Grupo de Trabajo en Red
Request for Comments: 2412
Categoría: Informativo

H. Orman
Department of Computer Science
University of Arizona
Noviembre 1998

Traducción al castellano:
Hugo Adrian Francisconi

Agosto 2005
<adrianfrancisconi@yahoo.com.ar>

El Protocolo de Determinación de Claves OAKLEY

Estado de este documento

Este documento proporciona información para la comunidad de Internet. No especifica un estándar de Internet de ninguna clase. La distribución de este documento es ilimitada.

Aviso de Copyright

Resumen

Este documento describe 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.

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.

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 por ejemplo en [RFC2401]. Un mecanismo de distribución de clave escalable y seguro para Internet es una necesidad. El objetivo de este protocolo es proporcionar ese mecanismo, además de gran cantidad de fuerza criptográfica.

El algoritmo de intercambio de clave Diffie-Hellman 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] proporciona una demostración 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.

Debido a que OAKLEY es un protocolo de intercambio de claves genérico, y debido a que las claves que este genera pueden ser utilizadas para encriptar datos con un largo tiempo de vida de privacidad, 20 años o más, es importante que los algoritmos subyacentes del protocolo puedan asegurar la seguridad de las claves para ese período de tiempo, basados en las mejores capacidades de predicción disponibles considerando el futuro matemático. El protocolo, por lo tanto, tiene dos opciones para agregar dificultades a un atacante que tenga gran cantidad de tráfico de intercambio de claves registrado a su disposición (un atacante pasivo). Estas opciones son útiles para obtener las claves que serán utilizadas para la encriptación.

El protocolo OAKLEY se relaciona con STS, compartiendo la similitud de la autentificación de la exponencial de Diffie-Hellman y usándolos para determinar una clave compartida, y también conseguir Perfect Forward Secrecy para las claves compartidas, pero se diferencia del protocolo STS de diferentes formas:

Primero con la adición de un mecanismo de validación de direcciones débiles ("cookies", descriptas por Phil Karn en el trabajo en curso del protocolo de intercambio de claves de Photuris) para ayudar a evitar ataques de denegación de servicio.

Segundo permitiendo 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.

Tercero, 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 algoritmo de sin encriptación. Esta flexibilidad es incluida para permitir que las partes usen las características más adecuadas a sus requisitos de seguridad y desempeño.

Este documento bosqueja ampliamente la metodología de trabajo de Photuris y del trabajo que está realizando Karn y Simpson (y de debates con los autores), específicos del documento de ISAKMP de Schertler y de otros. Como así también del documento del protocolo ISAKMP, y también fue influenciado por los escritos de Paul van Oorschot y de Hugo Krawcyzk.

2. Esquema del Protocolo

2.1 Observaciones Generales

El protocolo OAKLEY se utiliza para establecer una clave compartida con un identificador asignado y asociar las identidades autentificadas por las dos partes. El nombre de la clave se puede utilizar más adelante para obtener las asociaciones de seguridad para los protocolos AH (RFC 2402) y ESP (RFC 2406) o para conseguir otras metas de seguridad en la red.

Cada clave está asociada con los algoritmos utilizados para la autentificación, privacidad, y con las funciones unidireccionales [one-way function]. Éstos son algoritmos auxiliares para OAKLEY; su aparición en definiciones subsiguientes de SA obtenidas con otros protocolos no es requerida ni se prohíbe.

La especificación de los detalles de cómo aplicar un algoritmo a los datos se llama transformación. Este documento no provee las definiciones de las transformaciones; estas estarán en RFC separados.

Los tokens anti-saturación, o "cookies", proporcionan una forma débil 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 un puerto bien conocido (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.

La máquina que ejecuta OAKLEY debe proporcionar un buen generador de números aleatorios, 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 documento 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. Lo que sigue es una guía abreviada de la notación. Los formatos detallados y los valores asignados están en los Apéndices.

A fin de representar intercambios de mensajes, este documento utiliza una 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 (las constantes se definen en los apéndices).

Los números enteros de precisión variable con longitud cero son valores no validos (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, 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 el Apéndice B). NIDP significa no encriptados.

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 el Apéndice A) 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). Vea el Apéndice F 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 el Apéndice E), 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 (Apéndice A); 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 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 valor que caracteriza a la función 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.

El término "transformación" se utiliza para referirse a las funciones definidas en los RFC auxiliares. Las transformaciones de los RFC se

basarán en las definiciones del AH IPsec y ESP IPsec (véase el RFC 2401 para la arquitectura global incluida en esos protocolos)

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, un 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 el Apéndice B.

