la interpretación del DOI específico de contenido de los datos de la carga, incluyendo la SA y el Identificador de carga.

En general, ISAKMP propone los siguientes requerimientos para la definición de un DOI:

- Definir el esquema de nombramiento para los identificadores de protocolo para el DOI específico.
- Definir la interpretación para el campo Situación.
- Definir el conjunto de políticas de seguridad aplicables.
- Definir la sintaxis para los Atributos de las SAs (Fase 2) para el DOI específico.
- Definir la sintaxis para los contenidos de las cargas para el DOI específico.
- Definir tipos de Intercambios de Claves adicionales, si es necesario.
- Definir tipos de Mensajes de Notificación adicionales, si es necesario.

El resto del capítulo detalla las ejemplificaciones de estos requerimientos para usar el Protocolo de Seguridad IP (IPsec) para proporcionar autentificación, integrabilidad, y/o confidencialidad para los paquetes IP enviados entre sistemas host y/o firewalls.

2. **Esquema de Nombramiento IPsec**

Dentro de ISAKMP, todos los DOI beben estar registrados por la IANA en el RFC "Números Asignados" [STD-2]. El Número Asignado por la IANA para el DOI de Seguridad IP (DOI de IPsec) es uno (1). Dentro del DOI de IPsec, todos los identificadores DEBEN estar registrados por la IANA bajo el DOI de IPsec. A menos que se mencione lo contrario, todas las tablas de este capítulo hacen referencia a los Números Asignados por la IANA para el DOI de IPsec. Vea la Sección 10 para información adicional relacionada con el registro de la IANA para el DOI de IPsec.

Nota: Todos los valores binarios compuesto de varios octetos se almacenan en orden de byte de red.

3. **Dominio de Interpretación en la Carga SA de ISAKMP**

El DOI es un campo de 32-bits en la carga SA de ISAKMP, que identifica el dominio bajo el cual la negociación de la SA se esta llevando a cabo y es usado para interpretar las cargas de ISAKMP. Un DOI define los formatos de cargas, tipos de intercambio, y convenciones para nombrar información relevante a la seguridad tales como políticas de seguridad o algoritmos criptográficos y modos.

Dominio de Interpretación (DOI)	Valor	Referencia
ISAKMP	0	[ISAKMP]
IPSEC	1	[DOIPsec]
GDOI	2	[GDOI]

Un valor de DOI de cero (0) durante el intercambio de la Fase 1 específica una SA ISAKMP genérica la cual puede ser usada por cualquier protocolo durante el intercambio de la Fase 2. Un valor de DOI de 1 es asignado al DOI IPsec. El valor de 2 es asignado al DOI de ISAMKP para la gestión de claves de grupos multicast, denominado GDOI (Group Domain of Interpretation).

4. **Tipos De Carga Siguiente en la Cabecera de ISAKMP**

El Tipo de Carga Siguiente en la Cabecera de ISAKMP es una valor de 8 bits que indica el tipo de carga siguiente en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá el valor cero. Este campo proporciona la capacidad de "encadenamiento". El formato de cada carga (hasta la carga de valor 13, Carga de Identificación del Vendedor), se describió desde la Sección 3.4 hasta la Sección 3.16 del capítulo anterior, como así también en la Sección 5 de ese mismo capítulo describió el procesamiento para las cargas hasta la carga de valor 13 (Carga de Identificación del Vendedor).

Tipos de Carga Siguiente	Notación	Valor	Referencia
Ninguna		0	[ISAKMP]
Carga de Asociación de Seguridad	SA	1	[ISAKMP]
Carga de la Propuesta	P	2	[ISAKMP]
Carga de Transformación	T	3	[ISAKMP]
Carga de Intercambio de Claves	KE	4	[ISAKMP]
Carga de Identificación	ID	5	[ISAKMP]
Carga de Certificado	CERT	6	[ISAKMP]
Carga de Solicitud de Certificado	CR	7	[ISAKMP]
Carga de Hash	HASH	8	[ISAKMP]
Carga de Firma	SIG	9	[ISAKMP]
Carga de Nonce	NONCE	10	[ISAKMP]
Carga de Notificación	N	11	[ISAKMP]
Carga de Cancelación	D	12	[ISAKMP]
Carga de Identificación del Vendedor	VID	13	[ISAKMP]
Reservado, no debe ser usado		14	[IANA-2]
Carga de SA para la Clave de Encritación de Claves (KEK)	SAK	15	[GDOI]
Carga de SA para la Clave de Encritación de Tráfico (TEK)	SAT	16	[GDOI]
Carga de Descarga de Claves	KD	17	[GDOI]
Carga del Número de Secuencia	SEQ	18	[GDOI]
Carga de Posesión de la Prueba	POP	19	[GDOI]
Carga de Descubrimiento de NAT	NAT-D	20	[NATinIKE]
Carga de Dirección Original del NAT	NAT-OA	21	[NATinIKE]

Los valores de 22 a 127 están reservados por la IAMA para uso futuro. Los valores de 128 a 255 esta reservados para usarse en forma privada entre sistemas.

5. **Definición de la Situación para IPsec**

La Definición de Situación es un bitmask el cual representa el ambiente bajo el cual la propuesta SA IPsec y la negociación se están llevando a cabo. Dentro de ISAKMP, la Situación proporciona información que puede ser usada por el respondedor para elaborar una determinada política sobre como procesar la petición de la SA entrante. Para el DOI de IPsec, el campo

Situación es un bitmask de cuatro (4) octetos (32 bits) con los siguientes valores:

Situación	Valor
Situación de solo Identificación (SIT_IDENTITY_ONLY)	0x01
Situación secreta (SIT_SECRECY)	0x02
Situación integridad (SIT_INTEGRITY)	0x04

Los dos bit de orden superior están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevas situaciones deben estar acompañados por un RFC el cual describa la interpretación para los bit asociados.

5.1 Situación de Solo Identificación

El tipo, Situación de solo Identificación (SIT_IDENTITY_ONLY), especifica que la SA será identificada por la información de la identidad de origen presente en una Carga de Identificación (ver la Sección 3.8 del Capítulo 7) asociada. Vea la Sección 9.2 para una completa descripción de los diversos tipos de Identificadores. Toda implementación DOI de IPsec DEBE soportar SIT_IDENTITY_ONLY para incluir una Carga de Identificación en al menos uno de los intercambios Oakley de la Fase 1 (ver la Sección 4 del Capítulo 10) y DEBE abortar cualquier intento de establecer asociaciones que no incluyan una Carga de Identificación.

Si un iniciador no soporta SIT_SECRECY ni SIT_INTEGRITY, la situación consiste solo de la situación bitmap de 4 octetos y no incluye el campo Identificación del Identificador de Dominio (Figura 1, Sección 9.1) o ninguna información de identificación posterior. Por el contrario si el iniciador soporta SIT_SECRECY o SIT_INTEGRITY, el campo Identificación del Identificador de Dominio DEBE ser incluido en la carga de la Situación.

5.2 Situación Secreto

El tipo, Situación Secreto (SIT_SECRECY), especifica que la SA es negociada en un ambiente que requiere identificación de secreto. Si SIT_SECRECY está presente en la situación bitmap, el campo Situación estará seguido por datos de longitud variable que incluyen un nivel de sensibilidad y el sector bitmask. Ver Sección 9.1 para una completa descripción del formato de la Carga SA.

Si un iniciador no soporta SIT_SECRECY, SIT_SECRECY NO DEBE ser colocado en la Situación bitmap y no debe ser incluido el nivel de secreto o categorías bitmask.

Si un respondedor no soporta SIT_SECRECY, una Carga de Notificación conteniendo, SITUACIÓN NO SOPORTADA, DEBERÍA ser enviado y el establecimiento de la SA DEBE ser abortado.

5.3 Situación Integridad

El tipo, Situación Integridad (SIT_INTEGRITY), especifica que la SA es negociada en un ambiente que requiere identificación de integridad. Si SIT_INTEGRITY está presente en la Situación bitmap, el campo Situación estará seguido por datos de longitud variable que incluyen un nivel de integridad y el sector bitmask. Si SIT_SECRECY también es usado en la asociación, la información de integridad estará seguida de los datos del nivel de secreto (de longitud variable) y de las categorías. Ver Sección

9.1 para una completa descripción del formato de la Carga SA.

Si un iniciador no soporta SIT_INTEGRITY, SIT_INTEGRITY NO DEBE ser colocado en la Situación bitmap y no debe ser incluido el nivel de integridad o categorías bitmask.

Si un respondedor no soporta SIT_INTEGRITY, una Carga de Notificación conteniendo, SITUACIÓN NO SOPORTADA, DEBERÍA ser enviado y el establecimiento de la SA DEBE ser abortado.

6. **Requisitos para la Política de Seguridad de IPsec**

El DOI de IPsec no impone requisitos específicos para la política de seguridad en ninguna implementación. Los asuntos de políticas en sistemas host están fuera del alcance de este libro.

Sin embargo, las subsiguientes secciones tratan algunos de los aspectos que deben ser considerados cuando se diseña una implementación DOI de IPsec en host.

6.1 Cuestiones Sobre la Gestión de Claves

Se espera que muchos sistemas elijan implementar ISAKMP esforzándose por proporcionar un DOI protegido para un conjunto de demonios de administración de claves de IKE. En modo protegido, en sistemas operativos multiusuario, estos demonios de administración de claves, probablemente existan como procesos con privilegios separados.

En tales ambientes, puede ser conveniente que una API (Interfaz de Programa de Aplicación) realice la introducción del material clave dentro del kernel TCP/IP. La arquitectura de seguridad IP no tiene ningún requerimiento para la estructura o flujo entre un kernel TCP/IP host y estos proveedores de administración de claves.

6.2 Cuestiones Sobre las Claves Estáticas

Los sistemas host que implementen claves estáticas, para uso directo de IPsec, o para propósitos de autentificación (ver la Sección 4.1.4 del Capítulo 10), deberían tomar medidas para proteger el material clave estático cuando no se encuentre dentro de un área de memoria protegida o cuando lo esté usando el kernel TCP/IP.

Por ejemplo, en un ordenador portátil, puede que se prefiera guardar las claves estáticas en un depósito configurable, es decir, encriptadas bajo una contraseña privada.

Dependiendo del sistema operativo y del software instalado, puede que no sea posible proteger las claves estáticas una vez que estas están dentro del kernel TCP/IP, sin embargo no debería ser fácilmente recuperable encender inicialmente el sistema sin tener que satisfacer algunos requisitos adicionales de autentificación.

6.3 Cuestiones Sobre la Política en Host

No es realista asumir que la transmisión IPsec ocurrirá sin configuración. Los sistemas host deben estar dispuestos a implementar listas de políticas flexibles que describan que sistemas desean comunicarse en modo seguro y cuales de ellos requieren comunicaciones en modo seguro.

Una aproximación probable puede ser una lista estática de direcciones IP, máscaras de red, y una bandera o banderas con requisitos de seguridad.

Implementaciones más flexibles pueden consistir en una lista de nombres de DNS comodines (por ejemplo, '*.foo.bar'), una máscara de entrada/salida, y direcciones de firewall opcionales. Los nombres de DNS comodines podrían ser usados para hacer corresponder las direcciones IP entrantes o salientes, las máscaras IP podrían ser usadas para determinar si la seguridad será aplicada o no en esa determinada dirección y las direcciones de firewall opcionales podrían ser usadas para indicar si el modo túnel es o no necesario para comunicarse con el sistema de destino aunque exista un firewall intermedio.

6.4 Administración de Certificados

Sistemas host que implementen un esquema de autentificación de certificados necesitarán un mecanismo para obtener y administrar bases de datos de certificados.

Los DNS seguros son uno de los mecanismos de distribución de certificados, no obstante la disponibilidad permanente de zonas con DNS seguros, a corto plazo, es improbable por muchas razones. Lo que es mucho más probable es que los host necesitarán de un mecanismo para importar los certificados que adquieren a través de mecanismos seguros, mecanismos out-of-band (fuera de banda), así como también una capacidad para exportar sus propios certificados para que lo usen otros sistemas.

Sin embargo, la administración de certificados en forma manual no debería ser realizada para no imposibilitar la capacidad de introducir mecanismos dinámicos de descubrimiento de certificado y/o protocolos cuando sea posible.

7. **Números Asignados a IPsec**

7.1 Identificador de Protocolo de Seguridad para IPsec

El Identificador de Protocolo de Seguridad es un valor de 8 bit el cual identifica al conjunto de protocolos de seguridad que se esta negociado. La sintaxis de la propuesta de ISAKMP fue diseñada específicamente para poder negociar simultanea, múltiples Fases 2 de protocolos de seguridad dentro de una única negociación. Como consecuencia, la lista de conjuntos de protocolos de abajo forma el conjunto de protocolos que pueden ser negociados al mismo tiempo. Es decisión de la política del host qué conjuntos de protocolos pueden ser negociados conjuntamente.

La tabla siguiente lista los valores para los Identificadores de Protocolo de Seguridad referenciados en la Carga de la Propuesta de ISAKMP para el DOI de IPsec.

Identificador de Protocolo	Valor
Reservado	0
Protocolo ISAKMP (PROTO_IPCOMP)	1
Protocolo AH IPsec (PROTO_IPSEC_AH)	2
Protocolo ESP IPsec (PROTO_IPSEC_ESP)	3
Protocolo de Compresión IP (PROTO_IPCOMP)	4

Los valores de 249 a 255 están reservados para usarse en forma privada

entre sistemas. Pedidos de asignaciones de nuevos identificadores de protocolo de seguridad deben estar acompañados por un RFC el cual describa los requisitos del protocolo de seguridad. [AH] y [ESP] son ejemplos de documentos de protocolos de seguridad.

7.1.1 Protocolo ISAKMP

El tipo, Protocolo ISAKMP (PROTO_ISAKMP), especifica que se requiere protección de mensajes durante la Fase 1 de la negociación. El mecanismo de protección específico usado para el DOI de IPsec se describe en el Capítulo 10. Todas las implementaciones dentro del DOI de IPsec DEBEN soportar PROTO_ISAKMP.

NOTA: ISAKMP se reserva el valor uno (1) a través de todas las definiciones del DOI.

7.1.2 Protocolo AH IPsec

El tipo, Protocolo AH IPsec (PROTO_IPSEC_AH), especifica paquetes IP autentificados. La transformación AH por defecto proporciona autentificación del origen de los datos, protección de integridad, y detección de anti-replay. Debido a las consideraciones de control de exportación, la confidencialidad NO DEBE ser proporcionada por ninguna transformación PROTO_IPSEC_AH.

7.1.3 Protocolo ESP IPsec

El tipo, Protocolo ESP IPsec (PROTO_IPSEC_ESP), especifica confidencialidad de paquetes IP. La autentificación si es requerida, debe ser proporcionada como parte de la transformación ESP. La transformación ESP por defecto incluye autentificación del origen de los datos, protección de integridad, detección de anti-replay, y confidencialidad.

7.1.4 Protocolo de Compresión IP

El tipo, Protocolo de Compresión IP (PROTO_IPCOMP), especifica compresión de la carga IP como se define en [IPCOMP].

7.2 Identificador de Transformación ISAKMP IPsec

El Identificador de Transformación ISAKMP es un valor de 8 bit el cual identifica al protocolo de intercambio de claves usado para la negociación. Como parte de una negociación ISAKMP de la Fase 1, la elección del iniciador de ofrecer Intercambios de Claves se hace usando cierta descripción de la política del sistema host. La selección actual de mecanismos de Intercambios de Claves se realiza usando la Carga Propuesta de ISAKMP. La tabla siguiente lista los Identificadores de Transformación de la Fase 1 de ISAKMP definidos por la Carga Propuesta para el DOI de IPsec.

Transformación	Valor
Reservado	0
Clave IKE (KEY_IKE)	1

Los valores de 249 a 255 están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevos identificadores de transformación ISAKMP deben estar acompañados por un RFC el cual describa

los requisitos para el protocolo de intercambio de claves. [IKE] es un ejemplo de tal documento.

Dentro del marco de ISAKMP y del DOI de IPsec es posible definir protocolos para el establecimiento de claves aparte del IKE (Oakley). Versiones previas del [DOIPsec] definían tipos de claves manuales y diseños basándose en el uso de un Centro de Distribución de Claves (KDC) genérico. Estos identificadores se han quitado del [DOIPsec].

El DOI de IPsec todavía puede ser ampliado para incluir valores adicionales para protocolos de establecimiento de claves no Oakley para ISAKMP y IPsec, tales como Kerberos [RFC-1510] o como el Protocolo de Administración de Claves para Grupos (GKMP) [RFC-2093].

7.2.1 Clave IKE

El tipo, Clave IKE (KEY_IKE), especifica el intercambio de claves híbrido ISAKMP/Oakley Diffie-Hellman (IKE) tal como se define en el Capítulo 10. Todas las implementaciones dentro del DOI de IPsec DEBEN soportar KEY_IKE.

7.3 Identificador de Transformación AH IPsec

El Identificador de Transformación AH IPsec es un valor de 8 bit el cual identifica a un algoritmo particular usado para proporcionar protección de integridad para AH. El Protocolo AH (ver Capítulo 3) define una transformación obligatoria y varias trasformaciones opcionales usadas para proporcionar autentificación, integridad y detección de anti-replay. La tabla siguiente lista los Identificadores de Transformación de AH definidos para la Carga Propuesta de ISAKMP para el DOI de IPsec.

Nota: Los atributos del Algoritmo de Autentificación DEBEN ser especificados identificando el apropiado conjunto de protección AH. Por ejemplo, AH_MD5 puede ser interpretado como una transformación AH genérica usando MD5. Para solicitar la construcción de AH con HMAC, se especifica el Identificador de Transformación AH_MD5 junto con el conjunto de atributos de Algoritmos de Autentificación HMAC-MD5. Esto se ilustra usando la notación "Autentificación (HMAC-MD5)" en las siguientes secciones.

Identificador de Transformación	Valor	Referencia
Reservado	0-1	[DOIPsec]
AH con MD5 (AH_MD5)	2	[DOIPsec]
AH con SHA (AH_SHA)	3	[DOIPsec]
AH con DES (AH_DES)	4	[DOIPsec]
AH con SHA2 con 256 bits de longitud (AH_SHA2-256)	5	[IANA-2]
AH con SHA2 con 384 bits de longitud (AH_SHA2-384)	6	[IANA-2]
AH con SHA2 con 512 bits de longitud (AH_SHA2-512)	7	[IANA-2]
AH con RIPEMD (AH_RIPEMD)	8	[RIPEMD]
AH con AES-XCBC-MAC (AH_AES-XCBC-MAC)	9	[RFC3566]

Los valores de 249 a 255 están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevos identificadores de transformación AH deben estar acompañados por un RFC el cual describa como usar el algoritmo dentro del marco de AH ([AH]).

Nota: Todos los algoritmos de implementación obligatorios se listan cómo "DEBEN" ser implementados (por ejemplo AH_MD5). El resto de los algoritmos

son opcionales y PUEDEN ser implementados dentro de cualquier implementación particular.

7.3.1 AH con MD5

El tipo, AH con MD5 (AH_MD5), especifica una transformación AH genérica usando MD5. El conjunto de protección actual es determinado de acuerdo con una lista de atributos SA asociados.

Toda implementación dentro de DOI de IPsec DEBE soportar AH_MD5 junto con el atributo Autentificación (HMAC-MD5). Este conjunto es definido como la transformación HMAC-MD5-96 descripta en el Capítulo 6.

El tipo AH_MD5 junto con el atributo Autentificación (KPDK) especifica la transformación AH (clave/relleno/datos/clave) descripta en el RFC-1826.

El uso de AH_MD5 junto con algún otro valor de atributo de Algoritmo de Autentificación, actualmente no está definido.

7.3.2 AH con SHA

El tipo, AH con SHA (AH_SHA), especifica una transformación AH genérica usando SHA-1. El conjunto de protección actual es determinado de acuerdo con una lista de atributos SA asociados.

Toda implementación dentro de DOI de IPsec DEBE soportar AH_SHA junto con el atributo Autentificación (HMAC_SHA). Este conjunto es definido como la transformación HMAC-SHA-1-96 descripta en el Capítulo 6.

El uso de AH_SHA junto con algún otro valor de atributo de Algoritmo de Autentificación actualmente no está definido.

7.3.3 AH con DES

El tipo, AH con DES (AH_DES), especifica una transformación AH genérica usando DES. El conjunto de protección actual es determinado de acuerdo con una lista de atributos SA asociados.

El DOI de IPsec define que AH_DES junto con el atributo Autentificación (DES-MAC) es una transformación DES-MAC. Las implementaciones no requieren soportar este modo.

El uso de AH_DES junto con algún otro valor de atributo de Algoritmo de Autentificación actualmente no está definido.

7.3.4 AH con SHA2 con 256 Bits de Longitud

El tipo, AH con el algoritmo SHA2 con 256 bits de longitud (AH_SHA2-256), especifica una transformación AH genérica usando SHA2 con 256 bits de longitud.

7.3.5 AH con SHA2 con 384 Bits de Longitud

El tipo, AH con el algoritmo SHA2 con 384 bits de longitud (AH_SHA2-384), especifica una transformación AH genérica usando SHA2 con 384 bits de longitud .

7.3.6 AH SHA2 con 512 Bits de Longitud

El tipo, AH con el algoritmo SHA2 con 512 bits de longitud (AH_SHA2-512), especifica una transformación AH genérica usando SHA2 con 512 bits de longitud.

7.3.7 AH con RIPEMD

El tipo, AH con RIPEMD (AH_RIPEMD),especifica una transformación AH genérica usando el algoritmo RIPEMD definido en [RIPEMD].

7.3.8 AH con AES-XCBC-MAC

El tipo, AH con AES-XCBC-MAC (AH_AES-XCBC-MAC), especifica una transformación AH genérica usando el algoritmo AES-XCBC-MAC-96 definido en [RFC3566].

7.4 Identificador de Transformación ESP IPsec

El Identificador de Transformación ESP IPsec es un valor de 8 bit el cual identifica a un algoritmo particular usado para proporcionar protección de confidencialidad para ESP. La Carga de Seguridad Encapsulada (ESP) define una transformación obligatoria y varias trasformaciones opcionales usadas para proporcionar confidencialidad a los datos. La tabla siguiente lista los Identificadores de Transformación ESP definidos para la Carga de la Propuesta de ISAKMP para el DOI de IPsec.

Nota: cuando se requiere autentificación, protección de integridad, y detección de anti-replay, los atributos del Algoritmo de Autentificación DEBEN ser especificados para identificar el conjunto de protección ESP apropiado. Por ejemplo, si se requiere la autentificación HMAC-MD5 con 3DES, uno especifica el Identificador de Transformación ESP_3DES con el conjunto de atributos del Algoritmo de Autentificación HMAC-MD5. Para requerimientos de procesamiento adicional, ver la Sección 8 (Algoritmos de Autentificación)

Identificador de Transformación	Valor	Referencia
Reservado	0	[DOIPsec]
ESP con DES usando un IV de 64 bits (ESP_DES_IV64)	1	[DOIPsec]
ESP con DES (ESP_DES)	2	[DOIPsec]
ESP con 3DES (ESP_3DES)	3	[DOIPsec]
ESP con RC5 (ESP_RC5)	4	[DOIPsec]
ESP con IDEA (ESP_IDEA)	5	[DOIPsec]
ESP con CAST (ESP_CAST)	6	[DOIPsec]
ESP con BLOWFISH (ESP_BLOWFISH)	7	[DOIPsec]
ESP con 3IDEA (ESP_3IDEA)	8	[DOIPsec]
ESP con DES usando un IV de 32 bits (ESP_DES_IV32)	9	[DOIPsec]
ESP con RC4 (ESP_RC4)	10	[DOIPsec]
ESP con NULL (ESP_NULL)	11	[DOIPsec]
ESP con AES en modo CBC (ESP_AES-CBC)	12	[RFC3602]
ESP con AES en modo CTR (ESP_AES-CTR)	13	[RFC3686]
ESP con AES en modo CCM con un ICV de 8 octetos (ESP_AES-CCM_8)	14	[AES-CCM]
ESP con AES en modo CCM con un ICV de 12 octetos (ESP_AES-CCM_12)	15	[AES-CCM]
ESP con AES en modo CCM con un ICV de 16 octetos (ESP_AES-CCM_16)	16	[AES-CCM]
No Signado	17	[IANA-2]
ESP con AES en modo GCM con un ICV de 8 octetos (ESP_AES-GCM_8)	18	[AES-GCM]
ESP con AES en modo GCM con un ICV de 12 octetos (ESP_AES-GCM_12)	19	[AES-GCM]
ESP con AES en modo GCM con un ICV de 16 octetos (ESP_AES-GCM_16)	20	[AES-GCM]
ESP con SEED en modo CBC(ESP_SEED_CBC)	21	[SEED]
ESP con CAMELLIA (ESP_CAMELLIA)	22	[kato-ipsec]

Los valores de 249 a 255 están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevos identificadores de transformación ESP deben estar acompañados por un RFC el cual describa como usar el algoritmo dentro del marco de ESP ([ESP]).

Nota: Todos los algoritmos de implementación obligatorios se listan cómo "DEBEN" ser implementados (por ejemplo ESP_DES). El resto de los algoritmos son opcionales y PUEDEN ser implementados dentro de cualquier implementación particular.

7.4.1 ESP con DES Usando un IV de 64 Bits

El tipo, ESP con DES usando un IV de 64 bits (ESP_DES_IV64), especifica la transformación DES-CBC definida en el RFC-1827 y el RFC-1829 usando un Vector de Inicialización (IV) de 64 bits.

7.4.2 ESP con DES

El tipo, ESP con DES (ESP_DES), especifica una transformación DES genérica usando DES-CBC. El conjunto de protección actual es determinado de acuerdo con una lista de atributos SA asociados.

Toda implementación dentro del DOI de IPsec DEBE soportar ESP_DES junto con el atributo Autentificación (HMAC-MD5). Este conjunto es definido como la transformación [DES], suministrando autentificación y integridad a través de HMAC MD5, descripta en el Capítulo 6.

Nota del Grupo de Administración de Ingeniería de Internet (IESG): Trabajos resientes en el área del análisis criptográfico sugieren que el algoritmo DES puede no ser suficientemente fuerte para muchas aplicaciones. Por consiguiente, es muy probable que el IETF desestime el uso de ESP_DES como un algoritmo criptográfico obligatorio en un futuro cercano. Este permanecerá como de uso opcional en el Protocolo. Aunque el grupo de trabajo de IPsec y el IETF no se han decidido sobre el algoritmo alternativo (hay que tener en cuenta esta consideración de seguridad y funcionamiento), implementadores pueden desear tener en cuenta la recomendación de la Sección 7.4.3 sobre el uso de ESP_3DES.

7.4.3 ESP con 3DES

El tipo, ESP con 3DES (ESP_3DES), especifica una transformación DES triple genérica. El conjunto de protección actual es determinado de acuerdo con una lista de atributos SA asociados.

Toda implementación dentro de DOI de IPsec se le aconseja encarecidamente soportar ESP_3DES junto con el atributo Autentificación (HMAC-MD5). Este conjunto es definido como la transformación [CBC], suministrando autentificación y integridad por HMAC MD5, descripta en el Capítulo 6.

7.4.4 ESP con RC5

El tipo, ESP con RC5 (ESP_RC5), especifica la transformación RC5 definida en [CBC].

7.4.5 ESP con IDEA

El tipo, ESP con IDEA (ESP_IDEA), especifica la transformación IDEA definida en [CBC].

7.4.6 ESP con CAST

El tipo, ESP con CAST (ESP_CAST), especifica la transformación CAST definida en [CBC].

7.4.7 ESP con BLOWFISH

El tipo, ESP con BLOWFISH (ESP_BLOWFISH), especifica la transformación BLOWFISH definida en [CBC].

7.4.8 ESP con 3IDEA

El tipo, ESP con 3IDEA (ESP_3IDEA), esta reservado para IDEA triple.

7.4.9 ESP con DES Usando un IV de 32 Bits

El tipo, ESP con DES usando un IV de 32 bits (ESP_DES_IV32), especifica la transformación DES-CBC definida en el RFC 1827 y en el RFC 1829 usando un Vector de Inicialización (IV) de 32 bits.

7.4.10 ESP con RC4

El tipo, ESP con RC4 (ESP_RC4), está reservado para RC4.

7.4.11 ESP con NULL

El tipo, ESP con NULL (ESP_NULL), especifica que la confidencialidad no debe ser proporcionada por ESP. ESP_NULL se usa cuando ESP es usado en paquetes "tuneliados" que solamente requieren autentificación, protección de integridad, y detección de anti-replay.

Toda implementación dentro del DOI de IPsec DEBE soportar ESP_NULL. La transformación ESP NULL se define en el Capítulo 6. Ver la descripción de atributos de los Algoritmos de Autentificación en la Sección 8 para requerimientos adicionales relacionados con el uso de ESP_NULL.

7.4.12 ESP con AES en Modo CBC

El tipo, ESP con AES en modo CBC (ESP_AES-CBC), especifica una transformación usando el algoritmo AES en modo CBC definido en [RFC3602].

7.4.13 ESP con AES en Modo CTR

El tipo, ESP con AES en modo CTR (ESP_AES-CTR), especifica una transformación usando el algoritmo AES en modo CTR (Counter Mode) con un IV explicito, definido en [RFC3686].

7.4.14 ESP con AES en Modo CCM con un ICV de 8 Octetos

El tipo, ESP con AES en modo CCM con un ICV de 8 octetos (ESP_AES-CCM_8), especifica una transformación usando el algoritmo AES en modo CCM (Modo Counter con CBC-MAC) con un ICV (Valor de Comprobación de Integridad) de 8 octetos definido en [AES-CCM].

7.4.15 ESP con AES en Modo CCM con un ICV de 12 Octetos

El tipo, ESP con AES en modo CCM con un ICV de 12 octetos (ESP_AES-CCM_12), especifica una transformación usando el algoritmo AES en modo CCM con un ICV de 12 octetos definido en [AES-CCM].

7.4.16 ESP con AES en Modo CCM con un ICV de 16 Octetos

El tipo, ESP con AES en modo CCM con un ICV de 16 octetos (ESP_AES-CCM_16), especifica una transformación usando el algoritmo AES en modo CCM con un ICV de 16 octetos definido en [AES-CCM].

7.4.17 ESP con AES en Modo GCM con un ICV de 8 Octetos

El tipo, ESP con AES en modo GCM con un ICV de 8 octetos(ESP_AES-GCM_8), especifica una transformación usando el algoritmo AES en modo GCM (Modo Galois/Counter) con un ICV (Valor de Comprobación de Integridad) de 8 octetos definido en [AES-GCM].

7.4.18 ESP con AES en Modo GCM con un ICV de 12 Octetos

El tipo, ESP con AES en modo GCM con un ICV de 12 octetos(ESP_AES-GCM_12), especifica una transformación usando el algoritmo AES en modo GCM con un ICV de 12 octetos definido en [AES-GCM].

7.4.19 ESP con AES en Modo GCM con un ICV de 16 Octetos

El tipo, ESP con AES en modo GCM con un ICV de 16 octetos(ESP_AES-GCM_16), especifica una transformación usando el algoritmo AES en modo GCM con un ICV de 16 octetos definido en [AES-GCM].

7.4.20 ESP con SEED en Modo CBC

El tipo, ESP con SEED en modo CBC (ESP_SEED_CBC), especifica una transformación usando el algoritmo SEED en modo CBC con un IV explicito definido en [SEED].

7.4.21 ESP con CAMELLIA

El tipo, ESP con CAMELLIA (ESP_CAMELLIA), especifica una transformación usando el algoritmo de cifrado en bloque CAMELLIA en modo CBC con un IV explicito definido en [kato-ipsec].

7.5 Identificador de Transformación IPCOMP para IPsec

El Identificador de Transformación IPCOMP IPsec es un valor de 8 bit el cual identifica a un algoritmo particular usado para proporcionar compresión a nivel IP antes de ESP. La transformación de la compresión IP (IPCOMP) define algoritmos de compresión opcionales que pueden ser negociados para proporcionar compresión para la carga IP ([IPCOMP]). La tabla siguiente lista los Identificadores de Transformación IPCOMP definidos para la Carga de la Propuesta de ISAKMP dentro del DOI de IPsec.

Identificador de Transformación	Valor	Referencia
Reservado	0	[DOIPsec]
IPCOMP_OUI	1	[DOIPsec]
IPCOMP_DEFLATE	2	[DOIPsec]
IPCOMP_LZS	3	[DOIPsec]
IPCOMP_LZJH	4	[LZJH]

Los valores de 1 a 47 están reservados para algoritmos para los cuales un RFC ha sido aprobado para publicarse. Los valores de 48 a 63 están reservados para usarse en forma privada entre sistemas. Los valores de 64 a 255 están reservados para futuras ampliaciones. Pedidos de asignaciones de nuevos identificadores de transformación IPCOMP deben estar acompañados por un RFC el cual describa como usar el algoritmo dentro del marco de IPCOMP ([IPCOMP]). Además el algoritmo requerido debe ser publicado y de dominio público.

7.5.1 IPCOMP_OUI

EL tipo IPCOMP_OUI especifica una propiedad de transformación de compresión. El tipo IPCOMP_OUI debe estar acompañado por un atributo que identifique el algoritmo específico del vendedor.

7.5.2 IPCOMP_DEFLATE

El tipo IPCOMP_DEFLATE especifica el uso del algoritmo de compresión "zlib" como se especifica en [DEFLATE].

7.5.3 IPCOMP_LZS

El tipo IPCOMP_LZS especifica el uso del algoritmo Stac Electronics como se especifica en [LZS].

7.5.4 IPCOMP_LZJH

El tipo IPCOMP_LZJH especifica el uso del algoritmo de compresión de datos LZJH (también denominado, ITU-T V.44 Packet Method) como se describe en [LZJH].

8. **Atributos de la Asociación de Seguridad IPsec**

Los Atributos de la Asociación de Seguridad IPsec consiste de un tipo de 16 bit y de sus valores asociados. Los atributos SA IPsec son usados para pasar diversos valores entre los usuarios de ISAKMP. Las siguientes definiciones de los atributos de las SAs se usan en la negociación de la Fase 2 de IKE. Los tipos de Atributos pueden ser Básico (B) o Longitud-Variable (V). La codificación de estos atributos se definió en la Sección 3.3 de Capítulo 7 como Tipo/Valor (Básico) y Tipo/Longitud/valor (Variable).

La descripción de atributos como básico NO DEBE ser codificada como variable. El atributo longitud variable puede ser codificado como atributo básico si su valor puede entrar dentro de dos octetos. Ver el Capítulo 10 para información adicional sobre la codificación de los atributos en el DOI de IPsec. Todas las restricciones enumeradas dentro del Capítulo 10 también se aplican al DOI de IPsec.

Tipos de Atributos

Clase	Valor	Tipo	Referencia
Tipo de Vida de la SA	1	B	[DOIPsec]
Tiempo de Vida de la SA	2	V	[DOIPsec]
Descripción del Grupo	3	B	[DOIPsec]
Modo de Encapsulación	4	B	[DOIPsec]
Algoritmos de Autentificación	5	B	[DOIPsec]
Longitud de la Clave	6	B	[DOIPsec]
Ciclo de la clave	7	B	[DOIPsec]
Tamaño de la Compresión del Diccionario	8	B	[DOIPsec]
Algoritmo de Compresión Privado	9	V	[DOIPsec]
Túnel ECN (Notificación de Congestión Explícita)	10	B	[UdateIPsec]
Extensión del Número de Secuencia (a 64 bits)	11	B	[NumSequen]

Los valores de 32001 a 32767 están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevos Atributos SA IPsec deben describir la codificación del atributo (Básico/Longitud-Variable) y sus valores válidos. La Sección 8 proporciona un ejemplo de tal descripción.

Detalles de las Clases de Valores

- Tipo de Vida de la SA
- Tiempo de Vida de la SA

  Especifica el tiempo de vida para la SA. Cuando la SA expira, todas las claves negociadas bajo la asociación (AH o ESP) deben ser

renegociadas. Los valores para el tipo de vida son:

Tipo	Valor
RESERVADO	0
Segundos	1
kilobytes	2

Los valores de 3 a 61439 están reservados por la IANA. Los valores de 61440 a 65535 son para uso privado. Para un Tipo de Vida dado, el valor del atributo del Tiempo de Vida define la longitud actual del componente tiempo de vida -- un número de segundos, o un número en kilobytes que pueden ser protegidos.

Si no se especifica, se asumirá el valor por defecto el cual es de 28800 segundos (8 horas).

Un atributo Tiempo de Vida de la SA siempre DEBE estar seguido de un atributo Tipo de Vida que describa la unidad de duración.

Ver Sección 8.4 para información adicional relacionada con la notificación del tiempo de vida.

- Descripción del Grupo

  Especifica el Grupo Oakley usado en una negociación en Modo Rápido(QM) con PFS (Perfect Forward Secrecy). Para una lista de valores soportados, vea la Sección 10 del Capítulo 10.

- Modo de Encapsulación

  Los valores para el Modo de Encapsulación son:

Tipo	Valor	Referencia
RESERVADO	0	[DOIPsec]
Túnel	1	[DOIPsec]
Transporte	2	[DOIPsec]
Encapsulación UDP en modo Túnel	3	[NATinIKE]
Encapsulación UDP en modo Transporte	4	[NATinIKE]

  Los valores de 5 a 61439 están reservados por la IANA. Los valores de 61440 a 65535 son para uso privado.

  Si no se especifica, se asumirá el valor por defecto como no especificado (depende del host).

- Algoritmos de Autentificación

  Los valores para los Algoritmos de Autentificación son:

Tipo	Valor	Referencia
RESERVADO	0	[DOIPsec]
HMAC-MD5	1	[DOIPsec]
HMAC-SHA	2	[DOIPsec]
DES-MAC	3	[DOIPsec]
KPDK	4	[DOIPsec]
HMAC-SHA2-256	5	[IANA-2]
HMAC-SHA2-384	6	[IANA-2]
HMAC-SHA2-512	7	[IANA-2]
HMAC-RIPEMD	8	[RIPEMD]
AES-XCBC-MAC	9	[RFC3566]

Los valores de 10 a 61439 están reservados por la IANA. Los valores de 61440 a 65535 son para uso privado.

