(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 760 905**

(51) Int. CI.:

H04L 29/12 (2006.01)
H04L 29/06 (2006.01)
H04L 12/733 (2013.01)
H04L 12/707 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **29.10.1999 E 16165347 (2)**

(97) Fecha y número de publicación de la concesión europea: **11.09.2019 EP 3086533**

(54) Título: **Un protocolo de red agile para comunicaciones seguras con disponibilidad asegurada de sistema**

(30) Prioridad:

30.10.1998 US 106261 P
07.06.1999 US 137704 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
18.05.2020

(73) Titular/es:

VIRNETX INC. (100.0%)
308 Dorla Court, Suite 206, P.O. Box 439
Zephyr Cove, NV 89448-0439, US

(72) Inventor/es:

MUNGER, EDMUND COLBY;
SABIO, VINCENT, J;
SHORT, ROBERT, DUNHAM, III;
GLIGOR, VIRGIL, D. y
SCHMIDT, DOUGLAS, CHARLES

(74) Agente/Representante:

SÁEZ MAESO, Ana

DESCRIPCIÓN

Un protocolo de red agile para comunicaciones seguras con disponibilidad asegurada de sistema

Antecedentes de la invención

Se han propuesto e implementado una tremenda variedad de métodos para proporcionar seguridad y anonimato para las comunicaciones a través de Internet. La variedad proviene, en parte, de las diferentes necesidades de los diferentes usuarios de Internet. En la figura 1 se ilustra un marco heurístico básico para ayudar a discutir estas diferentes técnicas de seguridad. Dos terminales, un terminal 100 de origen y un terminal 110 de destino están en comunicación a través de Internet. Se desea que las comunicaciones sean seguras, es decir, inmunes a las escuchas. Por ejemplo, el terminal 100 puede transmitir información secreta al terminal 110 a través de Internet 107. Además, puede desear evitar que un espía descubra que el terminal 100 está en comunicación con el terminal 110. Por ejemplo, si el terminal 100 es un usuario y un terminal 110 aloja un sitio web, el usuario de la terminal 100 puede no querer que nadie en las redes intermedias sepa qué sitios web está “visitando”. El anonimato sería, por lo tanto, un problema, por ejemplo, para las empresas que desean mantener en privado sus intereses de investigación de mercado y, por lo tanto, preferirían evitar que personas externas sepan qué sitios web u otros recursos de Internet están “visitando”. Estos dos problemas de seguridad pueden denominarse seguridad de datos y anonimato, respectivamente.

La seguridad de los datos generalmente se aborda utilizando alguna forma de cifrado de datos. Se conoce una clave de cifrado 48 en los terminales 100 y 110 de origen y de destino. Las claves pueden ser privadas y públicas en los terminales 100 y 110 de origen y destino, respectivamente, o pueden ser claves simétricas (ambas partes utilizan la misma clave) cifrar y descifrar). Muchos métodos de cifrado son conocidos y utilizables en este contexto.

Para ocultar el tráfico de un administrador local o ISP, un usuario puede emplear un servidor proxy local para comunicarse a través de un canal cifrado con un proxy externo, de modo que el administrador local o ISP solo vea el tráfico cifrado. Los servidores proxy evitan que los servidores de destino determinen las identidades de los clientes de origen. Este sistema emplea un servidor intermedio interpuesto entre el cliente y el servidor de destino. El servidor de destino solo ve la dirección de Protocolo de Internet (IP) del servidor proxy y no el cliente de origen. El servidor de destino solo ve la dirección del proxy externo. Este esquema se basa en un servidor proxy externo de confianza. Además, los esquemas de proxy son vulnerables a los métodos de análisis de tráfico para determinar las identidades de los transmisores y receptores. Otra limitación importante de los servidores proxy es que el servidor conoce las identidades de las partes llamantes y llamadas. En muchos casos, un terminal de origen, como el terminal A, preferiría mantener su identidad oculta del proxy, por ejemplo, si el servidor proxy es proporcionado por un proveedor de servicios de Internet (ISP).

Para vencer el análisis del tráfico, un esquema llamado mezclas de Chaum emplea un servidor proxy que transmite y recibe mensajes de longitud fija, incluidos mensajes ficticios. Múltiples terminales de origen están conectadas a través de una mezcla (un servidor) a múltiples servidores de destino. Es difícil saber cuál de los terminales de origen se está comunicando con cuál de los servidores de destino conectados, y los mensajes ficticios confunden los esfuerzos de los espías para detectar pares de comunicación mediante el análisis del tráfico. Un inconveniente es que existe el riesgo de que el servidor mixto se vea comprometido. Una forma de lidiar con este riesgo es difundir la confianza entre múltiples mezclas. Si una mezcla se ve comprometida, las identidades de los terminales de origen y destino pueden permanecer ocultas. Esta estrategia requiere una serie de mezclas alternativas para que los servidores intermedios interpuestos entre los terminales de origen y destino no sean determinables, salvo comprometer más de una mezcla. La estrategia envuelve el mensaje con múltiples capas de direcciones cifradas. La primera mezcla en una secuencia puede descifrar solo la capa externa del mensaje para revelar la siguiente mezcla de destino en secuencia. La segunda mezcla puede descifrar el mensaje para revelar la próxima mezcla y así sucesivamente. El servidor de destino recibe el mensaje y, opcionalmente, una carga útil cifrada de múltiples capas que contienen información de retorno para enviar datos de la misma manera. La única forma de vencer ese esquema de mezcla es coludir entre mezclas. Si todos los paquetes son de longitud fija y se entremezclan con paquetes ficticios, no hay forma de hacer ningún tipo de análisis de tráfico.

Aún otra técnica de anonimato, llamada 'multitudes', protege la identidad del terminal de origen de los proxies intermedios al proporcionar que los terminales de origen pertenecen a grupos de proxies llamados multitudes. Los proxies de la multitud se interponen entre terminales de origen y destino. Cada proxy a través del cual se envía el mensaje es elegido aleatoriamente por un proxy en dirección de la red. Cada proxy intermedio puede enviar el mensaje a otro proxy elegido al azar en la “multitud” o al destino. Por lo tanto, incluso los miembros de la multitud no pueden determinar si un proxy anterior es el creador del mensaje o si simplemente se pasó desde otro proxy.

El protocolo IP anónimo ZKS (Zero-Knowledge Systems) permite a los usuarios seleccionar hasta cinco seudónimos diferentes, mientras que el software de escritorio cifra el tráfico saliente y lo envuelve en paquetes del Protocolo de Datagramas de Usuario (UDP). El primer servidor en un sistema de 2+- saltos obtiene los paquetes UDP, elimina una capa de cifrado para agregar otra, luego envía el tráfico al siguiente servidor, que elimina otra capa de cifrado y agrega una nueva. El usuario puede controlar el número de saltos. En el servidor final, el tráfico se descifra con una dirección IP no rastreable. La técnica se llama enrutamiento de cebolla. Este método puede ser derrotado utilizando análisis de

tráfico. Por un simple ejemplo, las ráfagas de paquetes de un usuario durante los períodos de poca actividad pueden revelar las identidades del remitente y el receptor.

Los cortafuegos intentan proteger las LAN del acceso no autorizado y la explotación hostil o el daño a los ordenadores conectadas a la LAN. Los cortafuegos proporcionan un servidor a través del cual debe pasar todo el acceso a la LAN. Los cortafuegos son sistemas centralizados que requieren una sobrecarga administrativa para mantenerse. Pueden verse comprometidos por aplicaciones de máquinas virtuales ("applets"). Infunden una falsa sensación de seguridad que conduce a violaciones de seguridad, por ejemplo, por parte de los usuarios que envían información confidencial a servidores fuera del cortafuegos o alientan el uso de módems para evadir la seguridad del cortafuegos. Los cortafuegos no son útiles para sistemas distribuidos como viajeros de negocios, extranets, equipos pequeños, etc.

El documento WO 98/31107 se refiere al denominado "enrutamiento de réplica" que dirige automáticamente los ordenadores de cliente que solicitan un servicio a una réplica del servidor para ese servicio. Un applet de cliente construye y envía una solicitud de enrutamiento de réplica a un enrutador de réplica. El enrutador de réplica calcula una lista de direcciones IP de destino posibles y envía un mensaje de respuesta que incluye la lista y los valores métricos de rendimiento correspondientes a cada elemento de la lista. El applet del cliente envía solicitudes de servicio a un número fijo de réplicas de servidor que tienen las mejores métricas de rendimiento.

Resumen de la invención

De acuerdo con la invención, se proporciona: un método para que un primer nodo establezca una sesión con un segundo nodo, como se describe en la reivindicación 1; un primer nodo según la reivindicación 11; un método para que un segundo nodo establezca una sesión con un primer nodo, como se describe en la reivindicación 12; un segundo nodo como se mencionó en la reivindicación 14.

Un mecanismo seguro para comunicarse a través de Internet, que incluye un protocolo denominado Protocolo de Enrutamiento Agile en Túnel (TARP), utiliza un formato único de cifrado de dos capas y enrutadores TARP especiales. Los enrutadores TARP tienen una función similar a los enrutadores IP normales. Cada enrutador TARP tiene una o más direcciones IP y utiliza el protocolo IP normal para enviar mensajes de paquetes IP ("paquetes" o "datagramas"). Los paquetes IP intercambiados entre terminales TARP a través de enrutadores TARP son en realidad paquetes cifrados cuya verdadera dirección de destino está oculta, excepto a los enrutadores y servidores TARP. El encabezado IP normal o "claro" o "externo" adjunto a los paquetes de TARP IP contiene solo la dirección de un enrutador de próximo salto o servidor de destino. Es decir, en lugar de indicar un destino final en el campo de destino del encabezado IP, el encabezado IP del paquete TARP siempre apunta al siguiente salto en una serie de saltos de enrutador TARP, o al destino final. Esto significa que no hay una indicación explícita de un paquete TARP interceptado del verdadero destino del paquete TARP ya que el destino siempre podría ser el enrutador TARP del siguiente salto, así como el destino final.

El verdadero destino de cada paquete TARP está oculto detrás de una capa de cifrado generada utilizando una clave de enlace. La clave de enlace es la clave de cifrado utilizada para la comunicación cifrada entre los saltos que intervienen entre un terminal TARP de origen y un terminal TARP de destino. Cada enrutador TARP puede eliminar la capa externa de cifrado para revelar el enrutador de destino para cada paquete TARP. Para identificar la clave de enlace necesaria para descifrar la capa externa de cifrado de un paquete TARP, un TARP receptor o un terminal de enrutamiento puede identificar el terminal remitente por los números IP del remitente/receptor en el encabezado IP de texto sin formato.

Una vez que se elimina la capa externa de cifrado, el enrutador TARP determina el destino final. Cada paquete 140 TARP se somete a un número mínimo de saltos para ayudar a frustrar el análisis del tráfico. Los saltos se pueden elegir al azar o por un valor fijo. Como resultado, cada paquete TARP puede realizar viajes aleatorios entre varios enrutadores geográficamente dispares antes de llegar a su destino. Es muy probable que cada viaje sea diferente para cada paquete que compone un mensaje dado porque cada viaje se determina aleatoriamente de forma independiente. Esta característica se llama enrutamiento agile. El hecho de que diferentes paquetes tomen diferentes rutas proporciona distintas ventajas al dificultar que un intruso obtenga todos los paquetes que forman un mensaje completo de múltiples paquetes. Las ventajas asociadas tienen que ver con la capa interna de cifrado que se analiza a continuación. El enrutamiento agile se combina con otra característica que promueve este propósito; Una función que garantiza que cualquier mensaje se divida en varios paquetes.