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 que, 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, proporcionando 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 no están requeridas, y si el secreto de identidad y la perfect forward secrecy para el material clave no se requieren, y si la firma (para no-repudio) es requerida, entonces el intercambio se puede completar 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 asociación de seguridad (SA) a usarse con los protocolos AH y ESP. Estos tipos de carga SA están especificados en el documento ISAKMP; los tipos de carga pueden ser protegidos mediante el material clave y algoritmos de OAKLEY, pero este documento no discute su uso.

2.3.1 Los 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 cada mensaje; si un campo no es relevante este puede tener un valor nulo (o no válido, no debe tenerse en cuenta ese valor, por simplicidad de ahora en adelante me referiré a el como valor "null") o no estar presente (no carga).

CKY-I	Cookie del originador.
CKY-R	Cookie del respondedor
MSGTYPE	Para el intercambio de claves, 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 (or g^y)	Representa un número entero de longitud variable o una potencia de un grupo generador
EHAO or 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. Phil Karn ha recomendado confeccionar 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 esta permitido. Las transformaciones OBLIGATORIAS y OPCIONALES incluirán cualquier que satisfaga estos criterios y estén definidas para usarse con ESP (RFC 2406).

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. Las transformaciones OBLIGATORIAS y OPCIONALES incluirán cualquiera que se defina para usarse con AH (RFC 2402).

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.

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 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

Todos los campos de los mensajes OAKLEY correspondientes a las cargas de los mensajes ISAKMP o a los componentes de las cargas. Los campos de cargas relevantes son la carga SA, la carga AUTH (autentificación), la carga del Certificado, y la carga de Intercambio de Claves. El marco del protocolo ISAKMP se encuentra en proceso de elaboración actualmente, y la asociación exacta de los campos de los mensajes Oakley para las cargas ISAKMP están también en proceso de elaboración (se lo conoce como el documento de Resolución).

Algunos de los campos de la cabecera y de la carga de ISAKMP tendrán valores constantes cuando se utilicen con OAKLEY. Los valores exactos que se utilizarán serán publicados en el documento Dominio de Interpretación (DOI) que acompañará al documento de la Resolución.

A continuación se indica donde aparecerá cada campo OAKLEY dentro de la estructura de los mensajes ISAKMP. Esto solo es una recomendación, el documento de la Resolución será la autoridad final sobre esta asociación.

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 (or g^y)	Carga de Intercambio de Claves, codificado como un número entero de precisión variable
EHAO and 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 de AUTH, Campo Identidad
ID(R)	Carga de AUTH, Campo Identidad
Ni	Carga de AUTH, Campo Nonce
Nr	Carga de AUTH, Campo Nonce
S{...}Kx	Carga de AUTH, Campo de Datos
prf{K,...}	Carga de 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 de OAKLEY depende de qué opciones desean el Iniciador y el Respondedor utilizar. 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 cookie (opcionalmente, sin estado)
2. Intercambio de la otra parte de la clave [half-key] de Diffie -Hellman (opcional, pero esencial para el perfect forward secrecy)
3. autentificación (opciones: 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, 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 por medio de ejemplos en esta sección.

El Iniciador es responsable de retransmitir los mensajes si el protocolo no termina a su debido tiempo. El Respondedor debe por lo tanto 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, 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 a 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; hay concatenaciones en los mensajes del protocolo actual usando su forma codificada como se especifica en el documento de la Resolución de SAKMP/Oakley.

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 de la siguiente forma:

Iniciación

El Iniciador genera un cookie único relacionado con la dirección IP esperada por el respondedor, y selecciona la información de estado: GRP (el identificador de grupo), un exponente x seleccionado seudo-aleatoriamente, g^x, una lista EHAO, nonce, identidades. La primera opción de autentificación en la lista EHAO es un algoritmo que soporta firmas digitales, y este es usado para firmar las identidades y la identidad del nonce y del grupo. 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 (para las alternativas, vea la Sección 6) y se acepta lo siguiente:

grupo con identificación GRP,
primera opción de autentificación (la cual debe ser una firma digital método usada para firmar los mensajes del Iniciador),
falta de perfect forward secrecy para el procesamiento de las identidades,
identidad ID(I) y identidad 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

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^y)^x (= 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 [half-keys] 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" aparecerá en el RFC que define la 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, pero no es 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, una identidad para el respondedor se indica en texto plano: ID(R'). Sin embargo, la ocultación de identidad con 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 por la encriptación con 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 Apéndice B.