No existe valor por defecto para el Algoritmo de Autentificación, se debe especificar para identificar correctamente la transformación AH o ESP aplicada, excepto en los siguientes casos:

Cuando ESP es negociado sin autentificación, el atributo Algoritmo de Autentificación NO DEBE ser incluido en la propuesta.

Cuando ESP es negociado sin confiabilidad, el atributo Algoritmo de Autentificación DEBE ser incluido en la propuesta y el identificador de transformación ESP debe ser ESP_NULL.

- Longitud de la Clave

  El valor para la Longitud de la Clave es:

Tipo	Valor
RESERVADO	0

  No existe valor por defecto para la Longitud de la Clave, se debe especificar para usar transformaciones de cifrado con longitudes de claves variables. Para los cifrados que tienen longitud fija, el atributo Longitud de la Clave NO DEBE ser enviado.

- Ciclo de la Clave

  El valor para el Ciclo de la Clave es:

Tipo	Valor
RESERVADO	0

  No existe valor por defecto para el Ciclo de la Clave, se debe especificar para usar transformaciones de cifrado con un número variable de ciclos.

- Tamaño de la Compresión del Diccionario

  El valor para la Compresión del Diccionario es:

Tipo	Valor
RESERVADO	0

Especifica el tamaño máximo de longitud del diccionario.

No existe valor por defecto para el tamaño del diccionario.

- Algoritmo de Compresión Privado

  Especifica un algoritmo de compresión de un vendedor privado. Los primeros tres (3) octetos deben ser una asignación del IEEE company_id (OUI). Los siguientes octetos pueden ser un subtipo específico de la compresión del vendedor, seguido de cero o más octetos de datos del vendedor.

- Túnel ECN (Notificación de Congestión Explícita)

Tipo	Valor	Referencia
RESERVADO	0	[UdateIPsec]
Permitido	1	[UdateIPsec]
Prohibido	2	[UdateIPsec]

Los valores de 3 a 61439 están reservados por la IANA. Los valores de 61440 a 65535 son para uso privado.
Si no está especificado, se asume Prohibido.

- Extensión del Número de Secuencia (a 64 bits)

Tipo	Valor	Referencia
RESERVADO	0	[NumSequen]
Número de Secuencia de 64 bits	1	[NumSequen]

8.1 Atributos SA Requeridos

Para garantizar interoperatividad, toda implementación DEBE estar preparada para negociar los siguientes atributos:

- Tipo de Vida de la SA
- Tiempo de Vida de la SA
- Algoritmo de Autentificación

8.2 Desglosamiento del Atributo Tipo de Vida y Tiempo de Vida

Para permitir flexibilidad en la semántica, el DOI de IPsec REQUIERE que una implementación ISAKMP desglose correctamente una lista de atributos que contengan múltiples instancias de la misma clase de atributo, siempre que las entradas de diferentes atributos no estén en conflicto con la de alguna otra. Actualmente, el único atributo que requiere este tratamiento es el Tipo de Vida y el Tiempo de Vida.

Para comprender por qué esto es importante, el siguiente ejemplo muestra la codificación en binario de una lista de 4 atributos de entrada que especifica un Tipo de Vida de la SA en 100MB o 24 horas. (Ver la Sección 3.3 del Capítulo 7 para una completa descripción del formato de la codificación de los atributos.)

```
Atributo N°1:
  0x80010001 (Formato del Atributo=1, tipo=Tipo de Vida de la SA,
              valor=segundos)

Atributo N°2:
  0x00020004 (Formato del Atributo=0, tipo=Tiempo de Vida de la SA,
              longitud=4 bytes)
  0x00015180 (valor=0x15180=86400 segundos=24 horas)

Atributo N°3:
  0x80010002 (Formato del Atributo=1, tipo=Tipo de Vida de la SA, valor=KB)

Atributo N°4:
  0x00020004 (Formato del Atributo=0, tipo=Tiempo de Vida de la SA,
              longitud=4 bytes)
  0x000186A0 (valor=0x186A0=100000KB=100MB)
```

Si se detecta conflicto en los atributos, una Carga de Notificación conteniendo, ATRIBUTOS NO SOPORTADOS, DEBERÍA ser enviado y el establecimiento de la SA DEBE ser abortado.

8.3 Negociación de Atributos

Si una implementación recibe una atributo DOI de IPsec especifico (o valor del atributo) el cual no es soportado, una Carga de Notificación conteniendo, ATRIBUTOS NO SOPORTADOS, DEBERÍA ser enviado y la instalación de la SA DEBE ser abortada, amenos que el valor del atributo este dentro del rango reservado.

Si una implementación recibe un valor de atributo dentro del rango reservado, una implementación PUEDE elegir continuar basándose en la política local.

8.4 Notificación del Tiempo de Vida

Cuando un iniciador ofrece un tiempo de vida para la SA mayor que lo que el respondedor desea basándose en su política local, el respondedor tiene 3 opciones:

1) cancelar la negociación entrante
2) completar la negociación pero usando un tiempo de vida más pequeño que el qué se había ofrecido.
3) completar la negociación y enviar un aviso de notificación al iniciador indicando el verdadero tiempo de vida al respondedor.

La decisión del respondedor depende de la implementación especifica y/o de la política local.

Para garantizar interoperabilidad en el último caso, solamente cuando el respondedor desea notificar al iniciador, el DOI de IPsec requiere que: si el iniciador ofrece un tiempo de vida de la SA mayor de lo que el respondedor está dispuesto a aceptar, el respondedor DEBERÍA incluir una Carga de Notificación ISAKMP en el intercambio que contiene la carga SA IPsec del respondedor. La Sección 9.3.1 define el diseño de la carga para el tipo de Mensaje de Notificación Tiempo de Vida del Respondedor (RESPONDER-LIFETIME) el cual DEBE ser usado para este propósito.

9. **Contenido de la Carga IPsec**

Las siguientes secciones describen las cargas ISAKMP cuya representación de los datos es dependiente del DOI aplicado.

9.1 Carga de la Asociación de Seguridad

El diagrama siguiente ilustra el contenido de la Carga SA para el DOI de IPsec. Ver la Sección 5 para una descripción de la Situación bitmap. La Figura 1 muestra el formato de la carga SA (donde, * = en bits, Cat. = Categoría; Integri = Interidad; Long = Longitud)

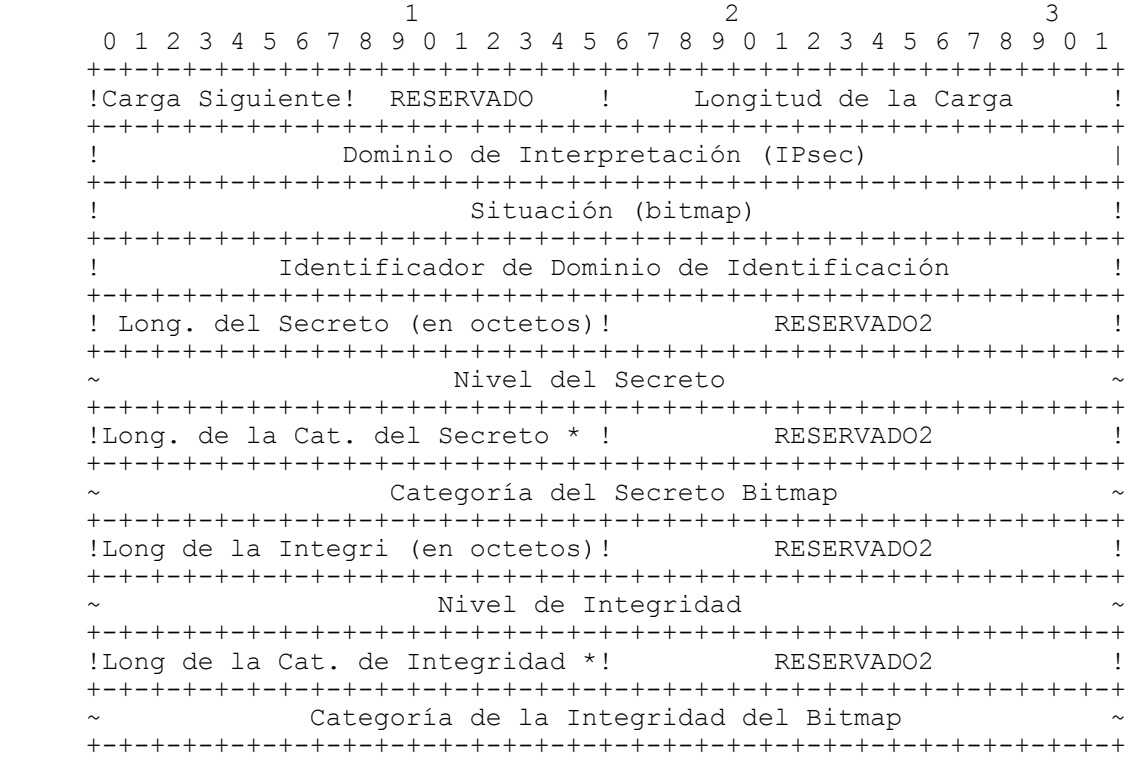


Figura 1: Formato de la Carga SA

La Carga SA se definen como sigue:

- Carga Siguiente (1 octeto): Identificador para el tipo de carga de la carga siguiente en el mensaje. Si la carga actual es la última en el mensaje, este campo contendrá ceros (0).
- RESERVADO (1 octeto): No usado, debe contener ceros (0).
- Longitud de la Carga (2 octetos): Longitud, en octetos, de la carga actual, incluida la cabecera de carga genérica.
- Dominio de Interpretación (4 octetos): Especifica el DOI de IPsec, el cuál ha sido asignado con el valor de uno (1).
- Situación (4 octetos): Bitmask usado para interpretar el resto de la Carga SA. Ver Sección 5 para una lista completa de valores.

- Identificador de Dominio de Identificación (4 octetos): Número Asignado por la IANA usado para interpretar la información del Secreto y de la Integridad.

- Longitud del Secreto (2 octetos): Especifica la longitud, en octetos, del identificador del nivel de secreto, excluyendo los bits de relleno.

- RESERVADO2 (2 octetos): No usado, debe contener ceros (0).

- Nivel del Secreto (longitud variable): Especifica el nivel de secreto requerido obligatoriamente. El nivel del secreto DEBE ser rellenado con ceros (0) para alinearlo a limites de 32 bit.

- Longitud de la Categoría del Secreto (2 octetos): Especifica la longitud, en bits, de la categoría (sector) bitmap, excluyendo los bits de relleno.

- Categoría del Secreto Bitmap (longitud variable): Un bitmap usado para designar categorías de secretos (sectores) que se requieren. El bitmap DEBE ser rellenado con ceros (0) para alinearlo a limites de 32 bit.

- Longitud de la Integridad (2 octetos): Especifica la longitud, en octetos, del identificador del nivel de integridad, excluyendo los bits de relleno.

- Nivel de Integridad (longitud variable): Especifica el nivel de integridad requerido obligatoriamente. El nivel de integridad DEBE ser rellenado con ceros (0) para alinearlo a limites de 32 bit.

- Longitud de la Categoría de integridad (2 octetos): Especifica la longitud, en bits, de la categoría de integridad (sector) bitmap, excluyendo los bits de relleno.

- Categoría de la Integridad del Bitmap (longitud variable): Un bitmap usado para designar categorías de integridad (sectores) que se requieran. El bitmap DEBE ser rellenado con ceros (0) para alinearlo a limites de 32 bit.

9.1.1 Identificadores de Dominio de Identificación de IPsec

El Identificador de Dominio de Identificación IPsec es un valor de 32 bit el cual identifica a un espacio de asignación de nombres (namespace) en el cual existen niveles de Confidencialidad, Integridad y valores de categorías. La tabla siguiente lista los valores asignados para el campo Identificador de Dominio de Identificación contenidos dentro del campo Situación en la Carga SA.

Dominio	Valor
RESERVADO	0

Los valores de 0x80000000 a 0xffffffff están reservados para usarse en forma privada entre sistemas. Pedidos de asignaciones de nuevos Identificadores de Dominio de Identificación se deberían conceder o demandar. No se requiere que lo acompañe documentación, sin embargo se aconsejan Drafts Internet cuando sea apropiado.

9.2 Contenido de la Carga de Identificación

La Carga de Identificación es usada para identificar al iniciador de la SA. La identificación del iniciador DEBERÍA ser usada por el respondedor para determinar los requisitos de la política de seguridad del sistema host adecuados para la asociación. Por ejemplo, un host puede elegir necesitar autentificación y integridad sin confidencialidad (AH) para un cierto conjunto de direcciones IP y brindar autentificación con confidencialidad (ESP) para otro rango de direcciones IP. La Carga de Identificación proporciona información que puede ser usada por el respondedor para tomar esta decisión.

Durante la Fase 1 de la negociación, el campo identificador de protocolo y puerto DEBEN estar en cero o el campo del puerto debe contener el valor 500. Si una implementación recibe algún otro valor, este DEBE ser considerado como un error y la carga SA DEBE ser abortada. Este evento DEBERÍA ser un evento auditable. La Figura 2 ilustra el contenido de la Carga de Identificación.

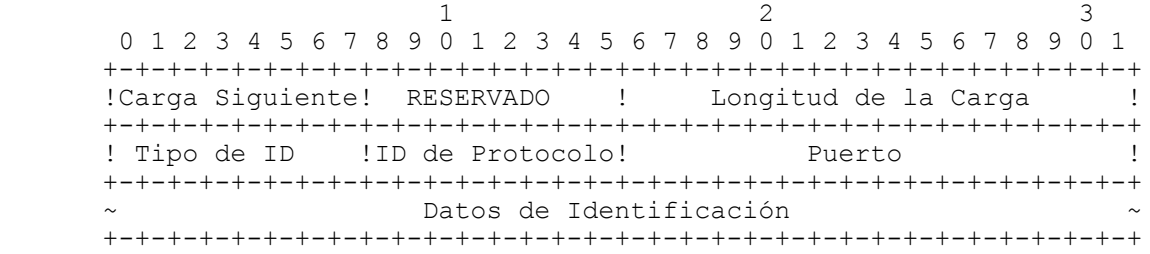


Figura 2: Formato de la Carga de Identificación

Los campos de la Carga de Identificación se definen de la siguiente forma:

- Carga Siguiente (1 octeto): Identificador para el tipo de carga de la carga siguiente en el mensaje. Si la carga actual es la última en el mensaje, este campo debe contener ceros (0).
- RESERVADO (1 octeto): No usado, debe contener ceros (0).
- Longitud de la Carga (2 octetos): Longitud, en octetos, de los datos de identificación, incluyendo la cabecera de carga genérica.
- Tipo de Identificador (1 octeto): Valor descriptivo de la información de identidad encontrada en el campo Datos de identificación.
- Identificador de Protocolo (1 octeto): Valor que especifica un identificador de protocolo IP asociado (por ejemplo UDP/TCP). Un valor de cero significa que el campo Identificador de Protocolo debería ser ignorado.
- Puerto (2 octetos): Valor que especifica un puerto asociado. Un valor de cero significa que el campo Puerto debería ser ignorado.
- Datos de Identificación (longitud variable): Valor, indicado por el Tipo de Identificador.

9.2.1 Tipo de Identificador en la Carga de Identificación de ISAKMP

El Tipo de Identificador IPsec es un valor de 8 bit el cual es usado como

un discriminante para interpretar la longitud variable de la Carga de Identificación. La tabla siguiente lista los valores asignados para el campo Tipo de Identificador encontrados dentro de la Carga de Identificación.

Tipo de Identificador	Valor	Referencia
RESERVADO	0	[DOIPsec]
Identificador de Dirección IPv4 (ID_IPV4_ADDR)	1	[DOIPsec]
Identificador de Nombre de Dominio Completamente Cuantificado (ID_FQDN)	2	[DOIPsec]
Identificador de Usuario de Nombre de Dominio Completamente Cuantificado (ID_USER_FQDN)	3	[DOIPsec]
Identificador de Dirección de Subred IPv4 (ID_IPV4_ADDR_SUBNET)	4	[DOIPsec]
Identificador de Dirección IPv6 (ID_IPV6_ADDR)	5	[DOIPsec]
Identificador de Dirección de Subred IPv6 (ID_IPV6_ADDR_SUBNET)	6	[DOIPsec]
Identificador de Rango de direcciones IPv4 (ID_IPV4_ADDR_RANGE)	7	[DOIPsec]
Identificador de Rango de direcciones IPv6 (ID_IPV6_ADDR_RANGE)	8	[DOIPsec]
Identificador DER ASN.1 de Nombre de distribución X.500 (ID_DER_ASN1_DN)	9	[DOIPsec]
Identificador DER ASN.1 de Nombre Generales X.500 (ID_DER_ASN1_GN)	10	[DOIPsec]
Identificador de Identificación Clave (ID_KEY_ID)	11	[DOIPsec]
Identificador de Lista (ID_LIST)	12	[IPsecSCTP]

Los valores de 249 a 255 están reservados para usarse en forma privada entre sistemas.

Para los tipos donde la entidad del identificador tiene una longitud variable, el tamaño de la entidad del identificador es calculado a partir del tamaño en la cabecera de carga de identificación.

Cuando un intercambio IKE es autentificado usando certificados (de cualquier formato), cualquier identificador usado para las decisiones de la política local de entrada DEBERÍA ser incluido en el certificado usado en la autentificación del intercambio.

9.2.2 Identificador de Dirección IPv4

El tipo, Identificador de Dirección IPv4 (ID_IPV4_ADDR), especifica una sola dirección IPv4 de cuatro (4) octetos.

9.2.3 Identificador de Nombre de Dominio Completamente Cuantificado

El tipo, Identificador de Nombre de Dominio Completamente Cuantificado (ID_FQDN), especifica una cadena de caracteres que contiene un nombre de dominio completamente cuantificado. Un ejemplo de un ID_FQDN es, "foo.bar.com". La cadena de caracteres no debería contener ningún terminador, como por ejemplo, un punto "." final.

9.2.4 Identificador de Usuario de Nombre de Dominio Completamente Cuantificado

El tipo, Identificador de Usuario de Nombre de Dominio Completamente Cuantificado (ID_USER_FQDN),especifica una cadena de caracteres que contiene un nombre de dominio completamente cuantificado. Un ejemplo de un ID_USER_FQDN es, "piper@foo.bar.com". La cadena de caracteres no debería contener ningún terminador, como por ejemplo, un punto "." final.

9.2.5 Identificador de Dirección de Subred IPv4

El tipo, Identificador de Dirección de Subred IPv4 (ID_IPV4_ADDR_SUBNET), especifica un rango de direcciones IPv4, representadas por dos valores de cuatro (4) octetos. El primer valor es una dirección IPv4. El segundo valor es una máscara de red IPv4. Note que unos (1s) en la máscara de red indican que el correspondiente bit en la dirección es fijo, mientras que ceros (0s) indican un bit "comodín".

9.2.6 Identificador de Dirección IPv6

El tipo, Identificador de Dirección IPv6 (ID_IPV6_ADDR), especifica una sola dirección IPv6 de dieciséis (16) octetos.

9.2.7 Identificador de Dirección de Subred IPv6

El tipo, Identificador de Dirección de Subred IPv6 (ID_IPV6_ADDR_SUBNET), especifica un rango de direcciones de IPv6 representados por dos valores de dieciséis (16) octetos. El primer valor es una dirección IPv6. El segundo valor es una máscara de red IPv6. Note que unos (1s) en la máscara de red indican que el correspondiente bit en la dirección es fijo, mientras que ceros (0s) indican un bit "comodín".

9.2.8 Identificador de Rango de Direcciones IPv4

El tipo, Identificador de Rango de Direcciones IPv4 (ID_IPV4_ADDR_RANGE), especifica un rango de direcciones IPv4, representados por dos valores de cuatro (4) octetos. El primer valor es el principio del rango de direcciones IPv4 (incluyendo este valor) y el segundo valor es el final del rango de valores de direcciones IPv4 (incluyendo este valor). Todas las direcciones que están dentro del rango se consideran dentro de la lista.

9.2.9 Identificador de Rango de Direcciones IPv6

El tipo, Identificador de Rango de Direcciones IPv6 (ID_IPV6_ADDR_RANGE), especifica un rango de direcciones IPv6, representados por dos valores de dieciséis (16) octetos. El primer valor es el principio del rango de direcciones IPv6 (incluyendo este valor) y el segundo valor es el final del rango de valores de direcciones IPv6 (incluyendo este valor). Todas las direcciones que están dentro del rango se consideran dentro de la lista.

9.2.10 Identificador DER ASN.1 de Nombre de distribución X.500

El tipo, Identificador DER ASN.1 de Nombre de distribución X.500 (ID_DER_ASN1_DN), especifica la codificación DER (Distinguished Encoding Rules - Regla de codificación de distribución) binaria del Nombre de distribución ASN.1 X.500 [X.501] de cuyos certificados se están intercambiando para el establecimiento de la SA.

9.2.11 Identificador DER ASN.1 de Nombre Generales X.500

El tipo, Identificador DER ASN.1 de Nombre Generales X.500 (ID_DER_ASN1_GN), especifica la codificación DER binaria del Nombre General ASN.1 X.500 [X.509] de cuyos certificados se están intercambiando para el establecimiento de la SA.

9.2.12 Identificador de Identificación Clave

El tipo, Identificador de Identificación Clave (ID_KEY_ID), especifica una cadena de bit oculta, la cual puede ser usada para enviar la información específica del vendedor necesaria para identificar que clave pre-compartida debería ser usada para autentificar la negociación en modo Agresivo.

9.2.13 Identificador de Identificador de Lista

El tipo, Identificador de Lista (ID_LIST)), se describe en [IpsecSCTP], el cual describe el uso el Protocolo de Trasmisión de Control de Flujo (SCTP) [SCTP]con IPsec.

9.3 Tipos de Mensaje de Notificación IPsec

El tipo, Mensajes de Notificación, es un valor de 16 bit tomado del rango de valores reservados por ISAKMP para cada DOI. ISAKMP define dos bloques de códigos de Mensajes de Notificación, uno para los Mensajes de Errores (8192 a 16383) y el otro para los Mensajes de Estado (24576 a 32767). ISAKMP también asigna una parte de cada bloque para uso privado dentro de un DOI. El DOI de IPsec define los siguientes tipos de mensaje privados para su propio uso.

Mensaje de Notificación – Tipos de Error	Valor
RESERVADO	8192

Mensaje de Notificación – Tipos de Estado	Valor
Tiempo de Vida del Respondedor (RESPONDER-LIFETIME)	24576
Estado del Anti-replay (REPLAY-STATUS)	24577
Contacto-Inicial (INITIAL-CONTACT)	24578

Los valores de 16001 a 16383 y los valores de 32001 a 32767 están reservados para usarse en forma privada entre sistemas.

Los Mensajes de Notificación de Estado DEBEN ser enviados bajo la protección de una SA ISAKMP ya sea: como una carga en el último intercambio del Modo Principal; o dentro de un Intercambio Informativo separado después de completarse el procesamiento del Modo Principal o el del Modo Agresivo; o como una carga dentro de cualquier intercambio de Modo Rápido. Estos mensajes NO DEBEN ser enviados dentro de un intercambio de Modo Agresivo, puesto que el Modo Agresivo no proporciona la protección necesaria para vincular el Mensaje de Notificación de Estado con el intercambio.

Nota: Una carga de Notificación está completamente protegida en Modo Rápido solo cuando la carga entera es incluida dentro del resumen (digest) HASH. En Modo Principal, la carga de Notificación es encriptada, esta no se incluye dentro del resumen HASH. Como resultado, un ataque activo por sustitución sobre el texto cifrado en Modo Principal podría provocar que el

tipo de mensaje de notificación de estado esté corrupto. Esto es así, en general, para el último mensaje de cualquier intercambio en Modo Principal. Mientras que existe menor riesgo de que el mensaje de notificación corrupto pueda causar que el receptor aborte la negociación entera pensando que el emisor encontró un error fatal.

Nota de Implementación: El protocolo ISAKMP no garantiza la entrega de los mensajes de Notificación de Estado cuando son enviados en un Intercambio Informativo ISAKMP. Para garantizar la recepción de cualquier mensaje, el emisor DEBERÍA incluir una Carga de Notificación en un intercambio de Modo Principal o de Modo Rápido, el cuál es protegido por un tiempo de retransmisión.

9.3.1 Tiempo de Vida del Respondedor

El mensaje de estado, Tiempo de Vida del Respondedor (RESPONDER-LIFETIME) se puede utilizar para comunicar el tiempo de vida de la SA IPsec seleccionado por el respondedor.

Cuando está presente, la Carga de Notificación DEBE tener el siguiente formato:

- Longitud de la Carga: determinado por la longitud de la carga más el tamaño de los datos (variable)
- DOI: determinado por el DOI de IPsec (1)
- Identificador de Protocolo: determinado por el Identificador de Protocolo seleccionado a partir de la SA seleccionada.
- Tamaño del SPI: determinado por los dieciséis (16)octetos (los dos cookies ISAKMP de ocho octetos) o por los cuatro (4) octetos (del SPI IPsec).
- Tipo de Mensaje de Notificación: determinado por el RESPONDER-LIFETIME (Ver Sección 9.3)
- SPI: Determinado por los dos cookies de ISAKMP o por los SPI IPsec entrantes del emisor.
- Datos de Notificación: contiene una lista de atributos ISAKMP con el/los tiempo/s de vida real de la SA del respondedor.

Nota de Implementación: decir que el campo Datos de Notificación contiene una lista de atributos es equivalente a decir que el campo Datos de Notificación tiene una longitud cero y la Carga de Notificación tiene una lista de atributos asociados.

9.3.2 Estado del Anti-Replay

El mensaje de estado, Estado del Anti-replay (REPLAY-STATUS) se puede utilizar para la confirmación positiva de la elección del respondedor de realizar o no la detección del anti-replay.

Cuando está presente, la Carga de Notificación DEBE tener el siguiente formato:

- Longitud de la Carga: determinado por la longitud de la carga más el tamaño de los datos (4)
- DOI: determinado por el DOI de IPsec (1)
- Identificador de Protocolo: determinado por el Identificador de Protocolo seleccionado a partir de la SA seleccionada.
- Tamaño del SPI: determinado por los dieciséis (16) octetos (los dos cookies ISAKMP de ocho octetos) o por los cuatro (4) octetos (del SPI IPsec).

- Tipo de Mensaje de Notificación: determinado por el REPLAY-STATUS
- SPI: Determinado por los dos cookies de ISAKMP o por los SPI IPsec entrantes del emisor.
- Datos de Notificación: un valor de 4 octetos:
  0 = detección de anti-replay desactivado
  1 = detección de anti-replay activado

9.3.3 Contacto Inicial

El mensaje de estado, Contacto Inicial (INITIAL-CONTACT) se puede utilizar cuando un lado desea informar a la otra parte que esta es la primera SA establecida con el sistema remoto. El receptor de este Mensaje de Notificación puede entonces escoger suprimir alguna de sus SA existentes que tiene para el sistema emisor bajo la suposición de que el sistema del emisor ha reiniciado y ya no tiene acceso a sus SA originales y a su material clave asociado. Cuando se usa, el contenido del campo, Datos de Notificación, DEBERÍA ser nulo (es decir la Longitud de la Carga debería estar determinada por la longitud de la Carga de Notificación)

Cuando está presente, la Carga de Notificación DEBE tener el siguiente formato:

- Longitud de la Carga: determinado por la longitud de la carga más el tamaño de los datos (0)
- DOI: determinado por el DOI de IPsec (1)
- Identificador de Protocolo: determinado por el Identificador de Protocolo seleccionado a partir de la SA seleccionada.
- Tamaño del SPI: determinado por los dieciséis (16) octetos (los dos cookies ISAKMP de ocho octetos).
- Tipo de Mensaje de Notificación: determinado por el INITIAL-CONTACT
- SPI: Determinado por las dos cookies de ISAKMP.
- Datos de Notificación: "no está incluido"

10. **Consideraciones de la IANA**

Este Capítulo contiene varios números que son mantenidos por la IANA. Todos los valores definidos no explícitamente en secciones anteriores están reservados por la IANA.

11. **Conclusiones**

Este capítulo se aplica al protocolo de Intercambio de Claves en Internet (Capítulo 10), combinado con ISAKMP (Capítulo 7) y con Oakley (Capítulo 9) para proporcionar la obtención del material clave de forma segura y autentificada.

El Protocolo de Gestión de Claves y Asociaciones de Seguridad en Internet (ISAKMP) define un marco para la administración de Asociaciones de Seguridad (SA) y el establecimiento de claves para Internet. Este marco consiste en la definición de intercambios, cargas, y de la elaboración de pautas que suceden dentro de un determinado Dominio de Interpretación (DOI). Este capítulo define el DOI de Seguridad IP (DOI de IPsec), que lo ejemplifica ISAKMP para usarse con IP cuando IP use ISAKMP para negociar asociaciones de seguridad.

Capítulo 9

El Protocolo OAKLEY

1. **Introducción**

El establecimiento de clave es la parte más importante de la protección de los datos que depende de la criptografía, y es un componente esencial del paquete de mecanismos de protección descriptos en el Capítulo 2. Este capítulo muestra el protocolo OAKLEY el cual proporciona un mecanismo de distribución de claves escalable y seguro para Internet y fuerza criptográfica.

El algoritmo de intercambio de clave Diffie-Hellman (vea el Capítulo 5) proporciona tal mecanismo. Permite que dos partes acuerden sobre un valor compartido sin que se requiera encriptación. El valor compartido está inmediatamente disponible para usarse en conversaciones subsiguientes encriptadas, por ejemplo transmisión de datos y/o autentificación. El protocolo STS [STS] es un ejemplo de cómo incluir el algoritmo en un protocolo de seguridad, primero garantizar que además de compartir un secreto de manera segura, las dos partes puedan estar seguras de las identidades de cada una de las partes, aún cuando exista un atacante activo.

El protocolo OAKLEY se relaciona con STS, en que comparten la similitud de la autentificación de la exponencial de Diffie-Hellman y lo usan para determinar una clave compartida, y para conseguir Perfect Forward Secrecy para las claves compartidas, pero se diferencia del protocolo STS en:

- La adición de un mecanismo de validación de direcciones "cookies" para ayudar a evitar ataques de denegación de servicio.
- Permite que las dos partes seleccionen juntas los algoritmos soportados para el protocolo: el método de encriptación, el método de obtención de claves y el método de autentificación.
- La autentificación no depende de la encriptación usando el exponencial de Diffie-Hellman; en vez de eso, la validación de la autentificación la vincula las exponenciales de las identidades de las partes.
- El protocolo no requiere que las dos partes calculen los exponenciales compartidos antes de la autentificación.
- Este protocolo agrega seguridad adicional a la obtención de claves usada con la encriptación (en comparación con la autentificación) incluyendo una dependencia de un algoritmo adicional. La obtención de claves para la encriptación se realiza en dependencia no sólo del algoritmo de Diffie-Hellman, sino también del método criptográfico usado para garantizar la autentificación de las partes que se comunican entre si.
- Finalmente, este protocolo define explícitamente como dos partes pueden seleccionar estructuras matemáticas (grupos de representación y operación) para realizar el algoritmo de Diffie-Hellman; las partes pueden utilizar grupos estándares o definir sus propios grupos. Los grupos definidos por el usuario proporcionan un grado adicional de seguridad a largo plazo.

OAKLEY tiene varias opciones para la distribución de las claves. Además del intercambio clásico de Diffie-Hellman, este protocolo se puede utilizar para derivar una nueva clave de una clave existente y para distribuir una clave

externamente derivada por medio de su encriptación.
El protocolo permite que dos partes utilicen todas o algunas de las características del anti-saturación (anti-clogging) y del Perfect Forward Secrecy. También permite el uso de autentificación basado en encriptación simétrica o en algoritmos sin encriptación. Esta flexibilidad es incluida para permitir que las partes usen las características más adecuadas a sus requisitos de seguridad y desempeño.

Para poder comprender este capítulo es esencial que el lector haya comprendido los capítulos precedentes en especial el Capítulo 5. Uno de los objetivos de este capítulo es sentar las bases para poder comprender el Capítulo 10 que deriva en gran parte de este capítulo.

2. **Esquema del Protocolo**

2.1 Observaciones Generales

El protocolo OAKLEY se utiliza para establecer una clave compartida con un identificador asignado y para asociar las identidades autentificadas por las dos partes. El nombre de la clave se puede utilizar más adelante para obtener las SA para los protocolos AH (ver el Capítulo 3) y ESP (ver el Capítulo 4) o para conseguir otros objetivos de seguridad.

Cada clave está asociada con los algoritmos utilizados para la autentificación, privacidad, y con las funciones unidireccionales. Éstos son algoritmos auxiliares para OAKLEY; su aparición en definiciones subsiguientes de SA obtenidas con otros protocolos no es requerida, como así tampoco se prohíbe.

Los tokens anti-saturación, o "cookies", proporcionan una forma de identificar la dirección de origen para ambas partes; el intercambio de cookies puede ser completado antes de que las partes realicen el costoso cálculo del protocolo (una exponenciación de un número entero igual o mayor a diez)

Es importante observar que OAKLEY utiliza las cookies para dos propósitos: anti-saturación y para el nombrado de claves. Las dos partes contribuyen con una cookie cada una en el inicio del establecimiento de la clave; el par de cookies se convierte en el identificador de clave (KEYID), un nombre reutilizable para el material clave. Debido a este rol dual, utilizaremos la notación para la concatenación de las cookies ("COOKIE-I, COOKIE-R") indistintamente mediante el símbolo "KEYID".

OAKLEY está diseñado para ser un componente compatible del protocolo ISAKMP [ISAKMP], que se ejecuta sobre el protocolo UDP usando el puerto 500 (véase el RFC sobre las asignaciones de puertos, STD02-RFC-1700). El único requisito técnico para el entorno del protocolo es que la pila de protocolos subyacente debe poder proveer la dirección de Internet de la parte remota para cada mensaje. Así, OAKLEY se podría, en teoría, utilizar directamente sobre el protocolo IP o sobre el UDP, si el protocolo adecuado o número de puerto asignado está disponible.

El sistema que ejecuta OAKLEY debe proporcionar un buen generador de números aleatorios (vea el Capítulo 5), según lo descripto en [RANDOM], como el origen de los números aleatorios requeridos en esta descripción del protocolo. Cualquier nombramiento de un "nonce" implica que el valor del nonce es generado por tal generador. Lo mismo ocurre en el caso de valores "seudo-aleatorios".

2.2 Notación

Esta sección describe la notación usada en este capítulo para las secuencias y contenido de los mensajes.

2.2.1 Descripciones de Mensajes

Los intercambios del protocolo se escriben en notación abreviada con la intención de expresar los elementos esenciales del intercambio de manera clara.

A fin de representar intercambios de mensajes, en este capítulo se utiliza la notación abreviada que describe cada mensaje en términos de su origen y destino y campos relevantes.

Las flechas ("->") indican que el mensaje es enviado del iniciador al respondedor, o viceversa ("<-").

Los campos en el mensaje se nombran y se separan por una coma ",". El protocolo utiliza la convención de que al principio varios campos constituyen un formato fijo de cabecera para todos los mensajes.

Por ejemplo, considere un intercambio de mensajes hipotético que involucre un mensaje con formato fijo, los cuatro campos fijos son las dos "cookies", el tercer campo es un nombre de tipo de mensaje, el cuarto campo es un número entero de precisión múltiple que representa una potencia de un número:

Iniciador		Respondedor
->	Cookie-I, 0, OK_KEYX, g^x	->
<-	Cookie-R, Cookie-I, OK_KEYX, g^y	<-

La notación describe una secuencia de dos mensajes. El iniciador comienza enviando un mensaje con 4 campos al respondedor; el primer campo tiene el valor de "Cookie-I" sin especificar, el segundo campo tiene el valor numérico 0, el tercer campo indica que el tipo de mensaje es OK_KEYX, el cuarto valor es un elemento de grupo abstracto g elevado a la potencia x.

La segunda línea indica que el respondedor contesta con valor "Cookie-R" en el primer campo, una copia del valor "Cookie-I" en el segundo campo, el tipo de mensaje OK_KEYX, y el número g elevado a la potencia y.

El valor OK_KEYX está en mayúsculas para indicar que es una constante única.

Los números enteros de precisión variable con longitud cero son valores no válidos (nulos) para el protocolo.

Algunas veces el protocolo indicará que una carga entera (generalmente la Carga de Intercambio de Claves) tiene valor nulo. La carga todavía está presente en el mensaje, con el fin de simplificar el análisis.

2.2.2 Guía de Símbolos

Cookie-I y Cookie-R (o CKY-I y CKY-R)
: Son números seudo-aleatorios de 64 bits. El método de generación debe asegurarse con alta probabilidad de que los números usados para cada dirección IP remota sean únicos sobre un cierto período de tiempo, tal como una hora.

KEYID
: Es la combinación de las cookies del iniciador y del respondedor y del dominio de interpretación; este es el nombre del material clave.

sKEYID
: se utiliza para indicar el material clave designado por el KEYID. Nunca se transmite, pero se utiliza en varios cálculos realizados por las dos partes.

OK_KEYX y OK_NEWGRP
: Son tipos de mensaje distintos.

IDP
: Es un bit que indica si el material después de los limites de la encriptación, es o no encriptado (véase la Sección 9). NIDP significa no encriptados.

NIDP
: Significa que la opción PFS para ocultar las identidades no es usada. Es decir, las identidades no son encriptadas usando una clave basada en g^xy.

g^x y g^y
: Es la codificación de los grupos elementales, donde g es un elemento especial del grupo indicado en la descripción del grupo (véase la Sección 8) y g^x indica que el elemento esta elevado a la potencia x. El tipo de codificación es un número entero de precisión variable o un par de tales números enteros, según lo indicado en la operación del grupo dentro de la descripción del grupo. Observe que escribiremos g^xy como abreviatura para g^(xy). Véase la Sección 13 para referencias que describen la implementación del cálculo de un número entero igual o mayor a diez y la relación entre varias definiciones de grupo y operaciones aritméticas básicas.

EHAO
: Es una lista de opciones de encriptación/hash/autentificación. Cada ítem es un par de valores: un nombre de clase y un nombre de algoritmo.

EHAS
: Es un conjunto de tres ítems seleccionados a partir de la lista de EHAO, uno de cada clase para la encriptación, el hash, y la autentificación.