La dirección IP de un enrutador TARP puede no permanecer constante; una característica llamada IP Agility. Cada enrutador TARP, independientemente o bajo la dirección de otro terminal o enrutador TARP, puede cambiar su dirección IP. También se define un identificador o dirección independiente e inmutable. Esta dirección, llamada dirección TARP, es conocida solo por los enrutadores y terminales TARP y puede estar correlacionada en cualquier momento por un enrutador TARP o un terminal TARP utilizando una tabla de búsqueda (LUT). Cuando un enrutador o terminal TARP cambia su dirección IP, actualiza los otros enrutadores y terminales TARP que a su vez actualizan sus respectivos LUT.

La carga útil del mensaje está oculta detrás de una capa interna de cifrado en el paquete TARP que solo puede desbloquearse utilizando una clave de sesión. La clave de sesión no está disponible para ninguno de los enrutadores TARP que intervienen. La clave de sesión se utiliza para descifrar las cargas útiles de los paquetes TARP que permiten reconstruir el flujo de datos.

La comunicación puede hacerse privada utilizando el enlace y las claves de sesión, que a su vez pueden compartirse y usarse de acuerdo con cualquier método deseado. Por ejemplo, se pueden utilizar claves públicas/privadas o claves simétricas.

Para transmitir un flujo de datos, un terminal de origen TARP construye una serie de paquetes TARP a partir de una serie de paquetes IP generados por un proceso de capa de red (IP). (Tenga en cuenta que los términos “capa de red”, “capa de enlace de datos”, “capa de aplicación”, etc., utilizados en esta especificación corresponden a la terminología de red de Interconexión de Sistemas Abiertos (OSI)). Las cargas útiles de estos paquetes se ensamblan en un bloque y bloque de cadena cifrado utilizando la clave de sesión. Esto supone, por supuesto, que todos los paquetes IP están destinados al mismo terminal TARP. El bloque se intercala y el bloque cifrado intercalado se divide en una serie de cargas útiles, una para cada paquete TARP que se generará. Los encabezados TARP especiales IP_T se agregan a cada carga útil utilizando los encabezados IP de los paquetes de flujo de datos. Los encabezados TARP pueden ser idénticos a los encabezados IP normales o personalizados de alguna manera. Deben contener una fórmula o datos para desintercalar los datos en el terminal TARP de destino, un parámetro de tiempo de vida (TTL) para indicar el número de saltos aún por ejecutar, un identificador de tipo de datos que indica si la carga útil contiene, por ejemplo, datos TCP o UDP, la dirección TARP del remitente, la dirección TARP de destino y un indicador de si el paquete contiene datos reales o simulado o una fórmula para filtrar datos simulados si los datos simulados se extienden de alguna manera a través de los datos de carga útil TARP.

Tenga en cuenta que, aunque el cifrado de bloque de cadena se trata aquí con referencia a la clave de sesión, se puede utilizar cualquier método de cifrado. Preferiblemente, como en el cifrado de bloque de cadena, se debe utilizar un método que dificulte el descifrado no autorizado sin un resultado completo del proceso de cifrado. Por lo tanto, al separar el bloque cifrado entre múltiples paquetes y dificultar que un intruso obtenga acceso a todos esos paquetes, el contenido de las comunicaciones proporciona una capa adicional de seguridad.

Se pueden agregar datos simulados o falsos a un flujo para ayudar a frustrar el análisis de tráfico al reducir la carga de red de pico a promedio. Puede ser deseable proporcionar al proceso TARP la capacidad de responder a la hora del día u otros criterios para generar más datos simulados durante períodos de poco tráfico para que las ráfagas de comunicación en un punto de Internet no puedan vincularse a las ráfagas de comunicación en otro punto para revelar los extremos de comunicación.

Los datos ficticios también ayudan a dividir los datos en un mayor número de paquetes de tamaño discreto, lo que permite aumentar el tamaño de la ventana intercalada mientras se mantiene un tamaño razonable para cada paquete. (El tamaño del paquete puede ser un único tamaño estándar o puede seleccionarse de un rango fijo de tamaños). Una razón principal para desear que cada mensaje se divida en múltiples paquetes es evidente si se usa un esquema de cifrado de bloque de cadena para formar la primera capa de cifrado antes de intercalar. Se puede aplicar un cifrado de bloque único a la porción, o la totalidad, de un mensaje, y esa porción o la totalidad se intercalan en varios paquetes separados. Teniendo en cuenta el enrutamiento IP Agile de los paquetes y la dificultad concomitante de reconstruir una secuencia completa de paquetes para formar un único elemento de mensaje cifrado en bloque, los paquetes simulados pueden aumentar significativamente la dificultad de reconstruir un flujo de datos completo.

El esquema anterior puede implementarse completamente mediante procesos que operan entre la capa de enlace de datos y la capa de red de cada servidor o terminal que participa en el sistema TARP. Debido a que el sistema de cifrado descrito anteriormente es insertable entre el enlace de datos y las capas de red, los procesos involucrados en el soporte de la comunicación cifrada pueden ser completamente transparentes para los procesos en la capa de IP (red) y superior. Los procesos TARP también pueden ser completamente transparentes a los procesos de la capa de enlace de datos. Por lo tanto, la inserción de la pila TARP no afecta a las operaciones en la capa de red o por encima de ella, o por debajo de la capa de enlace de datos. Esto proporciona seguridad adicional a todos los procesos en o por encima de la capa de red, ya que la dificultad de penetración no autorizada de la capa de red (por ejemplo, por un hacker) aumenta sustancialmente. Incluso los servidores recientemente desarrollados que se ejecutan en la capa de sesión dejan todos los procesos por debajo de la capa de sesión vulnerables al ataque. Tenga en cuenta que, en esta arquitectura, la seguridad se distribuye. Es decir, los ordenadores portátiles utilizados por los ejecutivos en el camino, por ejemplo, pueden comunicarse a través de Internet sin comprometer la seguridad.

Los cambios de dirección IP realizados por los terminales y enrutadores TARP se pueden hacer a intervalos regulares, a intervalos aleatorios o al detectar “ataques”. La variación de las direcciones IP dificulta el análisis del tráfico que podría revelar qué ordenadores se están comunicando, y también proporciona un grado de inmunidad contra los ataques. El nivel de inmunidad frente al ataque es aproximadamente proporcional a la velocidad a la que cambia la dirección IP del host.

Como se mencionó, las direcciones IP pueden cambiarse en respuesta a los ataques. Un ataque puede ser revelado, por ejemplo, por una serie regular de mensajes que indican que se está probando un enrutador de alguna manera. Al detectar un ataque, el proceso de la capa TARP puede responder a este evento cambiando su dirección IP. Además, puede crear un subproceso que mantenga la dirección IP original y continúe interactuando con el atacante de alguna manera.

Los paquetes simulados pueden ser generados por cada terminal TARP sobre una base determinada por un algoritmo. Por ejemplo, el algoritmo puede ser aleatorio y requiere la generación de un paquete de manera aleatoria cuando el terminal está inactivo. Alternativamente, el algoritmo puede responder a la hora del día o la detección de tráfico bajo para generar más paquetes simulados durante tiempos de tráfico bajo. Tenga en cuenta que los paquetes se generan preferiblemente en grupos, en lugar de uno por uno, y los grupos se dimensionan para simular mensajes reales. Además, para que los paquetes simulados se puedan insertar en las secuencias de mensajes TARP normales, el bucle de fondo puede tener un bloqueo que hace que sea más probable insertar paquetes simulados cuando se recibe una secuencia de mensajes. Alternativamente, si se recibe una gran cantidad de paquetes simulados junto con los paquetes TARP normales, el algoritmo puede aumentar la tasa de caída de paquetes simulados en lugar de reenviarlos. El resultado de descartar y generar paquetes simulados de esta manera es hacer que el tamaño aparente del mensaje entrante sea diferente del tamaño aparente del mensaje saliente para ayudar a frustrar el análisis del tráfico.

En varias otras realizaciones de la invención, se puede construir una versión escalable del sistema en la que se preasignan una pluralidad de direcciones IP a cada par de nodos comunicantes en la red. Cada par de nodos acuerda un algoritmo para “saltar” entre direcciones IP (tanto de envío como de recepción), de modo que un espía vea pares de direcciones IP aparentemente aleatorios (origen y destino) para los paquetes transmitidos entre el par. Las direcciones IP superpuestas o “reutilizables” pueden asignarse a diferentes usuarios en la misma subred, ya que cada nodo simplemente verifica que un paquete particular incluye un par de origen/destino válido del algoritmo acordado. Los pares de origen/destino preferiblemente no se reutilizan entre dos nodos durante una sesión de extremo a extremo, aunque los tamaños de bloque de IP limitados o las sesiones largas pueden requerirlo.

Breve descripción de los dibujos

La figura 1 es una ilustración de comunicaciones seguras a través de Internet según una realización de la técnica anterior.

La figura 2 es una ilustración de comunicaciones seguras a través de Internet de acuerdo con una realización de la invención.

La figura 3a es una ilustración de un proceso de formación de un paquete IP tunelizado de acuerdo con una realización de la invención.

La figura 3b es una ilustración de un proceso de formación de un paquete IP tunelizado de acuerdo con otra realización de la invención.

La figura 4 es una ilustración de una ubicación de capa OSI de procesos que pueden usarse para implementar la invención.

La figura 5 es un diagrama de flujo que ilustra un proceso para enrutar un paquete tunelizado de acuerdo con una realización de la invención.

La figura 6 es un diagrama de flujo que ilustra un proceso para formar un paquete tunelizado de acuerdo con una realización de la invención.

La figura 7 es un diagrama de flujo que ilustra un proceso para recibir un paquete tunelizado de acuerdo con una realización de la invención.

La figura 8 muestra cómo se establece y sincroniza una sesión segura entre un cliente y un enrutador TARP.

La figura 9 muestra un esquema de salto de dirección IP entre un ordenador cliente y un enrutador TARP utilizando tablas de transmisión y recepción en cada ordenador.

La figura 10 muestra la redundancia del enlace físico entre tres proveedores de servicios de Internet (ISP) y un ordenador de cliente.

La figura 11 muestra cómo se pueden incorporar múltiples paquetes IP en una única “trama”, como una trama de Ethernet, y además muestra el uso de un campo discriminador para camuflar receptores de paquetes verdaderos.

La figura 12A muestra un sistema que emplea direcciones de hardware con salto, direcciones IP con salto y campos discriminadores con salto.

La figura 12B muestra varios enfoques diferentes para saltar direcciones de hardware, direcciones IP y campos discriminadores en combinación.

La figura 13 muestra una técnica para restablecer automáticamente la sincronización entre el remitente y el receptor mediante el uso de un valor de sincronización parcialmente público.

La figura 14 muestra un esquema de “punto de control” para recuperar la sincronización entre un remitente y un receptor.

La figura 15 muestra detalles adicionales del esquema de punto de control de la figura 14.

La figura 16 muestra cómo dos direcciones pueden descomponerse en una pluralidad de segmentos para compararlos con vectores de presencia.