El resultado de este 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 de la siguiente forma:

Iniciación

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, identidad ID(I) y identidad ID(R).

El respondedor debe desencriptar la identificación y la información del nonce, usando la clave privada para la identificación del R (respondedor). Después de esto, la clave privada para la identificación de R será utilizada para desencriptar el campo 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 encriptación-hash-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^y)^x (= 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 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 [half-keys] 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.

La información dentro de mensajes violados es que las cookies no pueden ser usadas por ninguna operación de OAKLEY.

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. Vea el Apéndice B.

  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, como en Photuris, 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 esa ka, 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 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

La autentificación es primordial en cualquier esquema de intercambio de claves. La comunidad de Internet debe decidir un estándar escalable para solucionar este problema, y OAKLEY debe hacer uso de ese estándar. Al momento de la producción de este documento, no hay tal estándar, aunque están emergiendo varios. Este documento 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 dos partes han convenido un método confiable de distribución de claves secretas para su autentificación mutua, este puede ser utilizado para la autentificación. Esto tiene problemas obvios en sistemas grandes, 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 misma.

   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 sobre diferentes invocaciones de OAKLEY (después de un fallo de sistema, etc.), este debe ocupar un área reservada del espacio del KEYID en las dos partes. Algunos bits pueden ser reservados en el "espacio de la cookie" de cada parte para adecuar esto.

   No hay autoridad de certificación para las claves pre-compartidas. Cuando una clave pre-compartida se utiliza para generar una carga de autentificación, 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 un requisito para implementaciones de DNS seguros; extensiones para autorizar claves DSS opcionales es una posibilidad a mediano plazo.

El registro de CLAVE DNS tiene asociado los registros SIG que son firmados por una autoridad de la zona, una jerárquica de firmas hasta el servidor raíz que establece las bases para la confianza. 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 serán descriptos en un RFC separado. 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.1 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 Protocolo de Intercambio de Claves Pública discute la codificación y validación de X.509. El soporte para esto es OPCIONAL.

d.2 claves DSS con certificados: La codificación para los Estándares de Firmas Digitales con X.509 se describe en el draft 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 una 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 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 Credenciales de ISAKMP puede ser utilizada para adjuntar certificados útiles en los mensajes de OAKLEY. La Carga de Credenciales se define en el Apéndice B.

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.

El Protocolo de Intercambio de Claves Pública discute un protocolo completo que se puede utilizar con la codificación de certificados X.509.

2.6 Interfase para las Transformaciones Criptográficas

El material clave calculado para el intercambio de claves debería tener por lo menos 90 bits de entropía, que significa que debe tener por lo menos una longitud de 90 bits. Esto puede ser aproximado cuando se necesita para la clave la encriptación y/o transformaciones de funciones seudo-aleatorias.

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, 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, 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.

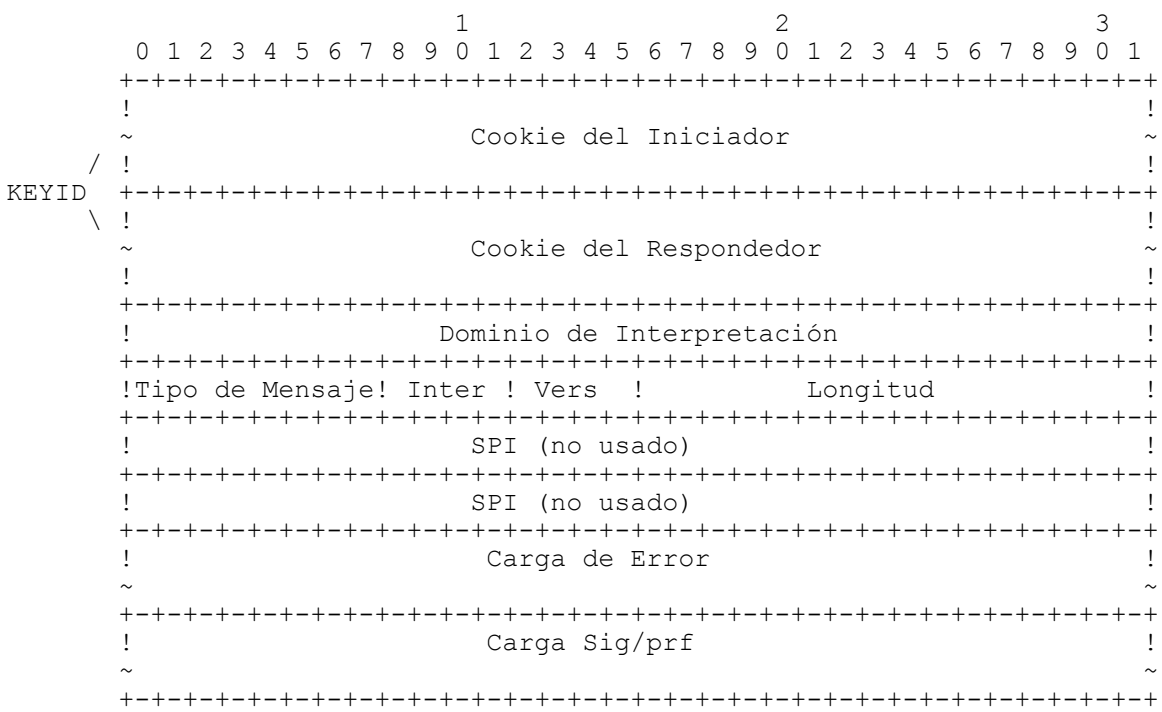
2.7 Retransmisión, Tiempo Agotado, y Mensajes de Error