GRP
: Es un nombre (un valor de 32 bits) para el grupo y sus parámetros relevantes: el tamaño de los números enteros, la aritmética operacional, y el elemento generador. Hay algunos GRP predefinidos (para grupos exponenciales modulares de 768 bits, de 1024 bits, de 2048 bits, curvas elípticas de 155 bits y 210 bits, véase la Sección 12), pero los participantes pueden compartir otras descripciones de grupo en una etapa posterior al protocolo (véase la Sección NUEVO GRUPO). Es importante separar la noción del GRP del descriptor de grupo (véase la Sección 8); el primero es un nombre para el segundo.

La barra vertical "|"
: Se utiliza para denotar la concatenación de cadenas de bits. Los campos se concatenan usando su forma codificada como aparece en su carga.

Ni y Nr
Son los nonces seleccionados por el iniciador y el respondedor, respectivamente.

ID(I) y ID(R)
Son las identidades usadas en la autentificación del iniciador y del respondedor respectivamente.

E{x}Ki
Indica la encriptación de x usando la clave pública del iniciador. La encriptación se realiza usando el algoritmo asociado con el método de autentificación; éste será generalmente RSA.

S{x}Ki
Indica la firma sobre x usando la clave privada (clave firmada) del iniciador. La firma se realiza usando el algoritmo asociado al método de autentificación; éste será generalmente RSA o DSS.

prf(a, b)
Denota el resultado de aplicar la función seudo-aleatoria "a" a los datos "b". Uno puede pensar en "a" como clave o como un valor que caracteriza a la función seudo aletoría (prf); en el segundo caso este es un índice hacia una familia de funciones. Cada función en la familia proporciona un "hash" o mezcla unidireccional de la entrada.

prf(0, b)
Denota la aplicación de una función unidireccional a los datos "b". La semejanza con la notación anterior es deliberada e indica que un solo algoritmo, por ejemplo MD5, podría ser usado para ambos propósitos. En el primer caso una "clave" MD5 de transformación sería usada con la clave "a"; en el segundo caso la transformación tendría el valor de clave fijado a cero, resultando en una función unidireccional.

La especificación de los detalles de cómo aplicar un algoritmo a los datos se llama "transformación". Las transformaciones fueron definidas y explicadas en los capítulos anteriores.

2.3 El Esquema General de los Mensajes de Intercambio de Claves

La meta del procesamiento del intercambio de claves es el establecimiento seguro del estado común de la información clave en las dos partes. Esta información de estado es un nombre de clave, material clave secreto, la identificación de las dos partes y tres algoritmos para usarse durante la autentificación: encriptación (para la privacidad de las identidades de las dos partes), hashing (una función seudo-aleatoria para proteger la integridad de los mensajes y para la autentificación de los campos del mensaje), y autentificación (el algoritmo en el cual la autentificación mutua de las dos partes se basa). Las codificaciones y los significados para estas opciones se presentan en la Sección 9.

El intercambio en modo principal tiene cinco características opcionales: intercambio de cookies sin estado, perfect forward secrecy para el material clave, secreto para las identidades, perfect forward secrecy para el secreto de las identidades, uso de firmas (para no repudio). Las dos partes pueden utilizar cualquier combinación de estas características.

La descripción general del proceso es: el iniciador del intercambio comienza por especificar tanta información como él lo desea en su primer mensaje. El

respondedor contesta, suministrando tanta información como él lo desee. Las dos partes intercambian mensajes, proveyendo cada vez más información, hasta que satisfagan sus requisitos.

La opción de cuánta información se incluye en cada mensaje depende de qué opciones son las que se desean. Por ejemplo, si las cookies sin estado, el secreto de identidad, y el perfect forward secrecy para el material clave no se requieren, pero la firma (para no repudio) es requerida, entonces el intercambio puede ser completado en tres mensajes.

Características adicionales pueden aumentar el número de viajes de ida y vuelta necesarios para la determinación del material clave.

ISAKMP proporciona los campos para especificar los parámetros de la SA a usarse con los protocolos AH y ESP. Los tipos de carga SA se especificaron en el Capítulo 7; los tipos de carga pueden ser protegidos mediante el material clave y algoritmos de OAKLEY.

2.3.1 Campos Esenciales de los Mensajes de Intercambio de Claves

Hay 12 campos en un mensaje de intercambio de claves OAKLEY. No todos los campos son relevantes en todos los mensajes; si un campo no es relevante este puede tener un valor nulo (o no válido, o no debe tenerse en cuenta ese valor, por simplicidad de ahora en adelante me referiré a el como de valor "null") o no estar presente (no carga).

Campos de los Mensajes de Intercambio de Claves	
Campo	Significado
CKY-I	Cookie del originador
CKY-R	Cookie del respondedor
MSGTYPE	Para el intercambio de claves el Tipo de Mensaje (MSGTYPE), debe ser ISA_KE&AUTH_REQ o ISA_KE&AUTH_REP; para la definición de un nuevo grupo debe ser ISA_NEW_GROUP_REQ o ISA_NEW_GROUP_REP
GRP	El nombre del grupo de Diffie-Hellman usado para el intercambio
g^x (o g^y)	Representa un número entero de longitud variable o una potencia de un grupo generador
EHAO o EHAS	Función de autentificación, hash, encriptación, ofrecida y seleccionada, respectivamente
IDP	Un indicador de que si la encriptación con g^xy sigue o no (el perfect forward secrecy para las identidades)
ID(I)	La identidad para el iniciador
ID(R)	La identidad para el respondedor
Ni	Nonce suministrado por el iniciador
Nr	Nonce suministrado por el respondedor

La construcción de las cookies es dependiente de la implementación. Pero se recomienda que las cookies sean el resultado de aplicar una función unidireccional a un valor secreto (cambiado periódicamente), la dirección IP local y remota, y el puerto UDP local y remoto. De esta manera, las cookies siguen sin tener estado y expiran periódicamente. Observe que con OAKLEY, esto causaría que las KEYID derivadas del valor secreto también expiren, haciéndose necesaria la renovación de cualquier información de estado asociada a él.

A fin de dar soporte a claves pre-distribuidas, se recomienda que las implementaciones reserven una cierta parte del área de su cookie para claves permanentes. La codificación de éstas solamente depende de la implementación.

Las funciones de encriptación usadas con OAKLEY deben ser transformaciones criptográficas que garanticen privacidad y integridad para los datos del mensaje. Usar DES en modo CBC no está permitido.

Las funciones (hash) unidireccionales usadas con OAKLEY deben ser transformaciones criptográficas que se puedan utilizar con cualquier clave hash (seudo-aleatoria) o transformación sin clave.

Donde se indique, los nonces serán números enteros de precisión variable con un valor de entropía que se corresponda con el atributo de la "fuerza" del GRP usado en el intercambio. Si no se indica ningún GRP, los nonces deben tener por lo menos una longitud de 90 bits. El generador seudo-aleatorio para el material nonce debería empezar con datos iniciales que tengan al menos 90 bits de entropía; véase el RFC 1750 par más detalles.

2.3.1.1 Consejos sobre el Exponente

Idealmente, los exponentes tendrán por lo menos 180 bits de entropía para cada intercambio de claves. Esto asegura completa independencia del material clave entre dos intercambios (observe que esto se aplica si solamente una de las partes elige un exponente aleatorio). En la práctica, los implementadores pueden desear basarse en varios intercambios de claves sobre la base de un solo valor de 180 bits de entropía y utilizar funciones hash unidireccionales para garantizar que la exposición de una clave no comprometerá a otras. En este caso, una buena recomendación es mantener separados los valores de base para los nonces y las cookies de los valores de bases para los exponentes, y reemplazar el valor base con 180 bits de entropía tan frecuentemente como sea posible.

Los valores 0 y p-1 no se deberían utilizar como valores del exponente; los implementadores deberían estar seguros al controlar estos valores, y deberían también negarse a aceptar los valores 1 y p-1 de las partes remotas (donde p es el número primo usado para definir un grupo modular exponencial).

2.3.2 Asociación de las Estructuras de los Mensajes ISAKMP

A continuación se indica donde podría aparecer cada campo OAKLEY dentro de la estructura de los mensajes ISAKMP. Los campos de las cargas relevantes son la carga SA, la carga de Autentificación (AUTH, el cual es un mecanismo de autentificación genérico, tal como firma (SIG) o hash (HASH)), la carga del Certificado (CERT), y la carga de Intercambio de Claves (KE). Esto solo es una recomendación.

CKY-I	Cabecera ISAKMP
CKY-R	Cabecera ISAKMP
MSGTYPE	Tipo de Mensaje en la cabecera de ISAKMP
GRP	Carga SA, en la sección de la Propuesta
g^x (o g^y)	Carga de Intercambio de Claves, codificado como un número entero de precisión variable
EHAO o EHAS	carga SA, en la sección de la Propuesta
IDP	Un bit en el campo RESERVADO en la cabecera AUTH (de autentificación)
ID(I)	Carga AUTH, Campo Identidad
ID(R)	Carga AUTH, Campo Identidad
Ni	Carga AUTH, Campo Nonce
Nr	Carga AUTH, Campo Nonce
S{...}Kx	Carga AUTH, Campo de Datos
prf{K,...}	Carga AUTH, Campo de Datos

2.4 El Protocolo de Intercambio de Claves

El número y contenido exacto de mensajes intercambiados durante un intercambio de claves OAKLEY depende de qué opciones deseen utilizar el iniciador y el respondedor. Un intercambio de claves puede ser completado en tres o más mensajes, dependiendo de esas opciones.

Los tres componentes del protocolo de determinación de clave son:

1. Intercambio de cookies (opcionalmente, sin estado)

2. Intercambio de la otra parte de la clave de Diffie-Hellman (opcional, pero esencial para el perfect forward secrecy)

3. Autentificación (opcionales: privacidad para las identidades, privacidad para las identidades con perfect forward secrecy, no repudio)

El iniciador puede suministrar tan poca información como una escueta petición de intercambio lo solicite, no llevando información adicional. Por otra parte el iniciador puede comenzar por suministrar toda la información necesaria para que el respondedor autentifique la petición y complete rápidamente la determinación de la clave, si el respondedor opta por este método. Si no, el respondedor puede responder con una mínima cantidad de información (el mínimo es una cookie).

El método de autentificación puede ser mediante firmas digitales, encriptación de clave pública, o una clave simétrica fuera de banda. Los tres métodos conducen a pequeñas variaciones en los mensajes, estas variaciones se describen el los ejemplos subsiguientes de estas sección.

El iniciador es responsable de retransmitir los mensajes si el protocolo no termina a su debido tiempo. Por lo tanto, el respondedor debe evitar desechar la información de la contestación hasta que es reconocida por el iniciador en el transcurso del protocolo.

El resto de esta sección contiene los ejemplos que muestran cómo utilizar las opciones de OAKLEY.

2.4.1 Un Ejemplo Dinámico (o Agresivo)

El ejemplo siguiente nuestra como dos partes pueden completar un intercambio de claves en tres mensajes. Las identidades no son secretas y el material clave obtenido es protegido por el perfect forward secrecy (PFS).

Usando firmas digitales, las dos partes tendrán una prueba de la comunicación que puede ser registrada y presentada posteriormente ante una tercera parte.

El material clave implícito por los grupos exponenciales no se necesita para completar el intercambio. Si se desea posponer el cálculo, las implementaciones pueden guardar el valor de "x" y de "g^y" y clasificarlo como material clave "sin-calcular". El cual puede ser calculado a partir de esta información posteriormente.

Iniciador		Respondedor
->	CKY-I, 0, OK_KEYX, GRP, g^x, EHAO, NIDP, ID(I), ID(R), Ni, 0, S{ID(I) | ID(R) | Ni | 0 | GRP | g^x | 0 | EHAO}Ki	->
<-	CKY-R, CKY-I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID(R), ID(I), Nr, Ni, S{ID(R) | ID(I) | Nr | Ni | GRP | g^y | g^x | EHAS}Kr	<-
->	CKY-I, CKY-R, OK_KEYX, GRP, g^x, EHAS, NIDP, ID(I), ID(R), Ni, Nr, S{ID(I) | ID(R) | Ni | Nr | GRP | g^x | g^y | EHAS}Ki	->

Nota: "NIDP" significa que la opción PFS para ocultar las identidades no es usada. Es decir, las identidades no son encriptadas usando una clave basada en g^xy.

Nota: Los campos se muestran separados por comas en este documento; cuando exista concatenaciones en los mensajes del protocolo actual se usará su forma codificada como se define en el Capítulo 10.

El resultado de estos intercambios es un clave con KEYID = CKY-I|CKY-R y de valor:

sKEYID = prf(Ni | Nr, g^xy | CKY-I | CKY-R)

El esquema de procesamiento para este intercambio es:

Inicio
El iniciador genera un cookie único relacionado con la dirección IP esperada por el respondedor y selecciona la información de estado: el GRP (el identificador de grupo), un exponente x seleccionado seudo-aleatoriamente, g^x, una lista EHAO, el nonce y las identidades. La primera opción de autentificación en la lista EHAO es un algoritmo que soporta firmas digitales, el cual es usado para firmar las identidades, la identidad del nonce y del grupo. Posteriormente el iniciador observa que la clave está en el estado inicial "sin autentificar" y se fija un tiempo para posibles retransmisiones y/o finalización de la petición.

Cuando el respondedor recibe el mensaje, puede elegir ignorar toda la información y tratarla simplemente como una respuesta para una cookie, creada sin estado. Si CKY-I no es previamente usada por la dirección de origen en la cabecera IP, el respondedor genera una cookie única, CKY-R. El siguiente paso depende de las preferencias del respondedor. La respuesta

mínima requerida es contestar con el primer campo de la cookie fijado en cero y CKY-R en el segundo campo. Para este ejemplo se asumirá que el respondedor es más dinámico (para las alternativas, vea la Sección 6) y se acepta lo siguiente:

un grupo con identificación GRP, primera opción de autentificación (la cual debe ser una firma digital usada para firmar los mensajes del iniciador), falta de perfect forward secrecy para el procesamiento de las identidades, del iniciador ID(I) y del respondedor ID(R).

En este ejemplo el respondedor decide aceptar toda la información ofrecida por el iniciador. La validación de la firma sobre la parte del mensaje firmado, y la relación del par (CKY-I, CKY-R) con la siguiente información de estado:

la dirección de red de origen y destino de los mensajes

la clave de estado "no autentificada"

el primer algoritmo de autentificación ofrecido

el grupo GRP, un valor del exponente "y" en el grupo GRP, y el g^x del mensaje

el nonce Ni y un valor Nr seleccionado seudo-aleatoriamente

un tiempo para posibles destrucciones del estado.

El respondedor calcula g^y, forma el mensaje de contestación, y firma la información de identificación y de nonce con la clave privada ID(R) y lo envía al iniciador. En todos los intercambios, cada parte debe cerciorarse de que ninguno de los dos ofrezca o valide el 1 (es decir, g^0, dado que g^0 = 1) o el g^(p-1) como exponencial.

En este ejemplo, para agilizar el protocolo, el respondedor implícitamente acepta el primer algoritmo en la clase de Autentificación de la lista EHAO. Esto se debe a que él no puede validar la firma del iniciador sin aceptar el algoritmo para realizar la firma. La lista EHAS del respondedor también reflejará su aceptación.

El iniciador recibe el mensaje de contestación y confirma que el CKY-I sea una asociación válida para la dirección de red del mensaje entrante,

agrega el valor CKY-R al estado para el par (CKI-I, dirección de red), asocia toda la información de estado con el par (CKY-I, CKY-R),

valida la firma del respondedor de la información del estado (si la validación falla, el mensaje es descartado)

agrega g^y para esta información de estado,

guarda el EHA seleccionado en el estado,

opcionalmente calcula g^xy (esto puede ser diferido hasta después de enviar el mensaje de contestación),

envía el mensaje de contestación, firma con la clave pública ID(I),

marca el KEYID (CKY-I|CKY-R) como autentificado,

y crea el mensaje de contestación y lo firma.

Cuando el respondedor recibe el mensaje del iniciador, y si la firma es válida, este marca la clave como estando en el estado autentificado. Se debería calcular g^xy y asociar esta con KEYID.

Observe que aunque el PFS para la protección de las identidades no se use, el PFS para la obtención del material clave debe estar presente debido a que se está intercambiando la otra parte de la clave de Diffie-Hellman g^x y g^y.

Aunque el respondedor solo acepta parte de la información del iniciador, el iniciador considerará que el protocolo esta en progreso. El iniciador debería asumir que los campos que no fueron aceptados por el respondedor no fueron registrados por el respondedor.

Si el respondedor no acepta el intercambio agresivo (dinámico) y selecciona otro algoritmo para la función A, entonces el protocolo no continuará usando el algoritmo firmado o el valor firmado del primer mensaje.

2.4.1.1 Campos Ausentes

Si el respondedor no acepta todos los campos ofrecidos por el iniciador, el respondedor debería incluir valores null para esos campos en su respuesta. La Sección 6 tiene pautas sobre cómo seleccionar los campos "de izquierda a derecha". Si un campo no es aceptado, entonces ese campo y todos los campos siguientes deben tener valores null.

El respondedor no debe registrar ningún tipo de información que él no haya aceptado. Si sus identificadores y nonces tienen valores nulos, no habrá una firma sobre esos valores nulos.

2.4.1.2 Firma Mediante Funciones Seudo-Aleatorias

El ejemplo agresivo esta escrito sugiriendo que la tecnología de clave pública se utiliza para las firmas. Sin embargo, una función seudo-aleatoria puede ser utilizada, si las partes previamente han convenido tal esquema y tienen una clave compartida.

Si la primera propuesta en la lista EHAO es un método de "clave existente", entonces el KEYID designado en esa propuesta suministrará el material clave para la "firma" el cual se calcula usando el algoritmo "H" asociado con el KEYID.

Suponga que la primera propuesta en EHAO es una
CLAVE-EXISTENTE, 32
y el algoritmo "H" para KEYID 32 es MD5-HMAC, por la negociación anterior. El material clave es una cadena de bits, llamado sK32. Entonces en el primer mensaje en el intercambio agresivo, donde la firma

S{ID(I), ID(R), Ni, 0, GRP, g^x, EHAO}Ki

se indica, el cálculo de la firma será realizado por
MD5-MAC_func(KEY=sK32, DATA = ID(I) | ID(R) | Ni | 0 | GRP | g^x | g^y | EHAO) (la definición exacta del algoritmo correspondiente a la "función-MD5-HMAC" aparece en el Capítulo 6 el cual define dicha transformación).

El resultado de este cálculo aparece en la carga de Autentificación.

2.4.2 Un Ejemplo Agresivo con Identidades Ocultadas

El siguiente ejemplo muestra cómo dos partes pueden completar un intercambio de claves sin usar firmas digitales. La criptografía de clave pública, oculta las identidades durante la autentificación. El grupo exponencial se intercambia y se autentifica, sin ser necesario que el material clave implícito (g^xy) se intercambie durante el intercambio.

Este intercambio tiene una diferencia importante del esquema de firmas anterior-- en el primer mensaje, la identidad para el respondedor se indica en texto plano: ID(R'). Sin embargo, la ocultación de la identidad en la criptografía de clave pública es diferente: ID(R). Esto se debe a que el iniciador debe de alguna manera decirle al respondedor qué par de claves pública/privada utilizará para la desencriptación, pero al mismo tiempo, la identidad se oculta con la encriptación de esa clave pública.

El iniciador puede elegir renunciar al secreto de la identidad del respondedor, pero esto es indeseable. En cambio, si hay una identidad bien conocida por el nodo respondedor, la clave pública de esa identidad puede ser usada para encriptar la identidad actual del respondedor.

Iniciador		Respondedor
->	CKY-I, 0, OK_KEYX, GRP, g^x, EHAO, NIDP, ID(R'), E{ID(I), ID(R), E{Ni}Kr}Kr'	->
<-	CKY-R, CKY-I, OK_KEYX, GRP, g^y, EHAS, NIDP, E{ID(R), ID(I), Nr}Ki, prf(Kir, ID(R) | ID(I) | GRP | g^y | g^x | EHAS)	<-
->	CKY-I, CKY-R, OK_KEYX, GRP, 0, 0, NIDP, prf(Kir, ID(I) | ID(R) | GRP | g^x | g^y | EHAS)	->

Kir = prf(0, Ni | Nr)

Nota: "NIDP" significa que la opción PFS para ocultar las identidades no es usada.

Nota: el valor ID(R') se incluye en la carga de Autentificación como se describe en el Sección 9.

El resultado de estos intercambio es una clave con KEYID = CKY-I|CKY-R y valor:

sKEYID = prf(Ni | Nr, g^xy | CKY-I | CKY-R)

El esquema de procesamiento para este intercambio es:

Inicio
El iniciador genera un cookie único relacionado con la dirección IP esperada por el respondedor, y selecciona la información de estado: GRP, g^x, una lista EHAO. La primera opción de autentificación en la lista EHAO es un algoritmo que soporta encriptación de clave pública. El iniciador también designa dos identidades a ser utilizadas para la conexión e ingresa éstos en el estado. Una identidad bien conocida para la máquina del respondedor es también elegida, y la clave pública para esta identidad se utiliza para encriptar el nonce Ni y las dos identidades de conexión. Posteriormente el iniciador observa que la clave esta en el estado inicial "sin autentificar" y se fija un tiempo

para posibles retransmisiones y/o finalización de la petición.

Cuando el respondedor recibe el mensaje, puede elegir ignorar toda la información y tratarla simplemente como una respuesta para una cookie, creada sin estado.
Si CKY-I no es previamente usada por la dirección de origen en la cabecera IP, el respondedor genera una cookie única, CKY-R. El siguiente paso depende de las preferencias del respondedor. La respuesta mínima requerida es contestar con el primer campo de la cookie fijado en cero y CKY-R en el segundo campo. Para este ejemplo se asumirá que el respondedor es más dinámico y se acepta lo siguiente:

grupo GRP, primera opción de autentificación (el cual debe ser un algoritmo de encriptación de clave pública usado para encriptar la carga), falta de perfect forward secrecy para el procesamiento de las identidades, del iniciador ID(I) y del respondedor ID(R)

El respondedor debe desencriptar la identificación y la información del nonce, usando la clave privada para la identificación del respondedor (R). Después de esto, la clave privada para la identificación del respondedor será utilizada para desencriptar el campo del nonce.

Ahora el respondedor asocia el par (CKY-I, CKY-R) con la siguiente información de estado:

la dirección de red de origen y destino de los mensajes

la clave de estado "no-autentificada"

el primer algoritmo de cada clase en la lista EHAO (algoritmos ofrecidos para la encriptación, el hash y la autentificación)

grupo GRP y una "y" y un valor g^y en el grupo GRP

el nonce Ni y un valor Nr seleccionado seudo-aleatoriamente

un tiempo para posibles destrucciones del estado.

Luego el respondedor encripta la información de estado con la clave pública ID(I), construye el valor prf, y lo envía al iniciador.

El iniciador recibe el mensaje de contestación y confirma que el CKY-I sea una asociación válida para la dirección de red del mensaje entrante,

agrega el valor CKY-R al estado para el par (CKI-I, dirección de red), asocia toda la información de estado con el par (CKY-I, CKY-R),

desencripta la información de identificación y nonce

comprueba el prf calculado (si la comprobación falla, el mensaje es descartado)

agrega g^y para esta información de estado,

guarda el EHA seleccionado en el estado,

opcionalmente calcula g^xy (esto puede ser diferido), y

envía el mensaje de contestación, encripta con la clave pública ID(I), y

marca el KEYID (CKY-I|CKY-R) como autentificado.

Cuando el respondedor recibe este mensaje, este marca la clave como estando en el estado autentificado. Si todavía no lo hace, debería calcular g^xy y asociar esta con KEYID.

El material clave secreto es: sKEYID = prf(Ni | Nr, g^xy | CKY-I | CKY-R)

Observe que aunque el PFS para la protección de las identidades no se use, el PFS para la obtención del material clave debe estar presente debido a que se está intercambiando la otra parte de la clave de Diffie-Hellman g^x y g^y.

2.4.3 Un Ejemplo Agresivo con Identidades Privadas y sin Diffie-Hellman

Considere el costo computacional que se puede evitar si el perfect forward secrecy no se requiriese para la derivación del material clave. Las dos partes pueden intercambiar nonces y partes de las claves secretas para lograr la autentificación y obtener el material clave. La privacidad a largo plazo de la protección de los datos por medio del material clave derivado dependerá de las claves de cada una de las partes.

En este intercambio, el GRP tiene el valor 0 y el campo para el grupo exponencial se utiliza para soportar un valor de nonce.

Como en la sección anterior, el primer algoritmo propuesto debe ser un sistema de encriptación de clave pública; respondiendo con una cookie y un campo exponencial diferente a cero, el respondedor acepta implícitamente la primera propuesta y la carencia de perfect forward secrecy para las identidades y para el material clave derivado.

Iniciador		Respondedor
->	CKY-I, 0, OK_KEYX, 0, 0, EHAO, NIDP, ID(R'), E{ID(I), ID(R), sKi}Kr', Ni	->
<-	CKY-R, CKY-I, OK_KEYX, 0, 0, EHAS, NIDP, E{ID(R), ID(I), sKr}Ki, Nr, prf(Kir, ID(R) | ID(I) | Nr | Ni | EHAS)	<-
->	CKY-I, CKY-R, OK_KEYX, EHAS, NIDP, prf(Kir, ID(I) | ID(R) | Ni | Nr | EHAS)	->

Kir = prf(0, sKi | sKr)

Nota: los valores sKi y sKr van dentro de los campos del nonce. El cambio en la notación tiene la intención de enfatizar que su entropía es crucial para determinar el material clave.

Nota: "NIDP" significa que la opción PFS para ocultar las identidades no es usada.

El resultado de este intercambio es una clave con KEYID = CKY-I|CKY-R y valor
sKEYID = prf (Kir, CKY-I | CKY-R).

2.4.4 Un Ejemplo Conservador

En este ejemplo las dos partes son poco dinámicas; utilizan el intercambio de cookies para denotar la creación del estado y utilizan perfect forward

secrecy para proteger las identidades. Este ejemplo usa encriptación de clave pública para la autentificación; también se puede usar firmas digitales o claves pre-compartidas, según lo ilustrado anteriormente. Este ejemplo no cambia el uso de los nonces, prfs, etc., pero sí la cantidad de información transmitida en cada mensaje.

El respondedor considera la capacidad del iniciador de repetir CKY-R como una débil evidencia de que el mensaje ha sido originado por el "verdadero" remitente y el remitente esta asociado con la dirección de red del iniciador. El iniciador realiza similares suposiciones cuando el CKY-I se repite en el iniciador.

Todos los mensajes deben tener cookies válidas o por lo menos una cookie cero. Si ambas cookies son cero, esto indica una solicitud de cookie; si solamente la cookie del iniciador es cero, es una respuesta a una solicitud de cookie.

Note que el iniciador y el respondedor deben estar de acuerdo sobre el conjunto de algoritmos EHA; no hay un conjunto para el respondedor y uno para el iniciador. El iniciador debe incluir por lo menos MD5 y DES en la oferta inicial.

Los campos no indicados tienen valores null.

Iniciador		Respondedor
->	0, 0, OK_KEYX	->
<-	0, CKY-R, OK_KEYX	<-
->	CKY-I, CKY-R, OK_KEYX, GRP, g^x, EHAO	->
<-	CKY-R, CKY-I, OK_KEYX, GRP, g^y, EHAS	<-
->	CKY-I, CKY-R, OK_KEYX, GRP, g^x, IDP*, ID(I), ID(R), E{Ni}Kr,	->
<-	CKY-R, CKY-I, OK_KEYX, GRP, 0, 0, IDP, E{Nr, Ni}Ki, ID(R), ID(I), prf(Kir, ID(R) | ID(I) | GRP | g^y | g^x | EHAS)	<-
->	CKY-I, CKY-R, OK_KEYX, GRP, 0, 0, IDP, prf(Kir, ID(I) | ID(R) | GRP | g^x | g^y | EHAS)	->

Kir = prf(0, Ni | Nr)

* cuando se lleva a cabo IDP, las cargas de autentificación se encriptan con el algoritmo de encriptación seleccionado usando el material clave prf (0, g^xy). La transformación define el algoritmo de encriptación que definirá cómo seleccionar los bits del material clave. Esta encriptación esta por encima y después de cualquier encriptación de clave pública. Para más detalle vea la Sección 9.

Note que en los primeros mensajes, varios campos no presentan descripción. Estos campos están presentes con valores nulos.

En el primer intercambio el respondedor puede usar cookies sin estado; si el respondedor genera cookies de un modo determinado que le permite validar sin guardar, entonces esto es posible. Si el iniciador incluye una cookie en su petición inicial, el respondedor aun puede usar cookies sin estado simplemente omitiendo el CKY-I de su respuesta y rechazando registrar la cookie del iniciador hasta que aparezca en un mensaje posterior.

Después de que el intercambio se haya completado, ambas partes calculan el

material clave compartido sKEYID como prf(Ni | Nr, g^xy | CKY-I | CKY-R) donde "prf" es la función seudo-aleatoria en la clase "hash" seleccionada en la lista EHA.

Como en el caso de las cookies, cada parte considera la capacidad de la otra parte de repetir el valor Ni o el de Nr como prueba de que la clave pública de una parte, hable por la parte remota y establezca su identidad.

En el análisis de este intercambio, es importante notar que aunque la opción IDP asegura que las identidades están protegidas con una clave efímera g^xy, la autentificación en sí no depende de g^xy. Es esencial que los pasos de la autentificación validen los valores g^x y g^y, y es imperativo que la autentificación no implique una dependencia circular en ellos. Una tercera parte podría intervenir con un esquema "hombre-en-el-medio" para convencer al iniciador y al respondedor de que utilicen valores diferentes de g^xy; aunque un ataque de este tipo puede dar lugar a revelar la identidad del fisgón, la autentificación podría fallar.

2.4.5 Fuerza Adicional para la Protección de Claves Encriptadas

Los nonces Ni y Nr se utilizan para proporcionar secreto (confidencialidad) adicional en la obtención de claves de sesión. Esto hace que el secreto de la clave dependa de dos problemas diferentes: del problema del logaritmo discreto en el grupo G, y del problema de quebrantamiento del esquema de encriptación del nonce. Si se utiliza la encriptación RSA, entonces este segundo problema es casi equivalente a factorizar las claves públicas RSA del iniciador y del respondedor.

Para la autentificación, el tipo de clave, el método de validación, y los requerimientos de certificación deben ser indicados.

2.5 Identidad y Autentificación

2.5.1 Identidad

En los intercambios OAKLEY el iniciador ofrece la identidad del iniciador y del respondedor-- la primera es la identidad demandada por el iniciador, y la segunda es la identidad solicitada por el respondedor.

Si no se especifica ninguna de las dos identidades, las identidades se toman de los campos toman de las direcciones de origen y destino de la cabecera IP.

Si el iniciador no proporciona una identidad para el respondedor, el respondedor puede contestar nombrando cualquier identidad que la política local permita. El iniciador puede rechazar la aceptación terminando el intercambio.

El respondedor también puede contestar con una identidad diferente de la que sugirió el iniciador; el iniciador puede aceptar esto implícitamente continuando el intercambio o rechazarlo terminando el intercambio (no contestando).

2.5.2 Autentificación

Esta sección procura describir cómo un puñado de estándares se podría incorporar en OAKLEY, sin procurar escoger y elegir entre ellos.

Los siguientes métodos pueden aparecer en ofertas de OAKLEY:

a. Claves pre-compartidas
   Cuando las dos partes han convenido en un método confiable de distribución de claves secretas para su autentificación mutua, este método puede ser utilizado para la autentificación. Esto tiene problemas obvios en sistemas de gran tamaño, pero es una solución intermedia aceptable para algunas situaciones. El soporte para claves pre-compartidas es REQUERIDO.

   La encriptación, el hash, y el algoritmo de autentificación a usarse con una clave pre-compartida, deben ser parte de la información de estado distribuida con la clave.

   Las claves pre-compartidas tienen un KEYID y material clave sKEYID; el KEYID se utiliza en una oferta de opción de autentificación de clave pre-compartida. Puede haber más de una oferta de clave pre-compartida en una lista.

   Debido a que el KEYID persiste a diferentes invocaciones de OAKLEY (después de un fallo de sistema, etc.), este debe ocupar un área reservada del espacio en las dos partes. Algunos bits pueden ser reservados en el "espacio de la cookie" de cada parte para adecuarlo a esto.

   No hay autoridad de certificación para las claves pre-compartidas. Cuando una clave pre-compartida se utiliza para generar una carga de Autentificación (AUTH), la autoridad de certificación es "Ninguna", el Tipo de Autentificación es "Pre-Compartida", y la carga contiene el KEYID, codificado con dos cantidades de 64 bits, y el resultado de aplicar la función hash seudo-aleatoria al cuerpo del mensaje con el sKEYID que forma la clave para la función

b. Claves públicas DNS
   Las extensiones de seguridad del protocolo DNS [DNSSEC] proporcionan una manera conveniente de tener acceso a la información de clave pública, especialmente para las claves públicas asociadas a los hosts. Las claves RSA son requeridas en implementaciones de DNS seguros; extensiones para autorizar claves DSS opcionales es una posibilidad a mediano plazo.

   El registro de CLAVES DNS tiene asociado registros de firmas (SIG) que son firmados por una autoridad de la zona. Los registros SIG indican el algoritmo usado para construir la firma.

   Las implementaciones de OAKLEY deben soportar el uso de registros SIG y de CLAVES DNS para la autentificación de las direcciones IPv4 e IPv6 y nombres de dominio completamente cuantificados. Sin embargo, las implementaciones no requieren soportar ningún algoritmo determinado (RSA, DSS, etc.).

c. Claves públicas RSA con y sin autoridad de certificación de firmas PGP [Zimmerman] utiliza claves públicas con un método informal para establecer confianza. El formato de las claves públicas PGP y los métodos de nombramiento están fuera del alcance de este libro, sin embargo una introducción a esto se puede encontrar en el Capítulo 5. El algoritmo RSA puede ser utilizado con claves PGP para firmar o encriptar; la opción de autentificación podría indicar RSA-SIG o RSA-ENC, respectivamente. El soporte para esto es OPCIONAL.

d. Claves públicas RSA con certificados: Hay varios formatos y convenciones de nombramiento para las claves públicas que son firmadas por una o más autoridades de certificación. El soporte para esto es OPCIONAL.

e. Claves DSS con certificados: La codificación para los Estándares de Firmas Digitales con X.509 se describe en el draft de Internet, draft-ietf-ipsec-dss-cert-00.txt. El soporte para esto es OPCIONAL; un Tipo de Autentificación ISAKMP será asignado.

2.5.3 Validación de Claves Autentificadas

La combinación del algoritmo de Autentificación, la Autoridad de Autentificación, el Tipo de Autentificación y la clave (usualmente la pública) definen la forma de validar los mensajes con respecto a la identidad demandada. La información de la clave estará disponible a partir de una clave pre-compartida, o de algún tipo de autoridad de certificación.

Generalmente la autoridad de certificación produce un certificado vinculado con el nombre de la entidad y una clave pública. Las implementaciones de OAKLEY deben estar preparadas para tomar y validar certificados antes de usar la clave pública para los propósitos de autentificación de OAKLEY.

La Carga de Autentificación de ISAKMP define el campo Autoridad de Autentificación para especificar la autoridad que debe ser visible en la jerarquía de confianza para la autentificación.

Una vez que se obtenga un certificado apropiado (véase la Sección 2.4.3), el método de validación dependerá del Tipo de Autentificación; si es PGP entonces las rutinas de validación de firma PGP se pueden invocar para satisfacer los requerimientos locales de la web de confianza; si es RSA con certificados X.509, el certificado debe ser examinado para comprobar si la firma de autoridad de certificación es válida, y si la jerarquía es reconocida por la política local.

2.5.4 Recuperando la Identidad de los Objetos

Además de interpretar el certificado o la otra estructura de datos que contiene una identidad, los usuarios de OAKLEY deben recuperar los certificados que vinculan una clave pública a un identificador y también recuperar los certificados auxiliares para las autoridades de certificación o co-firmantes (como en el web PGP de confianza).

La Carga de Certificados de ISAKMP puede ser utilizada para adjuntar certificados útiles en los mensajes de OAKLEY. La Carga de Certificados se define en la Sección 9.

El soporte para acceder y revocar certificados de claves públicas por medio del protocolo DNS Seguro [SECDNS] es OBLIGATORIO para las implementaciones de OAKLEY. Otros métodos de extracción pueden ser utilizados cuando la clase AUTH indica una preferencia.

2.6 Interfaz para las Transformaciones Criptográficas

El material clave calculado para el intercambio de claves debería tener por lo menos 90 bits de entropía, esto significa que debe tener por lo menos una longitud de 90 bits.

Las transformaciones utilizadas con OAKLEY deberían tener algoritmos

auxiliares que tomen un número entero de precisión variable y lo conviertan en el material clave de longitud apropiado. Por ejemplo, un algoritmo DES podría tomar los 56 bits de orden inferior y un algoritmo triple DES podría utilizar lo siguiente:

```
K1 = los 56 bits de orden inferior de md5(0|sKEYID)
K2 = los 56 bits de orden inferior de md5(1|sKEYID)
K3 = los 56 bits de orden inferior de md5(2|sKEYID)
```

Las transformaciones serán llamadas por medio del material clave codificado con un número entero de precisión variable, por la longitud de los datos, y el bloque de memoria de los datos. La conversión del material clave en una clave de transformación es responsabilidad de la transformación.

2.7 Retransmisión, Tiempo Agotado y Mensajes de Error

Si el iniciador no recibe una respuesta del respondedor, durante un periodo considerable de tiempo, el iniciador deberá retransmitir el mensaje. Estas retransmisiones deben ser manejadas por ambas partes; el respondedor debe conservar la información para retransmitir hasta que el iniciador se mueva al siguiente mensaje en el protocolo o termine el intercambio.