Descripción detallada de las realizaciones

Con referencia a la figura 2, un mecanismo seguro para comunicarse a través de Internet emplea una serie de enrutadores o servidores especiales, llamados enrutadores 122-127 TARP que son similares a los enrutadores 128-132 IP normales en el que cada uno tiene una o más direcciones IP y utiliza el protocolo IP normal para envíe mensajes de paquetes IP de aspecto normal, llamados paquetes 140 TARP. Los paquetes 140 TARP son idénticos a los mensajes de paquetes IP normales que son enrutados por los enrutadores IP 128-132 normales porque cada paquete 140 TARP contiene una dirección de destino como en un paquete IP normal. Sin embargo, en lugar de indicar un destino final en el campo de destino del encabezado IP, el encabezado 140 IP del paquete TARP siempre apunta al siguiente salto en una serie de saltos de enrutador TARP, o el destino final, el terminal 110 TARP. Porque el encabezado del paquete TARP contiene solo el destino del próximo salto, no hay una indicación explícita de un paquete TARP interceptado del verdadero destino del paquete 140 TARP ya que el destino siempre podría ser el enrutador TARP del siguiente salto, así como el destino final, terminal 110 TARP.

El verdadero destino de cada paquete TARP está oculto detrás de una capa externa de cifrado generada utilizando una clave 146 de enlace. La clave 146 de enlace es la clave de cifrado utilizada para la comunicación cifrada entre los extremos (terminales TARP o enrutadores TARP) de un solo enlace en la cadena de saltos que conectan el terminal 100 TARP de origen y el terminal 110 TARP de destino. Cada enrutador 122-127 TARP, que utiliza la clave 146 de enlace la utiliza para comunicarse con el salto anterior en una cadena, puede utilizar la clave de enlace para revelar el verdadero destino de un paquete TARP. Para identificar la clave de enlace necesaria para descifrar la capa externa de cifrado de un paquete TARP, un TARP receptor o un terminal de enrutamiento puede identificar el terminal remitente (que puede indicar la clave de enlace utilizada) por el campo remitente del encabezado IP transparente. Alternativamente, esta identidad puede estar oculta detrás de otra capa de cifrado en los bits disponibles en el encabezado IP transparente. Cada enrutador TARP, al recibir un mensaje TARP, determina si el mensaje es un mensaje TARP mediante el uso de datos de autenticación en el paquete TARP. Esto podría registrarse en bytes disponibles en el encabezado IP del paquete TARP. Alternativamente, los paquetes TARP podrían autenticarse intentando descifrar utilizando la clave 146 de enlace y determinando si los resultados son los esperados. El primero puede tener ventajas computacionales porque no implica un proceso de descifrado.

Una vez que la capa externa de descifrado se completa con un enrutador 122-127 TARP, el enrutador TARP determina el destino final. El sistema está diseñado preferiblemente para hacer que cada paquete 140 TARP experimente un número mínimo de saltos para ayudar a frustrar el análisis del tráfico. El contador de tiempo de vida en el encabezado IP del mensaje TARP se puede utilizar para indicar una serie de saltos de enrutador TARP que aún no se han completado. Cada enrutador TARP entonces disminuirá el contador y determinará si debe reenviar el paquete 140 TARP a otro enrutador 122-127 TARP o al terminal 110 TARP de destino. Si el tiempo de vida del contador es cero o inferior a cero después de la disminución, por un ejemplo de uso, el enrutador TARP que recibe el paquete 140 TARP puede reenviar el paquete 140 TARP al terminal 110 TARP de destino. Si el contador de tiempo de vida es superior a cero después de disminuir, para un ejemplo de uso, el enrutador TARP que recibe el paquete 140 TARP puede reenviar el paquete 140 TARP a un enrutador 122-127 TARP que el terminal TARP actual elige al azar. Como resultado, cada paquete 140 TARP se enruta a través de un número mínimo de saltos de enrutadores 122-127 TARP que se eligen al azar.

Por lo tanto, cada paquete TARP, independientemente de los factores tradicionales que determinan el tráfico en Internet, realiza viajes aleatorios entre varios enrutadores geográficamente dispares antes de llegar a su destino y es muy probable que cada viaje sea diferente para cada paquete que compone un mensaje dado porque cada viaje se determina de forma aleatoria independientemente como se describió anteriormente. Esta característica se llama *enrutamiento Agile*. Por razones que se aclararán en breve, el hecho de que diferentes paquetes tomen diferentes rutas proporciona distintas ventajas al dificultar que un intruso obtenga todos los paquetes que forman un mensaje

completo de múltiples paquetes. El enrutamiento Agile se combina con otra función que promueve este propósito, una función que garantiza que cualquier mensaje se divida en múltiples paquetes.

Un enrutador TARP recibe un paquete TARP cuando una dirección IP utilizada por el enrutador TARP coincide con la dirección IP en el encabezado IP del paquete TARP IP_C. Sin embargo, la dirección IP de un enrutador TARP puede no permanecer constante. Para evitar y gestionar ataques, cada enrutador TARP, independientemente o bajo la dirección de otro terminal o enrutador TARP, puede cambiar su dirección IP. También se define un identificador o dirección independiente e inmutable. Esta dirección, llamada dirección TARP, es conocida solo por los enrutadores y terminales TARP y puede estar correlacionada en cualquier momento por un enrutador TARP o un terminal TARP utilizando una tabla de búsqueda (LUT). Cuando un enrutador o terminal TARP cambia su dirección IP, actualiza los otros enrutadores y terminales TARP que a su vez actualizan sus respectivos LUT. En realidad, cada vez que un enrutador TARP busca la dirección de un destino en el encabezado cifrado, debe convertir una dirección TARP en una dirección IP real utilizando su LUT.

Si bien cada enrutador TARP que recibe un paquete TARP tiene la capacidad de determinar el destino final del paquete, la carga útil del mensaje está incorporada detrás de una capa interna de cifrado en el paquete TARP que solo puede desbloquearse utilizando una clave de sesión. La clave de sesión no está disponible para ninguno de los enrutadores 122-127 TARP que intervienen entre los terminales TARP de origen 100 y destino 110. La clave de sesión se utiliza para descifrar las cargas útiles de los paquetes 140 TARP permitiendo que se reconstruya un mensaje completo.

En una realización, la comunicación puede hacerse privada utilizando el enlace y las teclas de sesión, que a su vez pueden compartirse y usarse de acuerdo con cualquier método deseado. Por ejemplo, una clave pública o claves simétricas pueden comunicarse entre enlaces o extremos de sesión utilizando un método de clave pública. Cualquiera de una variedad de otros mecanismos para asegurar los datos para garantizar que solo los ordenadores autorizadas puedan tener acceso a la información privada en los paquetes 140 TARP se pueden utilizar como se desee.

En referencia a la figura 3a, para construir una serie de paquetes TARP, un flujo 300 de datos de paquetes IP 207a, 207b, 207c, etc., tales series de paquetes formados por un proceso de capa de red (IP), se dividen en una serie de segmentos de tamaño pequeño. En el presente ejemplo, los segmentos 1-9 de igual tamaño se definen y se utilizan para construir un conjunto de paquetes de datos intercalados A, B y C. Aquí se supone que el número de paquetes intercalados A, B y C formados es tres y que el número de paquetes IP 207a-207c utilizados para formar los tres paquetes intercalados A, B y C es exactamente tres. Por supuesto, el número de paquetes IP distribuidos en un grupo de paquetes intercalados puede ser cualquier número conveniente como puede ser el número de paquetes intercalados sobre los cuales se distribuye el flujo de datos entrantes. El último, el número de paquetes intercalados sobre los cuales se extiende el flujo de datos, se denomina *ventana intercalada*.

Para crear un paquete, el software de transmisión intercala los paquetes 207a IP normales et. seq. para formar un nuevo conjunto de datos 320 de carga útil intercalados. Estos datos 320 de carga útil se cifran utilizando una clave de sesión para formar un conjunto de datos 330 de carga útil cifrados con clave de sesión, cada uno de los cuales, A, B y C, formarán la carga útil de un paquete de TARP. Usando los datos de encabezado IP, a partir de los paquetes 207a-207c originales, nuevos encabezados TARP IP_T. Se forman los encabezados TARP IP_T pueden ser idénticos a los encabezados IP normales o personalizados de alguna manera. En una realización preferida, los encabezados TARP IP_T son encabezados IP con datos agregados que proporcionan la siguiente información requerida para el enrutamiento y la reconstrucción de mensajes, algunos de cuyos datos están normalmente, o pueden ser, contenidos en encabezados IP normales:

1. Un número de secuencia de ventana: un identificador que indica dónde pertenece el paquete en la secuencia del mensaje original.

2. Un número de secuencia de intercalación: un identificador que indica la secuencia de intercalación utilizada para formar el paquete, de modo que el paquete se pueda desintercalar junto con otros paquetes en la ventana de intercalación.

3. Un dato de tiempo de vida (TTL): indica el número de saltos de enrutador TARP que se ejecutarán antes de que el paquete llegue a su destino. Tenga en cuenta que el parámetro TTL puede proporcionar un dato para ser utilizado en una fórmula probabilística para determinar si enrutar el paquete al destino o a otro salto.

4. Identificador del tipo de datos: indica si la carga útil contiene, por ejemplo, datos TCP o UDP.

5. Dirección del remitente: indica la dirección del remitente en la red TARP.

6. Dirección de destino: indica la dirección del terminal de destino en la red TARP.

7. Simulado/Real - un indicador de si el paquete contiene datos de mensajes reales o datos falsos simulado o una combinación.

Obviamente, los paquetes que entran en una sola ventana de intercalación deben incluir solo paquetes con un destino común. Por lo tanto, se supone en el ejemplo representado que los encabezados IP de los paquetes 207a-207c IP contienen todos la misma dirección de destino o al menos serán recibidos por el mismo terminal para que puedan ser desintercalados. Tenga en cuenta que se pueden agregar paquetes o datos ficticios o simulados para formar una ventana de intercalación más grande de lo que sería requerido por el tamaño de un mensaje dado. Se pueden agregar datos simulados o falsos a una secuencia para ayudar a frustrar el análisis de tráfico al nivelar la carga en la red. Por lo tanto, puede ser deseable proporcionar al proceso TARP la capacidad de responder a la hora del día u otros criterios para generar más datos simulados durante los períodos de poco tráfico para que las ráfagas de comunicación en un punto de Internet no puedan vincularse a las ráfagas de comunicación en otro punto para revelar los extremos de comunicación.

Los datos ficticios también ayudan a dividir los datos en un mayor número de paquetes de tamaño discreto que permite aumentar el tamaño de la ventana intercalada mientras se mantiene un tamaño razonable para cada paquete. (El tamaño del paquete puede ser un único tamaño estándar o puede seleccionarse de un rango fijo de tamaños). Una razón principal para desear que cada mensaje se divida en múltiples paquetes es evidente si se usa un esquema de cifrado de bloque de cadena para formar la primera capa de cifrado antes de intercalar. Se puede aplicar un cifrado de bloque único a la porción, o la totalidad, de un mensaje, y esa porción o la totalidad se intercalan en varios paquetes separados.

En referencia a la figura 3b, en un modo alternativo de construcción de paquetes TARP, se acumula una serie de paquetes IP para formar una ventana de intercalación predefinida. Las cargas útiles de los paquetes se utilizan para construir un único bloque 520 para el cifrado de bloques en cadena utilizando la clave de sesión. Se presume que las cargas útiles utilizadas para formar el bloque están destinadas al mismo terminal. El tamaño del bloque puede coincidir con la ventana de intercalación como se representa en la realización de ejemplo de la figura 3b. Después del cifrado, el bloque cifrado se divide en cargas útiles y segmentos separados que se entrelazan como en la realización de la figura 3a. Los paquetes intercalados resultantes A, B y C se empaquetan luego como paquetes TARP con encabezados TARP como en el Ejemplo de la figura 3a. El proceso restante es como se muestra y se analiza con referencia a la figura 3a.