Si una respuesta del Respondedor no es obtenida en un determinado período de tiempo, el mensaje debería ser retransmitido por el Iniciador. 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. Un método posible para establecer tal clave se describe en el Apéndice B, bajo el uso de tipos de mensajes ISA_INIT.

El siguiente es un tipo de mensaje de error de OAKLEY, el KEYID proporciona el algoritmo H y la clave para autentificar el contenido del mensaje; este valor se transporta en la carga de Sig/Prf.

La carga de Error contiene el código de error y el contenido del mensaje rechazado.



Inter = Intercambio

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.

Los mensajes de error son solamente informativos, y la integridad del protocolo no depende de ellos.

Causas de error:

TIMEOUT	El intercambio ha tomado demasiado tiempo, destrucción del estado
AEH_ERROR	Un algoritmo desconocido aparece en una 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 fallida
RESOURCE_EXCEEDED	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 mutuamente, y siempre le sigue la definición de un nuevo grupo y esta 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.

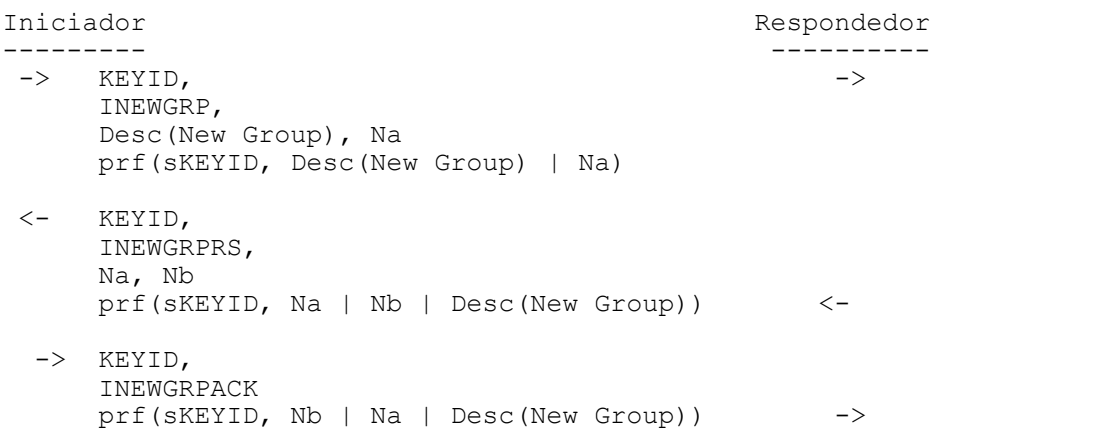
Solamente la descripción del nuevo grupo necesita ser encriptada en este intercambio. El algoritmo hash esta implícito debido a la sesión de OAKLEY designada por el grupo. La encriptación es la función de encriptación de la sesión de OAKLEY.

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:



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 el Apéndice B).

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 el Apéndice A 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. El KEYID podría ser uno que fue obtenido del Modo Principal, por ejemplo.

Por otra parte, la clave autentificada puede ser una clave manualmente distribuida, una que sea 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 se llevan en la Carga de Autentificación, y el valor prf se lleva en la Carga de Autentificación; 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 una clave y un identificador de clase asociado y la información de estado se han distribuido manualmente, entonces la clave puede ser usada para cualquier propósito de 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 KEYID representa, una clave de sesión de OAKLEY autentificada existente, y el sNEWKEYID representa el material clave generado.