Los mensajes informativos de error presentan un problema debido a que no pueden ser autentificados solamente usando la información presente en un intercambio incompleto; por esta razón, las partes pueden desear establecer una clave por defecto para los mensajes de error de OAKLEY. La Figura 1, muestra un tipo de mensaje de error de OAKLEY (donde, Inter = Intercambio, Vers = versión)

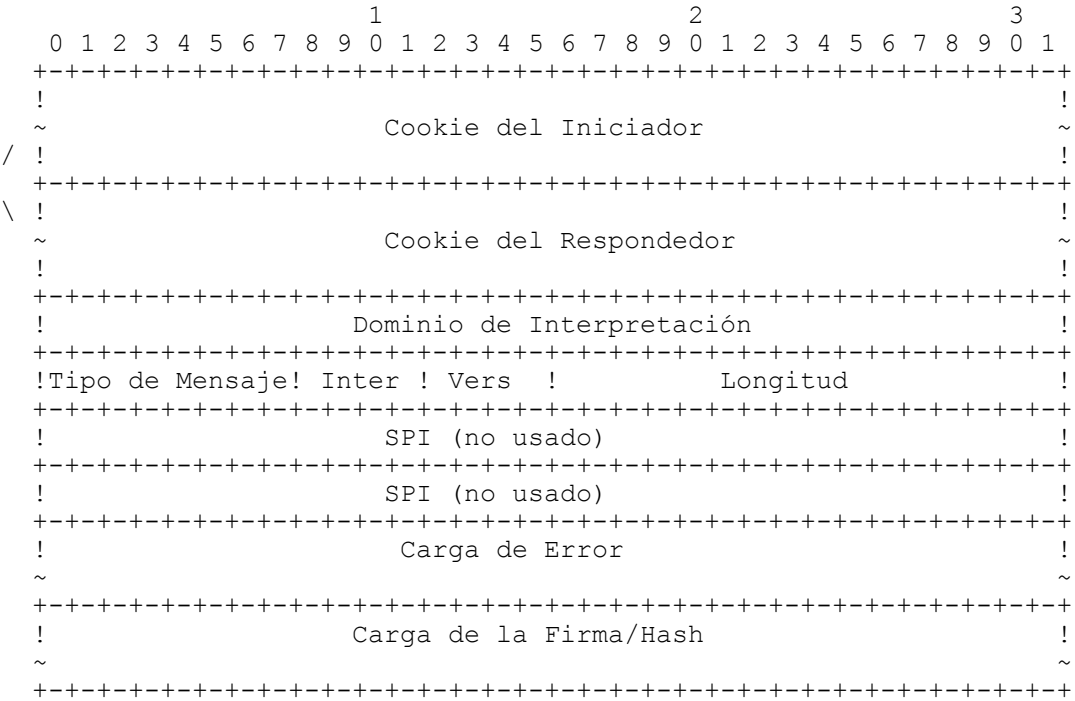


Figura 1: Mensaje de error de Oakley

El mensaje de error contendrá las cookies según lo presentado en el mensaje problemático, el tipo de mensaje OAKLEY_ERROR, y la causa del error, seguida por el mensaje rechazado.

El KEYID proporciona el algoritmo H y la clave para autentificar el contenido del mensaje; este valor es transportado en la carga de la Firma/Hash (Sig/Prf).

La carga de Error contiene el código de error y el contenido del mensaje rechazado.

Los mensajes de error son solamente informativos, y la integridad del protocolo no depende de ellos.

Causas de error:

TIMEOUT:	El tiempo para el intercambio a expirado, destrucción del estado.
AEH_ERROR:	Un algoritmo desconocido aparece en la oferta (propuesta).
GROUP_NOT_SUPPORTED:	GPR designado no soportado.
EXPONENTIAL_UNACCEPTABLE:	Exponencial demasiado pequeño/grande o es "±1
SELECTION_NOT_OFFERED:	La selección no aparece en la oferta.
NO_ACCEPTABLE_OFFERS	Ninguna de las ofertas reúne los requisitos del host.
AUTHENTICATION_FAILURE:	La función hash o firma a fallado.
RESOURCE_EXCEDED:	Demasiados intercambios o demasiados estados informativos.
NO_EXCHANGE_IN_PROGRESS:	Se recibió una respuesta sin que halla una petición en curso

2.8 Seguridad Adicional para las Claves Privadas: Grupos Privados

Si las dos partes necesitan utilizar un esquema de determinación de claves de Diffie-Hellman que no dependa de las definiciones de grupo estándares, estas tienen la opción de establecer un grupo privado. La autentificación no necesita ser repetida, debido a que esta etapa del protocolo será protegida por una clave de autentificación preexistente. Como medida de seguridad adicional, las dos partes establecerán un nombre privado para el material clave compartido, aun si ellas utilizan exactamente el mismo grupo para comunicarse con otros entes, la reutilización no será apreciable por los atacantes pasivos.

Los grupos privados tienen la ventaja de que son mucho más resistentes a extensos ataques pasivos aumentando el número de grupos que tendrían que ser analizados exhaustivamente para recuperar una gran cantidad de claves de sesión. En contraste con el caso de cuando solo uno o dos grupos se utilizan; en ese caso, uno esperaría que los años y años de claves de sesión estarán comprometidos.

Hay dos desafíos técnicos a enfrentar: ¿cómo puede un usuario determinado crear un grupo único y apropiado, y cómo puede una segunda parte asegurarse de que el grupo propuesto es razonablemente seguro?

La seguridad de un grupo exponencial modular depende del factor primo más grande del tamaño del grupo. Para maximizar esto, uno puede elegir números

primos "fuerte" o Sophie Germaine, P = 2Q + 1, donde P y Q son números primos. Pero si P = kQ + 1, donde k es pequeña, entonces la fuerza del grupo sigue siendo considerable. Estos grupos se conocen como subgrupos de Schnorr, y pueden estar basados en un menor esfuerzo computacional que los números primos de Sophie-Germaine.

Los subgrupos de Schnorr también pueden ser eficientemente validados usando pruebas de números primos probables.

Esto también facilita bastante la búsqueda de P, k, y Q de tal manera que puede comprobarse fácilmente que el factor primo más grande es Q.

Estimamos que tomaría cerca de 10 minutos encontrar un nuevo grupo de alrededor de 2^1024 elementos, y este se podría realizar una vez al día por un proceso programado; validar un grupo propuesto por la otra parte tomaría quizás un minuto en una máquina con un procesador RISC de 25 MHz o en una máquina con un procesador CISC de 66 MHz.

Observamos que la validación se hace solamente entre partes previamente autentificadas, y siempre le sigue la definición de un nuevo grupo y la cual está protegido por una clave establecida usando un grupo bien conocido. Hay cinco puntos a tener en cuenta:

a. La descripción y el identificador público para el nuevo grupo es protegido por el grupo bien conocido.

b. El respondedor puede rechazar la tentativa de establecer un nuevo grupo, porque está demasiado ocupado o porque no puede validar el factor primo más grande por ser excesivamente grande.

c. El generador y el módulo nuevo pueden estar en la caché por largos períodos de tiempo; no es seguridad crucial y no necesitan estar asociado con la actividad en curso.

d. La generación de un nuevo valor g^x será cada vez más costosa si hay muchos grupos en la caché; sin embargo, la importancia de generar nuevos valores de g^x normalmente se reduce, por ende el período de tiempo se puede prolongar correspondientemente.

e. Todos los grupos exponenciales modulares tienen subgrupos que son más débiles que el grupo principal. Para los números primos de Sophie Germain, si el generador esta elevado al cuadrado, entonces solamente hay dos elementos en ese subgrupo: 1 y g^(- 1) (es decir g^(p-1)) el cuál ya hemos recomendado evitar. Para los subgrupos de Schnorr con k diferente de 2, el subgrupo puede ser evitado controlando que el exponencial no sea una raíz de kth de 1 (e^k!= 1 mod p).

2.8.1 Definición de un Nuevo Grupo

Esta sección describe cómo definir un nuevo grupo. La descripción del grupo se oculta de fisgones, y el identificador asignado al grupo es único para las dos partes. El uso del nuevo grupo para los intercambios de clave de Diffie-Hellman se describe en la siguiente sección.

La confidencialidad de la descripción y del identificador incrementa la dificultad de un ataque pasivo, debido a que si la descripción del grupo no es conocida por el atacante, entonces no habrá una forma sencilla y eficiente de obtener información sobre las claves calculadas usando el grupo.

Solamente la descripción del nuevo grupo necesita ser encriptada en este intercambio. El algoritmo hash esta implícito debido a la sesión de OAKLEY designada por el grupo. La encriptación es la función de encriptación de la sesión de OAKLEY.

La descripción del nuevo grupo está codificada en la carga Nuevo Grupo. Los nonces se codifican en la Carga de Autentificación.

Los datos más allá del límite de encriptación se encriptan usando la transformación designada por el KEYID.

Los siguientes mensajes utilizan el Identificador de Intercambio de Claves ISAKMP de Nuevo Grupo de OAKLEY.

Para definir un nuevo grupo exponencial modular:

Iniciador		Respondedor
->	KEYID, INEWGRP, Desc(New Group), Na, prf(sKEYID, Desc(New Group) | Na)	->
<-	KEYID, INEWGRPRS, Na, Nb, prf(sKEYID, Na | Nb | Desc(New Group))	<-
->	KEYID, INEWGRPACK, prf(sKEYID, Nb | Na | Desc(New Group))	->

Estos mensajes se encriptan en el límite de la encriptación usando la clave indicada. El valor del hash se pone en el campo "Firma Digital" (véase la Sección 9).

Identificador de Nuevo GPR = trunc16(Na) | trunc16(Nb)

(trunc16 indica el truncamiento a los 16 bits; el iniciador y el respondedor deben utilizar nonces que tengan distintos bits de orden superior de los utilizados para los GRPID actuales).

Desc(G) es la codificación del descriptor para el descriptor del grupo (véase la Sección 8) para el formato de un descriptor de grupo)

Las dos partes deben guardar la asociación entre el identificador de nuevo grupo GRP y el descriptor Desc(New Group). Deben también observar las identidades usadas por el KEYID y copiar éstas al estado para el nuevo grupo.

Observe que uno podría tener el mismo descriptor de grupo asociado con varios KEYID. El cálculo previo de valores de g^x se puede realizar basándose solamente en el descriptor de grupo, no en el nombre del grupo privado.

2.8.2 Obtención de Claves Usando Grupos Privados

Una vez que se haya establecido un grupo privado, su identificador de grupo se puede utilizar en los mensajes de intercambio de claves en la posición GRP. No se requieren cambios en el protocolo.

2.9 Modo rápido: Nuevas Claves a partir de Claves Viejas

Cuando una KEYID autentificada y asociada con el material clave sKEYID existe, es fácil obtener KEYIDs adicionales y claves compartidas con

atributos similares (GRP, EHA, etc.) usando solamente funciones hash. Por ejemplo, el KEYID podría ser uno que fue obtenido del Modo Principal.

Por otra parte, la clave autentificada puede ser una clave manualmente distribuida, o una clave compartida por el iniciador y el respondedor vía algún medio externo a OAKLEY. Si el método de distribución ha formado el KEYID usando los valores únicos adecuados para las dos partes (CKY-I y CKY-R), entonces este método es aplicable.

En el siguiente esquema, el Identificador de Intercambio de Claves es el Modo Rápido de OAKLEY. Los nonces y el valor del prf se lleva en la Carga de Autentificación; donde la Autoridad de Autentificación es "Ninguna" y el tipo es "Pre-Compartido".

El protocolo es:

Iniciador		Respondedor
->	KEYID, INEWKRQ, Ni, prf(sKEYID, Ni)	->
<-	KEYID, INEWKRS, Nr, prf(sKEYID, 1 | Nr | Ni)	<-
->	KEYID, INEWKRP, 0, prf(sKEYID, 0 | Ni | Nr)	->

El Nuevo KEYID, NKEYID, es Ni | Nr

sNKEYID = prf(sKEYID, Ni | Nr)

Las identidades y los valores de EHA asociados con NKEYID son los mismos que los asociados a KEYID.

Cada parte debe validar los valores del hash antes de usar la nueva clave para cualquier propósito.

2.10 Definición y Uso de Claves Pre-Distribuidas

Si la clave, el identificador de clase asociado y la información de estado, se han distribuido manualmente, entonces la clave puede ser usada para cualquier propósito en OAKLEY. La clave debe estar asociada a la información de estado usual: Identificadores y algoritmos EHA.

La política local dictaminará cuando una clave manual puede ser incluida en la base de datos de OAKLEY. Por ejemplo, solamente los usuarios privilegiados se les permitiría introducir claves asociadas con los Identificadores privilegiados, un usuario no privilegiado podría introducir solamente las claves asociadas a su propia identificación

2.11 Distribución de una Clave Externa

Una vez establecida la clave de sesión de OAKLEY y los algoritmos auxiliares, el material clave y el algoritmo "H" se pueden utilizar para distribuir una clave externamente generada y asignarle a esta un KEYID.

En el siguiente esquema, el Identificador de Intercambio de Claves es el Modo Externo de OAKLEY. La Carga Intercambio de Claves contiene la nueva clave, la cual está protegida. El KEYID representa una clave de sesión autentificada de OAKLEY existente, y el sNEWKEYID representa el material clave generado.

Iniciador		Respondedor
->	KEYID, IEXTKEY, Ni, prf(sKEYID, Ni)	->
<-	KEYID, IEXTKEY, Nr, prf(sKEYID, 1 | Nr | Ni)	<-
->	KEYID, IEXTKEY, Kir xor sNEWKEYID*, prf(Kir, sNEWKEYID | Ni | Nr)	->

Kir = prf(sKEYID, Ni | Nr)

* este campo es transportado en la Carga de Intercambio de Claves.

Cada parte debe validar los valores del hash usando la función "H" en el estado del KEYID antes de intercambiar cualquier información de estado de la clave.

La nueva clave es recuperada por el respondedor calculando el XOR del campo de la Carga de Autentificación con el valor del Kir.

El identificador de la nueva clave, designa el sNEWKEYID del material clave, el cual es prf(sKEYID, 1 | Ni | Nr).

Observe que este intercambio no necesita encriptación.

2.11.1 Consideraciones de la Fuerza Criptográfica

La fuerza de la clave usada para distribuir la clave externa debe ser por lo menos igual a la fuerza de la clave externa. Generalmente, esto significa que la longitud del material sKEYID debe ser mayor o igual a la longitud del material del sNEWKEYID.

La obtención de la clave externa, su fuerza o uso pretendido no se trata en este capítulo; las partes que usen claves externas deben tener un método para determinar estas características.

A principios del año 1996, se observó que para 90 bits de fuerza criptográfica, uno debía utilizar módulos de un grupo exponencial modular de 2000 bits. Para 128 bits de fuerza, se requería módulos de 3000 bits.

3. **Especificación y Obtención de Asociaciones de Seguridad**

Obtener las claves para usarse con los protocolos IPsec por ejemplo ESP o AH es un tema que se trata en el Capítulo 10. Ese capítulo también describe cómo negociar un conjunto de parámetros aceptables y los identificadores para ESP y AH, y cómo calcular el material clave para cada instancia de los protocolos. El material clave básico definido aquí (g^xy) puede ser utilizado para obtener claves para varias instancias de ESP y AH, estos mecanismos usan funciones unidireccionales para convertir g^xy en varias claves únicas que son esenciales para el correcto uso.

4. **Compatibilidad con ISAKMP**

OAKLEY usa la cabecera de ISAKMP y los formatos de la carga, según lo descripto en este texto y en la Sección 9.

4.1 Autentificación con Claves Existentes

En el caso en que las dos partes no tengan los mecanismos de clave pública adecuados en su sitio para autentificar cada una a la otra parte, pueden utilizar claves distribuidas manualmente. Después del establecimiento de

estas claves y de asociar su estado en OAKLEY, pueden ser utilizadas para los modos de autentificación que dependen de firmas, por ejemplo el Modo Agresivo.

Cuando una clave existente aparece en una lista de ofertas, se debería indicar con un Algoritmo de Autentificación de ISAKMP_EXISTENTE.

Cuando el método de autentificación es ISAKMP_EXISTENTE, la autoridad de autentificación tendrá el valor ISAKMP_AUTH_EXISTENTE; el valor para este campo no debe estar en conflicto con ninguna otra autoridad de autentificación registrada en la IANA y definida en el RFC de ISAKMP.

La carga de autentificación tendrá dos partes:

el KEYID para la clave preexistente

el identificador para la parte a ser autentificada por la clave preexistente.

La función seudo-aleatoria "H" en la información de estado para el KEYID será el algoritmo de la firma, y utilizará el material clave para esa clave (sKEYID) cuando fue generada o controlará la validez de los datos del mensaje.

Por ejemplo, si la clave existente tiene un KEYID denotado por KID y 128 bits de material clave denotados por sKID y una transformación designada HMAC del algoritmo "H", entonces para generar una "firma" para un bloque de datos, la salida de HMAC(sKID, datos) será la carga correspondiente a la firma.

El estado del KEYID tendrá las identidades de las partes locales y partes remotas para los cuales el KEYID fue asignado; depende de la implementación de la política local decidir cuando es apropiado utilizar tal clave para autentificar a las otras partes. Por ejemplo, una clave distribuida para usarse entre el host A y B puede ser conveniente para autentificar todas las identidades de la forma "alice@A" y "bob@B".

4.2 Autentificación con Terceras Partes

Una política de seguridad local puede restringir la negociación de claves a partes confiables. Por ejemplo, dos demonios de OAKLEY ejecutándose con igual denominación de sensibilidad en dos máquinas pueden desear ser los únicos árbitros de los intercambios de claves entre los usuarios con esa misma denominación de sensibilidad. En este caso, una forma de autentificar la procedencia de las solicitudes de intercambios de clave es necesaria. Es decir, las identidades de los dos demonios deberían estar vinculadas a una clave, y esa clave será utilizada para formar una "firma" para los mensajes de intercambio de claves.

La Carga de la Firma, de la Sección 9, es para ese propósito. Esta carga designa un KEYID el cual existe antes del comienzo del intercambio actual. La transformación "H" para ése KEYID se utiliza para calcular un valor de integridad/autentificación para todas las cargas anteriores a la de la firma.

La política local puede dictaminar qué KEYID's son apropiados para los intercambios posteriores al de la firma.

4.3 Modo Nuevo Grupo

OAKLEY utiliza un nuevo KEI para el intercambio que define a nuevo grupo.

5. **Consideraciones de Seguridad**

Los ataques que tienen la capacidad de recuperar el valor del exponente usado en el cálculo de Diffie-Hellman han sido descriptos en [Kocher]. Para anular este tipo de ataques, los implementadores deben esforzarse por enmascarar la secuencia de las operaciones involucradas en la realización de la exponenciación modular.

Un "factor de enmascaramiento" puede obtenerse de la siguiente forma. Un elemento del grupo, r, se elige aleatoriamente. Cuando se elige un exponente x, el valor de r^(-x) también es calculado. Entonces, al calcular (g^y)^x, la implementación calculará la siguiente secuencia:

```
A = (rg^y)
B = A^x = (rg^y)^x = (r^x)(g^(xy))
C = B*r^(-x) = (r^x)(r^-(x))(g^(xy)) = g^(xy)
```

El factor de enmascaramiento se necesita solamente si el exponente x se utiliza más de 100 veces.

6. **Análisis Modular y Máquina de Estado de OAKLEY**

Hay muchos métodos con OAKLEY, pero estos siguen un orden de análisis modular de los campos del mensaje.

El iniciador opta por un mensaje inicial en el siguiente orden:

1. Ofrece una cookie. Esto no es necesario pero ayuda con los intercambios agresivos.

2. Escoge un grupo. La elección son los grupos bien conocidos o cualquier grupo privado que haya sido negociado. El inicio del primer intercambio entre dos demonios Oakley sin estado común debe involucrar un grupo bien conocido (0, significa ningún grupo, es un grupo bien conocido). Observe que el identificador de grupo (no el descriptor del grupo) es usado en el mensaje.

   Si se utiliza un grupo no nulo, este debe ser incluido en el primer mensaje especificando el EHAO. Este no necesita ser especificado hasta entonces.

3. Si se utiliza PFS, se escoge un exponente x y g^x.

4. Se Ofrece una lista de Autentificación, Hash y Encriptación.

5. Se usa el PFS para ocultar las identidades.

   Si el ocultamiento de identidad no es utilizado, entonces el iniciador tiene la siguiente opción:

6. Designa las identidades e incluye información de autentificación.

La información en la sección de autentificación depende de la primera oferta de autentificación. En un intercambio agresivo, el iniciador espera que el respondedor acepte toda la información ofrecida y el primer método

de autentificación. El método de autentificación determinará la carga de Autentificación de la siguiente forma:

1. Método de firma. La firma será aplicada a toda la información ofrecida.

2. Un método de encriptación de clave pública. El algoritmo que será utilizado para encriptar un nonce con la clave pública de la solicitud de identidad del respondedor. Hay dos casos posibles, dependiendo de que si se utiliza o no el ocultamiento de identidad:

   a. No hay ocultamiento de identidad. La identificación aparecerá en texto plano.

   b. Ocultamiento de identidad. Un identificador bien conocido, llamado R', aparecerá en texto plano en la carga de autentificación. Seguido por dos identificadores y un nonce; estos serán encriptados usando la clave pública para R'.

3. Un método de clave preexistente. La clave preexistente será utilizada para encriptar el nonce. Si se utiliza el ocultamiento de identidad, los identificadores estarán encriptados en la carga, usando el algoritmo "E" asociado con la clave preexistente.

El respondedor puede aceptar todo, parte o nada del mensaje inicial.

El respondedor acepta tantos campos como el lo desee, usando el mismo orden de decisión que el iniciador. En cualquier paso el respondedor puede parar, implícitamente rechazando los campos siguientes (los cuales contendrán valores nulos en su mensaje de respuesta). La respuesta mínima es una cookie y el GRP. El procesamiento del respondedor es el siguiente:

1. Acepta la cookie. El respondedor puede elegir no registrar la información de estado hasta que el iniciador conteste exitosamente con una cookie elegida por el respondedor. Si es así, el respondedor contesta con una cookie, el GRP, y ninguna otra información.

2. Acepta el GRP. Si el grupo no es aceptado, el respondedor no contestará. El respondedor puede enviar un mensaje de error indicando que el grupo no es aceptado (módulos demasiado pequeños, identificador desconocido, etc.). Observe que "no grupo" tiene dos significados durante el protocolo: puede denotar que el grupo aun no es especificado, o puede denotar que no se utilizará ningún grupo (y el PFS no será posible).

3. Acepta el valor del g^x. El respondedor indica su aceptación del valor del g^x incluyendo su propio valor g^y en su contestación. Él puede posponer esto ignorando el g^x y poniendo a cero el valor de la longitud de g^y en su contestación. Él puede también rechazar el valor del g^x por medio de un mensaje de error.

4. Acepta un elemento de cada una de las listas EHA. La aceptación se indica por una propuesta diferente de cero.

5. Si el PFS para ocultar las identidades es requerido, entonces no seguirán más datos.

6. Si la carga de autentificación está presente, y si el primer ítem en la clase ofrecida de autentificación es aceptado, entonces el respondedor debe validar/desencriptar la información en la carga de autentificación

y en la carga de la firma, si está presente. El respondedor deberá elegir un nonce y contestar con el mismo algoritmo de autentificación/hash que utilizó el iniciador.

El iniciador observa qué información ha aceptado el respondedor, valida/desencripta cualquier firma, hash, o campo encriptado, y si los datos son aceptados, contesta de acuerdo con el método EHA aceptado por el respondedor. La respuesta del iniciador se diferencia de su mensaje inicial por que tiene un valor de cookie diferente de cero para la cookie del respondedor.

El resultado de la firma o de la función prf será codificada como un número entero de precisión variable según lo descripto en la Sección 10. El KEYID indicará que KEYID designará el material clave y el Hash o función de Firma.

7. **La Carga de Certificado**

Los certificados con información de clave pública pueden ser añadidos a los mensajes de OAKLEY usando las Cargas de Certificado según lo definido en el Capítulo 7. Se debería notar que la opción de protección de identidad es aplicada a los certificados como así también a las identidades.

8. **Descriptores de Grupo**

Tres representaciones distintas de grupo se pueden usar con OAKLEY. Cada grupo es definido por su operación de grupo y por los campos subyacentes usados para representar los elementos del grupo. Los tres tipos son:

- El grupo de exponenciación modular, designado MODP.
- El grupo de curvas elípticas superiores al campo de Galois GF[2^N], designado EC2N.
- El grupo de curvas elípticas superiores al campo de Galois GF[P], designado ECP.

Para cada representación, hay distintas relaciones posibles, dependiendo de los parámetros seleccionados.

Salvo contadas excepciones, todos los parámetros se transmiten como si fuesen números enteros de precisión múltiple no negativos, usando el formato definido en esta sección (note, que esté es distinto que el codificado en la Sección 10). Cada número entero de precisión múltiple tiene una longitud de campo prefijada, incluso donde esta información es redundante.

Para el tipo de grupo EC2N, los parámetros están ideados más bien como campos de bit muy extensos, pero se representan como números enteros de precisión múltiple (mediante la longitud de los campos, y la adecuada justificación).

MODP significa el grupo exponencial modular clásico, donde la operación es calcular G^X (Módulo P). El grupo es definido por los parámetros numéricos P y G. P debe ser un número primo. G es frecuentemente 2, pero G puede variar de 2 hasta P-2 (vea el Capítulo 5).

ECP es un grupo de curvas elípticas, de módulo de un número primo P. La ecuación de definición para este tipo de grupo es Y^2 = X^3 + AX + B. La

función del grupo es tomar un múltiplo de un punto de la curva elíptica. El grupo está definido por 5 parámetros numéricos: El número primo P, dos parámetros de la curva A y B, y un generador (X,Y). A, B, X, Y codifican el módulo de P, y deben ser números enteros (no negativos) menores que P. Deben satisfacer la ecuación de definición, módulo P.

EC2N es un grupo de curvas elípticas, sobre el campo finito F[2^N]. La ecuación de definición para este tipo de grupo es Y^2 + XY = X^3 + AX^2 + B (esta ecuación se diferencia levemente del caso de la del Módulo P: tiene un término XY, y un término AX^2 en vez de un término AX.)

Debemos especificar el dominio de representación, y la curva elíptica. El dominio se especifica dando un polinomio irreducible (Módulo 2) de grado N. Este polinomio se representa como número entero de tamaño entre 2^N y 2^(N+1), como si el polinomio de definición fuera evaluado en el valor U=2.

Por ejemplo, el dominio definido para el polinomio U^155 + U^62 + 1 es representado por el número entero 2^155 + 2^62 + 1. El grupo es definido por 4 parámetros más, A, B, X, Y. Estos parámetros son elementos del dominio GF[2^N], y pueden ser interpretados como polinomios de menor grado que N, con coeficientes (Módulo 2). Se adapta a los campos de N bits, y se representan como números enteros menores a 2^N, como si el polinomio fuera evaluado en U=2. Por ejemplo, el elemento del dominio U^2 + 1 estaría representado por el número entero 2^2+1, que es 5. Los dos parámetros A y B definen la curva. Por lo general A tiene el valor cero (0). B no debe ser cero (0). Los parámetros X y Y seleccionan un punto en la curva. Los parámetros A, B, X, Y deben satisfacer la ecuación de definición, el módulo del polinomio de definición, y el Módulo 2.

La descripción del formato de un descriptor de grupo es:

- Tipo de Grupo: Un campo de dos bytes, los valores asignados para los tipos son "MODP", "ECP", "EC2N".
- Tamaño de un elemento de campo, en bits. Éste es como máximo el $\log_2 P$ o el grado del polinomio irreducible: un número entero de 32 bits.
- El número primo P o el polinomio irreducible del campo: un número entero de precisión múltiple.
- El generador: 1 o 2 valores, números enteros de precisión múltiple.
- Solamente para las Curvas Elípticas (EC): Los parámetros de la curva: 2 valores, números enteros de precisión múltiple.

Los siguientes parámetros son Opcionales, cada uno de estos puede aparecer independientemente (un valor de cero puede ser usado para representar un parámetro no especificado):

- El factor primo más grande: el valor codificado es decir el LPF (factor primo más grande) del tamaño del grupo, un número entero de precisión múltiple.
- Solamente para las Curvas Elípticas: El orden del grupo: números entero de precisión múltiple. (El tamaño del grupo para MODP es siempre P-1.)
- Fuerza del grupo: número entero de 32 bit. La fuerza del grupo es aproximadamente el número de bits de la clave protegida.

Para aclarar un poco esto se presentan los siguientes ejemplos:

- Este es un ejemplo genérico de un grupo "clásico" de exponenciación modular:
  - Tipo del grupo: "MODP"
  - Tamaño de un elemento de campo en bits: Log_2 p redondeado hacia arriba. Un número entero de 32 bits.
  - Definir un número primo P: un número entero de precisión múltiple.
  - Generador G: un número entero de precisión múltiple. G puede variar de 2 hasta P-2.

  Opcional:
  - El factor primo más grande de P-1: el número entero Q de precisión múltiple.
  - Fuerza del grupo: un número entero de 32 bits. Se especificará una fórmula para calcular este número (TBD).

- Este es un ejemplo genérico para un grupo de curvas elípticas, módulo P:
  - Tipo del grupo: "ECP"
  - Tamaño de un elemento de campo en bits: Log_2 p redondeado hacia arriba, un número entero de 32 bits.
  - Definir un número primo P: un número entero de precisión múltiple.
  - Generador (X,Y): 2 números enteros de precisión múltiple, cada uno menor que P.
  - Parámetros de la curva A, B: 2 números enteros de precisión múltiple, cada uno menor que P.

  Opcional:
  - El factor primo más grande del orden del grupo: un número entero de precisión
  múltiple.
  - Orden del grupo: un número entero de precisión múltiple.
  - Fuerza del grupo: un número entero de 32 bits. Fórmula TBD.

- Este es un ejemplo específico para un grupo de curvas elípticas:
  - Tipo del grupo: "EC2N"
  - Grado del polinomio irreducible: 155
  - Polinomio irreducible: U^155 + U^62 + 1, representado como el número entero de precisión múltiple.
  - Generador (X,Y): representado con 2 números enteros de precisión múltiple, cada uno menor a 2^155.
  - Para nuestra curva actual, éstos son 123 y 456 (en decimal). Cada uno representa un número entero de precisión múltiple.
  - Parámetros de la curva A, B: representados con 2 números enteros de precisión múltiple, cada uno menor a 2^155.
  - Para nuestra curva actual éstos son 0 y 471951 (en decimal), representan dos números enteros de precisión múltiple.

  Opcional:
  - El factor primo más grande del orden del grupo: 3805993847215893016155463826195386266397436443, representado como número entero de precisión múltiple.
  - El orden del grupo: 456719261665907161938655659143446351967692373 16 representado como número entero de precisión múltiple.
  - Fuerza del grupo: 76, representado como número entero de 32 bits.

Figura 2 muestra la codificación de un número entero de precisión variable para los campos del descriptor de grupo. Esto es una variación leve del

formato definido en la Sección 10 en el que un valor fijo de 16 bits se utiliza primero, y la longitud esta limitada a 16 bits. Sin embargo, la interpretación es idéntica.

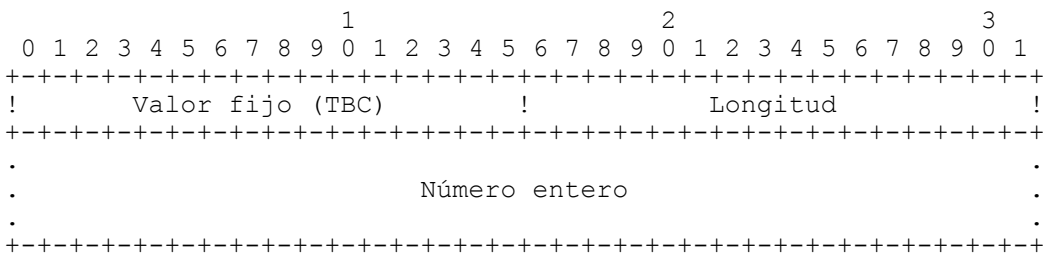


Figura 2: Codificación de un número entero de precisión variable para los campos del descriptor de grupo.

La Figura 3 muestra un ejemplo del formato un descriptor de grupo tipo MOD.

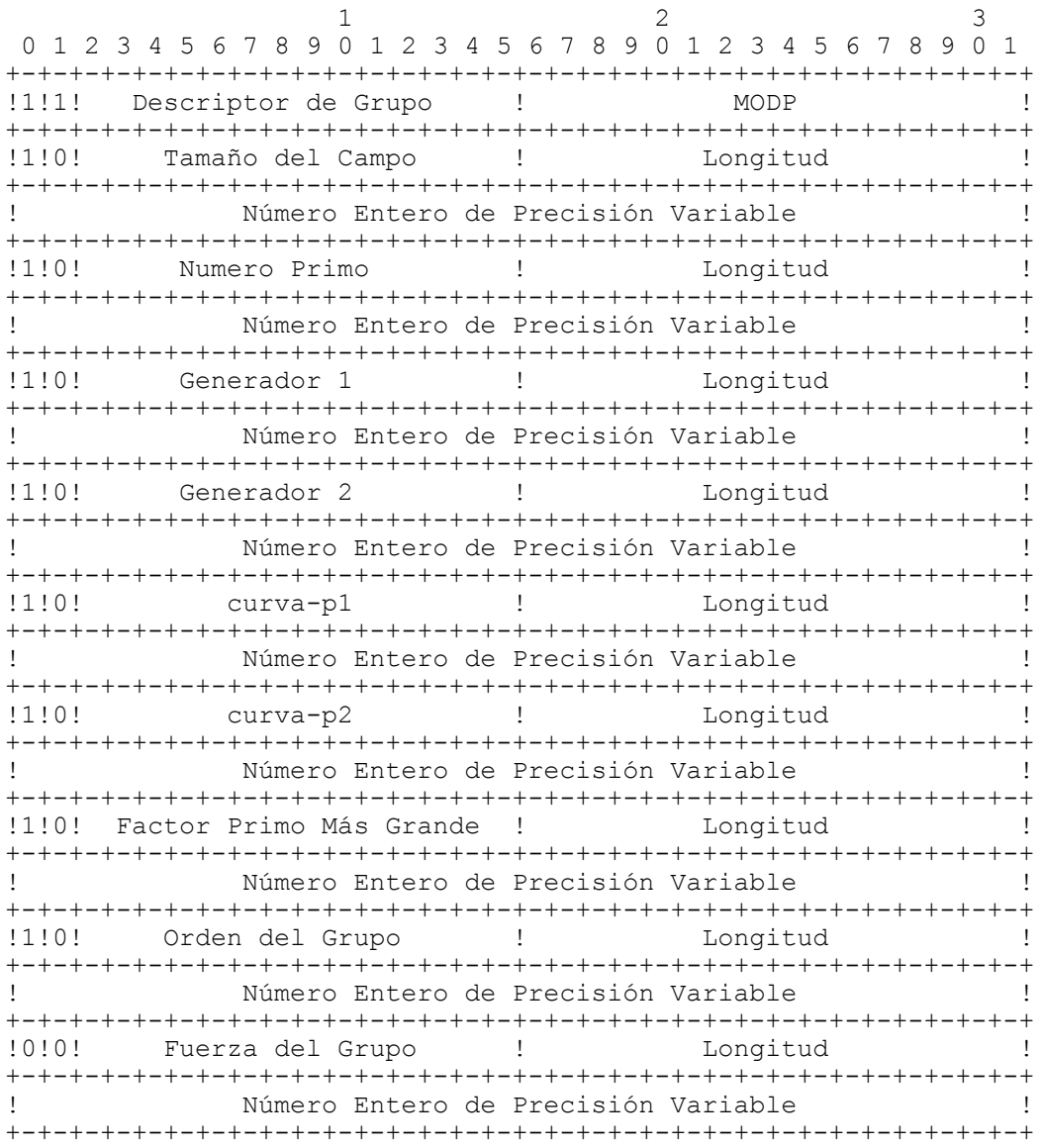


Figura 3: Ejemplo del formato un descriptor de grupo tipo MOD

9. **Formato de los mensajes**

El protocolo OAKLEY [OAKLEY] no define estos aspectos y solo menciona que deja la codificación de los mensajes de OAKLEY dentro de las cargas de ISAKMP en manos de IKE (Capítulo 10)

10. **Codificación de un Número Entero de Precisión Variable**

Los números enteros de precisión variable serán codificados en un campo con 32 bits de longitud seguido por una o más cantidades de 32 bits que contienen la representación del número entero, alineado con el bits más significativo en el primer ítem de 32 bits.

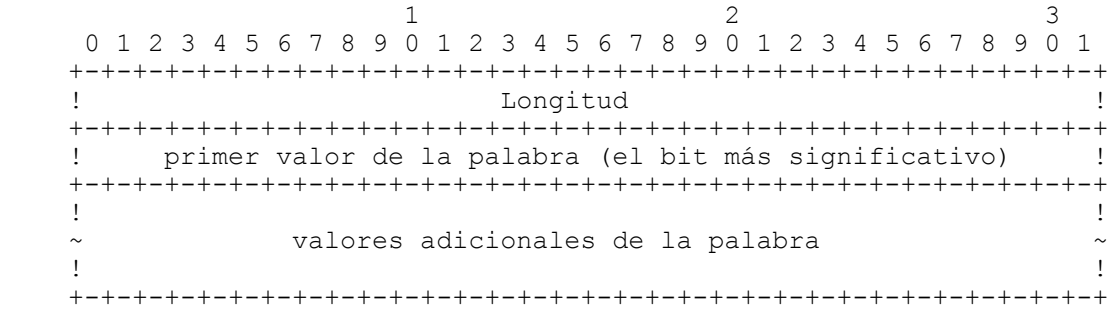


Figura 4: Codificación de un número entero de precisión variable

Un ejemplo de tal codificación se muestra en la Figura 5, para un número con 51 bits significativos. Al campo longitud le siguen 2 cantidades de 32 bits. El bit más significativo diferente de cero del número está en el bit 13 de la primera cantidad de 32 bits, el bit menos significativo de menor orden está en la segunda cantidad de 32 bits.