Una vez que se forman los paquetes 340 TARP, cada paquete 340 TARP completo, incluido el encabezado TARP IP_T, se cifra utilizando la clave de enlace para la comunicación con el enrutador TARP de primer salto. El enrutador TARP del primer salto se elige al azar. Se agrega un encabezado IP no cifrado final IP_C a cada paquete 340 TARP cifrado para formar un paquete 360 IP normal que se puede transmitir a un enrutador TARP. Tenga en cuenta que el proceso de construcción del paquete 360 TARP no tiene que hacerse en etapas como se describe. La descripción anterior es solo una heurística útil para describir el producto final, es decir, el paquete TARP.

Tenga en cuenta que, el encabezado TARP IP_T podría ser una configuración de encabezado completamente personalizada sin similitud con un encabezado IP normal, excepto que contiene la información identificada anteriormente. Esto es así ya que este encabezado es interpretado solo por los enrutadores TARP.

El esquema anterior puede implementarse completamente mediante procesos que operan entre la capa de enlace de datos y la capa de red de cada servidor o terminal que participa en el sistema TARP. Con referencia a la figura 4, un transceptor 405 TARP puede ser un terminal 100 de origen, un terminal 110 de destino o un enrutador 122-127 TARP. En cada Transceptor 405 TARP, se genera un proceso de transmisión para recibir paquetes normales de la capa de Red (IP) y generar paquetes TARP para la comunicación a través de la red. Se genera un proceso de recepción para recibir paquetes IP normales que contienen paquetes TARP y se generan a partir de estos paquetes IP normales que se “pasan” a la capa de red (IP). Tenga en cuenta que cuando el Transceptor 405 TARP es un enrutador, los paquetes 140 TARP recibidos no se procesan en un flujo de paquetes IP 415 porque solo necesitan autenticarse como paquetes TARP adecuados y luego pasar a otro enrutador TARP o un terminal 110 TARP de destino. El proceso intermedio, una “Capa TARP” 420, podría combinarse con la capa 430 de enlace de datos o la capa 410 de red. En cualquier caso, intervendría entre la capa 430 de enlace de datos para que el proceso recibiera paquetes IP regulares que contengan paquetes TARP integrados y “entregar” una serie de paquetes IP reensamblados a la capa 410 de red. Como ejemplo de combinación de la capa 420 de TARP con la capa 430 de enlace de datos, un programa puede aumentar los procesos normales que ejecutan una tarjeta de comunicaciones, por ejemplo , una tarjeta ethernet. Alternativamente, los procesos de la capa TARP pueden formar parte de un módulo cargable dinámicamente que se carga y ejecuta para soportar las comunicaciones entre la red y las capas de enlace de datos.

Como el sistema de cifrado descrito anteriormente puede insertarse entre el enlace de datos y las capas de red, los procesos involucrados en el soporte de la comunicación cifrada pueden ser completamente transparentes para los procesos en la capa de IP (red) y superior. Los procesos TARP también pueden ser completamente transparentes a los procesos de la capa de enlace de datos. Por lo tanto, ninguna operación en o sobre la capa de red, o en o debajo de la capa de enlace de datos, se ve afectada por la inserción de la pila TARP. Esto proporciona seguridad adicional a todos los procesos en o por encima de la capa de red, ya que la dificultad de penetración no autorizada de la capa de red (por ejemplo, por un hacker) aumenta sustancialmente. Incluso los servidores recientemente desarrollados que se ejecutan en la capa de sesión dejan todos los procesos por debajo de la capa de sesión vulnerables al ataque.

Tenga en cuenta que, en esta arquitectura, la seguridad se distribuye. Es decir, los ordenadores portátiles utilizadas por los ejecutivos en el camino, por ejemplo, pueden comunicarse a través de Internet sin comprometer la seguridad.

Tenga en cuenta que los cambios de dirección IP realizados por los terminales y enrutadores TARP se pueden realizar a intervalos regulares, a intervalos aleatorios o al detectar “ataques”. La variación de las direcciones IP dificulta el análisis de tráfico que podría revelar qué ordenadores se están comunicando, y también proporciona un grado de inmunidad contra ataques. El nivel de inmunidad contra ataques es aproximadamente proporcional a la velocidad a la que cambia la dirección IP del host.

Como se mencionó, las direcciones IP pueden cambiarse en respuesta a los ataques. Un ataque puede ser revelado, por ejemplo, por una serie regular de mensajes indica que se está probando un enrutador de alguna manera. Al detectar un ataque, el proceso de la capa TARP puede responder a este evento cambiando su dirección IP. Para lograr esto, el proceso TARP construirá un mensaje con formato TARP, al estilo de los datagramas del Protocolo de Mensajes de Control de Internet (ICMP) como ejemplo; Este mensaje contendrá la dirección TARP de la máquina, su dirección IP anterior y su nueva dirección IP. La capa TARP transmitirá este paquete a al menos un enrutador TARP conocido; luego, al recibir y validar el mensaje, el enrutador TARP actualizará su LUT con la nueva dirección IP para la dirección TARP indicada. El enrutador TARP formateará un mensaje similar y lo transmitirá a los otros enrutadores TARP para que puedan actualizar sus LUT. Dado que se espera que el número total de enrutadores TARP en cualquier subred sea relativamente pequeño, este proceso de actualización de las LUT debe ser relativamente rápido. Sin embargo, puede que no funcione tan bien cuando hay un número relativamente grande de enrutadores TARP y/o un número relativamente grande de clientes; Esto ha motivado un refinamiento de esta arquitectura para proporcionar escalabilidad; Este refinamiento ha llevado a una segunda realización, que se discute a continuación.

Al detectar un ataque, el proceso TARP también puede crear un subproceso que mantiene la dirección IP original y continúa interactuando con el atacante. Este último puede proporcionar una oportunidad para rastrear al atacante o estudiar los métodos del atacante (llamado “pecera”, basándose en la analogía de un pez pequeño en una pecera que “piensa” que está en el océano pero que en realidad está bajo observación cautiva). Una historia de la comunicación entre el atacante y la dirección IP abandonada (pecera) puede ser grabada o transmitida para análisis humano o sintetizada para responder de alguna manera.

Como se mencionó anteriormente, se pueden agregar datos o paquetes falsos o simulados a los flujos de datos salientes por los terminales o enrutadores TARP. Además de facilitar la distribución de datos en un número mayor de paquetes separados, estos paquetes simulados también pueden ayudar a nivelar la carga en partes inactivas de Internet para frustrar los esfuerzos de análisis de tráfico.

Los paquetes simulados pueden ser generados por cada terminal 100, 110 TARP o por cada enrutador 122-127, de alguna manera determinada por un algoritmo. Por ejemplo, el algoritmo puede ser aleatorio y requiere la generación de un paquete de manera aleatoria cuando el terminal está inactivo. Alternativamente, el algoritmo puede responder a la hora del día o la detección de tráfico bajo para generar más paquetes simulados durante tiempos de tráfico bajo. Tenga en cuenta que los paquetes se generan preferiblemente en grupos, en lugar de uno por uno, y los grupos se dimensionan para simular mensajes reales. Además, para que los paquetes simulados se puedan insertar en los flujos de mensajes TARP normales, el bucle de fondo puede tener un bloqueo que hace que sea más probable insertar paquetes simulados cuando se recibe una secuencia de mensajes. Es decir, cuando se reciben una serie de mensajes, se puede aumentar la tasa de generación de paquetes simulado. Alternativamente, si se recibe una gran cantidad de paquetes simulados junto con los paquetes TARP normales, el algoritmo puede aumentar la tasa de caída de paquetes simulados en lugar de reenviarlos. El resultado de descartar y generar paquetes simulados de esta manera es hacer que el tamaño aparente del mensaje entrante sea diferente del tamaño aparente del mensaje saliente para ayudar a frustrar el análisis del tráfico. La velocidad de recepción de paquetes, simulado o de otro modo, puede indicarse a los procesos de caída y generación de paquetes simulados a través de contadores de paquetes simulados perecederos y regulares. (Un contador perecedero es aquel que restablece o disminuye su valor en respuesta al tiempo, de modo que contiene un valor alto cuando se incrementa en sucesión rápida y un valor pequeño cuando se incrementa lentamente o un pequeño número de veces en sucesión rápida). Tenga en cuenta que ese terminal 110 TARP de destino puede generar paquetes simulados iguales en número y tamaño a los paquetes TARP recibidos para hacer que parezca que se trata simplemente de paquetes de enrutamiento y, por lo tanto, no es el terminal de destino.

En referencia a la figura 5, los siguientes pasos particulares pueden emplearse en el método descrito anteriormente para enrutar paquetes TARP.

• S0. Se realiza una operación de bucle de fondo que aplica un algoritmo que determina la generación de paquetes IP simulado. El bucle se interrumpe cuando se recibe un paquete TARP cifrado.

• S2. El paquete TARP puede probarse de alguna manera para autenticar el paquete antes de intentar descifrarlo utilizando la clave de enlace. Es decir, el enrutador puede determinar que el paquete es un paquete TARP auténtico al realizar una operación seleccionad a en algunos datos incluidos con el encabezado IP no cifrado adjunto al paquete TARP cifrado contenido en la carga útil. Esto permite evitar descifrar los paquetes que no son auténticos paquetes TARP.

• S3. El paquete TARP se descifra para exponer la dirección TARP de destino y una indicación de si el paquete es un paquete simulado o parte de un mensaje real.

• S4. Si el paquete es un paquete simulado, el contador simulado perecedero se incrementa.

• S5. Según el algoritmo de generación/caída simulado y el valor del contador simulado perecedero, si el paquete es un paquete simulado, el enrutador puede optar por tirarlo. Si el paquete recibido es un paquete simulado y se determina que debe desecharse (S6), el control vuelve al paso S0.

• S7. El parámetro TTL del encabezado TARP se reduce y se determina si el parámetro TTL es mayor que cero.

• S8. Si el parámetro TTL es mayor que cero, se elige aleatoriamente una dirección TARP de una lista de direcciones TARP mantenidas por el enrutador y la clave de enlace y la dirección IP correspondiente a esa dirección TARP memorizada para crear un nuevo paquete IP que contenga el paquete TARP .

• S9. Si el parámetro TTL es cero o menos, la clave de enlace y la dirección IP correspondiente a la dirección TARP del destino se memorizan para su uso en la creación del nuevo paquete IP que contiene el paquete TARP.

• S10. El paquete TARP se cifra utilizando la clave de enlace memorizada.

• S11. Se agrega un encabezado IP al paquete que contiene la dirección IP almacenada, el paquete TARP cifrado envuelto con un encabezado IP y el paquete completado se transmite al siguiente salto o destino.

Con referencia a la figura 6, los siguientes pasos particulares pueden emplearse en el método descrito anteriormente para generar paquetes TARP.

• S20. Una operación de bucle de fondo aplica un algoritmo que determina la generación de paquetes IP simulado. El bucle se interrumpe cuando se recibe una transmisión de datos que contiene paquetes IP para su transmisión.

• S21. Los paquetes IP recibidos se agrupan en un conjunto que consiste en mensajes con una dirección IP de destino constante. El conjunto se desglosa para coincidir con el tamaño máximo de una ventana de intercalación. El conjunto se cifra y se intercala en un conjunto de cargas útiles destinadas a convertirse en paquetes TARP.

• S22. La dirección TARP correspondiente a la dirección IP se determina a partir de una tabla de búsqueda y se almacena para generar el encabezado TARP. Se genera un recuento TTL inicial y se almacena en el encabezado. El recuento TTL puede ser aleatorio con valores mínimos y máximos o puede ser fijo o determinado por algún otro parámetro.