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.

```
 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 cambiar cualquier información de estado de la clave.

La nueva clave es recuperada por el Respondedor calculando el XOR del campo en la Carga de Autentificación con el valor de 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. Hugo Krawcyzk sugirió el método y advirtió sus ventajas.

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 protocolo; las partes que usen la clave deben tener otro método para determinar estas características.

A principios del año 1996, se observó que para 90 bits de fuerza criptográfica, uno debe utilizar módulos de un grupo exponencial modular de 2000 bits. Para 128 bits de fuerza, se requiere módulos de 3000 bits.

3. Especificación y Obtención de Asociaciones de Seguridad

Cuando una asociación de seguridad (SA) es definida, sólo el KEYID necesita ser dado. El respondedor debería ser capaz de buscar el estado asociado al valor del KEYID y encontrar el material clave apropiado, sKEYID.

Obtener las claves para usarse con los protocolos IPsec por ejemplo ESP o AH es un tema que se trata en el documento Resolución de ISAKMP/Oakley. Ese documento 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. Como el material clave básico definido aquí (g^xy) puede ser utilizado para obtener claves para varias instancias de ESP y AH, los mecanismos exactos 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 el texto y en el Apéndice B. Hay notables ampliaciones más allá del draft de la versión 4.

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. Este valor será asignado en el RFC de ISAKMP.

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 de 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, en el Apéndice B, 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. Notas de Implementaciones de Seguridad

Los ataques de tiempo que tienen la capacidad de recuperar el valor del exponente usado en el cálculo de Diffie-Hellman han sido descriptos por Paul Kocher [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 (estimación realizada por Richard Schroeppel).

6. Análisis Modular y Máquina de Estado de OAKLEY

Hay muchos caminos con OAKLEY, pero siguen un orden de izquierda a derecha el 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 esta 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 determina 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 un 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 él 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.

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 a cero.

5. Si el PFS para ocultar la identidad es requerido, entonces ningunos otros datos seguirán.