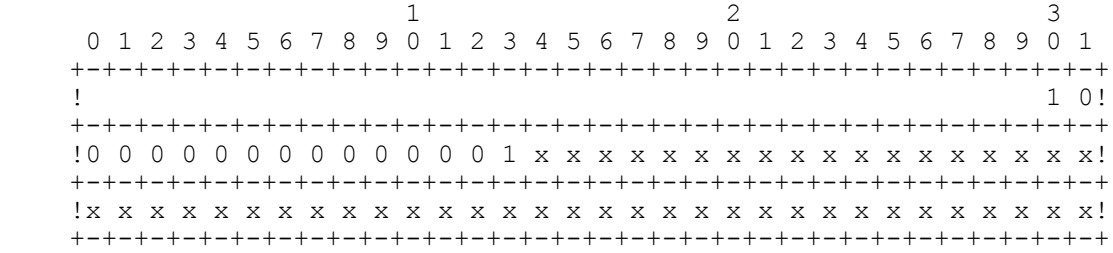


Figura 5: Ejemplo de la Codificación de un número entero de precisión variable

11. **Fuerza Criptográfica**

El algoritmo de Diffie-Hellman se utiliza para calcular las claves que serán utilizadas con los algoritmos simétricos. No debería ser más fácil romper el cálculo de Diffie-Hellman que hacer una búsqueda exhaustiva sobre el espacio de claves simétricas. Una recomendación reciente de un grupo de criptógrafos [Blaze] ha recomendado un tamaño de clave simétrica de 75 bits para un nivel práctico de seguridad. Para 20 años de seguridad, se recomiendan 90 bits.

De acuerdo con ese informe, una estrategia conservadora para los usuarios de OAKLEY sería asegurarse de que sus cálculos de Diffie-Hellman sean tan seguros conteniendo un espacio de clave de por lo menos 90 bits. Para lograr esto en los grupos exponenciales modulares, el tamaño del factor primo más grande del módulo debe ser de por lo menos 180 bits, y el tamaño del módulo debe ser de por lo menos 1400 bits. Para los grupos de curva elípticos, el LPF (factor primo más grande) debe ser de por lo menos 180 bits.

Si la confidencialidad a largo plazo de la clave criptográfica no es problema, entonces los siguientes parámetros se pueden utilizar para el grupo exponencial modular: 150 bits para el LPF, 980 bits para el tamaño de los módulos.

El tamaño de los módulos no es el único factor que determina la fuerza del cálculo de Diffie-Hellman; el tamaño de los exponentes usados en el cálculo de la potencia dentro del grupo también es importante. El tamaño del exponente en bits debería ser de por lo menos dos veces el tamaño de

cualquier clave simétrica que se pudiera obtener de él. Se recomienda que las implementaciones de ISAKMP utilicen por lo menos 180 bits de exponente (dos veces el tamaño de una clave simétrica de 20 años).

La justificación matemática para estas estimaciones se pueden encontrar en los textos que evalúan el esfuerzo para solucionar el problema del logaritmo discreto. Para más información vea [Stinson] y [Schneier].

12. **Los Grupos Bien Conocidos**

Los identificadores de grupo:

0 no hay grupo (usados como marcador de posición y para los intercambios no DH)
1 un grupo exponencial modular con un módulo de 768 bits
2 un grupo exponencial modular con un módulo de 1024 bits
3 un grupo exponencial modular con un módulo de 1536 bits (TBD)
4 un grupo de curvas elípticas superiores a GF[2^155]
5 un grupo de curvas elípticas superiores a GF[2^185]

Los valores 2^31 y superiores se utilizan para identificar a los grupos privados

Grupos exponenciales modulares de Diffie-Hellman clásicos

Los números primos para los grupos 1 y 2 fueron seleccionados para tener ciertas características. Los 64 bits de orden superior se fuerzan a 1. Esto ayuda al resto del algoritmo, porque el dígito del cociente de prueba siempre puede ser tomado como la palabra de orden superior del dividendo, posiblemente +1. Los 64 bits de orden inferior se fuerzan a 1. Esto ayuda a los algoritmos restantes, porque el dígito multiplicador siempre puede ser tomado como la palabra de orden inferior del dividendo. Los bits medios se toman de la extensión binaria de Π. Esto garantiza que son eficientemente aleatorios, mientras que evita cualquier sospecha de que los números primos se han seleccionado secretamente para ser débiles.

Debido a que ambos números primos se basan en el número Π, hay un gran sector de superposición en las representaciones hexadecimales de los dos números primos. Los números primos se eligen para ser números primos de Sophie Germain (es decir, (P-1)/2 es también un número primo), para tener más fuerza contra el ataque de la raíz cuadrada en el problema discreto del logaritmo.

Los números de prueba inicial fueron repetitivamente incrementados por un factor de 2^64 hasta que se localizaron números primos adecuados.

Debido a que estos dos números primos son congruentes a 7 (Módulo 8), 2 es un residuo cuadrático de cada uno de los números primos. Todas las potencias de 2 también serán residuos cuadráticos. Esto impide que un atacante sepa el bit de orden superior del exponente de Diffie-Hellman (También conocido como el problema del subgrupo confinado). Usar 2 como generador es eficiente en algunos algoritmos exponenciales modulares. Observe que 2 no es técnicamente un generador en el sentido de la teoría numérica, porque omite la mitad de los residuos posibles de módulo P. Desde el punto de vista criptográfico, esto es una virtud.

12.1 Grupo 1 Bien Conocido: Un número primo de 768 Bits

El número primo es 2^768 - 2^704 - 1 + 2^64 * { [2^638 Π] + 149686 }. Su valor decimal es:

```
155251809230070893513091813125848175563133404943451431320235
119490296623994910210725866945387659164244291000768028886422
915080371891804634263272761303128298374438082089019628850917
069131659317536746955176311984337163722100721057791 9
```

Esto ha sido rigurosamente verificado como un número primo.

La representación del grupo en OAKLEY es:

```
Tipo de grupo:                                    "MODP"
Tamaño del elemento del campo (en bits):          768
Módulo primo:                                     21 (en decimal)
   Longitud (en palabras de 32 bit):              24
   Datos (en hexadecimal):
      FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
      29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
      EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
      E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF
Generador:                                        22 (en decimal)
   Longitud (en palabras de 32 bit):              1
   Datos (en hexadecimal):                        2

Parámetros Opcionales:
El factor primo más grande del orden del grupo:   24 (en decimal)
   Longitud (en palabras de 32 bit):              24
   Datos (en hexadecimal):
      7FFFFFFF FFFFFFFF E487ED51 10B4611A 62633145 C06E0E68
      94812704 4533E63A 0105DF53 1D89CD91 28A5043C C71A026E
      F7CA8CD9 E69D218D 98158536 F92F8A1B A7F09AB6 B6A8E122
      F242DABB 312F3F63 7A262174 D31D1B10 7FFFFFFF FFFFFFFF
Fuerza del Grupo:                                 26 (en decimal)
   Longitud (en palabras de 32 bit):              1
   Datos (en hexadecimal):                        00000042
```

12.2 Grupo 2 Bien Conocido: Un número Primo de 1024 Bits

El número primo es 2^1024 - 2^960 - 1 + 2^64 * { [2^894 Π] + 129093 }. Su valor decimal es:

```
179769313486231590770839156793787453197860296048756011706444
423684197180216158519368947833795864925541502180565485980503
646440548199239100050792877003355816639229553136239076508735
759914822574862575007425302077447712589550957937778424442426
617334727629299387668709205606050270810842907692932019128194
467627007
```

El carácter primo del número ha sido rigurosamente verificado.

La representación del grupo en OAKLEY es:

```
Tipo de grupo:                                          "MODP"
Tamaño del elemento del campo (en bits):                1024
Módulo primo:                                           21 (en decimal)
   Longitud (en palabras de 32 bit):                    32
   Datos (en hexadecimal):
      FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
      29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
      EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
      E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
      EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
      FFFFFFFF FFFFFFFF
Generador:                                              22 (en decimal)
   Longitud (en palabras de 32 bit):                    1
   Datos (en hexadecimal):                              2

Parámetros Opcionales:
El factor primo más grande del orden del grupo:         24 (en decimal)
   Longitud (en palabras de 32 bit):                    32
   Datos (en hexadecimal):
      7FFFFFFF FFFFFFFF E487ED51 10B4611A 62633145 C06E0E68
      94812704 4533E63A 0105DF53 1D89CD91 28A5043C C71A026E
      F7CA8CD9 E69D218D 98158536 F92F8A1B A7F09AB6 B6A8E122
      F242DABB 312F3F63 7A262174 D31BF6B5 85FFAE5B 7A035BF6
      F71C35FD AD44CFD2 D74F9208 BE258FF3 24943328 F67329C0
      FFFFFFFF FFFFFFFF
Fuerza del Grupo:                                       26 (en decimal)
   Longitud (en palabras de 32 bit):                    1
   Datos (en hexadecimal):                              0000004D
```

12.3 Grupo 3 Bien Conocido: Una Definición de Grupos de Curvas Elípticas

La curva se basa en el campo de Galois GF[2^155] con 2^155 elementos de campo. El polinomio irreducible para el campo es u^155 + u^62 + 1. La ecuación para la curva elíptica es:

$$Y^2 + X Y = X^3 + A X + B$$

X, Y, A, B son elementos del campo

Para la curva especifica, A = 0 y

$$B = u^{18} + u^{17} + u^{16} + u^{13} + u^{12} + u^9 + u^8 + u^7 + u^3 + u^2 + u + 1$$

B se representa en binario como 1110011001110001111; en decimal es 471951, y en hexadecimal es 7338F.

El generador es un punto (X,Y) en la curva (que satisface la ecuación de la curva, el Módulo 2 y el módulo del polinomio del campo).

$$X = u^6 + u^5 + u^4 + u^3 + u + 1 \quad y \quad Y = u^8 + u^7 + u^6 + u^3$$

Las cadenas binarias de bits para X y Y son 1111011 y 111001000; en decimal son 123 y 456.

El orden del grupo (el número de puntos en la curva) es:

4567192616659071619386556591434463519676923731 6

el cual es 12 veces el número primo

3805993847215893016155463826195386266397436443.

El carácter primo del numero ha sido rigurosamente controlado. El punto generador (X,Y) tiene un orden de 4 veces el número primo; el generador es el triple en algún punto de la curva.
La representación en OAKLEY de este grupo es:

Tipo de grupo:	"EC2N"
Tamaño del elemento del campo (en bits):	155
Campo del polinomio irreducible	21 (en decimal)
Longitud (en palabras de 32 bit):	5
Datos (en hexadecimal):	
08000000 00000000 00000000 40000000 00000001	
Generador:	
Coordenada X:	22 (en decimal)
Longitud (en palabras de 32 bit):	1
Datos (en hexadecimal):	7B
Coordenada X:	22 (en decimal)
Longitud (en palabras de 32 bit):	1
Datos (en hexadecimal):	1C8
Parámetros de la curva elíptica:	
Parámetro A:	23 (en decimal)
Longitud (en palabras de 32 bit):	1
Datos (en hexadecimal):	0
Parámetro B:	23 (en decimal)
Longitud (en palabras de 32 bit):	1
Datos (en hexadecimal):	7338F
Parámetros Opcionales:	
El factor primo más grande del orden del grupo:	24 (en decimal)
Longitud (en palabras de 32 bit):	5
Datos (en hexadecimal):	
00AAAAAA AAAAAAAA AAAAB1FC F1E206F4 21A3EA1B	
Orden del Grupo:	25 (en decimal)
Longitud (en palabras de 32 bit):	5
Datos (en hexadecimal):	
08000000 00000000 000057DB 56985371 93AEF944	
Fuerza del Grupo:	26 (en decimal)
Longitud (en palabras de 32 bit):	1
Datos (en hexadecimal):	0000004C

12.4 Grupo 4 Bien conocido: Una Definición General de Grupos de Curvas Elípticas

Esta curva se basa en el campo de Galois GF[2^185] con 2^185 elementos de campo.
El polinomio irreducible para el campo es:

$$u^{185} + u^{69} + 1.$$

La ecuación para la curva elíptica es:

$$Y^2 + X\,Y = X^3 + A\,X + B.$$

X, Y, A, B, son elementos del campo. Para la curva especifica, A=0 y

$$B = u^{12} + u^{11} + u^{10} + u^9 + u^7 + u^6 + u^5 + u^3 + 1.$$

B se representa en binario como 1111011101001; en decimal es 7913, y en hexadecimal es 1EE9.

El generador es un punto (X,Y) en la curva (que satisface la ecuación de la curva, el Módulo 2 y el módulo del polinomio del campo).

X = u^4 + u^3 y Y = u^3 + u^2 + 1

Las cadenas binarias de bits para X y Y son 11000 y 1101; en decimal son 24 y 13. El orden del grupo (el número de puntos en la curva) es:

490398573077084434674671048576526822480523850010450531 16,

que es 4 veces el número primo

122599643269271108668667762144131705620130962502612632 79.

El carácter primo del numero ha sido rigurosamente controlado.

El punto generador (X,Y) tiene un orden de 2 veces el número primo; el generador es el doble en algún punto de la curva.

La representación en OAKLEY de este grupo es:

```
Tipo de grupo:                                          "EC2N"
Tamaño del elemento del campo (en bits):                185
Campo del polinomio irreducible                         21 (en decimal)
   Longitud (en palabras de 32 bit):                    6
   Datos (en hexadecimal):
      02000000 00000000 00000000 00000020 00000000 00000001
Generador:
   Coordenada X:                                        22 (en decimal)
      Longitud (en palabras de 32 bit):                 1
      Datos (en hexadecimal):                           18
   Coordenada X:                                        22 (en decimal)
      Longitud (en palabras de 32 bit):                 1
      Datos (en hexadecimal):                           D
Parámetros de la curva elíptica:
   Parámetro A:                                         23 (en decimal)
      Longitud (en palabras de 32 bit):                 1
      Datos (en hexadecimal):                           0
   Parámetro B:                                         23 (en decimal)
      Longitud (en palabras de 32 bit):                 1
      Datos (en hexadecimal):                           1EE9

Parámetros Opcionales:
El factor primo más grande del orden del grupo:         24 (en decimal)
   Longitud (en palabras de 32 bit):                    6
   Datos (en hexadecimal):
      007FFFFF FFFFFFFF FFFFFFFF F6FCBE22 6DCF9210 5D7E53AF
Orden del Grupo:                                        25 (en decimal)
   Longitud (en palabras de 32 bit):                    6
   Datos (en hexadecimal):
      01FFFFFF FFFFFFFF FFFFFFFF DBF2F889 B73E4841 75F94EBC
Fuerza del Grupo:                                       26 (en decimal)
   Longitud (en palabras de 32 bit):                    1
   Datos (en hexadecimal):                              0000005B
```

12.5 Grupo 5 Bien Conocido: Un número Primo de 1536 Bits

El número primo es 2^1536 - 2^1472 - 1 + 2^64 * { [2^1406 Π] + 741804 }. Su valor en decimal es:

```
241031242692103258855207602219756607485695054850245994265411
694195810883168261222889009385826134161467322714147790401219
650364895705058263194273070680500922306273474534107340669624
601458936165977404102716924945320037872943417032584377865919
814376319377685986952408894019557734611984354530154704374720
774996976375008430892633929555996888245787241299381012913029
459299994792636526405928464720973038494721168143446471443848
852094012745984428885933652689632091963391 9
```

El carácter primo del número ha sido rigurosamente verificado.

La representación del grupo en OAKLEY es:

```
Tipo de grupo:                                      "MODP"
Tamaño del elemento del campo (en bits):            1536
Módulo primo:                                       21 (en decimal)
   Longitud (en palabras de 32 bit):                48
   Datos (en hexadecimal):
      FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
      29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
      EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
      E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
      EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
      C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
      83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
      670C354E 4ABC9804 F1746C08 CA237327 FFFFFFFF FFFFFFFF
Generador:                                          22 (en decimal)
   Longitud (en palabras de 32 bit):                1
   Datos (en hexadecimal):                          2

Parámetros Opcionales:
El factor primo más grande del orden del grupo:     24 (en decimal)
   Longitud (en palabras de 32 bit):                48
   Datos (en hexadecimal):
      7FFFFFFF FFFFFFFF E487ED51 10B4611A 62633145 C06E0E68
      94812704 4533E63A 0105DF53 1D89CD91 28A5043C C71A026E
      F7CA8CD9 E69D218D 98158536 F92F8A1B A7F09AB6 B6A8E122
      F242DABB 312F3F63 7A262174 D31BF6B5 85FFAE5B 7A035BF6
      F71C35FD AD44CFD2 D74F9208 BE258FF3 24943328 F6722D9E
      E1003E5C 50B1DF82 CC6D241B 0E2AE9CD 348B1FD4 7E9267AF
      C1B2AE91 EE51D6CB 0E3179AB 1042A95D CF6A9483 B84B4B36
      B3861AA7 255E4C02 78BA3604 6511B993 FFFFFFFF FFFFFFFF
Fuerza del Grupo:                                   26 (en decimal)
   Longitud (en palabras de 32 bit):                1
   Datos (en hexadecimal):                          0000005B
```

13. **Implementación de funciones de Grupo**

El funcionamiento del grupo debe estar implementado como una secuencia de operaciones aritméticas; las operaciones exactas dependen del tipo de grupo. Para grupos exponenciales modulares, la operación es la multiplicación de números enteros de precisión variable y los restos multiplicados por grupos modulares. Vea [Knuth] para una discusión de cómo implementar éstos para números enteros más grandes. Las recomendaciones de implementación de

funciones de curvas elípticas sobre el campo GF[2^N] se describen en [Schroeppel].

14. **Conclusiones**

Este capítulo describió un protocolo, llamado OAKLEY, por el cual dos partes autentificadas pueden convenir en el material clave seguro y secreto. El mecanismo básico es el algoritmo de intercambios de claves de Diffie-Hellman, descripto genéricamente en el Capítulo 5 y profundizado en este capítulo.

El protocolo OAKLEY soporta Perfect Forward Secrecy, compatibilidad con el protocolo ISAKMP para la administración de Asociaciones de Seguridad, estructuras abstractas definidas por grupos de usuarios para usarse con el algoritmo de Diffie-Hellman, actualizaciones de claves, y la incorporación de claves distribuidas vía mecanismos fuera de banda.

Cabe recalcar que uno de los objetivos principales de este capítulo fue sentar las bases para poder comprender el Capítulo 10 (IKE) que deriva en gran parte de este capítulo.

Capítulo 10

El Protocolo de Intercambio de Claves en Internet (IKE)

1. **Introducción**

Como se estudió en el capítulo 7,ISAKMP proporciona un marco para la autentificación y el intercambio de claves pero no los define. ISAKMP esta diseñado para ser un intercambio de claves independiente; es decir, está diseñado para soportar una gran cantidad de intercambios de claves diferentes.

Como se estudió en el capítulo 9, Oakley describe una serie de intercambios de claves, llamados "modos" y detalla los servicios proporcionados por cada uno (por ejemplo, perfect forward secrecy para claves, protección de identidad y autentificación).

SKEME [SKEME] describe una técnica de intercambio de claves que proporciona anonimato, repudiabilidad, y renovación rápida de claves.

Este capítulo describe un protocolo usando partes de Oakley y partes de SKEME (solamente el método de encriptación de claves públicas para la autentificación y el concepto de rápido recambio de claves usando un intercambio nonce) conjuntamente con ISAKMP para obtener material clave autentificado para usarse con ISAKMP, y para otras Asociaciones de Seguridad (SA) tales como las de AH y ESP del DOI de IPsec de la IETF.

El propósito de este protocolo híbrido es negociar (y proporcionar el material clave autentificado para) las SA de un modo protegido.

Los procedimientos que implementa este capítulo pueden ser utilizados en las negociaciones de las Redes Privadas Virtuales (VPNs), como así también para proporcionar a un usuario remoto (cuya dirección IP no necesita ser conocida de antemano) acceso a un host o red de forma segura.

Este protocolo soporta la negociación de cliente. El modo cliente es donde las partes negociantes no son la de los extremos para los cuales la negociación de la SA se lleva a cabo. Cuando se usa el modo cliente, las identidades de las partes de los extremos quedan ocultas.

IKE no demanda conformidad con el protocolo entero de Oakley ni es dependiente de ninguna forma del protocolo Oakley ni del protocolo SKEME.

2. **Notación**

La siguiente notación se utiliza a través de todo este capítulo en todos los diagramas donde se represente un intercambio.

Notación	Significado
HDR	Es una cabecera de ISAKMP cuyo tipo de intercambio es el modo.
HDR*	La encriptación del mensaje (es denotado por "*" después de la cabecera de ISAKMP) DEBE comenzar inmediatamente después de la cabecera de ISAKMP. Cuando se protege la comunicación, todas las cargas que le siguen a la cabecera de ISAKMP DEBEN estar encriptadas. Las claves de encriptación son generadas por SKEYID_e el cual es definido por cada algoritmo.
SA	Es una Carga SA de negociación con una o más propuestas. El iniciador PUEDE proporcionar múltiples propuestas para la negociación; el respondedor solamente DEBE contestar una.

Notación	Significado
<P>_b	Indica el cuerpo de la carga.
<P>--	La cabecera de la carga genérica de ISAKMP no está incluida.
SAi_b	Es el cuerpo entero de la Carga SA (menos la cabecera de carga genérica de ISAKMP); es decir el DOI, la situación, todas las propuestas y todas las transformaciones ofrecidas por el Iniciador.
CKY-I	Es la cookie del Iniciador en la cabecera de ISAKMP.
CKY-R	Es la cookie del Respondedor en la cabecera de ISAKMP.
g^xi	ES el valor público de Diffie Hellman [DH] del Iniciador.
g^xr	Es el valor público de Diffie Hellman [DH] del Respondedor
g^xy	Es el secreto compartido de Diffie Hellman.
KE	Es la Carga de Intercambio de claves, la cual contiene la información pública intercambiada en un intercambio de Diffie Hellman. No hay una codificación particular (por ejemplo, TLV o TV) usada para los datos de una carga KE.
Ni	Es la Carga Nonce de ISAKMP para el Iniciador.
Nr	Es la Carga Nonce de ISAKMP para el Respondedor.
IDii	Es la Carga de Identificación de ISAKMP para el Iniciador durante la Fase 1 de la negociación.
IDir	Es la Carga de Identificación de ISAKMP para el Respondedor durante la Fase 1 de la negociación.
IDui	Es la Carga de Identificación de ISAKMP para el Iniciador durante la Fase 2 de la negociación.
IDur	Es la Carga de Identificación de ISAKMP para el Iniciador durante la Fase 2 de la negociación.
SIG	Es la Carga de la Firma. Los datos a firmar son un intercambio específico.
CERT	Es la Carga de Certificado.
HASH o HASH(2) o HASH_I	Es la Carga HASH. Los contenidos del Hash son específicos del método de autentificación.
prf(clave, msg)	Es una función de claves seudo aleatoria (a menudo una función de claves hash) usada para generar una salida determinista que aparece seudo aleatoriamente. Las funciones seudo aletorias (prf) utilizan para derivar claves (de una clave obtener varias) y para la autentificación (tal como una clave MAC, vea [HMAC]).
SKEYID	Es una cadena derivada del material secreto conocido solamente por los participantes activos en el intercambio.
SKEYID_e	Es el material clave usado por la SA de ISAKMP para proporcionar confidencialidad a sus mensajes
SKEYID_a	Es el material clave usado por la SA de ISAKMP para autentificar sus mensajes.
SKEYID_d	Es el material clave usado para derivar las claves para las SA no ISAKMP.
<x>y	Indica que "x" está encriptado con la clave "y".
-->	Significa comunicación del "iniciador al respondedor" (peticiones).
<--	Significa comunicación del "respondedor al iniciador" (respuestas).
|	Significa concatenación de la información (por ejemplo X | Y es la concatenación de X con Y).
[x]	Indica que x es opcional.
,	Los campos en el mensaje se nombran y se separan por una coma ",".

3. **Arquitectura del Protocolo**

Oakley y SKEME definen un método para establecer y autentificar intercambios de claves, el cual incluye la construcción de las cargas, el trasporte de cargas informativas, el orden en la cual se procesan y cómo se utilizan.

The relationship between the two is very straightforward and IKE presents different exchanges as modes which operate in one of two phases.

Como se vio anteriormente Oakley define "Modos" y ISAKMP define "Fases". Mientras que IKE presenta diferentes intercambios definidos como modos, que funcionan en una de las dos fases.

La Fase 1 es donde los dos usuarios de ISAKMP establecen un canal seguro y autentificado en el cual comunicarse. Este se llama SA de ISAKMP (o SA ISAKMP). El "Modo Principal" y el "Modo Agresivo" se llevan a cabo en un intercambio de Fase 1. El "Modo Principal" y el "Modo Agresivo" SOLAMENTE SE DEBEN utilizar en la Fase 1.

La Fase 2 es donde las SAs se negocian en representación de servicios tales como IPsec (por ejemplo AH IPsec o ESP IPsec) o cualquier otro servicio que necesite el material clave y/o negociación de parámetros. El "Modo Rápido" se lleva a cabo en un intercambio de Fase 2. EL "Modo Rápido" SOLAMENTE SE DEBE utilizar en la Fase 2.

El "Modo Nuevo Grupo" no está realmente ni en la Fase 1 ni en la Fase 2, le sigue a la Fase 1, y sirve para establecer un nuevo grupo que puede ser utilizado en futuras negociaciones. El "Modo Nuevo Grupo" SE DEBE utilizar SOLAMENTE después de la Fase 1.

La SA ISAKMP es bidireccional. Es decir, una vez establecida, cualquier parte puede iniciar intercambios en Modo Rápido, Informativos y Modo Nuevo Grupo. Como se vio en el Capítulo 7, la SA ISAKMP se identifica por la cookie del Iniciador y por la cookie del Respondedor. El papel desempeñado por cada parte en el intercambio de la Fase 1 dictaminará cual es la cookie del Iniciador y cual la del Respondedor. El orden de la cookie establecida por el intercambio de la Fase 1 continúa identificando la SA de ISAKMP sin importar la dirección de los intercambios de Modo Rápido, Informativo, o Nuevo Grupo. Es decir, las cookies NO DEBEN alternar lugares cuando la dirección de la SA ISAKMP cambia.

Con el uso de las fases de ISAKMP, una implementación puede lograr claves muy rápidamente cuando sea necesario. Una simple negociación de Fase 1 se puede utilizar para más de una negociación de Fase 2. Una simple negociación de Fase 2 puede solicitar múltiples SA. Con estas optimizaciones, una implementación puede perder menos de un viaje de ida y vuelta por SA, así como también menos de una exponenciación de DH (Diffie Hellman) por SA. El "Modo Principal" para la Fase 1 proporciona protección de identidad. Cuando la protección de identidad no es necesaria, el "Modo Agresivo" se puede utilizar para reducir futuros viajes de ida y vuelta. Se debe notar que usar la encriptación de clave pública para autentificar un intercambio de Modo Agresivo proporcionará protección de identidad.

Este protocolo no define su propio DOI. La SA ISAKMP, establecida en la Fase 1, PUEDE usar el DOI y la situación de un servicio no ISAKMP (tal como el DOI de IPsec de la IETF descripto en el Capítulo 8). En este caso una implementación PUEDE elegir restringir el uso de la SA ISAKMP para el establecimiento de SAs para los servicios del mismo DOI. Alternativamente,

la SA ISAKMP SE PUEDE establecer con el valor de cero (0) para el campo DOI y situación (véase el capítulo 7 para una descripción de estos campos) y en este caso las implementaciones serán libres de establecer los servicios de seguridad para cualquier DOI definido usando esta SA ISAKMP. Si un DOI de valor cero se utiliza para el establecimiento de una SA de Fase 1, la sintaxis de las cargas de identidad usadas en la Fase 1 es la definida en el Capítulo 7 y no la de cualquier DOI (por ejemplo la definida en el Capítulo 8) la cual pueda ampliar la sintaxis y semántica de futuras identidades.

Los siguientes atributos son utilizados por IKE y son negociados como parte de la SA ISAKMP. Estos atributos pertenecen solamente a la SA ISAKMP y no a cualquier SA que ISAKMP pueda negociar en nombre de otros servicios.

- Algoritmo de encriptación

- Algoritmo de hash

- Método de autentificación

- Información sobre un grupo al cual realizarle Diffie Hellman.

Todos estos atributos son obligatorios y DEBEN ser negociados. Además, es posible negociar opcionalmente una función seudo-aleatoria ("prf"). Los valores de atributo de uso privado pueden ser usados para negociar prf entre las partes concernientes. Si un "prf" no es negociado, la versión HMAC (véase [HMAC]) del algoritmo hash negociado es usado como una función seudo-aleatoria. Otros atributos no obligatorios se describen en la Sección 10. El algoritmo hash seleccionado DEBE soportar el modo nativo y el modo HMAC.

El grupo de Diffie Hellman DEBE ser especificado usando una descripción definida de grupo (Sección 5) o definiendo todos los atributos de un grupo (Sección 4.3). Los atributos de grupo (tales como Tipo o Numero Primo, vea la Sección 10) NO SE DEBEN ofrecer conjuntamente con un grupo previamente definido (una descripción del grupo reservado o una descripción de uso privado que es establecida después de la finalización de un intercambio de Modo Nuevo Grupo).

Las implementaciones de IKE DEBEN soportar los siguientes valores de atributo:

- DES [DESAN] en modo CBC con claves débiles, semi-débiles y control de claves (las claves débiles y semi-débiles se describieron en el Capítulo 5). La claves se derivan según lo establecido en la Sección 11.

- MD5 [MD5] y SHA [SHA].

- Autentificación mediante claves pre-compartidas.

- El Grupo de Exponenciación Modular (MODP) número uno (1).

Además, las implementaciones IKE DEBERÍAN soportar: 3DES para la encriptación; Tiger [TIGER] para el hash; la Firma Digital Estándar (DSS), firmas RSA [RSA] y autentificación con encriptación de clave pública RSA; y el grupo MODP número dos (2). Las implementaciones IKE PUEDEN soportar cualquier algoritmo de encriptación definido en la Sección 10 y DEBERÍAN soportar los grupos ECP y EC2N.

Los modos de IKE descriptos aquí DEBEN ser implementados siempre que se implemente el DOI de IPsec de la IETF (vea el Capítulo 8). Otros DOIs PUEDEN utilizar los modos descriptos aquí.

4. **Intercambios**

Hay dos métodos básicos usados para establecer un intercambio de claves autentificado: El Modo Principal y el Modo Agresivo. Cada uno genera material clave autentificado a partir de un intercambio de Diffie Hellman. El Modo Principal DEBE ser implementado, en cambio el Modo Agresivo DEBERÍA ser implementado. El Modo Rápido SE DEBE implementar como mecanismo para generar nuevo material clave y para negociar servicios de seguridad no ISAKMP. Además, el Modo Nuevo Grupo SE DEBERÍA implementar como mecanismo para definir grupos privados para intercambios de Diffie Hellman. Las implementaciones NO DEBEN cambiar los tipos de intercambio cuando se esta realizando un intercambio.

Los intercambios se atienen a la sintaxis de la carga, codificación de los atributos, intervalo y retransmisiones de mensajes, y mensajes informativos de ISAKMP; por ejemplo un mensaje de notificación es enviado cuando, por ejemplo, una propuesta es inaceptable, o una verificación de firma o una desencriptación no fue exitosa, etc.

La carga SA DEBE preceder al resto de las cargas en un intercambio de Fase 1. Excepto donde se indique lo contrario, donde no existan requisitos para las cargas de ISAKMP en ningún mensaje cuando estén en algún orden en particular.

El valor público de Diffie Hellman colocado en la carga de Intercambio de Claves (KE), en un intercambio de Fase 1 o de Fase 2, DEBE tener la longitud del grupo negociado de Diffie Hellman, en caso de necesidad, rellenar el valor pre-pendiente con ceros.

La longitud de la carga nonce DEBE ser entre 8 y 256 bytes inclusive.

El Modo Principal es una ejemplificación del Intercambio de Protección de Identidad de ISAKMP (Sección 4.5 del Capítulo 7). Los primeros dos mensajes negocian la política; los dos siguientes intercambian los valores públicos de Diffie Hellman y datos auxiliares (por ejemplo nonces) necesarios para el intercambio; y los dos últimos mensajes autentifican el Intercambio de Diffie Hellman. El método de autentificación negociado como parte inicial del intercambio de ISAKMP influencia en la composición de las cargas pero no en su propósito. El XCHG (Intercambio) para el Modo Principal es el Intercambio de Protección de Identidad de ISAKMP.

El Modo Agresivo es una ejemplificación del Intercambio Agresivo de ISAKMP (Sección 4.7 del Capítulo 7). Los primeros dos mensajes negocian la política, intercambian los valores públicos de Diffie Hellman y los datos auxiliares necesarios para el intercambio, y las identidades. Además el segundo mensaje autentifica al respondedor. El tercer mensaje autentifica al iniciador y proporciona una prueba de la participación en el intercambio. El XCHG (Intercambio) para el Modo Agresivo es el Intercambio Agresivo de ISAKMP. El mensaje final NO PUEDE ser enviado bajo la protección de la SA ISAKMP dado que cada parte pospone la exponenciación, si lo desea, hasta que la negociación de este intercambio se haya completado. Las descripciones gráficas del Modo Agresivo muestran la carga final en limpio, lo cual no necesita ser así.

Los intercambios en IKE tienen un número fijo de mensajes. La Recepción de una carga de Petición de Certificado NO DEBE ampliar el número de mensajes transmitidos o esperados.

La negociación de SA está limitada en el Modo Agresivo, debido a los requerimientos en la construcción de mensajes, el grupo en el cual el intercambio de Diffie Hellman se ejecuta no puede ser negociado. Además, métodos de autentificación diferentes pueden limitar aun más la negociación de los atributos. Por ejemplo, la autentificación con encriptación de clave pública no puede ser negociada y cuando se usa el método revisado de encriptación de clave pública para autentificar, el cifrado y el hash no pueden ser negociados. Para las situaciones donde se requiere la capacidad de negociar gran cantidad de atributos en IKE, el Modo Principal puede ser requerido

El Modo Rápido y el Modo Nuevo Grupo no tienen ningún análogo en ISAKMP. Los valores XCHG para el Modo Rápido y el Modo Nuevo Grupo se definen en la Sección 10.

El Modo Principal, el Modo Agresivo, y el Modo Rápido realizan la negociación de la SA. Las ofertas SA consisten en carga/s de Transformación encapsuladas en la carga/s de la Propuesta encapsuladas en carga/s SA. Si múltiples ofertas se realizan para los intercambios de Fase 1 (Modo Principal/Modo Agresivo) estas DEBEN consistir de múltiples Cargas de Transformación para una sola Carga de la Propuesta en una sola carga SA. En otras palabras, para los intercambios de la Fase 1 NO DEBE haber múltiples cargas de la Propuesta para una sola carga SA y NO DEBE haber múltiples cargas SA. No prohibiéndose tal postura en ofertas en intercambios de Fase 2.

No hay límite en el número de ofertas que el iniciador puede enviar al respondedor pero las implementaciones PUEDEN elegir limitar el número de ofertas que examinará por razones de rendimiento.

Durante la negociación de la SA, los iniciadores presentan posibles ofertas de SA a los respondedores. Los respondedores NO DEBEN modificar los atributos de ninguna oferta, exceptuando la codificación de los atributos (véase la Sección 10). Si el iniciador de un intercambio nota que los valores de los atributos han cambiado o que se han agregado atributos o se han suprimido de una oferta realizada, esa respuesta DEBE ser rechazada.

Los cuatro métodos de autentificación permitidos en el Modo Principal o en el Modo Agresivo son: Firma Digital, Claves Pre-compartidas y dos métodos de Autentificación con Encriptación de Clave Pública. El valor del SKEYID es calculado por separado para cada método de autentificación.

- Para las firmas: SKEYID = prf(Ni_b | Nr_b, g^xy)
- Para la encriptación de clave pública: SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)
- Para las claves pre-compartidas: SKEYID = prf(pre-shared-key, Ni_b | Nr_b)

El resultado del Modo Principal o del Modo Agresivo son tres grupos de material clave autentificado,

```
    SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0),
    SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1),
    SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2),
```

y una política acordada para proteger subsiguientes comunicaciones. Los valores de 0, 1, y 2 de arriba son representados por un solo octeto. La clave usada para la encriptación es derivada a partir de SKEYID_e de un algoritmo especifico (véase la Sección 11).

Para autentificar cualquier intercambio el iniciador del protocolo genera HASH_I y el respondedor genera HASH_R donde:

```
HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b )
HASH_R = prf(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b )
```

Para la autentificación con las firmas digitales, se firman y se verifican HASH_I y HASH_R; para la autentificación con encriptación de clave pública o claves pre-compartidas, HASH_I y HASH_R autentifican directamente el intercambio. La carga entera de identificación (incluyendo tipo de identificador, acceso y protocolo; pero excluyendo la cabecera de carga genérica) es hasheada en HASH_I y HASH_R.

Según lo mencionado anteriormente, el método de autentificación negociado influencia el contenido y el uso de los mensajes para los modos de la Fase 1, pero no su propósito. Al usar claves públicas para la autentificación, el intercambio de la Fase 1 se puede realizar usando firmas o usando la encriptación de clave pública (si el algoritmo lo soporta). A continuación se presentan intercambios de Fase 1 con diversas opciones de autentificación.

4.1 Intercambios de la Fase 1 de IKE

4.1.1 Autentificación con Firmas Digitales

Usando firmas, la información auxiliar intercambiada durante el segundo viaje de ida y vuelta son nonces; el intercambio es autentificado firmando un hash mutuamente obtenido. El Modo Principal con autentificación de firmas se describe de la siguiente forma:

Iniciador		Respondedor
HDR, SA	-->	
	<--	HDR, SA
HDR, KE, Ni	-->	
	<--	HDR, KE, Nr
HDR*, IDii, [CERT,] SIG_I	-->	
	<--	HDR*, IDir, [CERT,] SIG_R

El Modo Agresivo con firmas en conjunto con ISAKMP se describe de la siguiente forma:

Iniciador		Respondedor
HDR, SA, KE, Ni, IDii	-->	
	<--	HDR, SA, KE, Nr, IDir, [CERT], SIG_R
HDR, [CERT,] SIG_I	-->	

En ambos modos, los datos firmados, SIG_I o SIG_R, son el resultado de la negociación del algoritmo de firma digital aplicado a HASH_I o a HASH_R respectivamente.

En general la firma será sobre HASH_I y HASH_R usando el prf negociado, o la versión de HMAC de la función negociada de hash (si no se negocia ningún prf). Sin embargo, esto se puede evitar para la construcción de la firma si el algoritmo de la firma se vincula a un determinado algoritmo de hash (por ejemplo DSS se define solamente con el algoritmo SHA con una salida de 160 bit). En este caso, la firma será sobre HASH_I y HASH_R, excepto que se usa la versión de HMAC del algoritmo de hash asociado con el método de la firma. El prf y la función hash negociadas continuarían siendo utilizados por el resto de las funciones seudo-aleatorias.