• S23. Los números de secuencia de ventana y los números de secuencia de intercalación se registran en los encabezados TARP de cada paquete.

• S24. Se elige aleatoriamente una dirección de enrutador TARP para cada paquete TARP y la dirección IP correspondiente se almacena para su uso en el encabezado IP transparente. La clave de enlace correspondiente a este enrutador se identifica y se utiliza para cifrar paquetes TARP que contienen encabezados TARP y datos cifrados e intercalados.

• S25. Se genera un encabezado IP no cifrado con la dirección IP real del enrutador del primer salto y se agrega a cada uno de los paquetes TARP cifrados y los paquetes resultantes.

En referencia a la figura 7, los siguientes pasos particulares pueden emplearse en el método descrito anteriormente para recibir paquetes TARP.

• S40. Se realiza una operación de bucle de fondo que aplica un algoritmo que determina la generación de paquetes IP simulado. El bucle se interrumpe cuando se recibe un paquete TARP cifrado.

• S42. El paquete TARP puede probarse para autenticar el paquete antes de intentar descifrarlo utilizando la clave de enlace.

• S43. El paquete TARP se descifra con la clave de enlace apropiada para exponer la dirección TARP de destino y una indicación de si el paquete es un paquete simulado o parte de un mensaje real.

• S44. Si el paquete es un paquete simulado, el contador simulado perecedero se incrementa.

• S45. Según el algoritmo de generación/caída simulado y el valor del contador simulado perecedero, si el paquete es un paquete simulado, el receptor puede optar por tirarlo.

• S46. Los paquetes TARP se almacenan en caché hasta que se reciben todos los paquetes que forman una ventana intercalada.

• S47. Una vez que se reciben todos los paquetes de una ventana de intercalación, los paquetes se desintercalan.

• S48. El bloque de paquetes combinados que define la ventana de intercalación se descifra utilizando la clave de sesión.

• S49. El bloque descifrado se divide utilizando los datos de secuencia de la ventana y los IP_T encabezados se convierten en encabezados IP_C normales. Los números de secuencia de la ventana están integrados en los encabezados IPC.

• S50. Los paquetes se entregan a los procesos de la capa IP.

Mejoras de escalabilidad

La característica de agilidad IP descrita anteriormente se basa en la capacidad de transmitir cambios de dirección IP a todos los enrutadores TARP. Las realizaciones que incluyen esta característica se denominarán realizaciones “boutique” debido a limitaciones potenciales en la ampliación de estas características para una red grande, como Internet. (Sin embargo, las realizaciones “boutique” serían robustas para su uso en redes más pequeñas, como pequeñas redes privadas virtuales, por ejemplo). Un problema con las realizaciones de boutique es que, si los cambios de dirección IP se producen con frecuencia, el tráfico de mensajes requerido para actualizar todos los enrutadores lo suficientemente rápido crea una carga seria en Internet cuando el enrutador TARP y/o la población de clientes aumenta. La carga de ancho de banda agregada a las redes, por ejemplo, en paquetes ICMP, que se utilizaría para actualizar todos los enrutadores TARP podría abrumar a Internet para una implementación a gran escala que se acercara a la escala de Internet. En otras palabras, la escalabilidad del sistema boutique es limitada.

Se puede construir un sistema que intercambie algunas de las características de las realizaciones anteriores para proporcionar los beneficios de la agilidad de IP sin la carga adicional de mensajes. Esto se logra mediante el salto de direcciones IP de acuerdo con algoritmos compartidos que gobiernan las direcciones IP utilizadas entre enlaces que participan en sesiones de comunicaciones entre nodos, como los nodos TARP. (Tenga en cuenta que la técnica de salto de IP también es aplicable a la realización de boutique.) La característica de agilidad de IP discutida con respecto al sistema de boutique se puede modificar para que se descentralice bajo este régimen escalable y se rija por el algoritmo compartido descrito anteriormente. Otras características del sistema boutique se pueden combinar con este nuevo tipo de agilidad IP.

La nueva realización tiene la ventaja de proporcionar agilidad IP gobernada por un algoritmo local y un conjunto de direcciones IP intercambiadas por cada par de nodos comunicantes. Esta gobernanza local es independiente de la sesión, ya que puede gobernar las comunicaciones entre un par de nodos, independientemente de la sesión o los puntos de extremo que se transfieren entre el par de nodos que se comunican directamente.

En las realizaciones escalables, se asignan bloques de direcciones IP a cada nodo en la red. (Esta escalabilidad aumentará en el futuro, cuando las direcciones del Protocolo de Internet se aumenten a campos de 128 bits, aumentando enormemente el número de nodos claramente direccionables). Por lo tanto, cada nodo puede utilizar cualquiera de las direcciones IP asignadas a ese nodo para comunicarse con otros nodos en la red. De hecho, cada par de nodos de comunicación puede utilizar una pluralidad de direcciones IP de origen y direcciones IP de destino para comunicarse entre sí.

Cada par de nodos comunicantes en una cadena que participa en cualquier sesión almacena dos bloques de direcciones IP, llamados “netblocks”, y un algoritmo y semilla de aleatorización para seleccionar, de cada bloque de red, el siguiente par de direcciones IP de origen/destino que será utilizado para transmitir el siguiente mensaje. En otras palabras, el algoritmo gobierna la selección secuencial de pares de direcciones IP, una dirección IP del remitente y un receptor, de cada netblock. La combinación de algoritmo, semilla y netblock (bloque de dirección IP) se denominará “bloque de salto”. Un enrutador emite bloques de transmisión y recepción por separado para sus clientes. La dirección de envío y la dirección de recepción del encabezado IP de cada paquete saliente enviado por el cliente se completan con las direcciones IP de envío y recepción generadas por el algoritmo. El algoritmo es “sincronizado” (indexado) por un contador para que cada vez que se use un par, el algoritmo genere un nuevo par de transmisión para el siguiente paquete que se enviará.

El bloque de salto de recepción del enrutador es idéntico al bloque de salto de transmisión del cliente. El enrutador utiliza el bloque de salto de recepción para predecir cuál será el par de direcciones IP de envío y recepción para el próximo paquete esperado de ese cliente. Dado que los paquetes pueden recibirse fuera de orden, el enrutador no puede predecir con certeza qué par de direcciones IP estará en el siguiente paquete secuencial. Para tener en cuenta este problema, el enrutador genera un rango de predicciones que abarcan el número de posibles direcciones de envío/recepción de paquetes transmitidos, de los cuales el siguiente paquete recibido podría avanzar. Por lo tanto, si

existe una probabilidad muy pequeña de que un paquete determinado llegue al enrutador antes de los 5 paquetes transmitidos por el cliente antes del paquete dado, entonces el enrutador puede generar una serie de 6 pares de direcciones IP de envío/recepción (o “salto” ventana “) para comparar con el siguiente paquete recibido. Cuando se recibe un paquete, se marca en la ventana de salto como tal, de modo que se descarte un segundo paquete con el mismo par de direcciones IP. Si un paquete fuera de secuencia no llega dentro de un período de tiempo de espera predeterminado, se puede solicitar su retransmisión o simplemente descartarlo de la tabla de recepción, dependiendo del protocolo en uso para esa sesión de comunicaciones, o posiblemente por convención.

Cuando el enrutador recibe el paquete del cliente, compara las direcciones IP de envío y recepción del paquete con los próximos N pares de direcciones IP de envío y recepción previstos y rechaza el paquete si no es miembro de este conjunto. Los paquetes recibidos que no tienen las direcciones IP de origen/destino pronosticadas que caen en la ventana son rechazados, frustrando así a posibles hackers. (Con el número de combinaciones posibles, incluso una ventana bastante grande sería difícil de encontrar al azar). Si es miembro de este conjunto, el enrutador acepta el paquete y lo procesa aún más. Esta estrategia de salto de IP basada en enlaces, denominada “IHOP”, es un elemento de red independiente y no necesariamente va acompañado de elementos del sistema de boutique descrito anteriormente. Si la característica de agilidad de enrutamiento descrita en relación con la realización boutique se combina con esta estrategia de salto de IP basada en enlaces, el siguiente paso del enrutador sería descifrar el encabezado TARP para determinar el enrutador TARP de destino para el paquete y determinar cuál debería ser el próximo salto para el paquete. El enrutador TARP reenviaría el paquete a un enrutador TARP aleatorio o al enrutador TARP de destino con el cual el enrutador TARP de origen tiene establecida una comunicación de salto de IP basada en enlaces.

La figura 8 muestra cómo un ordenador de cliente 801 y un enrutador 811 TARP pueden establecer una sesión segura. Cuando el cliente 801 busca establecer una sesión IHOP con el enrutador 811 TARP, el cliente 801 envía el paquete 821 de solicitud de “sincronización segura” (“SSYN”) al enrutador 811 TARP. Este paquete 821 SYN contiene la credencial de autenticación 801 del cliente, y puede ser enviado al enrutador 811 en un formato cifrado. Los números IP de origen y destino en el paquete 821 son la dirección IP fija actual 801 del cliente y una dirección IP fija “conocida” para el enrutador 811. (Por razones de seguridad, puede ser deseable rechazar cualquier paquete desde fuera del local red destinada a la dirección IP fija conocida del enrutador). Al recibir y validar el paquete 821 SSYN 801 del cliente, el enrutador 811 responde enviando una “confirmación de sincronización segura” (“SSYN ACK”) 822 cifrado al cliente 801. Este SSYN ACK 822 contendrá los bloques de salto de transmisión y recepción que utilizará el cliente 801 cuando se comunique con el enrutador 811 TARP. El cliente 801 reconocerá el paquete de respuesta 811 del enrutador TARP 822 generando un paquete 823 SSYN ACK ACK cifrado que se enviará desde la dirección IP fija 801 del cliente y hasta la dirección IP fija conocida del enrutador 811 TARP. El cliente 801 generará simultáneamente un paquete SSYN ACK ACK; este paquete SSYN ACK, denominado paquete 824 de iniciación de sesión segura (SSI), se enviará con el primer par IP {remitente, receptor} en la tabla 921 de transmisión del cliente (FIG. 9), como se especifica en el bloque de salto de transmisión proporcionado por el enrutador 811 TARP en el paquete SSYN ACK 822. El enrutador 811 TARP responderá al paquete SSI 824 con un paquete SSI ACK 825, que se enviará con el primer par IP {remitente, receptor} en la tabla 923 de transmisión del enrutador TARP. Una vez que estos paquetes se han intercambiado con éxito, se establece la sesión de comunicaciones seguras, y todas las comunicaciones seguras adicionales entre el cliente 801 y el enrutador 811 TARP se llevarán a cabo a través de esta sesión segura, siempre que se mantenga la sincronización. Si se pierde la sincronización, entonces el cliente 801 y el enrutador 802 TARP pueden restablecer la sesión segura mediante el procedimiento descrito en la Figura 8 y descrito anteriormente.

Mientras la sesión segura está activa, tanto el cliente 901 como el enrutador 911 TARP (figura 9) mantendrán sus respectivas tablas 921, 923 de transmisión y recibirán las tablas 922, 924, según lo dispuesto por el enrutador TARP durante la sincronización 822 de sesión. Es importante que la secuencia de pares de IP en la tabla 921 de transmisión del cliente sea idéntica a la de la tabla 924 de recepción del enrutador TARP; de manera similar, la secuencia de pares de IP en la tabla 922 de recepción del cliente debe ser idéntica a la de la tabla 923 de transmisión del enrutador. Esto es necesario para mantener la sincronización de la sesión. El cliente 901 necesita mantener solo una tabla 921 de transmisión y una tabla 922 de recepción durante el curso de la sesión segura. Cada paquete secuencial enviado por el cliente 901 empleará el siguiente par de direcciones IP {enviar, recibir} en la tabla de transmisión, independientemente de la sesión TCP o UDP. El enrutador 911 TARP esperará que cada paquete que llegue desde el cliente 901 lleve el siguiente par de direcciones IP que se muestra en su tabla de recepción.