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 que 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 el Apéndice C. 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 documento de ISAKMP. Se debería notar que la opción de protección de identidad es aplicada a los certificados como así también a las identidades.

Consideraciones de Seguridad

El tema de este documento es seguridad; por lo tanto las consideraciones de seguridad invaden este documento.

Dirección del Autor

Hilarie K. Orman Departamento de Ciencias de la Computación de la Universidad de Arizona

EMail: ho@darpa.mil

APÉNDICE A 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 como MODP), el grupo de curvas elípticas superiores al campo GF[2^N] (designado como EC2N) y el grupo de curvas elípticas superiores a GF[P] (designado como 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 este apéndice (note, que esté es distinto que el codificado en el Apéndice C). 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 [right-justified]). Ésta es la codificación natural.

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 puede ser un número más grande. 2 <= G <= P-2.

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. A es frecuentemente 0. B no debe ser 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.

Formatos de los descriptores de grupo:

Tipo de grupo: Un campo de dos bytes,
los valores asignados para los tipos son "MODP", "ECP", "EC2N" los cuales serán definidos (véase ISAKMP-04).
Tamaño de un elemento de campo, en bits. Éste es Maximo(log2 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 EC (Curvas Elípticas): 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 como propietario de ese lugar para representar un parámetro no especifico, cualquier número de parámetros se puede enviar, desde el 0 hasta el 3.

El factor primo más grande: el valor codificado es decir el LPF del tamaño del grupo, un número entero de precisión múltiple.

Solamente para EC : 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.
Es determinado por el log2 del esfuerzo de atacar al grupo. Puede cambiar cuando aprendamos más sobre criptografía.

Este es un ejemplo genérico de un grupo "clásico" de exponenciación modular: Tipo del grupo: "MODP".
Tamaño de un elemento de campo en bits: Log2 (p) redondeado *por 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. 2 <= G <= 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: Log2 (p) redondeado *por 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 2^155 + 2^62 + 1.
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:

380599384721589301615546382619538626639743644 3,

representado como número entero de precisión múltiple.

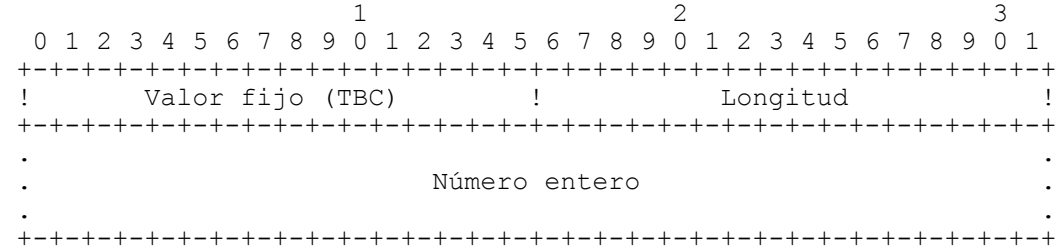
El orden del grupo:

456719261665907161938655659143446351967692373 16

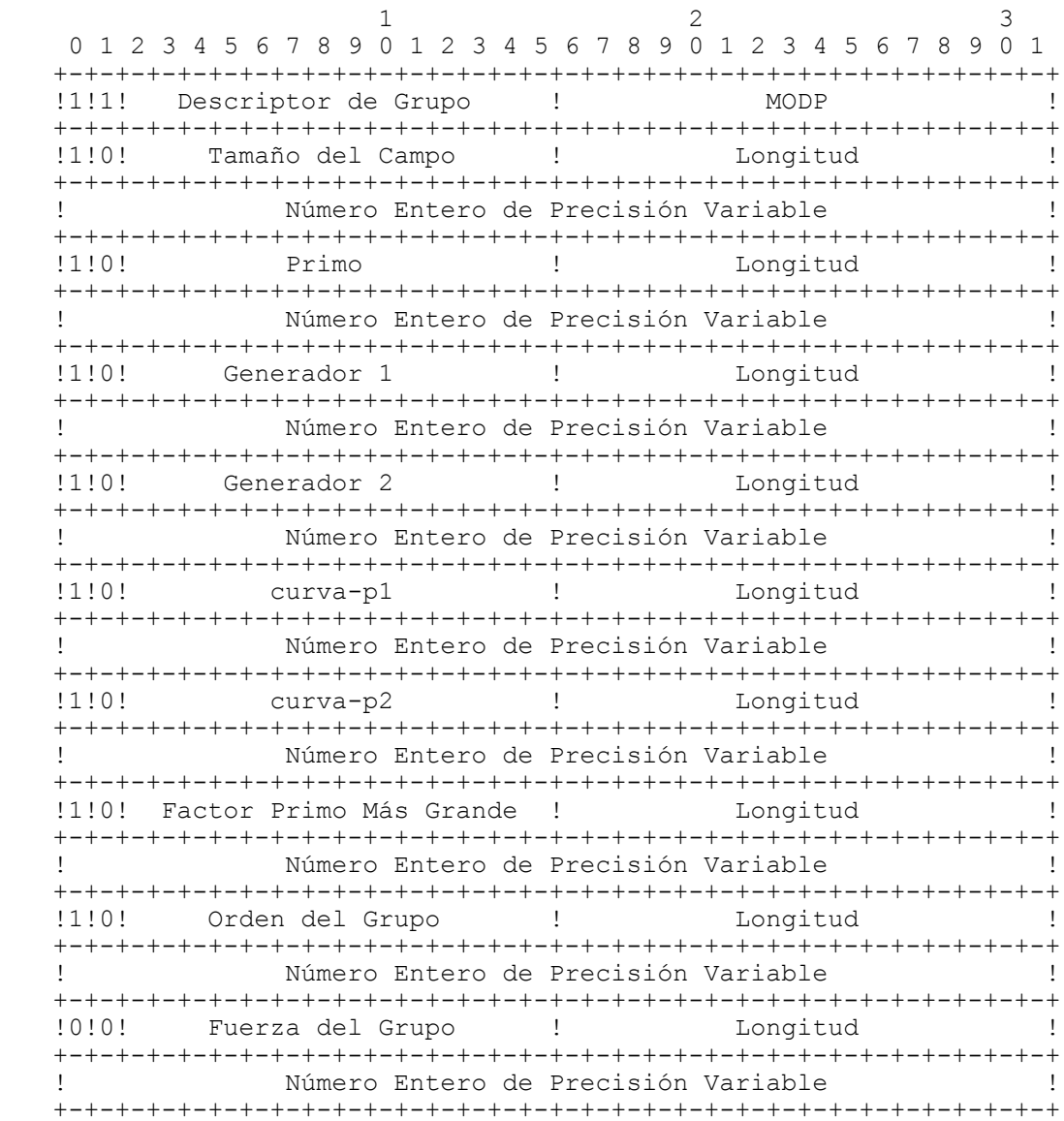
representado como número entero de precisión múltiple.

Fuerza del grupo: 76, representado como número entero de 32 bits.