Puesto que el algoritmo de hash usado ya se conoce no hay necesidad de codificar su OID (Identificador de Objeto) en la firma. Además, no hay vínculos entre los OIDs usado para firmas RSA en PKCS N°1 y los usados en este capítulo. Por lo tanto, las firmas RSA DEBEN estar codificadas con encriptación de clave pública en formato PKCS N°1 y no con firma en el formato PKCS N°1 (que incluye el OID del algoritmo hash).

Opcionalmente una o más cargas de certificado PUEDEN ser enviadas.

4.1.2 Autentificación con Encriptación de Clave Pública

Usando la encriptación de clave pública para autentificar el intercambio, la información auxiliar intercambiada son nonces encriptados. Cada parte habilitada para la reconstrucción del hash (comprobando que la otra parte desencriptó el nonce) autentifica el intercambio.

Para realizar la encriptación de la clave pública, el iniciador debe tener la clave pública del respondedor. En el caso de que el respondedor tenga múltiples claves públicas, un hash del certificado del iniciador es utilizado para encriptar la información auxiliar, la cual es pasada como parte del tercer mensaje. De esta manera el respondedor puede determinar la correspondiente clave privada usada para desencriptar las cargas encriptadas y la protección de identidad es mantenida.

Además del nonce, las identidades de las partes (IDii e IDir) también se encriptan con la clave pública de la otra parte. Si el método de autentificación es la encriptación de clave pública, las carga Nonce y la de Identificación SE DEBEN encriptar con la clave pública de la otra parte. Solamente el cuerpo de las cargas es encriptado, las cabeceras de la carga se dejan sin encriptar.

Al usar la encriptación para la autentificación, el Modo Principal se describe de la siguiente forma:

Iniciador		Respondedor
HDR, SA	-->	
	<--	HDR, SA
HDR, KE, [HASH(1)], <IDii_b>PubKey_r, <Ni_b>PubKey_r	-->	
	<--	HDR, KE, <IDir_b>PubKey_i, <Nr_b>PubKey_i
HDR*, HASH_I	-->	
	<--	HDR*, HASH_R

El Modo Agresivo autentificado con encriptación se describe de la siguiente forma:

Iniciador		Respondedor
HDR, SA, [HASH(1)], KE, <IDii_b>Pubkey_r, <Ni_b>Pubkey_r	-->	
	<--	HDR, SA, KE, <IDir_b>PubKey_i, <Nr_b>PubKey_i, HASH_R
HDR, HASH_I	-->	

Donde HASH(1) es el hash (usando la función negociada de hash) del certificado que el iniciador está utilizando para encriptar el nonce y la identidad.

La encriptación RSA SE DEBE codificar en formato PKCS N°1. Mientras que solamente el cuerpo de la carga Nonce y la de Identificación son encriptadas, los datos encriptados deben estar precedidos por una cabecera genérica de ISAKMP. La longitud de la carga es la longitud de la carga entera encriptada más la cabecera. La codificación de PKCS N°1 permite la determinación de la longitud actual de la carga de texto plano sobre la desencriptación.

Usar encriptación para la autentificación proporciona un intercambio posiblemente rechazable. No hay prueba (como con una firma digital) de que la conversación alguna vez tuvo lugar porque cada parte puede reconstruir totalmente ambos lados del intercambio. Para reforzar la seguridad, se agrega la generación del secreto, puesto que un atacante tendría que quebrar exitosamente no solo el intercambio de Diffie Hellman sino también la encriptación RSA. Este intercambio fue motivado por [SKEME].

Observe que, a diferencia de otros métodos de autentificación, la autentificación con encriptación de clave pública permite la protección de la identidad en Modo Agresivo.

4.1.3 Autentificación con un Modo Revisado de Encriptación de Clave Pública

La autentificación con encriptación de clave pública tiene ventajas significativas sobre la autentificación con firmas, como se vio en la sección anterior. Desafortunadamente, a costa de 4 operaciones de claves públicas (dos encriptaciones de clave pública y dos desencriptaciones de clave privada). Este modo de autentificación conserva las ventajas de la autentificación usando la encriptación de clave pública pero lo hace con la mitad de las operaciones de clave pública.

En este modo, el nonce todavía es encriptado usando la clave pública del usuario, no obstante la identidad del usuario (y el certificado si es enviado) es encriptado usando el algoritmo de encriptación simétrico negociado (de la carga SA) con una clave derivada del nonce. Esta solución agrega una mínima complejidad pero esta condición economiza dos costosas operaciones de clave pública en cada uno de los extremos. Además, la carga de Intercambio de Claves también es encriptada usando la misma clave derivada. Esto proporciona protección adicional contra el análisis criptográfico en el intercambio de Diffie Hellman.

Como con el método de autentificación con encriptación de clave pública (Sección 4.1.2), una carga HASH puede ser enviada para identificar un certificado si el respondedor tiene múltiples certificados, los cuales contienen claves públicas utilizables (por ejemplo si el certificado no es

para las firmas solamente, debido a las restricciones del certificado o a las restricciones algorítmicas). Si se envía la carga HASH esta DEBE ser la primera carga del segundo intercambio de mensajes y DEBE estar seguida por el nonce encriptado. Si la carga HASH no es enviada, la primera carga del segundo intercambio del mensaje DEBE ser el nonce encriptado. Además, el iniciador puede enviar opcionalmente una carga de certificado para proporcionar al respondedor una clave pública con la cual responder.

Al usar el modo revisado de encriptación para la autentificación, el Modo Principal es definido de la siguiente forma:

Iniciador		Respondedor
HDR, SA	-->	
	<--	HDR, SA
HDR, [HASH(1)], <Ni_b>Pubkey_r, <KE_b>Ke_i, <Idii_b>Ke_i, [<<Cert-I_b>Ke_i]	-->	
	<--	HDR, <Nr_b>PubKey_i, <KE_b>Ke_r, <IDir_b>Ke_r
HDR*, HASH_I	-->	
	<--	HDR*, HASH_R

El Modo Agresivo autentificado con el método revisado de encriptación se describe de la siguiente forma:

Iniciador		Respondedor
HDR, SA, [HASH(1)], <Ni_b>Pubkey_r, <KE_b>Ke_i, <Idii_b>Ke_i, [<Cert-I_b>Ke_i]	-->	
	<--	HDR, SA, <Nr_b>PubKey_i, <KE_b>Ke_r, <IDir_b>Ke_r, HASH_R
HDR, HASH_I	-->	

Donde HASH(1) es el hash (usando la función negociada de hash) del certificado que el iniciador está utilizando para encriptar el nonce y la identidad. Ke_i y Ke_r son las claves para el algoritmo de encriptación simétrico negociado en el intercambio de la carga SA. Solamente el cuerpo de las cargas son encriptados (con claves públicas y operaciones simétricas), las cabeceras de carga genérica se dejan en limpio. La longitud de la carga incluye el relleno para realizar la encriptación.

Las claves simétricas encriptadas se derivan de los nonces desencriptados como se describe a continuación. Primero se calculan los valores Ne_i y Ne_r:

```
Ne_i = prf(Ni_b, CKY-I)
Ne_r = prf(Nr_b, CKY-R)
```

Las claves Ke_i y Ke_r se extraen de Ne_i y de Ne_r respectivamente, según se describe en la Sección 11, usando las claves simétricas derivadas para usarse con el algoritmo de encriptación negociado. Si la longitud de salida del prf negociado es mayor o igual que la longitud de la clave requerida de cifrado, Ke_i y Ke_r se derivan de los bits más significativos de Ne_i y de Ne_r respectivamente. Si la longitud deseada de Ke_i y Ke_r excede la longitud de salida del prf el número necesario de bits es obtenido introduciendo repetitivamente el resultado del prf nuevamente dentro de sí

mismo y concatenando el resultado hasta que se ha alcanzado el número necesario. Por ejemplo, si el algoritmo de encriptación negociado requiere 320 bits de clave y la salida del prf es de solamente 128 bits, Ke_i es los 320 bits de K más significativos, donde:

```
K = K1 | K2 | K3
Donde:
K1 = prf(Ne_i, 0)
K2 = prf(Ne_i, K1)
K3 = prf(Ne_i, K2)
```

Por brevedad, solamente se muestra la derivación de Ke_i, siendo la derivación de Ke_r idéntica. La longitud del valor cero (0) en el cálculo de K1 es de solo un octeto. Observe que Ne_i, Ne_r, Ke_i, y Ke_r son todas de corta vida (efímeras) y DEBEN ser descartadas después de usarse.

Excepto los requisitos de la localización de la carga Hash (opcional) y de la carga Nonce (obligatoria) no hay otros requisitos de carga. Todas las cargas (independientemente del orden) detrás de la del nonce encriptado SE DEBEN encriptar con Ke_i o Ke_r dependiendo de la dirección.

Si el modo CBC se utiliza para la encriptación simétrica entonces los Vectores de Inicialización (IVs) se determinan de la siguiente forma:

> El IV para la encriptación de la primera carga que sigue a la carga Nonce se fija a cero (0). El IV para las subsiguientes cargas encriptadas con la clave cifrada secreta efímera, Ke_i, es el último bloque de texto cifrado de las cargas previas. Las cargas encriptadas se rellenan hasta alcanzar el tamaño de bloque más cercano. Todo los bytes de relleno, con excepción del último, contienen 0x00. El último byte de relleno contiene el número de bytes de relleno usado, excluyendo el último. Observe que esto significa que siempre habrá relleno.

4.1.4 Autentificación con Claves Pre-Compartidas

Una clave derivada por un cierto mecanismo fuera de banda también se puede utilizar para autentificar el intercambio. El establecimiento de esta clave está fuera del alcance de este libro.

Cuando se realiza una Autentificación con Clave Pre-Compartida, el Modo Principal se define de la siguiente forma:

Iniciador		Respondedor
HDR, SA	-->	
	<--	HDR, SA
HDR, KE, Ni	-->	
	<--	HDR, KE, Nr
HDR*, IDii, HASH_I	-->	
	<--	HDR*, IDir, HASH_R

El Modo Agresivo con una clave pre-compartida se describe como sigue:

Iniciador		Respondedor
HDR, SA, KE, Ni, IDii	-->	
	<--	HDR, SA, KE, Nr, IDir, HASH_R
HDR, HASH_I	-->	

Al usar la autentificación con clave pre-compartida en el Modo Principal la clave solo se puede identificar por la dirección IP de los usuarios, puesto que HASH_I debe ser calculado antes de que el iniciador haya procesado el IDir. El Modo Agresivo permite una gama más amplia de identificadores de confidencialidad pre-compartidos para utilizarse. Además, el Modo Agresivo permite que dos partes mantengan múltiples, claves pre-compartidas diferentes y seleccionar la correcta para un intercambio determinado.

4.2 Intercambios de la Fase 2 de IKE

4.2.1 Modo Rápido

El Modo Rápido no es un intercambio íntegro (en cuanto a que está limitado por el intercambio de la Fase 1), pero se utiliza como parte del proceso de negociación de la SA (Fase 2) para derivar el material clave y negociar la política compartida para las SAs no ISAKMP. La información intercambiada en el Modo Rápido DEBE estar protegida por la SA ISAKMP, es decir todas las cargas excepto la cabecera de ISAKMP están encriptadas. En Modo Rápido, una carga HASH DEBE seguir inmediatamente a la cabecera ISAKMP y una carga SA DEBE seguir inmediatamente a la de Hash. Este hash autentifica el mensaje y también proporciona prueba de la actividad.

El identificador del mensaje en la cabecera de ISAKMP identifica el Modo Rápido en curso para una SA ISAKMP determinada, la cual a su vez es identificada por las cookies en la cabecera de ISAKMP. Puesto que cada instancia de Modo Rápido utiliza un Vector de Inicialización único (véase la Sección 11) es posible tener simultáneamente múltiples Modos Rápidos, basados solo en la SA ISAKMP en curso en cualquier momento.

El Modo Rápido es esencialmente una negociación de SA y un intercambio de nonces que proporciona protección anti-replay. Los nonces se utilizan para generar el material clave actualizado y impedir que ataques de reenvío generen SA falsas. Una carga de Intercambio de Claves opcional puede ser intercambiada para permitir un intercambio de Diffie Hellman adicional y una exponenciación por Modo Rápido. A pesar de que el uso de la carga de Intercambio de Claves (KE)con el Modo Rápido es opcional este DEBE ser soportado.

El Modo Rápido (sin la carga KE) actualiza el material clave derivado de la exponenciación la Fase 1. Esto no proporciona PFS. Usando la carga KE opcional, se realiza una exponenciación adicional y el PFS es proporcionado para el material clave.

Las identidades de la SAs negociadas en Modo Rápido se asume implícitamente que son las direcciones IP de los usuarios de ISAKMP, sin ninguna restricción implícita en el protocolo o en la cantidad de accesos permitidos, a menos que los identificadores del cliente se especifiquen en Modo Rápido. Si ISAKMP está actuando como un cliente negociador en nombre de otra parte, las identidades de las partes SE DEBEN pasar como IDci y IDcr. La política local dictaminará si las propuestas son aceptables para las identidades especificadas. Si las identidades del cliente no son aceptadas por el respondedor en Modo Rápido (debido a la política o a otras razones), una carga de Notificación conteniendo el tipo de mensaje de Notificación, INFORMACIÓN DEL IDENTIFICADOR NO VÁLIDO (18), DEBERÍA ser enviado.

Las identidades del cliente se utilizan para identificar y para dirigir el tráfico al túnel apropiado en caso de que existan múltiples túneles entre

dos usuarios y también para permitir SAs únicas y compartidas con diferentes niveles de modularidad.

Todas las ofertas hechas durante un Modo Rápido están lógicamente relacionadas y deben ser consistentes. Por ejemplo, si se envía una carga KE, el atributo que describe al grupo de Diffie Hellman (véase el Capítulo 8) SE DEBE incluir en cada transformación de cada propuesta de cada SA que es negociada. Semejantemente, si se utilizan las identidades del cliente, DEBEN aplicarse a cada SA en la negociación.

Se define el Modo Rápido como sigue:

Iniciador		Respondedor
HDR*, HASH(1), SA, Ni, [KE], [IDci, IDcr]	-->	
	<--	HDR*, HASH(2), SA, Nr, [KE], [IDci, IDcr]
HDR*, HASH(3)	-->	

Donde HASH(1) es el prf sobre el Identificador del Mensaje (M-ID) de la cabecera de ISAKMP concatenada con el mensaje entero que sigue al Hash incluyendo todas las cabeceras de carga, pero excluyendo cualquier relleno agregado para la encriptación. HASH(2) es idéntico al HASH(1) excepto por el nonce del iniciador (Ni, menos la carga de la cabecera) que se agrega después del M-ID pero antes del mensaje completo. La suma del nonce en HASH (2) está para una prueba de actividad. HASH(3) (para la prueba de actividad) es el prf sobre el valor cero (0) representado como un solo octeto, seguido por una concatenación de identificadores de mensajes y de dos nonces (el del iniciador seguido por el del respondedor) menos la carga de la cabecera. Es decir, los hashes para el intercambio antedicho son:

```
HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [| KE] [| IDci | IDcr])
HASH(2) = prf(SKEYID_a, M-ID | Ni_b | SA | Nr [| KE] [| IDci | Idcr])
HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)
```

Con excepción del HASH, la SA, y de las cargas de Identificación opcionales, no hay restricciones para el ordenamiento de las cargas en Modo Rápido. HASH(1) y HASH(2) pueden diferenciarse de la ilustración de arriba si el orden de las cargas en el mensaje difieren del ejemplo ilustrado o si cualquier carga opcional, por ejemplo la carga de Notificación, se ha añadido al mensaje.

Si el PFS no es necesario, y las cargas KE no se intercambian, el nuevo material clave es definido de la siguiente forma:

```
KEYMAT = prf(SKEYID_d, protocol | SPI | Ni_b | Nr_b)
```

Si se desea el PFS y las cargas KE son intercambiadas, el nuevo material clave es definido de la siguiente forma:

```
KEYMAT = prf(SKEYID_d, g(qm)^xy | protocolo | SPI | Ni_b | Nr_b)
```

donde g(qm)^xy es el secreto compartido del intercambio efímero de Diffie Hellman en Modo Rápido.

En ambos casos, el "protocolo" y el "SPI" son la de la carga de la Propuesta de ISAKMP (ver la Sección 3.5 del Capítulo 7) que contiene la Transformación negociada.

Una sola negociación de SA da lugar a dos SAs (una de entrada y una de salida). Diferentes SPIs para cada SA (uno elegido por el iniciador, y el otro por el respondedor) garantizan una clave diferente para cada dirección. El SPI elegido por el destinatario de la SA se utiliza para derivar KEYMAT para esa SA.

Para las situaciones donde la cantidad de material clave deseado es mayor que el proporcionado por el prf, el KEYMAT es obtenido introduciendo repetitivamente el resultado del prf nuevamente dentro de sí mismo y concatenando el resultado hasta que se ha alcanzado el tamaño del material clave requerido. Es decir

```
KEYMAT = K1 | K2 | K3 | ...
Done:
K1 = prf(SKEYID_d, [ g(qm)^xy | ] protocolo | SPI | Ni_b | Nr_b)
K2 = prf(SKEYID_d, K1 | [ g(qm)^xy | ] protocolo | SPI | Ni_b | Nr_b)
K3 = prf(SKEYID_d, K2 | [ g(qm)^xy | ] protocolo | SPI | Ni_b | Nr_b)
etc.
```

Este material clave (con PFS o sin PFS, derivado directamente o a través de la concatenación) SE DEBE utilizar con la SA negociada. Depende del servicio definir la forma en que las claves son derivadas del material clave.

En el caso de un intercambio efímero de Diffie Hellman dentro del Modo Rápido, el exponencial (g(qm)^xy) se quita del estado actual y el SKEYID_e y el SKEYID_a (derivados de la negociación de la Fase 1) continúan protegiendo y autentificando la SA ISAKMP y el SKEYID_d se continúa utilizando para derivar las claves.

Usando el Modo Rápido, múltiples SAs y claves pueden ser negociadas con el siguiente intercambio:

Iniciador		Respondedor
HDR*, HASH(1), SA0, SA1, Ni, [KE], [IDci, IDcr]	-->	
	<--	HDR*, HASH(2), SA0, SA1, Nr, [KE], [IDci, IDcr]
HDR*, HASH(3)	-->	

Como se observa en el diagrama en este caso en particular (donde se están negociando dos cargas SA) el resultado sería cuatro Asociaciones de Seguridad (dos para cada una de las SAs). El material clave se deriva idénticamente como en el caso de una sola SA.

4.3 Modo Nuevo Grupo

El Modo Nuevo Grupo NO DEBE ser utilizado antes del establecimiento de la SA ISAKMP. La descripción de un Nuevo Grupo solamente se DEBE llevar a cabo luego de la negociación de Fase 1. El Modo Nuevo Grupo no es un intercambio de Fase 2 y la negociación se realiza de la siguiente forma:

Iniciador		Respondedor
HDR*, HASH(1), SA	-->	
	<--	HDR*, HASH(2), SA

Donde HASH(1) es la salida del prf, usando el SKEYID_a como la clave, y el identificador del mensaje de la cabecera de ISAKMP concatenado con la propuesta SA entera (el cuerpo y la cabecera), como los datos; HASH(2) es la salida del prf, usando SKEYID_a como la clave, y el identificador del mensaje de la cabecera de ISAKMP concatenado con la contestación como los datos. Es decir, los hashes para el intercambio antedicho son:

```
HASH(1) = prf(SKEYID_a, M-ID | SA)
HASH(2) = prf(SKEYID_a, M-ID | SA)
```

La propuesta especificará las características del grupo (véase la Sección 10). Las descripciones de grupo para los Grupos privados DEBEN ser mayor o igual a 2^15. Si el grupo no es aceptado, el respondedor DEBE contestar con una carga de Notificación conteniendo el tipo de mensaje de Notificación, ATRIBUTOS NO SOPORTADOS (13).

Las implementaciones de ISAKMP PUEDEN requerir que grupos privados expiren con la SA bajo la cual fueron establecidos.

Los grupos pueden ser negociados directamente en la propuesta SA con el Modo Principal. Para hacer esto las partes que lo componen (para un grupo MODP, Tipo, Número Primo y Generador; para un grupo EC2N, Tipo, Polinomio Irreducible, Primer Grupo Generado, Segundo Grupo Generado, Grupo Curva A, Grupo Curva B y Orden del Grupo) se pasan como atributos SA (véase la Sección 10). Alternativamente, la naturaleza del grupo se puede ocultar usando el Modo Nuevo Grupo y solamente el identificador del grupo se pasa sin encriptar durante la negociación de la Fase 1.

4.4 Intercambios Informativos de ISAKMP

Este protocolo protege a los Intercambios Informativos de ISAKMP cuando es posible. Una vez que la SA de ISAKMP ha sido establecida (y se han generado SKEYID_e y SKEYID_a) los Intercambios Informativos de ISAKMP, cuando se usan con este protocolo, son de la siguiente forma:

Iniciador		Respondedor
HDR*, HASH(1), N/D	-->	

Donde N/D es una Carga de Notificación de ISAKMP o una Carga de Cancelación de ISAKMP y HASH(1) es la salida del prf, usando SKEYID_a como la clave, y un M-ID único para este intercambio concatenado con la carga Informativa (una Notificación o Cancelación) como los datos. Es decir, el hash para el intercambio antedicho es:

```
HASH(1) = prf(SKEYID_a, M-ID | N/D)
```

Como se observó el Identificador del Mensaje en la cabecera de ISAKMP (y utilizado en el cálculo del prf) es único para este intercambio y NO DEBE ser igual que el Identificador de Mensaje de otro intercambio de Fase 2 que generó este intercambio informativo. La derivación del Vector de Inicialización, usado con SKEYID_e para encriptar este mensaje, se describe en la Sección 11.

Si la SA de ISAKMP aún no se ha establecido al momento del Intercambio Informativo, el intercambio debe hacerse en "limpio" (sin encriptación)sin una carga HASH adicional.

5. **Grupos de Oakley**

Por medio de IKE, se negocia el grupo dentro del cual se realiza el intercambio de Diffie Hellman. Cuatro grupos, con valores de 1 a 4, se definen debajo. Estos grupos provienen del Protocolo OAKLEY y las características de estos grupos se describieron en la Sección 12 del Capítulo 9, estos grupos se llaman "Grupos de Oakley". La clasificación del atributo para el "grupo" se define en la Sección 10. Todos los valores de 2^15 y superiores se utilizan para los identificadores de grupos privados. Para una discusión sobre la robustez de los grupos de Oakley por defecto vea la Sección 9.

5.1 Grupo 1 de Oakley (768)

Las implementaciones de IKE DEBEN soportar un el Grupo de Exponenciación Modular (MODP) con el siguiente número primo y el generador (vea la Sección 12.1 del Capítulo 9 para más detalles). Este grupo es asignado al identificador uno (1).

El número primo es: 2^768 - 2 ^704 - 1 + 2^64 * { [2^638 Π] + 149686 }
Su valor hexadecimal es:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF
```

El generador es: 2 (en decimal).

5.2 Grupo 2 de Oakley (1024)

Las implementaciones de IKE DEBERÍAN soportar un grupo MODP con el siguiente número primo y el generador (vea la Sección 12.2 del Capítulo 9). Este grupo es asignado al identificador dos (2).

El número primo es: 2^1024 - 2^960 - 1 + 2^64 * { [2^894 Π] + 129093 }
Su valor hexadecimal es:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF
```

El generador es: 2(en decimal).

5.3 Grupo 14 de Oakley (2048)

Las implementaciones de IKE DEBERÍAN soportar un grupo MODP con el siguiente número primo y el generador (vea la Sección 3 del RFC 3526). Este grupo es asignado al identificador dos (14).

El número primo es: 2^2048 - 2^1984 - 1 + 2^64 * { [2^1918 pi] + 124476 }

Su valor hexadecimal es:

```
FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D
C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F
83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D
670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B
E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9
DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510
15728E5A 8AACAA68 FFFFFFFF FFFFFFFF
```

El generador es: 2(en decimal).

5.4 Grupo 3 de Oakley

Las implementaciones IKE DEBERÍAN soportar el grupo de curvas elípticas superiores al campo de Galois GF[2^N], designado EC2N, con las siguientes características (vea la Sección 12.3 del Capítulo 9 para más detalles). Este grupo es asignado al identificador tres (3). La curva se basa en el Campo de Galois GF[2^155]. El tamaño del campo es de 155 bits. El polinomio irreducible para el campo es de:

$$u^{155} + u^{62} + 1$$

La ecuación para la curva elíptica es:

$$y^2 + xy = x^3 + ax^2 + b$$

```
Tamaño del campo (en bits):         155
Grupo Primo/Polinomio Irreducible:
                0x080000000000000000000004000000000000001
Primer Grupo Generado:              0x7b
Grupo Curva A:                      0x0
Grupo Curva B:                      0x07338f

Orden del Grupo: 0X080000000000000000057db5698537193aef944
```

Los datos en la carga KE cuando se usa este grupo es el valor x de la solución (x, y), del punto en la curva seleccionado tomando el secreto aleatoriamente escogido ka y calculando ka*P, donde * es la repetición de la suma del grupo y de las operaciones dobles, P es el punto de la curva con coordenada X igual al generador 1 y la coordenada Y determinada de la ecuación definida. La ecuación de la curva es conocida implícitamente por el Tipo de Grupo y los coeficientes A y B. Hay dos valores posibles para la coordenada Y; cada uno puede ser utilizado exitosamente (las dos partes no necesitan convenir en la selección).

5.5 Grupo 4 de Oakley

Las implementaciones de IKE DEBERÍAN soportar el grupo de curvas elípticas superiores al campo de Galois GF[P], designado ECP, con las siguientes características (vea la Sección 12.4 del Capítulo 9 para más detalles). Este grupo es asignado al identificador cuatro (4). La curva se basa en el Campo de Galois GF[2^155]. El tamaño del campo es de 185 bits. El polinomio irreducible para el campo es de:

$$u^{185} + u^{69} + 1$$

la ecuación para la curva elíptica es:

$$y^2 + xy = x^3 + ax^2 + B.$$

Tamaño del campo (en bits): 185
Grupo Primo/Polinomio Irreducible:
0x020000000000000000000000000000200000000000000001
Primer Grupo Generado: 0x18
Grupo Curva A: 0x0
Grupo Curva B: 0x1ee9

Orden del Grupo: 0X01ffffffffffffffffffffdbf2f889b73e484175f94ebc

Los datos en la carga de KE cuando se usa este grupo son idénticos a cuando se usa el grupo tres (3) de Oakley.

Otros grupos pueden ser definidos usando el Modo Nuevo Grupo. Las características de éstos números primos se describieron en el Capítulo 9.

6. **Ejemplificación de las Cargas en un Intercambio IKE**

Esta sección ilustra cómo se utiliza el protocolo IKE para:

- Establecer un canal seguro y autentificado entre procesos ISAKMP (Fase 1); y

- Generar material clave para una SA IPsec (Fase 2), y negociar SA IPsec (Fase 2).

6.1 Fase 1 Utilizando el Modo Principal

La Figura 1 ilustra las cargas intercambiadas entre dos entidades durante en el primer intercambio de ida y vuelta. El iniciador PUEDE proponer varias propuestas; el respondedor DEBE contestar solo una.

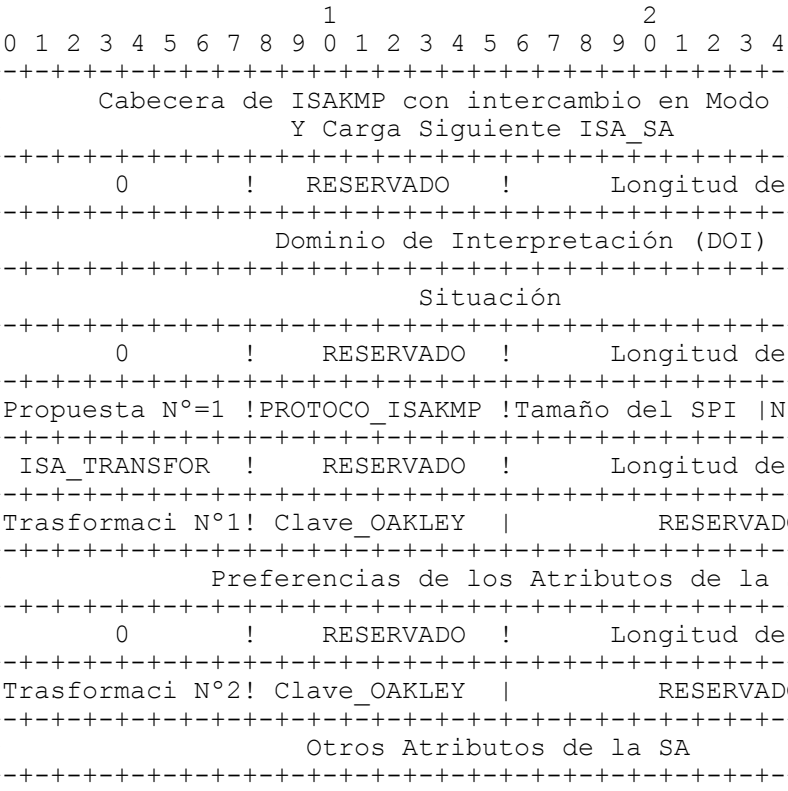


Figura 1: Cargas intercambiadas durante el primer intercambio de la Fase 1 utilizando el Modo Principal.

El respondedor contesta el tipo pero selecciona, y retorna, solo una propuesta de transformación (los atributos de la SA ISAKMP).

El segundo intercambio consiste en las siguientes cargas:

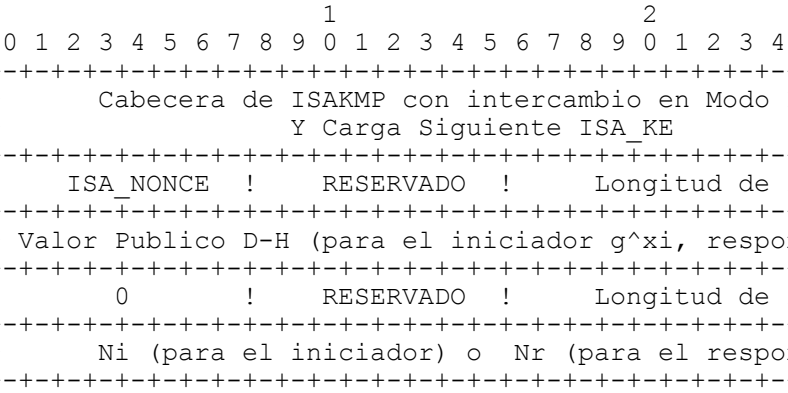


Figura 2: Cargas intercambiadas durante el segundo intercambio de la Fase 1 utilizando el Modo Principal.

Las claves compartidas, SKEYID_e y SKEYID_a, ahora se utilizan para proteger y autentificar todas las futuras comunicaciones. Observe que SKEYID_e y SKEYID_a no están autentificados.

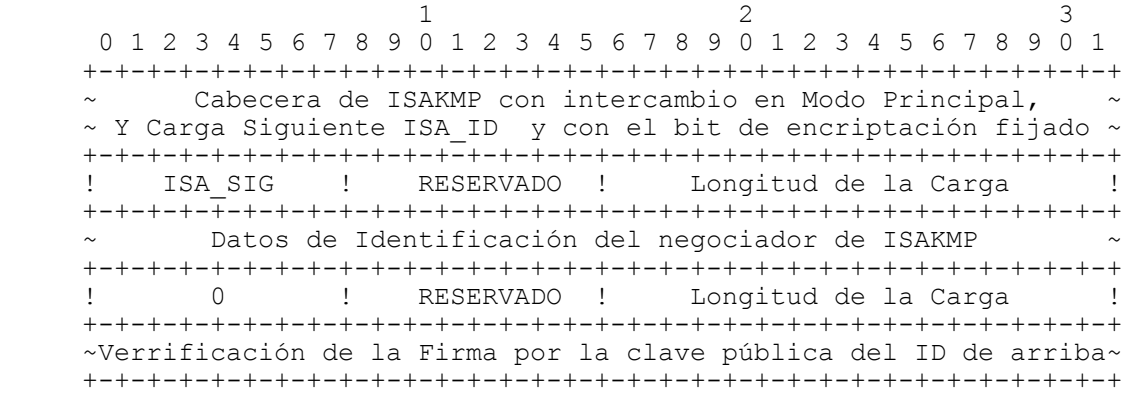


Figura 3: Cargas intercambiadas durante el tercer intercambio de la Fase 1 utilizando el Modo Principal.

El intercambio de clave es autentificado por el hash firmado según lo descripto en la Sección 4.1.1. Una vez que se haya verificado la firma usando el algoritmo de autentificación negociado como parte de la SA ISAKMP, las claves compartidas, SKEYID_e y SKEYID_a se las pueden referir como autentificadas. (Por brevedad, las cargas de certificación no fueron intercambiadas).

6.2 Fase 2 Utilizando el Modo Rápido

Las cargas siguientes se intercambian en el primer ciclo del Modo Rápido con la SA ISAKMP negociada. En este intercambio hipotético, los negociadores de ISAKMP son representantes de otras partes que han solicitado la autentificación.

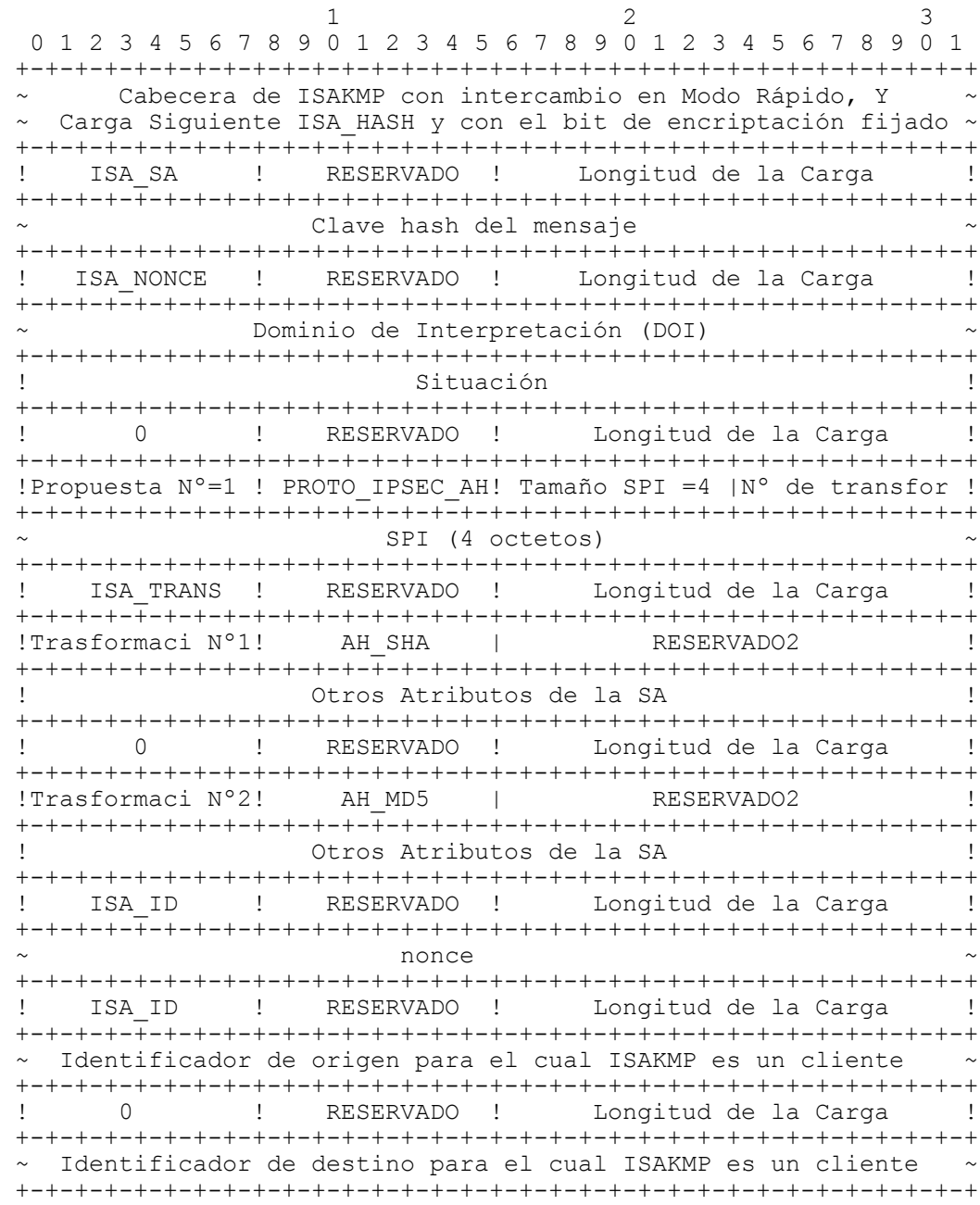


Figura 4: Cargas intercambiadas durante el primer intercambio de la Fase 2.

Donde los contenidos del hash se describen en la Sección 4.2.1. El respondedor contesta con un mensaje similar que contiene solamente una transformación (la transformación seleccionada AH). Al recibir el mensaje, el iniciador puede proporcionar el proceso clave con la SA negociada y el material clave. Como una comprobación contra ataques de anti-replay, el respondedor espera hasta recibir el siguiente mensaje.

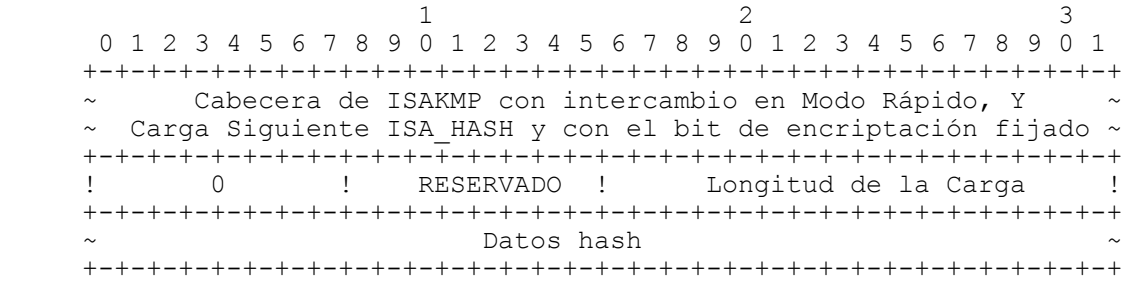


Figura 5: Cargas intercambiadas durante el Segundo intercambio de la Fase 2.