Como los paquetes pueden llegar fuera de orden, sin embargo, el enrutador 911 puede mantener un búfer “de anticipación” en su tabla de recepción, y marcará los pares de IP recibidos previamente como inválidos para futuros paquetes; cualquier paquete futuro que contenga un par de IP que esté en el búfer de anticipación, pero esté marcado como recibido previamente será descartado. Las comunicaciones del enrutador 911 TARP al cliente 901 se mantienen de manera idéntica; en particular, el enrutador 911 seleccionará el siguiente par de direcciones IP de su tabla 923 de transmisión cuando construya un paquete para enviar al cliente 901, y el cliente 901 mantendrá un búfer de anticipación de los pares de IP esperados en los paquetes que está recibiendo. Cada enrutador TARP mantendrá pares separados de tablas de transmisión y recepción para cada cliente que esté actualmente en una sesión segura con o a través de ese enrutador TARP.

Mientras que los clientes reciben sus bloques de salto del primer servidor que los conecta a Internet, los enrutadores intercambian bloques de salto. Cuando un enrutador establece un régimen de comunicación de salto de IP basado en enlaces con otro enrutador, cada enrutador del par intercambia su bloque de salto de transmisión. El bloque de salto de transmisión de cada enrutador se convierte en el bloque de salto de recepción del otro enrutador. La comunicación entre enrutadores se rige como se describe en el ejemplo de un cliente que envía un paquete al primer enrutador.

Aunque la estrategia anterior funciona bien en el entorno de IP, muchas redes locales que están conectadas a Internet son sistemas ethernet. En ethernet, las direcciones IP de los dispositivos de destino deben traducirse a direcciones de hardware y viceversa, utilizando procesos conocidos ("protocolo de resolución de direcciones" y "protocolo de resolución de direcciones inversas"). Sin embargo, si se emplea la estrategia de salto de IP basada en enlaces, el proceso de correlación se volvería explosivo y oneroso. Se puede emplear una alternativa a la estrategia de salto de IP basada en enlaces dentro de una red ethernet. La solución es proporcionar que el nodo que conecta Internet a Ethernet (llámelo el nodo de borde) use el régimen de comunicación de salto de IP basado en enlaces para comunicarse con nodos fuera de la LAN de Ethernet. Dentro de la LAN Ethernet, cada nodo TARP tendría una única dirección IP que se abordaría de la manera convencional. En lugar de comparar los pares de direcciones IP {remitente, receptor} para autenticar un paquete, el nodo TARP intra-LAN usaría uno de los campos de extensión de encabezado IP para hacerlo. Por lo tanto, el nodo de borde utiliza un algoritmo compartido por el nodo TARP intra-LAN para generar un símbolo que se almacena en el campo libre en el encabezado IP, y el nodo TARP intra-LAN genera un rango de símbolos basado en su predicción de siguiente paquete esperado que se recibirá de esa dirección IP de origen particular. El paquete se rechaza si no cae dentro del conjunto de símbolos predichos (por ejemplo, valores numéricos) o se acepta si lo hace. Las comunicaciones desde el nodo TARP intra-LAN al nodo límite se realizan de la misma manera, aunque el algoritmo será necesariamente diferente por razones de seguridad. Por lo tanto, cada uno de los nodos comunicantes generará tablas de transmisión y recepción de manera similar a la de la Figura 9; la tabla de transmisión de los nodos TARP intra-LAN será idéntica a la tabla de recepción del nodo TARP, y la tabla de recepción del nodo TARP intra-LAN será idéntica a la tabla de transmisión del nodo fronterizo.

El algoritmo utilizado para el salto de dirección IP puede ser cualquier algoritmo deseado. Por ejemplo, el algoritmo puede ser un generador de números pseudoaleatorio dado que genera números del rango que cubre las direcciones IP permitidas con una semilla dada. Alternativamente, los participantes de la sesión pueden asumir un cierto tipo de algoritmo y especificar simplemente un parámetro para aplicar el algoritmo. Por ejemplo, el algoritmo supuesto podría ser un generador de números pseudoaleatorio particular y los participantes de la sesión podrían simplemente intercambiar valores semilla.

Tenga en cuenta que no existe una distinción física permanente entre los nodos terminales de origen y destino. Cualquiera de los dispositivos en cualquier punto final puede iniciar una sincronización del par. Tenga en cuenta también que la solicitud de autenticación/sincronización (y confirmación) y el intercambio de bloques de salto pueden ser atendidos por un solo mensaje, por lo que puede que no sea necesario intercambiar mensajes separados.

Como otra extensión de la arquitectura establecida, un cliente puede utilizar múltiples rutas físicas para proporcionar redundancia de enlace y frustrar aún más los intentos de denegación de servicio y monitoreo de tráfico. Como se muestra en la Figura 10, por ejemplo, el cliente 1001 puede establecer tres sesiones simultáneas con cada uno de los tres enrutadores TARP proporcionados por diferentes ISP 1011, 1012, 1013. Como ejemplo, el cliente 1001 puede utilizar tres líneas 1021, 1022, 1023 telefónicas diferentes para conectarse a los ISP, o dos líneas telefónicas y un módem de cable, etc. En este esquema, los paquetes transmitidos se enviarán de manera aleatoria entre las diferentes rutas físicas. Esta arquitectura proporciona un alto grado de redundancia de comunicaciones, con inmunidad mejorada contra ataques de denegación de servicio y monitoreo de tráfico.

Extensiones adicionales

A continuación, se describen diversas extensiones de las técnicas, sistemas y métodos descritos anteriormente. Como se describió anteriormente, la seguridad de las comunicaciones que se producen entre ordenadores en una red informática (como Internet, Ethernet u otras) se puede mejorar utilizando direcciones de Protocolo de Internet (IP) de origen y destino aparentemente aleatorias para los paquetes de datos transmitidos a través de la red. Esta característica evita que los espías determinen qué ordenadores de la red se comunican entre sí, al tiempo que permite que los dos ordenadores que se comunican reconozcan fácilmente si un paquete de datos recibido es legítimo o no. En una realización de los sistemas descritos anteriormente, se usa un campo de extensión de encabezado IP para autenticar los paquetes entrantes en una Ethernet.

Varias extensiones de las técnicas descritas anteriormente divulgadas en el presente documento incluyen: (1) uso de hardware saltado o direcciones "MAC" en una red de tipo difusión; (2) una técnica de auto-sincronización que permite que una computadora recupere automáticamente la sincronización con un remitente; (3) algoritmos de sincronización que permiten transmitir y recibir ordenadores para restablecer rápidamente la sincronización en caso de pérdida de paquetes u otros eventos; y (4) un mecanismo de rechazo rápido de paquetes para rechazar paquetes no válidos. Cualquiera o todas estas extensiones se pueden combinar con las características descritas anteriormente de varias maneras.

A. Salto de dirección de hardware

Las técnicas de comunicación basadas en el protocolo de Internet en una LAN, o en cualquier medio físico dedicado, típicamente incorporan los paquetes IP en paquetes de nivel inferior, a menudo denominados “tramas”. Como se muestra en la figura 11, por ejemplo, una primera trama 1150 Ethernet comprende un encabezado 1101 de trama y dos paquetes IP integrados IP1 e IP2, mientras que una segunda trama 1160 Ethernet comprende un encabezado 1104 de trama diferente y un único paquete IP IP3. Cada encabezado de trama generalmente incluye una dirección 1101A de hardware de origen y una dirección 1101B de hardware de destino; otros campos bien conocidos en encabezados de trama se omiten de la figura 11 para mayor claridad. Dos nodos de hardware que se comunican a través de un canal de comunicación física insertan las direcciones de hardware de origen y destino apropiadas para indicar qué nodos en el canal o red deberían recibir la trama.

Puede ser posible que un oyente nefasto adquiera información sobre el contenido de una trama y/o sus comunicantes al examinar las tramas en una red local en lugar de (o además de) los paquetes IP en sí. Esto es especialmente cierto en los medios de difusión, como Ethernet, donde es necesario insertar en el encabezado del marco la dirección de hardware de la máquina que generó el marco y la dirección de hardware de la máquina a la que se envía el marco. Todos los nodos en la red pueden potencialmente “ver” todos los paquetes transmitidos a través de la red. Esto puede ser un problema para las comunicaciones seguras, especialmente en los casos en que los comunicantes no desean que ningún tercero pueda identificar quién participa en el intercambio de información. Una forma de abordar este problema es llevar el esquema de salto de dirección a la capa de hardware. De acuerdo con diversas realizaciones de la invención, las direcciones de hardware se “saltan” de una manera similar a la utilizada para cambiar las direcciones IP, de modo que un oyente no puede determinar qué nodo de hardware generó un mensaje particular ni qué nodo es el receptor previsto.

La figura 12A muestra un sistema en el que las direcciones de hardware de Control de acceso al medio (“MAC”) se “saltan” para aumentar la seguridad en una red como una Ethernet. Si bien la descripción se refiere al caso ejemplar de un entorno Ethernet, los principios inventivos son igualmente aplicables a otros tipos de medios de comunicación. En el caso de Ethernet, la dirección MAC del remitente y el receptor se insertan en la trama de Ethernet y pueden ser observados por cualquier persona en la LAN que esté dentro del rango de transmisión para esa trama. Para comunicaciones seguras, es deseable generar tramas con direcciones MAC que no sean atribuibles a ningún remitente o receptor específico.

Como se muestra en la figura 12A, dos nodos 1201 y 1202 de ordenador se comunican a través de un canal de comunicación como un Ethernet. Cada nodo ejecuta uno o más programas 1203 y 1218 de aplicación que se comunican transmitiendo paquetes a través del software 1204 y 1217 de comunicación, respectivamente. Ejemplos de programas de aplicación incluyen videoconferencia, correo electrónico, programas de procesamiento de texto, telefonía y similares. El software 1204 y 1217 de comunicación puede comprender, por ejemplo, una arquitectura en capas OSI o “pila” que estandariza varios servicios proporcionados en diferentes niveles de funcionalidad.

Los niveles más bajos de software 1204 y 1217 de comunicación se comunican con los componentes 1206 y 1214 de hardware respectivamente, cada uno de los cuales puede incluir uno o más registros 1207 y 1215 que permiten que el hardware se reconfigure o controle de acuerdo con varios protocolos de comunicación. Los componentes de hardware (una tarjeta de interfaz de red Ethernet, por ejemplo) se comunican entre sí a través del medio de comunicación. Cada componente de hardware suele tener previamente asignada una dirección de hardware fija o un número MAC que identifica el componente de hardware a otros nodos en la red. Uno o más controladores de interfaz controlan el funcionamiento de cada tarjeta y, por ejemplo, se pueden configurar para aceptar o rechazar paquetes de ciertas direcciones de hardware. Como se describirá con más detalle a continuación, varias realizaciones de los principios inventivos proporcionan “saltar” diferentes direcciones utilizando uno o más algoritmos y una o más ventanas móviles que rastrean un rango de direcciones válidas para validar los paquetes recibidos. Los paquetes transmitidos de acuerdo con uno o más de los principios inventivos generalmente se denominarán paquetes “seguros” o “comunicaciones seguras” para diferenciarlos de los paquetes de datos ordinarios que se transmiten no cifrados utilizando direcciones ordinarias relacionadas con la máquina.