La codificación del número entero de precisión variable para los campos del descriptor de grupo es el siguiente. Esto es una variación leve del formato definido en el Apéndice C 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.



El formato de un descriptor de grupo es:

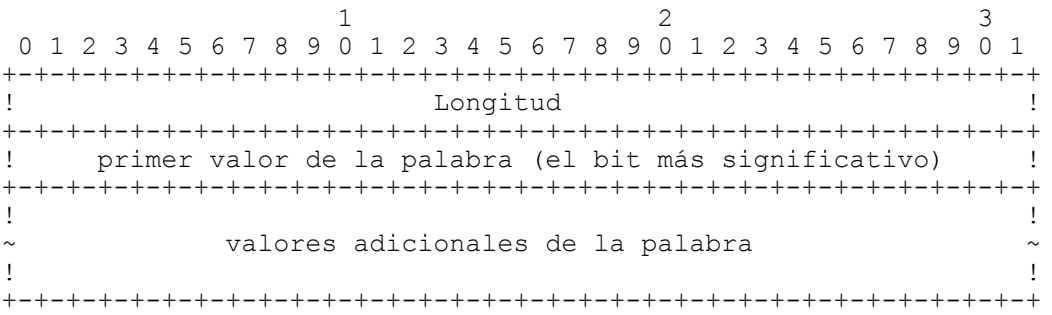

1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
!1!1! Descriptor de Grupo ! MODP !
!1!0! Tamaño del Campo ! Longitud !
! Número Entero de Precisión Variable !
!1!0! Primo ! Longitud !
! Número Entero de Precisión Variable !
!1!0! Generador 1 ! Longitud !
! Número Entero de Precisión Variable !
!1!0! Generador 2 ! Longitud !
! Número Entero de Precisión Variable !
!1!0! curva-p1 ! Longitud !
! Número Entero de Precisión Variable !
!1!0! curva-p2 ! Longitud !
! Número Entero de Precisión Variable !
!1!0! Factor Primo Más Grande ! Longitud !
! Número Entero de Precisión Variable !
!1!0! Orden del Grupo ! Longitud !
! Número Entero de Precisión Variable !
!0!0! Fuerza del Grupo ! Longitud !
! Número Entero de Precisión Variable !

APÉNDICE B Formato de los Mensajes

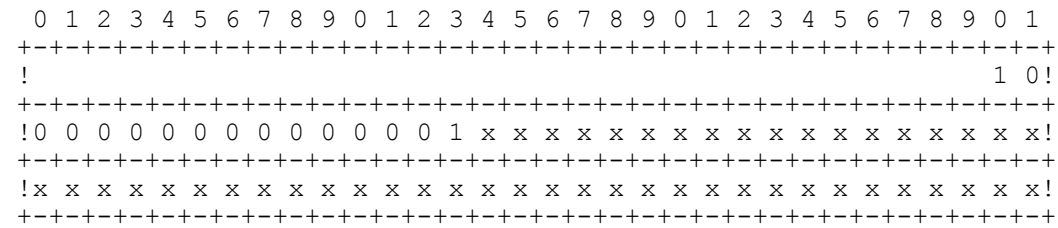
Las codificaciones de los mensajes Oakley en las cargas de ISAKMP se definirá en el documento de la Resolución de ISAKMP/Oakley.

APÉNDICE C 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.



Un ejemplo de tal codificación se da abajo, 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.



APÉNDICE D 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 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. Recomendamos 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 (log) discreto, una tarea que se relaciona fuertemente con la eficacia del uso del Cernidor del Campo Numérico [Number Field Sieve] para factorizar números enteros grandes. Para más información vea [Stinson] y [Schneier].

APÉNDICE E 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 [over] GF[2^155]
5 un grupo de curvas elípticas superiores a [over] GF[2^185]

los valores 2^31 y superiores se utilizan para identificar a los grupos privados.

Richard Schroeppel realizó todo el trabajo matemático y computacional para este apéndice.

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 digito del cociente de prueba [trial quotient digit] 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 a los del estilo de Montgomery, 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 pi. 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 pi, 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 (AKA 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 un punto de vista criptográfico, esto es una virtud.)

E.1. Grupo 1 Bien Conocido: Un número primo de 768 bits

El número primo es 2^768 - 2^704 - 1 + 2^64 * {[2^638 pi] + 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
```