7. **Ejemplo de Perfect Forward Secrecy (PFS)**

El término Perfect Forward Secrecy (PFS), se refiere a la noción que compromete a una única clave que permitirá el acceso solamente a los datos protegidos derivados de esa única clave. Para que el PFS exista la clave usada para proteger la transmisión de los datos NO DEBE ser usada para derivar claves adicionales, y si la clave usada para proteger la transmisión de los datos fue derivada de otro material clave, ese material NO DEBE ser usado para derivar más claves.

Este protocolo puede proporcionar PFS para las claves, las identidades de ambos usuarios de ISAKMP y, si fuera aplicable, a las identidades de quienes lo están negociando se las puede proteger con PFS.

Para proporcionar PFS para las claves y todas las identidades, las dos partes realizarían lo siguiente:

- Un Intercambio en Modo Principal para proteger las identidades de los usuarios de ISAKMP. (Esto establece una SA ISAKMP.)

- Un Intercambio en Modo Rápido para negociar la protección de otro Protocolo de Seguridad. (Esto establece una SA en cada extremo para este protocolo.)

- Elimine la SA ISAKMP y su estado asociado.

Puesto que la clave para usarse en la SA no ISAKMP fue derivada de solo un intercambio efímero de Diffie Hellman, se preserva el PFS.

Para proporcionar PFS solamente a las claves de una SA no ISAKMP, no se necesita hacer un intercambio de Fase 1 si existe una SA ISAKMP entre dos usuarios. Un solo Modo Rápido en el cual se pasa la carga de Intercambio de Claves, y se realiza un intercambio adicional de Diffie Hellman, es todo lo que se requiere. En este punto el estado derivado de este Modo Rápido debe suprimir la SA ISAKMP según lo descripto en la Sección 4.2.1.

8. **Sugerencias para la Implementación de IKE**

Utilizar una única negociación ISAKMP de Fase 1 hace que las subsiguientes negociaciones de Fase 2 sean más rápidas. Siempre y cuando el estado de la Fase 1 permanezca oculto, y el PFS no sea necesario, la Fase 2 puede proceder sin exponenciación. La cantidad de negociaciones de Fase 2 que se pueden realizar para una única Fase 1 es un asunto de la política local. La decisión dependerá de la fuerza de los algoritmos que son utilizados y de

los niveles de confianza en el sistema de usuarios.

Una implementación puede desear negociar una serie de SAs cuando realiza el Modo Rápido. Haciendo esto se puede acelerar el "recambio de claves". El Modo Rápido define la forma en la que se define KEYMAT para una serie de SAs. Cuando un usuario considera que es tiempo de cambiar las SAs simplemente utiliza la siguiente dentro del rango indicado. Una serie de SAs pueden ser establecidas por múltiples SAs negociadas (idénticos atributos, diferentes SPIs) mediante un único Modo Rápido.

Una optimización que es a menudo útil es establecer SA con los usuarios antes de que estas sean necesarias de modo que cuando sean necesarias ya estén en su sitio. Esto asegura que no habrá retrasos debido a la administración de claves antes de la transmisión inicial de datos. Esta optimización es fácilmente implementada estableciendo más de una SA por usuario por cada SA solicitada y ocultando esas SAs no usadas inmediatamente.

Si una implementación ISAKMP es alertada que una SA pronto será necesitada (por ejemplo para sustituir una SA existente que expirará en un futuro próximo), se puede establecer una nueva SA antes de que esta sea necesaria.

En el Capítulo de ISAKMP se describió las condiciones en las cuales una parte puede informar a la otra parte de cierta actividad (tal como la cancelación de una SA o en respuesta a un cierto error en el protocolo tal como una verificación de firma fallida o una carga no desencriptada). Se sugiere encarecidamente que estos intercambios Informativos no sean contestados bajo ninguna circunstancia. Tal condición puede dar lugar a "conflicto de notificaciones" en el sentido de que no se entendería un mensaje resultante de una notificación a un usuario quien no puede entenderlo y envía su propia notificación la cual tampoco es entendida.

9. **Consideraciones de Seguridad**

Este capítulo discute un protocolo híbrido, el cual combinando partes de Oakley y partes de SKEME con ISAKMP, para negociar y derivar el material clave para las SAs de forma segura y autentificada.

La confidencialidad es asegurada con el uso de un algoritmo de encriptación negociado. La autentificación es asegurada con el uso de un método negociado: un algoritmo de firma digital; un algoritmo de clave pública que soporta encriptación; o una clave pre-compartida. La confidencialidad y la autentificación de este intercambio son solamente tan buenos como los atributos negociados como parte de la SA de ISAKMP.

La reiteración del recambio de claves usando el Modo Rápido puede comprometer la entropía del secreto compartido de Diffie Hellman. Los implementadores deberían observar este hecho y fijar un límite de Intercambios en Modo Rápido por medio de la exponenciación.

El PFS para el material clave y para las identidades es factible con este protocolo. Especificando un grupo de Diffie Hellman y pasando los valores públicos en cargas KE, los usuarios de ISAKMP pueden establecer el PFS para las claves-- las identidades serían protegidas por el SKEYID_e de la SA ISAKMP y por lo tanto no serían protegidas con PFS. Si se desea el PFS para el material clave y para las identidades, una ISAKMP de usuario DEBE establecer solamente una SA no ISAKMP (por ejemplo una SA HA o una SA ESP) por SA ISAKMP. El PFS para las claves y para las identidades es llevado a cabo cancelando la SA ISAKMP (y opcionalmente enviando un mensaje de

CANCELACIÓN) al crearse una SA no ISAKMP. De esta forma una negociación de Fase 1 esta unívocamente vinculada a una negociación de Fase 2, y la SA ISAKMP establecida durante la Fase 1 de la negociación nunca más es usada.

La fuerza de una clave derivada de un intercambio de Diffie Hellman usando cualquiera de los grupos definidos aquí depende de la fuerza inherente del grupo, del tamaño del exponente usado, y de la entropía proporcionada por el generador de números aleatorios usado. Debido a estas entradas de información es difícil determinar la fuerza de una clave para cualquiera de los grupos definidos. El grupo de Diffie Hellman por defecto (el primer grupo) cuando esta utilizado con un fuerte generador de números aleatorios y un exponente no menor de 160 bits, es suficiente para utilizar DES. Los grupos dos a cuatro proporcionan gran seguridad. Las implementaciones deberían notar estas conservadoras estimaciones al establecer la política y negociar parámetros de seguridad.

Observe que estas limitaciones están en los grupos de Diffie Hellman. No hay nada en IKE que prohíba el uso de grupos más fuertes ni que disminuya la fuerza obtenida a partir de grupos más fuertes. De hecho, el marco extensible de IKE alienta la definición de más grupos; el uso de grupos de curvas elípticas aumentará en gran medida la fuerza usando números mucho más pequeños.

Para las situaciones donde los grupos definidos proporcionan fuerza insuficiente, el Modo Nuevo Grupo se puede utilizar para intercambiar un grupo de Diffie Hellman que proporcione la fuerza necesaria. Es responsabilidad de las implementaciones controlar el carácter primo de los grupos que se ofrecen y la estimación de la fuerza del grupo.

Se asume que los exponentes de Diffie Hellman de los intercambios son borrados de la memoria después de ser usados. En particular, estos exponentes no deben ser derivados a partir de secretos permanentes como la seed de un generador seudo-aleatorio.

Los intercambios IKE mantienen activos los Vectores de Inicialización (IV) donde el último bloque de texto cifrado del último mensaje es el IV para el siguiente mensaje. Para prevenir retransmisiones (o mensajes falsificados con cookies válidas) que produzcan intercambios fuera del sincronismo de IKE, las implementaciones NO DEBERÍAN actualizar sus IV hasta que el mensaje desencriptado ha pasado por una comprobación de coherencia y se determine realmente adelantar la máquina de estado de IKE (es decir no es una retransmisión).

Mientras que el último viaje de ida y vuelta en Modo Principal (y opcionalmente el ultimo mensaje en Modo Agresivo) está encriptado, en sentido estricto, no está autentificado. Un ataque activo de substitución contra el texto cifrado podría resultar en la corrupción de la carga. Si tal ataque corrompe las cargas obligatorias sería detectado por un error en la autentificación, pero si corrompe alguna de las cargas opcionales (por ejemplo cargas de Notificación añadidos en el último mensaje de un intercambio en Modo Principal) es posible que no sea perceptible.

10. **Atributos**

10.1 Números Asignados a los Atributos

Los atributos negociados en IKE son identificados por su clase. Los atributos de la Fase 2 se definen en la especificación pertinente al DOI (por ejemplo, los atributos del DOI de IPsec se definieron en el Capítulo

8), a excepción de una descripción de grupo cuando el Modo Rápido incluye un intercambio efímero de Diffie Hellman. Los tipos de atributo pueden ser Básico (B) o Longitud-Variable (V). La codificación de estos atributos se definió en la Sección 3.3 del Capítulo 7 como Tipo/Valor (Básico) y Tipo/Longitud/Valor (Variable).

La descripción de atributos como básico NO DEBE ser codificada como variable. El atributo longitud variable PUEDE ser codificado como atributo básico si su valor puede caber dentro de dos octetos. Si éste es el caso, un atributo ofrecido como variable (o básico) por el iniciador de este protocolo PUEDE regresar al iniciador como básico (o variable).

10.2 Clases de Atributos

Los Atributos negociados durante la Fase 1 usan las siguientes definiciones.

clase	Valor	Tipo
Algoritmo de Encriptación	1	B
Algoritmo Hash	2	B
Método de Autentificación	3	B
Descripción del Grupo	4	B
Tipo de Grupo	5	B
Grupo Primo/Polinomio Irreducible	6	V
Primer Grupo Generado	7	V
Segundo Grupo Generado	8	V
Grupo Curva A	9	V
Grupo Curva B	10	V
Tipo de Vida	11	B
Tiempo de Vida	12	V
Función seudo Aleatoria (PRF)	13	B
Longitud de la Clave	14	B
Tamaño del Campo	15	B
Orden del Grupo	16	V

Los valores de 17 a 16383 están reservados por la IANA. Los valores de 16384 a 32767 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Algoritmos de Encriptación

  Los valores de la Clase de Algoritmo de Encriptación definen un algoritmo de encriptación para utilizarse con IKE cuando es requerido.

Clase	Valor	Referencia
DES-CBC	1	[DES]
IDEA-CBC	2	[IKE]
Blowfish-CBC	3	[IKE]
RC5-R16-B64-CBC	4	[IKE]
3DES-CBC	5	[IKE]
CAST-CBC	6	[IKE]
AES-CBC	7	[RFC3602]
CAMELLIA-CBC	8	[kato-ipsec]

Los valores de 9 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Algoritmos Hash

Los valores de la Clase de Algoritmo Hash definen un algoritmo de hash para utilizarse con IKE cuando sea requerido. Debido a la derivación de claves y al uso de expansión de claves de tipo HMAC de los algoritmos en IKE, los pedidos de asignación de nuevos valores de algoritmo hash deben considerar las características criptográficas, por ejemplo la resistencia a la colisión del algoritmo de hash.

Clase	Valor	Referencia
MD5	1	[MD5]
SHA	2	[FIPS-180-1]
Tiger	3	[TIGER]
SHA2-256	4	[IANA-1]
SHA2-384	5	[IANA-1]
SHA2-512	6	[IANA-1]

Los valores de 7 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Método de Autentificación

Los valores del la Clase de Método de Autentificación definen un algoritmo de autentificación para utilizarse con IKE cuando es requerido.

Clase	Valor	Referencia
Claves pre-compartidas	1	[IKE]
Firmas DSS	2	[IKE]
Firmas RSA	3	[IKE]
Encriptación con RSA	4	[IANA-1]
Encriptación revisada con RSA	5	[IANA-1]
Encriptación con El-Gamal	6	[IANA-1]
Encriptación Revisada con El-Gamal	7	[IANA-1]
Firmas ECDSA	8	[IANA-1]

Los valores de 9 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Descripción del Grupo

Los valores de la Clase de Descripción de Grupo identifican a un grupo para utilizarse en un intercambio de Diffie Hellman.

Clase	Valor	Referencia
Grupo MODP por defecto 768-bit	1	[IKE]
Grupo MODP alternativo 1024-bit	2	[IKE]
Grupo EC2N sobre el GP[2^155]	3	[IKE]
Grupo EC2N sobre el GP[2^185]	4	[IKE]
Grupo MODP de 1536-bit	5	[RFC3526]
Grupo EC2N sobre el GP[2^163]	6	[IANA-1]
Grupo EC2N sobre el GP[2^163]	7	[IANA-1]
Grupo EC2N sobre el GP[2^283]	8	[IANA-1]
Grupo EC2N sobre el GP[2^283]	9	[IANA-1]
Grupo EC2N sobre el GP[2^409]	10	[IANA-1]
Grupo EC2N sobre el GP[2^409]	11	[IANA-1]
Grupo EC2N sobre el GP[2^571]	12	[IANA-1]
Grupo EC2N sobre el GP[2^571]	13	[IANA-1]
Grupo MODP de 2048-bit	14	[RFC3526]
Grupo MODP de 3072-bit	15	[RFC3526]
Grupo MODP de 4096-bit	16	[RFC3526]
Grupo MODP de 6144-bit	17	[RFC3526]
Grupo MODP de 8192-bit	18	[RFC3526]

Los valores de 19 a 32767 están reservados por la IANA. Los valores de 32768 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Tipo de Grupo

Los valores de la Clase Tipo de Grupo definen el tipo de grupo.

Clase	Valor
MODP (grupo exponencial modular)	1
ECP (grupo curva elíptica sobre GF[P])	2
EC2N (grupo curva elíptica sobre GF[2^N])	3

Los valores de 4 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Tipo de Vida

Para un Tipo de Vida dado, el valor del atributo del Tiempo de Vida define la longitud actual del componente tiempo de vida, en un número de segundos, o un número en kilobytes que pueden ser protegidos. Los valores de la Clase Tipo de Vida definen un tipo de tiempo de vida el cual es aplicado a la SA ISAKMP.

Clase	Valor
Segundos	1
Kilobytes	2

Los valores de 3 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado. Para un "Tipo de Vida" dado el valor del atributo "Tiempo de Vida" define la duración de la SA, un número de segundos, o un número en Kbytes.

- Función Seudo Aleatoria (PRF)

  No hay actualmente funciones seudo-aleatorias definidas.

  Los valores de 1 a 65000 están reservados por la IANA. Los valores de 65001 a 65535 son para usarse privadamente entre usuarios que mutuamente lo han acordado.

- Longitud de la Clave

  Al usar un algoritmo de encriptación que tiene una longitud de clave variable, este atributo especifica la longitud de la clave en bits. Se DEBE utilizar el orden de byte de red. Este atributo NO DEBE ser utilizado cuando el algoritmo de encriptación especificado use una clave de longitud fija.

- Tamaño del Campo

  El tamaño del campo, en bits, de un grupo de Diffie Hellman.

- Orden del Grupo

  El Orden del Grupo de un grupo de curvas elípticas. Observe que la longitud de este atributo depende del tamaño del campo.

- Los Intercambios Adicionales Definidos

Clase	valores del XCHG
Modo Rápido	32
Modo Nuevo Grupo	33

11. **Encriptación de Mensajes ISAKMP**

Esta sección describe los detalles de encriptación que se utilizarán SOLAMENTE al encriptar mensajes ISAKMP. Cuando un servicio (tal como una transformación IPsec) utiliza ISAKMP para generar material clave, todos los detalles específicos del algoritmo de cifrado (tales como generación de claves, IV, relleno, etc...) DEBEN estar definidos por ese servicio. ISAKMP no pretende generar siempre las claves que son convenientes para cualquier algoritmo de encriptación. ISAKMP produce la cantidad solicitada de material clave del cual el servicio DEBE generar una clave conveniente. Los detalles, tales como control de claves, son responsabilidad del servicio.

El uso de Algoritmos seudo-Aleatoríos (PRFs) negociados puede requerir que la salida del PRF se amplié debido al mecanismo de retroalimentación del PRF empleado en IKE. Por ejemplo, si él DOORAK-MAC (algoritmo ficticio) requiere 24 bytes de clave pero produce solamente 8 bytes de salida, la salida se debe ampliar tres veces antes de que sea utilizada como clave por otra instancia de este. La salida de un PRF es ampliada retroalimentando los resultados del PRF para generar bloques sucesivos. Se concatenan estos bloques hasta que el número indispensable de bytes se ha alcanzado. Por ejemplo, para la autentificación de claves pre-compartidas con DOORAK-MAC el PRF negociado es:

```
    BLOCK1-8 = prf(clave pre-compartida, Ni_b | Nr_b)
    BLOCK9-16 = prf(clave pre-compartida, BLOCK1-8 | Ni_b | Nr_b)
    BLOCK17-24 = prf(clave pre-compartida, BLOCK9-16 | Ni_b | Nr_b)
entonces
```

```
  SKEYID = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

por consiguiente para derivar SKEYID_d:

```
  BLOCK1-8 = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
  BLOCK9-16 = prf(SKEYID, BLOCK1-8 | g^xy | CKY-I | CKY-R | 0)
  BLOCK17-24 = prf(SKEYID, BLOCK9-16 | g^xy | CKY-I | CKY-R | 0)
```

entonces

```
  SKEYID_d = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

Subsiguientes PRF son derivados de forma similar.

Las claves usadas para proteger la SA ISAKMP es derivada a partir de SKEYID_e de un algoritmo específico. Cuando SKEYID_e no es suficientemente largo para proveer todo el material clave necesario que un algoritmo requiere, la clave se deriva a partir de la concatenación de los resultados de una función seudo-aleatoria dentro de sí misma, concatenando los resultados, y tomando los bits de orden superior necesarios.

Por ejemplo, si el algoritmo AKULA (ficticio) requiere 320 bits de clave (y no tiene ningún control de clave débil) y el prf usado para generar SKEYID_e genera solamente 120 bits de material, la clave para AKULA, sería los primeros 320 bits de ka, donde:

```
    Ka = K1 | K2 | K3
y
    K1 = prf(SKEYID_e, 0)
    K2 = prf(SKEYID_e, K1)
```

Donde el prf es el prf negociado o la versión HMAC de la función hash negociada (si no se negoció ningún prf) y el cero (0) es representado por un solo octeto. Cada resultado del prf proporciona 120 bits de material para un total de 360 bits. AKULA utilizaría los primeros 320 bits de esa cadena de 360 bits.

En la Fase 1, el material para el Vector de Inicialización (IV) para el algoritmo de encriptación en modo CBC es derivado a partir de un hash de una concatenación del valor público de Diffie Hellman del iniciador y del valor público de Diffie Hellman del respondedor usando el algoritmo hash negociado. Esto se utiliza solamente para el primer mensaje. Cada mensaje debería ser rellenado hasta el tamaño de bloque más cercano usando bytes que contengan 0x00. La longitud del mensaje en la cabecera DEBE incluir la longitud del relleno, puesto que éste refleja el tamaño del texto cifrado. Los mensajes subsiguientes DEBEN utilizar el último bloque CBC encriptado del mensaje anterior como su IV.

En la Fase 2, el material para el IV para la encriptación en modo CBC del primer mensaje en un intercambio en Modo Rápido es derivado a partir de un hash de una concatenación del último bloque CBC de salida de la Fase 1 y del identificador del mensaje de la Fase 2 usando el algoritmo hash negociado. EL IV para los subsiguientes mensajes dentro de un intercambio en Modo Rápido es el CBC del bloque de salida del mensaje anterior. El relleno y los IVs para los mensajes subsiguientes se realizan como en la Fase 1.

Después de que se haya autentificado la SA ISAKMP todos los Intercambios Informativos se encriptan usando SKEYID_e. El IV para estos intercambios se deriva exactamente de la misma manera que para el Modo Rápido; es decir, se deriva a partir de un hash de una concatenación del último bloque CBC de salida de la Fase 1 y del identificador de mensaje de la cabecera ISAKMP del Intercambio Informativo (no del identificador de mensaje que pudo haber originado el Intercambio Informativo).

Observe que al final de la Fase 1 el bloque CBC de salida, resultante de la encriptación/desencriptación del último mensaje de la Fase 1, se debe conservar en el estado SA ISAKMP para permitir la generación de los IVs únicos para cada Modo Rápido. Cada intercambio posterior a la Fase 1 (los Modos Rápidos e Intercambios Informativos) generan IVs independientes para evitar que IVs se salga de sincronismo cuando dos intercambios diferentes comienzan simultáneamente.

En todos estos casos, hay un solo contexto bidireccional de cifrado/IV. Considerando que cada Modo Rápido e Intercambio Informativo mantienen un único contexto evitando que el IVs se salga de sincronismo.

La clave para DES-CBC se deriva de los primeros 8 bytes no débiles ni semi débiles de SKEYID_e (ver la Sección 10). El IV es los primeros 8 bytes del material IV derivado anteriormente.

La clave para IDEA-CBC se deriva de los primeros 16 bytes de SKEYID_e. El IV es los primeros 8 bytes del material del IV derivado anteriormente.

La clave para el blowfish-CBC es el tamaño de la clave negociada, o los primeros 56 bytes de una clave (sino se negocia ningún tamaño de clave) derivado a partir del método seudo-aleatorio de retroalimentación de la función ya mencionada. El IV es los primeros 8 bytes del material IV derivado anteriormente.

La clave para RC5-R16-B64-CBC es el tamaño de la clave negociada, o los primeros 16 bytes de una clave (sino se negocia ningún tamaño de clave) derivado a partir del método seudo-aleatorio de retroalimentación de la función ya mencionada si fuera necesario. El IV es los primeros 8 bytes del material del IV derivado anteriormente. El número de ciclos DEBE ser 16 y el tamaño de bloque DEBE ser de 64.

La clave para 3DES-CBC es los primeros 24 bytes de una clave derivado a partir del método seudo-aleatorio de retroalimentación de la función ya mencionada. 3DES-CBC es una operación de encriptación-desencriptación-encriptación que usa la primera, mitad, y los últimos 8 bytes de la clave entera de 3DES-CBC. El IV es los primeros 8 bytes del material del IV derivado anteriormente.

La clave para CAST-CBC es el tamaño de la clave negociada, o los primeros 16 bytes de una clave derivada a partir del método seudo-aleatorio de retroalimentación de la función ya mencionada. El IV es los primeros 8 bytes del material del IV derivado anteriormente.

El soporte para otros algoritmos, a excepción del DES-CBC, es puramente opcional. Algunos algoritmos opcionales pueden depender de derechos de propiedad intelectual.

12. **Algoritmos para IKE**

Los algoritmos requeridos y sugeridos en la especificación original del

Protocolo de Intercambio de Claves en Internet versión 1 (IKEv1) [IKE], no reflejan la realidad actual de las necesidades de IPsec. Esta sección pretende resumir las principales diferencias existentes entre [IKE] y el RCF [IKEupdate] que actualizó los algoritmos para IKE. A continuación se muestra una tabla comparativa con las diferencias y actualizaciones de los requerimientos actuales.

Algoritmo	RFC 2409	RFC 4109
DES para la encriptación	DEBE	PUEDE (1)
3DES para la encriptación	DEBERÍA	DEBE
AES-128 en modo CBC [RFC3602] para la encriptación	No lo menciona	DEBERÍA
MD5 para el hashing y para HMAC	DEBE	PUEDE (1)
SHA1 para el hashing y para HMAC	DEBE	DEBE
Tiger para el hashing	DEBERÍA	PUEDE (2)
AES-XCBC-MAC-96 ([RFC3566] y [RFC3664]) para el PRF	No lo menciona	DEBERÍA
Secretos pre-compartidos	DEBE	DEBE
RSA para al autentificación con firmas	DEBERÍA	DEBERÍA
DSA para al autentificación con firmas	DEBERÍA	PUEDE (2)
RSA para al autentificación con encriptación	DEBERÍA	PUEDE (2)
Grupo 1 de D-H (768)	DEBE	PUEDE (1)
Grupo 2 de D-H (1024)	DEBERÍA	DEBE
Grupo 14 de D-H (2048) [RFC3526]	No lo menciona	DEBERÍA
Grupo de Curvas elípticas de D-H	DEBERÍA	PUEDE (2)

(1)= Debido a debilidades criptograficas
(2)= Debido a la falta de implementaciones y/o interoperatividad

Nota: Se recuerda que las palabras DEBE (MUST), NO DEBE (MUST NOT), REQUERIDO (REQUIRED), PODER (SHALL), NO PODER (SHALL NOT), DEBERÍA (SHOULD), NO DEBERÍA (SHOULD NOT), RECOMENDADO (RECOMMENDED), PUEDE (MAY) y OPCIONAL (OPTIONAL), cuando aparezcan en este libro deben interpretarse como se describe en [Bra97].

Existe una versión 2 de IKE (IKEv2) [IKEv2] que al momento de realizar esta versión del libro todavía no se ha convertido en un estándar (es un draft de internet). El cual de convertirse en estándar puede que deje obsoleto los RFC 2407 (el del DOI de IPsec para ISAKMP), 2408 (el de ISAKMP), 2409 (el de IKEv1), debido a pretende englobar y actualizar esos RFCs.

Apéndice A - Acrónimos

Acrónimo	Significado
AAA	Autenticación, Autorización y Contabilidad (Authentication, Authorization and Accounting)
AES	Estándar de Cifrado Avanzado (Advanced encryption standard)
AH	Cabecera de autentificación (Authentication Header)
API	Interfaz de Programa de Aplicación (Application Program Interface)
BITS	Puesto en la Pila (Bump-in-the-stack)
BITW	Puesto en el Cable (Bump-in-the-wire)
CA	Autoridad de Certificación (Certificate Athority)
CBC	Concatenación de Bloques Cifrados (cipher block chiang)
CCA	Autoridad de Certificación de País (country certification authority).
CFB	Modo de Retroalimentación Cifrado (cipher feedback mode).
CIDR	Enrutamiento entre Dominios sin Clase (Classless Inter-Domain Routing)
CRL	Lista de Renovación de Certificados
D-H	Algoritmo de Diffie-Hellman
DES	Estándar de Cifrado de Datos (Data encryption standard)
DH	Algoritmo de Diffie-Hellman
DHCP	Protocolo de Configuración Dinámica de Host (Dynamic host configuration protocol)
DNS	Sistema de Nombres de Dominio (Domain Name System)
DOI	Dominio de Interpretación (Domain of Interpretation)
DoS	Denegación de servicio (denial of service)
DSS	Estándar de Firma Digital (Digital Signature Standard)
ECB	Bloque de Código Electrónico (Electronic Code Block)
ESP	Carga de Seguridad Encapsulada (Encapsulation security payload)
FDDI	Interfaz de Datos Distribuidos por Fibra (Fiber Distributed Data Interface)
FP	Prefijo de Formato (Format Prefix)
GKMP	Protocolo de Administración de Claves para Grupos (Group Key Management Protocol)
HASH	Función que genera un número a partir de una cadena de texto
HMAC	Hash con claves para la autenticación de mensajes (Keyed Hashing for Message Authentication, [ARCH])

Acrónimo	Significado
IANA	Autoridad de Asignación de Números en Internet (Internet Assigned Numbers Authority)
ICMP	Protocolo de Mensajes de Control de Internet (Internet Control Message Protocol)
ICV	Valor de Comprobación de Integridad (Integrity Check Value)
IDEA	Algoritmo Internacional de Encriptado de Datos (international data encryption algorithm)
IEEE	Instituto de Ingeniería Eléctrica y Electrónica (Institute of Electrical and Electronic Engineers)
IETF	Grupo de Trabajo de Ingeniería de Internet (Internet Engineering Task Force)
IGMP	Protocolo de Administración de Grupos de Internet (Internet Group Management Protocol)
IKE	Intercambio de Claves en Internet (Internet Key Exchange)
IP	Protocolo Internet (Internet Protocol)
IPng	la siguiente generación de IP (IP-The Next Generation)
IPRA	Autoridad de Registración de Políticas en Internet (Internet Policy Registration Authority)
IPsec	Seguridad IP (IP Security)
IPv4	IP versión 4 (IP Version 4)
IPv6	IP versión 6 (IP Version 6)
ISAKMP	Protocolo de Manejo de Claves y Asociaciones de Seguridad en Internet (Internet security association and key management protocol)
ISP	Proveedor de Servicios de Internet (Internet service provider)
IV	Vector de Inicialización (Initialization vector)
KDC	Centro de Distribución de Claves (Key Distribution Center)
KEK	Clave de Encritación de Claves (Key Encrypting Keys)
L2TP	Protocolo de Túnel de Capa 2 (Layer 2 Tunnelling Protocol)
LAN	Red de Área Local (Local area network)
MAC	Control de Acceso al Medio (Media Access Control
MAC	Código de Autentificación de Mensaje (Message Authentication Code)
MD4/5	Condensado (o Resumen) de Mensaje 4/5 (Message Digest)
MDC	Códigos de Detección de Modificación (Modification Detection Codes)
MID	Identificador de Mensaje (Message ID)
MLD	Descubrimiento de Escucha de Multidifusión (Multicast Listener Discovery)
MLS	Seguridad Multinivel (Multi-Level Security)
NAT	Traductor de Direcciones de Red (Network Address Translator)

Acrónimo	Significado
NIST	Instituto Nacional de Estándares y Tecnología (National Institute of Standards and Technology (US))
NLA	Agregador de Nivel Siguiente (Next-Level Aggregator)
OAKLEY	Protocolo de Determinación de Claves (Key Determination Protocol)
PCA	Autoridades de Certificación de Políticas (Policy Certification Authorities)
PFS	Perfect Forward Secrecy (No tiene traducción exacta al castellano ver el Apendice Glosario)
PGP	Privacidad Bastante Buena (Pretty Good Privacy)
PKI	Infraestructura de Clave Pública (Public-key infrastructure)
PKIX	PKI Basada en X.509v3 (Public-key infrastructure (X.509))
prf	función seudo aletoría (pseudo-random function)
PRF	Función Seudo Aleatoría (Pseudo-random Function)
QoS	Calidad de Servicio (Quality of Service)
RADIUS	Remote authentication dial-in user service
RC2	Cifrado de Rivest (Rivest's Cipher)
RSA	Rivest, Shamir y Adleman Algoritmo de Clave Pública (Rivest-Shamir-Adleman public-key system)
S/MIME	Extensiones de Correo Internet Multipropósito Seguras (Secure multipurpose Internet mail extension)
SA	Asociaciones de Seguridad (Security association)
SAD	Base de Datos de Asociaciones de Seguridad (Security Association Databases)
SG	Pasarela de Seguridad (security gateway)
SHA	Algoritmo de Troceo Seguro (Secure Hash Algorithm)
SHA-1	Algoritmo de Generación Numérica Seguro N°1 (Secure Sash Algorithm 1)
SLA	Agregador de nivel de sitio (Site-Level Aggregator)
SML	Capa de Gestión de Servicio (service management layer).
SNAP	Protocolo de Acceso a Subredes (Sub-Network Access Protocol)
SPD	Base de Datos de Políticas de Seguridad (Security Policy Database)
SPI	Índice de Parámetros de Seguridad (Security Parameters Index)
SSH	Secure Shell
SSL	Capa de Conector Seguro (Secure socket layer)
TCP	Protocolo de Control de Transporte (Transmission Control Protocol)
TEK	Clave de Encritación de Tráfico (Traffic Encryption Keys)
TLA	Agregador de Nivel Superior (Top Level Aggregator)
TLS	Seguridad en la Capa de Trasporte (Transport layer security)
TOS	Tipo de Servicio (Type of Service)

Acrónimo	Significado
UDP	Protocolo de Datagramas de Usuario (User datagram protocol)
URL	Localizador de Recursos Universal (Universal resource locator)
VPN	Red privada virtual (Virtual private network)
X.509	Certificados de Clave Pública por el Estandad ITU (ITU standard for public key certificates)

Apéndice B - Glosario

Esta sección provee definiciones para términos claves que son utilizados en este documento. Otros documentos proporcionan definiciones adicionales y información de trasfondo relacionadas a esta tecnología como por ejemplo [VK83], [HA94]. En este glosario se incluyen términos de servicios de seguridad genéricos y de mecanismos de seguridad, más términos específicos de IPsec.

Accesibilidad
: La accesibilidad, cuando es vista como un servicio de seguridad, trata las preocupaciones de seguridad generadas por ataques contra redes que deniegan o degradan servicios. Por ejemplo en el contexto IPsec, el uso de mecanismo de anty-replay en AH y en ESP soportan accesibilidad [ARCH].

Algoritmo de Hash
: Algoritmo que genera un valor hash de algún dato, como una clave de mensaje o de sesión. Con un buen algoritmo de hash, los cambios que se produzcan en los datos de entrada pueden cambiar todos los bits del valor hash resultante, por lo que estos valores son útiles para detectar cualquier modificación en un objeto de datos, como un mensaje. Además, un buen algoritmo de hash hace que sea computacionalmente imposible crear dos entradas independientes que tengan el mismo valor hash. Los algoritmos de hash comunes son MD2, MD4, MD5 y SHA-1. Estos algoritmos también se llaman funciones de hash [win2003].

Algoritmo de hash seguro (SHA-1)
: Algoritmo que genera un valor hash de 160 bits a partir de una cantidad arbitraria de datos de entrada. SHA-1 se utiliza con el Algoritmo de firma digital (DSA) en el Estándar de firma digital (DSS), entre otros sitios.[win2003]

Algoritmo
: En criptografía, un proceso matemático que se utiliza en operaciones criptográficas como el cifrado y la firma digital de datos. Los algoritmos suelen utilizarse con una clave de cifrado para mejorar la seguridad [win2003].

Algoritmos criptográficos Rivest-Shamir-Adleman (RSA)
: Conjunto de algoritmos de claves públicas ampliamente utilizados publicados por RSA Data Security, Inc. y admitidos por los proveedores de servicios criptográficos básicos y de servicios criptográficos mejorados de Microsoft.

Análisis del tráfico
: El análisis del flujo de tráfico de red para propósitos de deducir información que le es útil al adversario. Ejemplo de tal información son: frecuencia de transmisión, identidades de las partes, tamaño de los paquetes, identificadores de flujo, etc. [Sch94].

Anti-replay
: Ver "Integridad"

Anycast
: un paquete hacia un solo hosts de la lista

Asociación de seguridad (SA)
: Una Asociación de Seguridad es un conjunto de parámetros específicos del protocolo de seguridad que definen completamente los servicios y mecanismos necesarios para proteger el tráfico en ese lugar del protocolo de seguridad. Estos parámetros pueden incluir identificadores de algoritmos, modos, claves criptográficas, etc. La SA hace referencia a su protocolo de seguridad asociado (por ejemplo "SA ISAKMP", "SA ESP", "SA TLS")[ISAKMP]. Las SA IPsec (SA ESP, SA AH) es una conexión lógica unidireccional, que ofrece servicios de seguridad al tráfico transportado por este. Una SA es identificada unívocamente por un trío que consiste en: un Índice de Parámetros de Seguridad (SPI), una Dirección IP de Destino, y un identificador de protocolo de seguridad (por ejemplo, AH o ESP).

Ataque de Denegación de Servicio
: Ataque en el que un intruso aprovecha un defecto o una limitación de diseño de un servicio de red para sobrecargar o detener el servicio, de forma que éste no está disponible para su uso. Generalmente, este tipo de ataque se inicia para impedir que otros usuarios utilicen un servicio de red, como un servidor Web o un servidor de archivos. [win2003]

Ataque por Usuario Interpuesto
: Ataque a la seguridad en el que un intruso intercepta, y posiblemente modifica, datos que se transmiten entre dos usuarios. El intruso pretende hacerse pasar por la otra persona para cada uno de los usuarios. En un ataque por usuario interpuesto con éxito, los usuarios desconocen que hay un intruso entre ellos, que intercepta y modifica sus datos. También se conoce como ataque de la brigada de bomberos (bucket brigade attack).[win2003]

Autentificación del Origen de los Datos
: La autentificación del origen de los datos es un servicio de seguridad que verifica la identidad de origen de los datos. Este servicio usualmente trabaja en conjunto con el servicio de integridad sin conexión [ARCH].

Autentificación
: Este termino se usa informalmente para referirse a la combinación de dos servicios de seguridad distintos, autentificación del origen de los datos y integridad sin conexión. Ver las definiciones debajo para cada uno de estos servicios [ARCH].

Autoridad de Certificación (CA)
: Entidad encargada de establecer y avalar la autenticidad de las claves públicas pertenecientes a sujetos (normalmente usuarios o equipos) u otras entidades emisoras de certificados. Entre las actividades de una entidad emisora de certificados se encuentran enlazar claves públicas a nombres completos mediante certificados firmados, administrar los números de serie de los certificados y revocar certificados. [win2003]

birthday attack (Ataque del Día de Cumpleaños)
: El nombre deriva de la probabilidad de que dos o más personas en un grupo de 23 personas, compartan la misma fecha de cumpleaños es menor que 0.5, (conocida como paradoja del cumpleaños). El birthday attack es un nombre usado para referirse a una clase de ataque por fuerza bruta. Para una función hash que tiene como salida una cadena de 160 bits, es necesario recorrer entonces 2^80 mensajes para poder encontrar dos con el mismo hash, o sea encontrar una colisión.

Bitmask
: Una serie de bits destinados a realizar una comparación lógica con un valor bit existente (Definición extraída del Diccionario de IBM Corp.).

BITS (Puesto en la Pila)
: IPsec se implementa "por debajo" de una implementación existente de una pila IP, entre el IP nativo y los drivers locales de la red. El acceso al código fuente para la pila IP no es requerido en este contexto, este contexto es apropiado para los sistemas antiguos. Este método, cuando se adopta, se emplea generalmente en hosts [ARCH.

BITW (Puesto en el cable).
: El uso de un procesador criptográfico externo es una característica de diseño común de los sistemas de seguridad de red usados por los militares, y en algunos sistemas comerciales. Tales implementaciones se pueden diseñar para asistir a un host o un gateway (o a ambos). El dispositivo BITW generalmente tiene una IP direccionable. Cuando asiste a un único host, puede resultar análogo a una implementación BITS, pero en un router o en un firewall debe funcionar como un security gateway [ARCH].