Un método sencillo de generar direcciones MAC no atribuibles es una extensión del esquema de salto de IP. En este escenario, dos máquinas en la misma LAN que desean comunicarse de manera segura intercambian generadores y semillas de números aleatorios, y crean secuencias de direcciones MAC cuasialeatorias para el salto sincronizado. Los problemas de implementación y sincronización son similares a los del salto de IP.

Este enfoque, sin embargo, corre el riesgo de utilizar direcciones MAC que están actualmente activas en la LAN, lo que, a su vez, podría interrumpir las comunicaciones para esas máquinas. Dado que una dirección MAC Ethernet tiene actualmente 48 bits de longitud, la posibilidad de mal uso aleatorio de una dirección MAC activa es en realidad bastante pequeña. Sin embargo, si esa cifra se multiplica por una gran cantidad de nodos (como se encontraría en una LAN extensa), por una gran cantidad de cuadros (como podría ser el caso con paquetes de voz o transmisión de video), y por una gran cantidad de Redes Privadas Virtuales (VPN) concurrentes, entonces la posibilidad de que la dirección MAC de una máquina no segura pueda usarse en un marco de salto de dirección puede volverse no trivial. En resumen, cualquier esquema que corra incluso un pequeño riesgo de interrumpir las comunicaciones para otras máquinas en la

LAN seguramente recibirá resistencia de los posibles administradores del sistema. Sin embargo, es técnicamente factible y puede implementarse sin riesgo en una LAN en la que hay un pequeño número de máquinas, o si todas las máquinas en la LAN están involucradas en comunicaciones con salto de MAC.

El salto de la dirección MAC sincronizada puede generar una sobrecarga en el curso del establecimiento de la sesión, especialmente si hay múltiples sesiones o múltiples nodos involucrados en las comunicaciones. Un método más simple de aleatorizar direcciones MAC es permitir que cada nodo reciba y procese cada trama incidente en la red. Por lo general, cada controlador de interfaz de red verificará la dirección MAC de destino en el encabezado de cada trama incidente para ver si coincide con la dirección MAC de esa máquina; Si no hay coincidencia, la trama se descarta. Sin embargo, en una realización, estas comprobaciones pueden deshabilitarse, y cada paquete incidente se pasa a la pila TARP para su procesamiento. Esto se denominará modo “promiscuo”, ya que se procesa cada trama incidente. El modo promiscuo permite al remitente utilizar direcciones MAC completamente aleatorias y no sincronizadas, ya que la máquina de destino está garantizada para procesar la trama. La decisión sobre si el paquete realmente estaba destinado a esa máquina es manejado por la pila TARP, que verifica las direcciones IP de origen y destino para encontrar una coincidencia en sus tablas de sincronización IP. Si no se encuentra ninguna coincidencia, el paquete se descarta; si hay una coincidencia, el paquete se desenvuelve, se evalúa el encabezado interno y si el encabezado interno indica que el paquete está destinado a esa máquina, el paquete se reenvía a la pila IP; de lo contrario, se descarta.

Una desventaja del salto de direcciones MAC puramente aleatorio es su impacto en la sobrecarga de procesamiento; es decir, dado que cada trama incidente debe procesarse, la CPU de la máquina se activa con mucha más frecuencia que si el controlador de interfaz de red discrimina y rechaza los paquetes unilateralmente. Un enfoque de compromiso es seleccionar una sola dirección MAC fija o un pequeño número de direcciones MAC (por ejemplo, una para cada red privada virtual en una Ethernet) para utilizar en las comunicaciones con salto MAC, independientemente del receptor real para el que se envía el mensaje destinado a. En este modo, el controlador de interfaz de red puede verificar cada trama incidente con una (o algunas) direcciones MAC preestablecidas, liberando así a la CPU de la tarea de discriminación de paquetes de capa física. Este esquema no revela ninguna información útil a un intruso en la LAN; en particular, cada paquete seguro ya puede identificarse por un tipo de paquete único en el encabezado externo. Sin embargo, dado que todas las máquinas involucradas en comunicaciones seguras estarían utilizando la misma dirección MAC, o estarían seleccionando de un pequeño grupo de direcciones MAC predeterminadas, la asociación entre una máquina específica y una dirección MAC específica se rompe efectivamente.

En este esquema, la CPU se activará más a menudo de lo que lo haría en comunicaciones no seguras (o en saltos de direcciones MAC sincronizadas), ya que el controlador de interfaz de red no siempre puede discriminar unilateralmente entre paquetes seguros destinados a esa máquina y paquetes seguros de otras VPN. Sin embargo, el tráfico no seguro se elimina fácilmente en la interfaz de red, lo que reduce la cantidad de procesamiento requerido por la CPU. Existen condiciones límite en las que estas declaraciones no se cumplirían, por supuesto, por ejemplo, si todo el tráfico en la LAN es tráfico seguro, entonces la CPU estaría activada en el mismo grado que en el caso de salto de dirección puramente aleatorio; alternativamente, si cada VPN en la LAN usa una dirección MAC diferente, entonces la interfaz de red puede discriminar perfectamente las tramas seguras destinadas a la máquina local de aquellas que constituyen otras VPN. Estas son compensaciones de ingeniería que podrían manejarse mejor al proporcionar opciones administrativas para los usuarios al instalar el software y/o establecer VPN.

Sin embargo, incluso en este escenario, sigue existiendo un ligero riesgo de seleccionar direcciones MAC que están siendo utilizadas por uno o más nodos en la LAN. Una solución a este problema es asignar formalmente una dirección o un rango de direcciones para utilizar en comunicaciones con salto de MAC. Esto normalmente se realiza a través de una autoridad de registro de números asignada; por ejemplo, en el caso de Ethernet, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) asigna rangos de direcciones MAC a los proveedores. Un rango de direcciones asignado formalmente garantizaría que las tramas seguras no entren en conflicto con ninguna máquina configurada correctamente y que funcione correctamente en la LAN.

Ahora se hará referencia a las Figs. 12A y 12B para describir las muchas combinaciones y características que siguen los principios inventivos. Como se explicó anteriormente, se supone que dos nodos 1201 y 1202 de ordenador se comunican a través de una red o medio de comunicación como un Ethernet. Un protocolo de comunicación en cada nodo (1204 y 1217, respectivamente) contiene un elemento 1205 y 1216 modificado que realiza ciertas funciones que se desvían de los protocolos de comunicación estándar. En particular, el nodo 1201 de ordenador implementa un primer algoritmo 1208X de “salto” que selecciona direcciones IP de origen y destino aparentemente aleatorias (y, en una realización, campos discriminadores de encabezado de IP aparentemente aleatorios) para transmitir cada paquete al otro nodo de ordenador. Por ejemplo, el nodo 1201 mantiene una tabla 1208 de transmisión que contiene tripletes de origen (S), destino (D) y campos discriminadores (DS) que se insertan en los encabezados de paquetes IP salientes. La tabla se genera mediante el uso de un algoritmo apropiado (por ejemplo, un generador de números aleatorios que se siembra con una semilla apropiada) que el nodo 1202 receptor conoce. A medida que se forma cada nuevo paquete IP, la siguiente entrada secuencial sale de la tabla 1208 de transmisión del remitente se usa para llenar el campo de origen de IP, destino de IP y extensión de encabezado de IP (por ejemplo, campo discriminador). Se apreciará que la tabla de transmisión no necesita ser creada de antemano, sino que podría crearse sobre la marcha ejecutando el algoritmo cuando se forma cada paquete.

En el nodo 1202 receptor, se mantiene el mismo algoritmo 1222X de salto de IP y se usa para generar una tabla 1222 de recepción que enumera tripletes válidos de dirección IP de origen, dirección IP de destino y campo discriminador. Esto se muestra en virtud de que las primeras cinco entradas de la tabla 1208 de transmisión coinciden con las segundas cinco entradas de la tabla 1222 de recepción. (Las tablas pueden estar ligeramente desplazadas en cualquier momento particular debido a paquetes perdidos, paquetes mal ordenados o retrasos en la transmisión). Además, el nodo 1202 mantiene una ventana de recepción W3 que representa una lista de campos de origen, destino de IP y discriminador de IP válidos que se aceptarán cuando se reciban como parte de un paquete de IP entrante. A medida que se reciben los paquetes, la ventana W3 se desliza hacia abajo en la lista de entradas válidas, de modo que las posibles entradas válidas cambian con el tiempo. Se aceptarán dos paquetes que lleguen fuera de orden pero que coincidan con las entradas dentro de la ventana W3; los que caigan fuera de la ventana W3 serán rechazados como inválidos. La longitud de la ventana W3 se puede ajustar según sea necesario para reflejar los retrasos de la red u otros factores.

El nodo 1202 mantiene una tabla 1221 de transmisión similar para crear paquetes IP y tramas destinadas al nodo 1201 utilizando un algoritmo 1221X de salto potencialmente diferente, y el nodo 1201 mantiene una tabla 1209 de recepción coincidente utilizando el mismo algoritmo 1209X. A medida que el nodo 1202 transmite paquetes al nodo 1201 utilizando campos de fuente IP, destino IP y/o discriminador aparentemente aleatorios, el nodo 1201 hace coincidir los valores de los paquetes entrantes con los que caen dentro de la ventana W1 mantenida en su tabla de recepción. En efecto, la tabla 1208 de transmisión del nodo 1201 está sincronizada (es decir, las entradas se seleccionan en el mismo orden) para recibir la tabla 1222 del nodo 1202 de recepción. De manera similar, la tabla 1221 de transmisión del nodo 1202 está sincronizada para recibir la tabla 1209 del nodo 1201. Se apreciará que, aunque se muestra un algoritmo común para los campos fuente, destino y discriminador en la figura 12A (usando, por ejemplo, una semilla diferente para cada uno de los tres campos), un algoritmo completamente diferente podría de hecho usarse para establecer valores para cada uno de estos campos. También se apreciará que uno o dos de los campos se pueden “saltar” en lugar de los tres como se ilustra.

De acuerdo con otro aspecto de la invención, se saltan direcciones de hardware o “MAC” en lugar de o además de las direcciones IP y/o el campo discriminador para mejorar la seguridad en un área local o red de tipo difusión. Con ese fin, el nodo 1201 mantiene además una tabla 1210 de transmisión utilizando un algoritmo 1210X de transmisión para generar direcciones de hardware de origen y destino que se insertan en encabezados de trama (por ejemplo, campos 1101A y 1101B en la figura 11) que se sincronizan con una tabla 1224 de recepción correspondiente en el nodo 1202. Del mismo modo, el nodo 1202 mantiene una tabla 1223 de transmisión diferente que contiene las direcciones de hardware de origen y destino que se sincroniza con una tabla 1211 de recepción correspondiente en el nodo 1201. De esta manera, las tramas de hardware salientes parecen originarse y llegar a completamente nodos aleatorios en la red, a pesar de que cada receptor puede determinar si un paquete determinado está destinado o no. Se apreciará que la función de salto de hardware se puede implementar en un nivel diferente en el protocolo de comunicaciones que la función de salto de IP (por ejemplo, en un controlador de tarjeta o en una tarjeta de hardware para mejorar el rendimiento).