E.2. Grupo 2 Bien Conocido: Un número Primo de 1024 bits

El número primo es 2^1024 - 2^960 - 1 + 2^64 * {[2^894 pi] + 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
```

E.3. Grupo 3 Bien Conocido: Una Definición de Grupos de Curvas Elípticas

La curva se basa en el campo 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

y

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:
456719261665907161938655659143446351967692373l6
el cual es 12 veces el número primo

380599384721589301615546382619538626639743644 3.

(Este número primo se ha probado rigurosamente.) 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
```

E.4 Grupo 4 Bien conocido: Una Definición General de Grupos de Curvas Elípticas

Esta curva se basa en el campo 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:

     4903985730770844346746710485765268224805238500104505311 6,

que es 4 veces el número primo

     1225996432692711086686677621441317056201309625026126327 9.

(Este número primo ha sido probado rigurosamente.)

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
```

E.5. Grupo 5 Bien Conocido: Un número Primo de 1536 bits

El número primo es 2^1536 - 2^1472 - 1 + 2^64 * { [2^1406 pi] + 741804 }. Su valor en decimal es:

```
            24103124269210325885520760221975660748569505485024599426541169419581088316826122288900938582613416146732271414779040121965036489570505826319427307068050092230627347453410734066962460145893616597740410271692494532003787294341703258437786591981437631937768598695240889401955773461198435453015470437472077499697637500843089263392955599688824578724129938101291302945929999479263652640592846472097303849472116814344647144384885209401274598442888593365268963209196339191
```

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
```

APÉNDICE F Implementación de funciones de Grupo

El funcionamiento del grupo debe estar implementado como 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 [remainders] por grupos modulares. Vea Knuth vol. 2 [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].

BIBLIOGRAFÍA

[RFC2401] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2406] Atkinson, R., "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2402] Atkinson, R., "IP Authentication Header", RFC 2402, November 1998.

[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

[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

[SECDNS] Eastlake, D. and C. Kaufman, "Domain Name System Security Extensions", RFC 2065, January 1997.

[Random] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[Kocher] Kocher, Paul, Timing Attack, http://www.cryptography.com/timingattack.old/timingattack.html

[Knuth] Knuth, Donald E., The Art of Computer Programming, Vol. 2, Seminumerical Algorithms, Addison Wesley, 1969.

[Krawcyzk] Krawcyzk, Hugo, SKEME: A Versatile Secure Key Exchange Mechanism for Internet, ISOC Secure Networks and Distributed Systems Symposium, San Diego, 1996

[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).

[Stinson] Stinson, Douglas, Cryptography Theory and Practice. CRC Press, Inc., 2000, Corporate Blvd., Boca Raton, FL, 33431-9868, ISBN 0-8493-8521-0, 1995

[Zimmerman] Philip Zimmermann, The Official Pgp User's Guide, Published by MIT Press Trade, Publication date: June 1995, ISBN: 0262740176

Declaración de Copyright Completa

Notas del Traductor

Los Términos que aparecen entre "[]" que no sean referencias reflejan la palabra/s en inglés de las palabra/s que se encuentran (en español) a la izquierda, debido a que NO ESTOY SEGURO de que sea la correcta traducción del término o simplemente para que no se pierda el VERDADERO sentido del texto.

En este presente texto cuando hace referencia al "Documento de Resolución" o "Documento de la Resolución de SAKMP/Oakley" hacia referencia al draft-ietf-ipsec-isakmp-oakley-05.txt que hoy por hoy se convirtió en el RFC 2409 (IKE). Para prueba de lo antedicho vean draft-ietf-ipsec-isakmp-gss-auth-00 y draft-ietf-ipsec-isakmp-gss-auth-01 en http://www.ietf.org.

Esta presente traducción fue realizada por Hugo Adrian Francisconi para mi tarjado de tesis de "Ingeniero en Electrónico" en la Facultad U.T.N. (Universidad Nacional Tecnología) Regional Mendoza - Argentina. Si le interesa IPsec y quieres saber más podes bajarte mi trabajo de tesis, "IPsec en Ambientes IPv4 e IPv6" de http://codarec6.frm.utn.edu.ar, para el cual traduje varios RFCs al español relacionados con IPsec. Cualquier sugerencia debate o comentario sobre este presente tema o traducción será bien recibida en adrianfrancisconi@yahoo.com.ar

Se a realizado el máximo esfuerzo para hacer de esta traducción sea 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 traductor no será responsable ante cualquier persona o entidad con respecto a cualquier pérdida o daño que pudiera resultar emergente de la información contenida en está traducción.

Datos del Traductor

Nombre y Apellido del Traductor: Hugo Adrian Francisconi
Domicilio: Carril Godoy Cruz 2801, Villa Nueva-Guay Mallen-Mendoza-Argentina
Código Postal: 5500
Tel: 054-0261-4455427
E-mail: adrianfrancisconi@yahoo.com.ar