Bootstrapping
: Es generalmente un término más extenso para el arranque o proceso de inicio de un sistema. El bootstrapping también puede referirse al proceso por el cual una referencia inicial del servicio de nombramiento es obtenida .

Broadcast (difusión)
: Paquete de datos enviado a todos los nodos de una red. Los broadcasts se identifican mediante una dirección de broadcast

Capa Superior
: una capa de protocolo inmediatamente encima de la capa IP Ejemplos son los protocolos transporte tal como el TCP y el UDP, protocolos control tal como el ICMP, protocolos enrutamiento tal como el OSPF, y protocolos internet o de capa inferior que están siendo "tunelizados" sobre (es decir, encapsulados dentro) IP tal como el IPX, el AppleTalk, o el mismo IP.

Carga
: ISAKMP define varios tipos de cargas, que son usadas para transferir información según los datos de la SA, o los datos del intercambio de claves, dentro de las formas definidas en el DOI. Una carga consiste en una cabecera de carga genérica y en octetos encadenados que están ocultos para ISAKMP. ISAKMP usa funcionalidades específicas del DOI para sintetizar e interpretar esas cargas. Múltiples cargas pueden ser enviadas en un único mensaje de ISAKMP. Ver la Sección 3 del Capítulo 7 para más detalles de tipos de carga, y [IPDOI] para los formatos de seguridad de cargas DOI IP IETF [ISAKMP].

Cifrado de Clave Pública
: Método de cifrado que utiliza dos claves de cifrado relacionadas matemáticamente. Una se denomina clave privada y es confidencial. La otra se denomina clave pública y se entrega libremente a todos los posibles destinatarios. En una situación típica, un remitente utiliza la clave pública del destinatario para cifrar un mensaje. Sólo el destinatario tiene la clave privada correspondiente para descifrar el mensaje. La complejidad de esta relación entre la clave pública y la clave privada supone que, siempre que ambas tengan una longitud apropiada, resulta computacionalmente imposible determinar una a partir de la otra. También se denomina cifrado asimétrico [win2003].

Cifrado Simétrico
: Algoritmo de cifrado que requiere el uso de una misma clave secreta tanto en el cifrado como en el descifrado. Gracias a su velocidad, el cifrado simétrico se utiliza normalmente cuando el remitente de un mensaje necesita cifrar grandes volúmenes de datos. También se denomina cifrado de clave secreta [win2003].

Cifrado
: Proceso de camuflar un mensaje o datos de forma que se oculte su contenido. Método para formar un mensaje oculto. El cifrado se utiliza para transformar un mensaje legible, denominado texto sin formato (también denominado texto no cifrado) en un mensaje ilegible, codificado u oculto denominado texto cifrado. Solamente aquel usuario con una clave de descodificación secreta puede convertir dicho texto en el texto original [win2003].

Clave (Key)
: Valor utilizado junto con un algoritmo para cifrar o descifrar datos. [win2003]

Clave Privada
: Mitad secreta de una pareja de claves criptográficas que se utiliza con un algoritmo de clave pública. Las claves privadas se usan, normalmente, para descifrar una clave de sesión simétrica, firmar datos digitalmente o descifrar datos que han sido cifrados con la clave pública correspondiente [win2003].

Clave Pública
: Mitad no secreta de una pareja de claves criptográficas que se utiliza con un algoritmo de clave pública. Las claves públicas se utilizan normalmente para cifrar una clave de sesión, comprobar una firma digital o cifrar datos que han sido descifrados con la clave privada correspondiente [win2003].

Clave Simétrica
: Una sola clave que se utiliza con los algoritmos de cifrado simétrico en el cifrado y el descifrado [win2003].

Código de Autenticación de Mensajes (MAC)
: Algoritmo que garantiza la calidad de un bloque de datos [win2003].

Confidencialidad
: La confidencialidad es un servicio de seguridad que protege los datos de la exposición (divulgación) no autorizada. El principal interés de la confidencialidad en muchos de los casos es la exposición no autorizada de los datos en el nivel de aplicación, pero la exposición de características externas de comunicación también son de interés en ciertas circunstancias. La confidencialidad del flujo de tráfico es el servicio que trata este último tema encubriendo las direcciones de origen y destino, la longitud del mensaje, frecuencia de comunicación. En el contexto IPsec, usando ESP en modo túnel, especialmente en una security gateway, puede proporcionar algunos niveles de confidencialidad del flujo de tráfico. (ver también análisis de tráfico abajo) [ARCH].

Conjunto de Protección
: Un conjunto de protección es una lista de servicios de seguridad que debe ser aplicada por varios protocolos de seguridad. Por ejemplo, un conjunto de protección puede consistir de encriptación DES en ESP IP, y una clave MD5 en AH IP. Todas las protecciones en un conjunto deben ser tratadas como una unidad simple. Esto es necesario porque los servicios de seguridad en diferentes protocolos de seguridad pueden tener sutiles interaccionés, y los efectos de un conjunto deben ser analizados y verificados como un todo [ISAKMP].

Control de Acceso
: Es un servicio de seguridad que impide el uso no autorizado de un recurso, incluyendo la prevención de el uso de recursos en forma no autorizados. En el contexto IPsec, el recurso cuyo acceso esta siendo controlado es con frecuencia:

  - Para un host, ciclos o datos computacionales
  - Para un security gateway, una red que esta detrás de un gateway o ancho de banda de esa red [ARCH].

Criptografía Fuerte
: Este tipo de criptografía es muy segura y es prácticamente imposible desencriptar mensajes encriptados con este tipo de criptografía [win2003].

Dirección
: Un identificador de capa IPv6 o IPv4 para una interface o un conjunto de interfaces.

Distancia de Hamming
: El número de las posiciones de dígito en las cuales los dígitos correspondientes de dos palabras binarias de la misma longitud son diferentes. Nota 1: La distancia de Hamming entre 1011101 y 1001001 es dos. Nota 2: El concepto se puede ampliar a otros sistemas de la notación. Por ejemplo, la distancia de Hamming entre 2143896 y 2233796 es tres. (según: ITS-Institute for Telecommunication Sciences)

Dominio de Interpretación (DOI)
: Un DOI define los formatos de cargas, tipos de intercambio, y convenciones para nombrar información relevante a la seguridad tales como políticas de seguridad, algoritmos criptográficos y modos. En ISAKMP un identificador de DOI es usado para interpretar las cargas de ISAKMP. Un sistema DEBERÍA soportar múltiples DOI simultáneamente. El concepto del DOI se extiende más allá de la interpretación de etiquetas de seguridad para incluir el nombramiento y la interpretación de los servicios de seguridad. Un DOI define:
  - Una "situación": el conjunto de información que será usado para determinar los servicios de seguridad requeridos.
  - El conjunto de políticas de seguridad que deben o podrían ser soportados.
  - La sintaxis para la especificación de los propósitos de los servicios de seguridad sugeridos.
  - Un esquema para nombrar información relativa a la seguridad, incluyendo algoritmos de encriptación, algoritmos de intercambio de claves, atributos de política de seguridad y autoridades de certificación.
  - Los formatos específicos de los contenidos de las diversas cargas.
  - Tipos de intercambio adicionales, si son requeridos.

Encriptación
: La encriptación es un mecanismo de seguridad usado para trasformar datos desde una forma inteligible (texto plano) en una forma ininteligible (texto cifrado), para proporcionar confidencialidad. El proceso de transformación inverso se denomina "desencriptación". Algunas veces el término "encriptación" es usado para referirse genéricamente a ambos procesos [ARCH].

Enlace
: una facilidad de comunicación o medio sobre el cual los nodos pueden comunicarse en la capa de enlace, es decir, la capa inmediatamente debajo del IPv6 o IPv4. Ejemplos son las Ethernets (simples o de puentes); enlaces PPP; X.25, Frame Relay, o redes ATM; y "túneles" de capa internet (o superior), tal como los túneles sobre IPv4 o sobre el mismo IPv6.

Estándar de cifrado de datos (DES)
: Un algortimo de cifrado que utiliza una clave de 56 bits y asigna un bloque de entrada de 64 bits a un bloque de salida de 64 bits. Aunque la clave parece tener 64 bits, un bit de cada ocho bytes se utiliza para la paridad impar, lo que da un resultado de 56 bits útiles [win2003].

Firma Digital
: Es una secuencia de caracteres calculados a partir del documento original mediante funciones de resumen (digest) o Hash que acompaña a un documento (o fichero), acreditando quién es su autor ("autenticación") y que no ha existido ninguna manipulación posterior de los datos ("integridad"). Para firmar un documento digital, su autor utiliza su propia clave secreta, cualquier persona puede verificar la validez de una firma si dispone de la clave pública del autor.

Granularidad
: Grado de modularidad de un sistema, cuanto mayor sea la granularidad, más personalizable o flexible será el sistema.

Hash
: Función que genera un resumen criptográfico a partir de una terminada cadena de información. Un número generado a partir de una cadena de caracteres que es usado para garantizar que el mensaje trasmitido llegue intacto (Definición extraída del Diccionario de IBM Corp.).

Host
: cualquier nodo que no es un enrutador.

Infraestructura de Claves Públicas (PKI)
: Las leyes, directivas, estándares y software que regulan o controlan los certificados y las claves públicas y privadas. En la práctica, se trata de un sistema de certificados digitales, entidades emisoras de certificados y demás entidades de registro que comprueban y autentican la validez de cada parte implicada en una transacción electrónica [win2003].

Integridad
: La integridad es un servicio de seguridad que asegura que la modificación de los datos sea perceptible. La integridad tiene diversas formas para corresponderse con los requerimientos de las aplicaciones. IPsec soporta dos formas de integridad: sin conexión y una forma de integridad de la secuencia parcial. La integridad sin conexión es un servicio que detecta la modificación de un datagrama IP individual, sin considerar el orden de los datagramas cuando estos llegan. La forma de integridad de la secuencia parcial ofrecida en IPsec es referida como integridad anti-replay, y detecta la llagada de datagramas IP duplicados (dentro de una ventana acotada). Esto esta en oposición de la integridad orientada a la conexión, que impone requerimientos más estrictos en el tráfico, por ejemplo, para poder detectar mensajes perdidos o reordenados. Aunque los servicios de autentificación e integridad son frecuentemente citados por separado, en la práctica están relacionados íntimamente y casi siempre ofrecidos en conjunto [ARCH].

Interface
: Lo que acopla un nodo a un enlace.

IV (Vector de Inicialización)
: Para evitar que dos mensajes idénticos se codificarán de la misma forma usando el modo CBC. Más aún, dos mensajes que empiecen igual se codificarán igual hasta llegar a la primera diferencia entre ellos. Para evitar esto se emplea un Vector de Inicialización, que puede ser un bloque aleatorio, como bloque inicial de la transmisión. Este vector será descartado en destino, pero garantiza que siempre los mensajes se codifiquen de manera diferente, aunque tengan partes comunes.

MD2
: Algoritmo de hash, desarrollado por RSA Data Security, Inc., que crea un valor de hash de 128 bits [win2003].

MD4
: Algoritmo de hash, desarrollado por RSA Data Security, Inc., que crea un valor de hash de 128 bits [win2003].

MD5
: Esquema de hash normalizado unívoco de 128 bits desarrollado por RSA Data Security, Inc. y utilizado por varios proveedores de Protocolo punto a punto (PPP) para autenticación cifrada. Un esquema de hash es un método de transformación de datos (por ejemplo, una contraseña) en el que el resultado es único y no se puede devolver a su forma original. El Protocolo de autenticación por desafío mutuo (CHAP) utiliza un mecanismo de desafío y respuesta con hash MD5 unívoco en la respuesta. De esta forma, puede probar al servidor que conoce la contraseña sin enviarla realmente a través de la red. [win2003]

Modo de autenticación de mensajes basado en hash (HMAC)
: Mecanismo de autenticación de mensajes mediante funciones de hash cifradas. Se puede utilizar HMAC con cualquier función criptográfica e iterativa de hash, como MD5 o SHA-1, junto con una clave secreta compartida. La solidez criptográfica de HMAC depende de las propiedades de la función de hash subyacente [win2003].

Multicast
: Paquetes únicos copiados por la red y enviados a un subconjunto específico de direcciones de red. Estas direcciones se especifican en el campo de direcciones de destino. Comparar con broadcast y unicast.

Multihomed
: Que tiene muchas direcciones en el Internet (relacionado a varios puntos de interfase)

Nodo
: Un dispositivo que implementa el IP.

Nombre canónico
: Nombre completo de un objeto presentado primero con la raíz y sin las etiquetas de atributos del Protocolo compacto de acceso a directorios (LDAP) (como: CN=, DC=). Los segmentos del nombre están delimitados por una barra diagonal (/). Por ejemplo:
CN=MisDocumentos,OU=MiUO,DC=Microsoft,DC=Com
se presenta como
microsoft.com/MiUO/MisDocumentos
en forma canónica [win2003].

Nombre Completo
: Nombre que identifica de forma única un objeto mediante su nombre completo relativo y los nombres de los objetos y dominios contenedores que lo contienen. El nombre completo identifica el objeto y su ubicación en un árbol. Cada uno de los objetos de Active Directory posee un nombre completo. Un nombre completo típico podría ser
CN=MiNombre,CN=Usuarios,DC=Microsoft,DC=Com
Esto identifica el objeto de usuario MiNombre en el dominio microsoft.com [win2003]

Nonce
: Aleatoriamente, cadena de texto única que es encriptada junto con datos y que luego es usada para detectar ataques contra el sistema que envía el dato encriptado. Un nonce es usado específicamente para la autentificación y para asegurar que el dato encriptado es diferente cada vez que es encriptado. (Definición extraída del Diccionario de IBM Corp.)

Paquete
: Una cabecera IPv6 o IPv4 más carga útil.

Perfect Forward Secrecy (PFS)
: En criptografía, en un protocolo de establecimiento de clave, la noción que compromete a una única clave que permitirá el acceso solamente a los datos protegidos derivados de esa única clave. Para que el PFS exista la clave usada para proteger la transmisión de datos NO DEBE ser usada para derivar claves adicionales, y si la clave usada para proteger la transmisión de los datos fue derivada de otro material clave, ese material NO DEBE ser usado para derivar más claves [IKE]. Este término se lo pude encontrar traducido como: máxima confidencia en el reenvío (según la traducción de U.S. Robotics Corporation); confidencialidad directa perfecta (según la traducción de Microsoft), confidencialidad anticipada perfecta o Confidencialidad respaldada correctamente.

Proposición
: Una proposición es una lista, en orden decreciente de preferencia, del conjunto de protecciones que un sistema considera aceptable para proteger el tráfico bajo una situación dada [ISAKMP].

Protocolo de Negociación de Claves de Diffie-Hellman
: Mecanismo criptográfico que permite a dos partes establecer una clave secreta compartida sin necesidad de ningún secreto preestablecido entre ellos. Diffie-Hellman suele utilizarse para establecer las claves secretas compartidas que utilizan las aplicaciones comunes de criptografía, como IPSec. No suele utilizarse para la protección de datos [win2003].

Protocolo de Seguridad
: Un Protocolo de Seguridad consiste en una entidad de un solo extremo en la pila de red, realizando un servicio de seguridad para las comunicaciones de red. Por ejemplo, ESP IPsec, AH IPsec, son dos diferentes protocolos de seguridad. TLS es otro ejemplo. Los protocolos de seguridad pueden proporcionar más de un servicio, por ejemplo proporcionar integrabilidad y confidencialidad en un solo módulo [ISAKMP].

Retry Counte
: Contador decreciente, por ejemplo se inicializa en 30 y con cada intento se va decrementando.

Router
: Un nodo que reenvía paquetes IP no explícitamente destinados hacia sí mismo.

SA ISAKMP
: Una SA usada por los servidores ISAKMP para proteger su propio tráfico. Las Secciones 2.3 y 2.4 del Capítulo 7 proporcionan más detalles acerca de las SAs ISAKMP [ISAKMP].

SAD (La Base de Datos de Asociaciones de seguridad)
: Contiene los parámetros que se asocian con cada SA (activa).

Security Gateway
: Un security gateway es un sistema intermedio que actúa como interfaz de comunicaciones entre dos redes. El conjunto de host (y redes) en el lado externo de la security gateway es visto como no confiable (o menos confiable), mientras que las redes y host en el lado interno son vistas como confiables (o más confiables). Las subredes internas y host que están proporcionados por una security gateway son presuntos de ser confiables en virtud de que comparten una administración de seguridad común (ver "Subredes Confiables" debajo). En el contexto IPsec una security gateway es un punto en el cual AH y/o ESP es implementado para proporcionar un conjunto de host internos, proporcionando servicios de seguridad para estos host cuando se comunican con host externos que también implementan IPsec (directamente o a través de otra security gateway)[ARCH].

Seed
: Un valor que añade aleatoriedad a la creación de números seudo-aleatorios (Definición extraída del Diccionario de IBM Corp.).

Selector
: Es un conjunto de campos con valores de protocolos de capas superiores y de la capa IP que son usados por la SPD para asignar el tráfico a una política, es decir, a una SA (o grupo de SA).

simplex
: Capacidad de transmisión en una sola dirección entre una estación emisora y una estación receptora. (Estudiantes Redes Cisco, Babylon)

Situación
: Una situación contiene toda la información relevante a la seguridad que un sistema considera necesaria para decidir los servicios de seguridad requeridos para proteger las sesiones que están siendo negociadas. La situación puede incluir direcciones, clasificaciones de seguridad, modos de operación, (normal vs. emergencia), etc [ISAKMP].

Sniffer
: Programa que monitorea y analiza el tráfico de una red para detectar problemas o cuellos de botella. Su objetivo es mantener la eficiencia del tráfico de datos. Pero también puede ser usado ilegítimamente para capturar datos en una red. (El diccionario informático de Clarin)

Socket
: Abstracción que permite a un programa de aplicación accesar a los puertos TCP/UDP.

SPD (La Base de Datos de Políticas de Seguridad)
: Especifica las políticas que determinan el tratamiento de todo el tráfico IP entrante o saliente en un host, security gateway, o en implementaciones IPsec BITS o BITW.

SPI (Índice de parámetros de seguridad)
: Un identificador para una Asociación de Seguridad, relativo a algún protocolo de seguridad. Cada protocolo de seguridad tiene su propio "espacio-SPI". Un par (protocolo de seguridad, SPI) pueden identificar unívocamente a una SA. La unívocariedad (exclusividad) de la SPI es dependiente de la implementación, pero puede estar basada en sistemas, en protocolos, u otras opciones. Dependiendo del DOI, información adicional (por ejemplo, las direcciones de los host) puede ser necesarias para identificar a una SA. El DOI también determinará cuales SPIs (es decir, los SPIs del iniciador o del respondedor) son enviados durante la comunicación [ISAKMP].

SPI de AH y de ESP
: La combinación de la dirección de destino, protocolo de seguridad y SPI identifican unívocamente a la SA. El SPI es trasportado por los protocolos AH y ESP, para permitir que el nodo receptor seleccione la SA bajo la cual un paquete recibido será procesado. Un SPI solo tiene significado localmente, como lo define el creador de la SA (usualmente el receptor del paquete que lleva el SPI); por lo tanto un SPI es generalmente visto como una secuencia de bits ocultos. Sin embargo el creador de una SA puede elegir interpretar los bits en un SPI para facilitar el procesamiento local [ARCH].

Subredes confiables
: Una subred que contiene host y routers que confían mutuamente no se ocupan de ataques pasivos o activos. También hay suposición de que el canal de comunicación subyacente (por ejemplo, una LAN o CAN) no esta siendo atacado por otros métodos [ARCH].

Tampering
: Violación de seguridad en la comunicación, en la cual la información en transito es cambiada o reemplazada y es enviada hacia el receptor. (Definición extraída del Diccionario de IBM Corp.)

Tipos de Intercambios
: Un tipo de intercambio es una especificación de un número de mensajes en un intercambio ISAKMP, y los tipos de carga que están contenidos en cada uno de estos mensajes. Cada tipo de intercambio está diseñado para proporcionar, un conjunto especifico de servicios de seguridad, tales como el anonimato de los participantes, perfect forward secrecy del material clave, autentificación para los participantes, etc. En la Sección 4.1 del Capítulo 7 se define el conjunto por defecto de tipos de intercambio ISAKMP. Otros tipos de intercambio se pueden agregar para soportar intercambios adicionales de claves, si es requerido [ISAKMP].

Triple DES (3DES)
: Implementación del Cifrado estándar de datos (DES) que emplea tres iteraciones de operaciones criptográficas en cada segmento de datos. Cada iteración utiliza una clave de 56 bits en el cifrado, lo que supone un cifrado de 168 bits para los datos. Si bien 3DES es más lento que DES debido a los cálculos criptográficos adicionales, su protección es mucho más segura que la de DES[win2003].

Tupla
: Lista ordenada de valores, los elementos tienen un orden definido.

unicast
: Mensaje que se envía a un solo destino de red. Comparar con broadcast y multicast.

Referencias

[AES-CCM] R. Housley, "Using AES CCM Mode With IPsec ESP", <draft-ietf-ipsec-ciph-aes-ccm-05.txt>, November 2003.

[AES-GCM] J. Viega, D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, June 2005.

[AH] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[ANSI] ANSI, X9.42: Public Key Cryptography for the Financial Services Industry -- Establishment of Symmetric Algorithm Keys Using Diffie-Hellman, Working Draft, April 19, 1996.

[ARCH] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[ATAH] Atkinson, R., "The IP Authentication Header", RFC 1826, August 1995.

[ATESP] Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 1827, August 1995.

[BC] Ballardie, A., and J. Crowcroft, Multicast-specific Security Threats and Countermeasures, Proceedings of 1995 ISOC Symposium on Networks & Distributed Systems Security, pp. 17-30, Internet Society, San Diego, CA, February 1995.

[Bell95] Bellovin, S., "An Issue With DES-CBC When Used Without Strong Integrity", Presentation at the 32nd Internet Engineering Task Force, Danvers Massachusetts, April 1995.

[Bell96] Bellovin, S., "Problem Areas for the IP Security Protocols", Proceedings of the Sixth Usenix Security Symposium, July 1996.

[Bell97] Bellovin, S., "Probable Plaintext Cryptanalysis of the IP Security Protocols", Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, pp. 155-160, February 1997 (also http://www.research.att.com/~smb/papers/probtxt.{pspdf}).

[Bellare96a] Bellare, M., Canetti, R., and H. Krawczyk, "Keying Hash Functions for Message Authentication", Advances in Cryptography, Crypto96 Proceeding, June 1996.

[Berge] Berge, N., "UNINETT PCA Policy Statements", RFC 1875, December 1995.

[BL73] Bell, D.E. & LaPadula, L.J., "Secure Computer Systems: Mathematical Foundations and Model", Technical Report M74-244, The MITRE Corporation, Bedford, MA, May 1973.

[Blaze] Blaze, Matt et al., MINIMAL KEY LENGTHS FOR SYMMETRIC CIPHERS TO PROVIDE ADEQUATE COMMERCIAL SECURITY. A REPORT BY AN AD HOC GROUP OF CRYPTOGRAPHERS AND COMPUTER SCIENTISTS... -- http://www.bsa.org/policy/encryption/cryptographers.html

[BLOW] Schneier, B., "The Blowfish Encryption Algorithm", Dr. Dobb's Journal, v. 19, n. 4, April 1994.

[Bra97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.

[BS93] Biham, E., and A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard", Berlin: Springer-Verlag, 1993.

[CAST] Adams, C., "The CAST-128 Encryption Algorithm", RFC 2144, May 1997.

[CBC] Periera, R., and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.

[CICESE] Protocolos de Seguridad e Instrumentación de IPSec en Escenarios Experimentales de Internet 2 en México. Trabajo de tesis, Centro de Investigación Científica y de Educación Superior de Ensenada, México Enero de 2002 María Concepción Mendoza Díaz

[CN94] Carroll, J.M., and S. Nudiati, "On Weak Keys and Weak Data: Foiling the Two Nemeses", Cryptologia, Vol. 18 No. 23 pp. 253-280, July 1994.

[Cripseg] Criptografía y Seguridad en Computadores Tercera Edición (Versión 2.01). Marzo de 2003. Manuel José Lucena López, Profesor de la Universidad de Jaén, mlucena@ujaen.es

[CW87] Clark, D.D. and D.R. Wilson, A Comparison of Commercial and Military Computer Security Policies, Proceedings of the IEEE Symposium on Security & Privacy, Oakland, CA, 1987, pp. 184-193.

[DEFLATE] Pereira, R., "IP Payload Compression Using DEFLATE", RFC 2394, August 1998.

[DES] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.

[DESAN] ANSI X3.106, "American National Standard for Information Systems-Data Link Encryption", American National Standards Institute, 1983.

[DH] Diffie, W., and Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, V. IT-22, n. 6, June 1977.

[IPv6] Deering, S., and B. Hinden, "Internet Protocol version 6 (IPv6) Specification", RFC 2460, December 1998.

[DNSSEC] D. Eastlake III, Domain Name System Protocol Security Extensions, Work in Progress.

[DoD85] US National Computer Security Center, "Department of Defense Trusted Computer System Evaluation Criteria", DoD 5200.28-STD, US Department of Defense, Ft. Meade, MD.,December 1985.

[DoD87] US National Computer Security Center, "Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria", NCSC-TG-005, Version 1, US Department of Defense, Ft. Meade, MD., 31 July 1987.

[DOIPsec] Piper, D., "The Internet IP Security Domain Of Interpretation for ISAKMP", RFC 2407, November 1998.

[DOW92] Diffie, W., M.Wiener, P. Van Oorschot, Authentication and Authenticated Key Exchanges, Designs, Codes, and Cryptography, 2, 107-125, Kluwer Academic Publishers, 1992.

[DSS] NIST, "Digital Signature Standard", FIPS 186, National Institute of Standards and Technology, U.S. Department of Commerce, May, 1994.

[ECMWF] European Centre for Medium-Range Weather Forecasts Sección de Red y Seguridad, División de Informática Estudio de viabilidad sobre IPsec, Mayo de 2003.

[ESP] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload", RFC 2406, November 1998.

[ESPNULL] Glenn, R., and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

[FIPS-180-1] NIST, FIPS PUB 180-1: Secure Hash Standard, April 1995. http://csrc.nist.gov/fips/fip180-1.txt (ascii) http://csrc.nist.gov/fips/fip180-1.ps (postscript)

[FIPS-46-2] US National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 46-2, December 1993, http://www.itl.nist.gov/div897/pubs/fip46-2.htm (supercedes FIPS-46-1).

[FIPS-74] US National Bureau of Standards, "Guidelines for Implementing and Using the Data Encryption Standard", Federal Information Processing Standard (FIPS) Publication 74, April 1981, http://www.itl.nist.gov/div897/pubs/fip74.htm.

[FIPS-81] US National Bureau of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, December 1980, http://www.itl.nist.gov/div897/pubs/fip81.htm.

[GDOI] Baugher, M., Hardjono, T., Harney, H., and B. Weis, "The Group Domain of Interpretation", RFC 3547, July 2003.

[HA94] Haller, N., and R. Atkinson, "On Internet Authentication", RFC 1704, October 1994.

[hispa] Criptografía. José Angel de Bustos Pérez, jadebustos@augcyl.org joseangel.bustos@hispalinux.es

[HM97] Harney, H., and C. Muckenhirn, "Group Key Management Protocol (GKMP) Architecture", RFC 2094, July 1997.

[HMAC] Krawczyk, K., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[HMACMD5] Madson, C., and R. Glenn, "The Use of HMAC-MD5 within ESP and AH", RFC 2403, November 1998.

[HMACSHA] Madson, C., and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[IAB] Bellovin, S., "Report of the IAB Security Architecture Workshop", RFC 2316, April 1998.

[IAMA-3] Internet Assigned Numbers Authority. Attribute Assigned Numbers. (http://www.iana.org/assignments/icmp-parameters), ICMP TYPE NUMBERS, last updated 27 January 2005.

[IAMA-4] Internet Assigned Numbers Authority. Attribute Assigned Numbers. (http://www.iana.org/assignments/ip-parameters), IP OPTION NUMBERS, last updated 2001-06-29.

[IANA-1] Internet Assigned Numbers Authority. Attribute Assigned Numbers. (http://www.iana.org/assignments/ipsec-registry), from RFC 2409 (IKE), last updated 10 June 2005.

[IANA-2] Internet Assigned Numbers Authority. Attribute Assigned Numbers. (http://www.iana.org/assignments/isakmp-registry), from RFC 2407 and RFC 2408, last updated 10 June 2005.

[IB93] John Ioannidis and Matt Blaze, "Architecture and Implementation of Network-layer Security Under Unix", Proceedings of USENIX Security Symposium, Santa Clara, CA, October 1993.

[IBK93] John Ioannidis, Matt Blaze, & Phil Karn, "swIPe: Network-Layer Security for IP", presentation at the Spring 1993 IETF Meeting, Columbus, Ohio.

[ICMPv4] Postel, J., "Internet Control Message Protocol Specification" STD 5, RFC 792, September 1981.

[ICMPv6] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

[IDEA] Lai, X., "On the Design and Security of Block Ciphers," ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992

[IKE] Harkins, D., and D. Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[IKEupdate] P. Hoffman, "Algorithms for Internet Key Exchange version 1 (IKEv1)", RFC 4109, May 2005.

[IKEv2] Charlie Kaufman Kaufman, "Internet Key Exchange (IKEv2) Protocol", <draft-ietf-ipsec-ikev2-17.txt>, September 23, 2004.

[inco] El Protocolo IPv6 y sus extensiones de seguridad IPSec Gabriel Verdejo Alvarez,Univercidad Autonoma de Barcelona, Febrero del 2000.

[IPCOMP] Shacham, A., Monsour, R., Pereira, R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 2393, August 1998.

[IPsecSCTP] S. Bellovin, J. Ioannidis, A. Keromytis, and R. Stewart, "On the Use of SCTP with IPsec", RFC 3554, July 2003.

[IPv4] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[ISAKMP] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[ISO] ISO/IEC JTC1/SC6, Network Layer Security Protocol, ISO-IEC DIS 11577, International Standards Organisation, Geneva, Switzerland, 29 November 1992.

[itu] Unión Internacional de Telecomunicaciones La seguridad de las telecomunicaciones-Diciembre de 2003 http://www.itu.int

[JaHu] James Hughes (jim_hughes@stortek.com) y Harry Varnis (hgv@anubis.network.com)

[Karn] Karn, P., and B. Simpson, Photuris: Session Key Management Protocol, Work in Progress.

[kato-ipsec] A. Kato, S. Moriai, M. Kanda, "The Camellia Cipher Algorithm and Its Use With IPsec", <draft-kato-ipsec-ciph-camellia-01.txt>, March 2005.

[Ken91] Kent, S., "US DoD Security Options for the Internet Protocol", RFC 1108, November 1991.

[Kent94] Steve Kent, IPSEC SMIB, e-mail to ipsec@ans.net, August 10, 1994.

[Knuth] Knuth, Donald E., The Art of Computer Programming, Vol. 2, Seminumerical Algorithms, Addison Wesley, 1969.

[Kocher] Kocher, Paul, Timing Attack, http://www.cryptography.com/timingattack.old/timingattack.html

[LZJH] Heath, J. and J. Border, "IP Payload Compression Using ITU-T V.44 Packet Method", RFC 3051, January 2001.

[LZS] Friend, R., and R. Monsour, "IP Payload Compression Using LZS", RFC 2395, August 1998.

[Matsui94] Matsui, M., "Linear Cryptanalysis method for DES Cipher", Advances in Cryptology -- Eurocrypt '93 Proceedings, Berlin: Springer-Verlag, 1994.

[MD5] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, April 1992.

[NATinIKE] T. Kivinen, A. Huttunen, B. Swander, and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[NumSequen] Stephen Kent, "Extended Sequence Number Addendum to IPsec DOI for ISAKMP", <draft-ietf-ipsec-esn-addendum-03.txt>, February 2004.

[OAKLEY] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[PKCS1] RSA Laboratories, "PKCS #1: RSA Encryption Standard", November 1993.

[PMTU] Mogul, J. and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.

[popu] Redes de computadores: Un Enfoque Descendente Basado en Internet, Segunda Edición. Jim Kurose, Keith Ross.

[Principi] Criptografía Para Principiantes (Versión 1.0). José de Jesús Angel Angel. jesus@seguridata.com

[RANDOM] Eastlake, D., Crocker, S., and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[RC5] Rivest, R., "The RC5 Encryption Algorithm", Dr. Dobb's Journal, v. 20, n. 1, January 1995.

[RFC-1422] Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management", RFC 1422, February 1993.

[RFC-1810] Touch, J., "Report on MD5 Performance", RFC 1810, June 1995.

[RFC-1949] Ballardie, A., "Scalable Multicast Key Distribution", RFC 1949, May 1996.

[RFC-2026] Bradner, S., "The Internet Standards Process – Revision 3", BCP 9, RFC 2026, October 1996.

[RFC-2040] Baldwin, R., and R. Rivest, "The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms", RFC 2040, October 1996.

[RFC-2093] Harney, H., and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification", RFC 2093, July 1997.

[RFC-2202] Cheng, P., and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1", RFC 2202, March 1997.

[RFC1063] J.C. Mogul, C.A. Kent, C. Partridge, K. McCloghrie, "IP MTU discovery options", RFC 1063, Jul 1988.

[RFC1122] R. Braden, Ed., "Requirements for Internet Hosts - Communication Layers", RFC 1122 October 1989.

[RFC1256] S. Deering, Ed., "ICMP Router Discovery Messages", RFC 1256, September 1991.

[RFC1393] G. Malkin, "Traceroute Using an IP Option", RFC 1393, January 1993.

[RFC1475] R. Ullmann, "TP/IX: The Next Internet", RFC 1475, June 1993.

[RFC1770] C. Graff, "IPv4 Option for Sender Directed Multi-Destination Delivery", RFC 1770, March 1995.

[RFC1812] F. Baker, Ed., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.

[RFC3566] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.

[RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.

[RFC3664] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 3664, January 2004.

[RFC950] J.C. Mogul, J. Postel, "Internet Standard Subnetting Procedure", RFC950, August 1985.

[RIPEMD] Keromytis, A. and N. Provos, "The Use of HMAC-RIPEMD-160-96 within ESP and AH", RFC 2857, June 2000.

[ROAD] Thayer, R., Doraswamy, N., and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.

[RoutAlert] D. Katz, "IP Router Alert Option", RFC 2113, February 1997.

[RSA] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.

[Sch94] Bruce Schneier, Applied Cryptography, Section 8.6, John Wiley & Sons, New York, NY, 1994.

[Schneier] Schneier, Bruce, Applied cryptography: protocols, algorithms, and source code in C, Second edition, John Wiley & Sons, Inc. 1995, ISBN 0-471-12845-7, hardcover. ISBN 0-471-11709-9, softcover.

[Schroeppel] Schroeppel, Richard, et al.; Fast Key Exchange with Elliptic Curve Systems, Crypto '95, Santa Barbara, 1995. Available on-line as ftp://ftp.cs.arizona.edu/reports/1995/TR95-03.ps (and . Z).

[SCTP] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[SDNS] SDNS Secure Data Network System, Security Protocol 3, SP3, Document SDN.301, Revision 1.5, 15 May 1989, published in NIST Publication NIST-IR-90-4250, February 1990.

[SECDNS] Eastlake, D. and C. Kaufman, "Domain Name System Security Extensions", RFC 2065, January 1997.

[SEED] Hyangjin Lee, Jaeho Yoon, Seoklae Lee, Jaeil Lee, "The SEED Cipher Algorithm and Its Use With IPSec", <draft-lee-ipsec-cipher-seed-01.txt>, February 2005.

[SHA] NIST, "Secure Hash Standard", FIPS 180-1, National Institue of Standards and Technology, U.S. Department of Commerce, May 1994.

[SKEME] Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.

[STD-2] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

[Stinson] Stinson, Douglas, Cryptography Theory and Practice. CRC Press, Inc., 2000, Corporate Blvd., Boca Raton, FL, 33431-9868, ISBN 0-8493-8521-0, 1995

[STS] W. Diffie, P.C. Van Oorschot, and M.J. Wiener, "Authentication and Authenticated Key Exchanges," in Designs, Codes and Cryptography, Kluwer Academic Publishers, 1992, pp. 107

[TIGER] Anderson, R., and Biham, E., "Fast Software Encryption", Springer LNCS v. 1039, 1996.

[UdateIPsec] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[VK83] V.L. Voydock & S.T. Kent, "Security Mechanisms in High-level Networks", ACM Computing Surveys, Vol. 15, No. 2, June 1983.

[Wiener94] Wiener, M.J., "Efficient DES Key Search", School of Computer Science, Carleton University, Ottawa, Canada, TR-244, May 1994. Presented at the Rump Session of Crypto '93. [Reprinted in "Practical Cryptography for Data Internetworks", W.Stallings, editor, IEEE Computer Society Press, pp.31-79 (1996). Currently available at ftp://ripem.msu.edu/pub/crypt/docs/des-key-search.ps.]

[win2003] Ayuda de Microsoft Windows 2003

[www1] Luciano Moreno. Departamento de diseño web de BJS Software. http://www.htmlweb.net/seguridad/cripto/cripto_1.html

[www2] José de Jesús Angel Angel. Director de Investigación y Desarrollo de SeguriDATA http://www.htmlweb.net/seguridad/cripto_p/cripto_princ_1.html

[www3] Grupos Diffie-Hellman. http://www.microsoft.com/windows2000/es/advanced/help/default.asp?url=/windows2000/es/advanced/help/sag_IPSECkeyexchgsm.htm

[www4] Infraestructura de claves públicas de MS Windows 2000 http://www.eu.microsoft.com/latam/technet/articulos/windows2k/2000pk/default.asp

[www5] Seguridad distribuida http://www.microsoft.com/latam/technet/articulos/199911/art02/

[X.501] ISO/IEC 9594-2, "Information Technology - Open Systems Interconnection - The Directory: Models", CCITT/ITU Recommendation X.501, 1993.

[X.509] ISO/IEC 9594-8, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT/ITU Recommendation X.509, 1993.

[Zimmerman] Philip Zimmermann, The Official Pgp User's Guide, Published by MIT Press Trade, Publication date: June 1995, ISBN: 0262740176.