La figura 12B muestra tres realizaciones o modos diferentes que pueden emplearse utilizando los principios mencionados anteriormente. En un primer modo denominado modo “promiscuo”, todos los nodos de la red utilizan una dirección de hardware común (por ejemplo, una dirección fija para el origen y otra para el destino) o una dirección de hardware completamente aleatoria, de modo que un paquete particular no se puede atribuir a ningún nodo. Cada nodo debe aceptar inicialmente todos los paquetes que contienen la dirección de hardware común (o aleatoria) e inspeccionar las direcciones IP o el campo discriminador para determinar si el paquete está destinado a ese nodo. A este respecto, las direcciones IP o el campo discriminador o ambos pueden variarse de acuerdo con un algoritmo como se describió anteriormente. Como se explicó anteriormente, esto puede aumentar la sobrecarga de cada nodo ya que se trata de un procesamiento adicional para determinar si un paquete dado tiene direcciones de hardware de origen y destino válidas.

En un segundo modo denominado modo “promiscuo por VPN”, se usa un pequeño conjunto de direcciones de hardware fijas, con una dirección de hardware de origen/destino fija utilizada para todos los nodos que se comunican a través de una red privada virtual. Por ejemplo, si hay seis nodos en una Ethernet, y la red se dividirá en dos redes virtuales privadas de modo que los nodos en una VPN puedan comunicarse solo con los otros dos nodos en su propia VPN, entonces dos conjuntos de direcciones de hardware podría usarse: un conjunto para la primera VPN y un segundo conjunto para la segunda VPN. Esto reduciría la cantidad de sobrecarga involucrada en la verificación de tramas válidas, ya que solo los paquetes que llegan desde la VPN designada necesitarían ser verificados. Las direcciones IP y uno o más campos discriminadores aún podrían saltarse como antes para una comunicación segura dentro de la VPN. Por supuesto, esta solución compromete el anonimato de las VPN (es decir, un extraño puede decir fácilmente qué tráfico pertenece a qué VPN, aunque no puede correlacionarlo con una máquina/persona específica). También requiere el uso de un campo discriminador para mitigar la vulnerabilidad a ciertos tipos de ataques DoS. (Por ejemplo, sin el campo discriminador, un atacante en la LAN podría transmitir tramas que contengan las direcciones MAC que usa la VPN; rechazar esas tramas podría generar una sobrecarga de procesamiento excesiva. El campo discriminador proporcionaría un medio de baja sobrecarga para rechazar los paquetes falsos).

En un tercer modo denominado modo de “salto de hardware”, las direcciones de hardware varían como se ilustra en la figura 12A, de modo que las direcciones de origen y destino de hardware se cambian constantemente para proporcionar un direccionamiento no atribuible. Por supuesto, las variaciones en estas realizaciones son posibles, y la invención no pretende estar limitada en ningún aspecto por estos ejemplos ilustrativos.

B. Ampliar el espacio de direcciones

El salto de direcciones proporciona seguridad y privacidad. Sin embargo, el nivel de protección está limitado por el número de direcciones en los bloques que se saltan. Un bloque de salto denota un campo o campos modulados en forma de paquete con el fin de proporcionar una VPN. Por ejemplo, si dos nodos se comunican con el salto de direcciones IP utilizando bloques de salto de 4 direcciones (2 bits) cada uno, habría 16 combinaciones posibles de pares de direcciones. Una ventana de tamaño 16 daría como resultado que la mayoría de los pares de direcciones sean aceptados como válidos la mayor parte del tiempo. Esta limitación se puede superar mediante el uso de un campo discriminador además de o en lugar de los campos de dirección saltados. El campo discriminador se saltaría exactamente de la misma manera que los campos de dirección y se usaría para determinar si un paquete debe ser procesado por un receptor.

Suponga que dos clientes, cada uno con bloques de salto de cuatro bits, desearían el mismo nivel de protección brindado a los clientes que se comunican mediante salto de IP entre dos bloques A (24 bits de dirección elegibles para salto). Un campo discriminador de 20 bits, utilizado junto con los 4 bits de dirección elegibles para saltar en el campo de dirección IP, proporciona este nivel de protección. Un campo discriminador de 24 bits proporcionaría un nivel de protección similar si los campos de dirección no se saltaran o ignoraran. El uso de un campo discriminador ofrece las siguientes ventajas: (1) se puede proporcionar un nivel arbitrariamente alto de protección, y (2) el salto de direcciones es innecesario para proporcionar protección. Esto puede ser importante en entornos donde el salto de direcciones causaría problemas de enrutamiento.

C. Técnicas de sincronización

En general, se supone que una vez que un nodo remitente y un nodo receptor hayan intercambiado algoritmos y semillas (o información similar suficiente para generar tablas de origen y destino cuasialeatorias), la comunicación posterior entre los dos nodos se realizará sin problemas. Sin embargo, de manera realista, dos nodos pueden perder la sincronización debido a demoras o interrupciones de la red u otros problemas. En consecuencia, es deseable proporcionar medios para restablecer la sincronización entre nodos en una red que ha perdido la sincronización.

Una técnica posible es exigir que cada nodo proporcione una confirmación tras la recepción exitosa de cada paquete y, si no se recibe confirmación dentro de un cierto período de tiempo, volver a enviar el paquete no acreditado. Sin embargo, este enfoque aumenta los costes generales y puede ser prohibitivo en entornos de alto rendimiento, como la transmisión de video o audio, por ejemplo.

Un enfoque diferente es emplear una técnica de sincronización automática a la que se hará referencia en este documento como “auto-sincronización”. En este enfoque, la información de sincronización se incrusta en cada paquete, lo que permite al receptor volver a sincronizarse al recibir un solo paquete si determina que ha perdido la sincronización con el remitente. (Si las comunicaciones ya están en progreso, y el receptor determina que todavía está sincronizado con el remitente, entonces no hay necesidad de volver a sincronizar). Un receptor podría detectar que no estaba sincronizado, por ejemplo, empleando un temporizador “hombre muerto” que expira después de un cierto período de tiempo, en el que el temporizador se restablece con cada paquete válido. Una marca de tiempo podría agregarse al campo de sincronización pública (ver más abajo) para evitar ataques de reintento de paquetes.

En una realización, se agrega un “campo de sincronización” al encabezado de cada paquete enviado por el remitente. Este campo de sincronización podría aparecer no cifrado o como parte de una parte cifrada del paquete. Suponiendo que un remitente y un receptor hayan seleccionado un generador de números aleatorios (RNG) y un valor inicial, esta combinación de RNG e inicial puede usarse para generar una secuencia de números aleatorios (RNS). El RNS se usa para generar una secuencia de pares de IP de origen/destino (y, si se desea, campos discriminadores y direcciones de origen y destino de hardware), como se describió anteriormente. Sin embargo, no es necesario generar la secuencia completa (o los primeros valores N-1) para generar el enésimo número aleatorio en la secuencia; si se conoce el índice de secuencia N, el valor aleatorio correspondiente a ese índice puede generarse directamente (ver más abajo). Se podrían utilizar diferentes RNG (y semillas) con diferentes períodos fundamentales para generar las secuencias de IP de origen y destino, pero los conceptos básicos seguirían siendo válidos. En aras de la simplicidad, la siguiente discusión supondrá que los pares de direcciones de origen y destino IP (solo) se saltan utilizando un único mecanismo de secuencia RNG.

De acuerdo con una característica de “auto-sincronización”, un campo de sincronización en cada encabezado de paquete proporciona un índice (es decir, un número de secuencia) en el RNS que se está utilizando para generar pares de IP. Al conectar este índice en el RNG que se está utilizando para generar el RNS, se obtiene un valor de número aleatorio específico, que a su vez produce un par de IP específico. Es decir, se puede generar un par de IP directamente a partir del conocimiento del RNG, la semilla y el número de índice; no es necesario, en este esquema,

generar la secuencia completa de números aleatorios que preceden al valor de secuencia asociado con el número de índice proporcionado.

Como los comunicantes presumiblemente han intercambiado previamente RNG y semillas, la única información nueva que debe proporcionarse para generar un par de IP es el número de secuencia. Si el remitente proporciona este número en el encabezado del paquete, entonces el receptor solo necesita enchufar este número en el RNG para generar un par de IP, y así verificar que el par de IP que aparece en el encabezado del paquete sea válido. En este esquema, si el remitente y el receptor pierden la sincronización, el receptor puede volver a sincronizarse inmediatamente al recibir un solo paquete simplemente comparando el par IP en el encabezado del paquete con el par IP generado a partir del número de índice. Por lo tanto, las comunicaciones sincronizadas se pueden reanudar al recibir un solo paquete, lo que hace que este esquema sea ideal para las comunicaciones de multidifusión. Llevado al extremo, podría obviar la necesidad de tablas de sincronización por completo; es decir, el remitente y el receptor podrían simplemente confiar en el número de índice en el campo de sincronización para validar el par de IP en cada paquete y, por lo tanto, eliminar las tablas por completo.

El esquema antes mencionado puede tener algunos problemas de seguridad inherentes asociados a él, a saber, la colocación del campo de sincronización. Si el campo se coloca en el encabezado externo, un intruso podría observar los valores del campo y su relación con la secuencia de IP. Esto podría comprometer el algoritmo que se utiliza para generar la secuencia de direcciones IP, lo que comprometería la seguridad de las comunicaciones. Sin embargo, si el valor se coloca en el encabezado interno, entonces el remitente debe descifrar el encabezado interno antes de poder extraer el valor de sincronización y validar el par de IP; esto abre el receptor a ciertos tipos de ataques de denegación de servicio (DoS), como la reproducción de paquetes. Es decir, si el receptor debe descifrar un paquete antes de que pueda validar el par de IP, entonces podría verse obligado a gastar una cantidad significativa de procesamiento en el descifrado si un atacante simplemente retransmite paquetes previamente válidos. Otras metodologías de ataque son posibles en este escenario.

Un posible compromiso entre la seguridad del algoritmo y la velocidad de procesamiento es dividir el valor de sincronización entre un encabezado interno (cifrado) y externo (no cifrada). Es decir, si el valor de sincronización es lo suficientemente largo, podría dividirse en una parte que cambia rápidamente y que se puede ver en claro, y una parte fija (o que cambia muy lentamente) que debe protegerse. La parte que se puede ver no cifrada se denominará parte de “sincronización pública” y la parte que debe protegerse se denominará parte de “sincronización privada”.

Se necesitan ambas partes, la sincronización pública y la sincronización privada, para generar el valor de sincronización completo. Sin embargo, la parte privada se puede seleccionar de modo que sea fija o cambie solo ocasionalmente. Por lo tanto, el receptor puede almacenar el valor de sincronización privada, evitando así la necesidad de descifrar el encabezado para recuperarlo. Si el remitente y el receptor han acordado previamente la frecuencia con la que cambiará la parte privada de la sincronización, entonces el receptor puede descifrar selectivamente un solo encabezado para extraer la nueva sincronización privada si la brecha de comunicación que ha llevado a la sincronización perdida excedió la vida útil de la sincronización privada anterior. Esto no debería representar una cantidad pesada de descifrado y, por lo tanto, no debería abrir el receptor al ataque de denegación de servicio simplemente en función de la necesidad de descifrar ocasionalmente un solo encabezado.

Una implementación de esto es utilizar una función de hashing con un mapeo uno a uno para generar las porciones de sincronización privadas y públicas a partir del valor de sincronización. Esta implementación se muestra en la figura 13, donde (por ejemplo) un primer ISP 1302 es el remitente y un segundo ISP 1303 es el receptor. (Otras alternativas son posibles a partir de la figura 13.) Un paquete transmitido comprende un encabezado 1305 público o “externo” que no está cifrado, y un encabezado 1306 privado o “interno” que está cifrado usando, por ejemplo, una clave de enlace. El encabezado 1305 externo incluye una porción de sincronización pública mientras que el encabezado interno 1306 contiene la porción de sincronización privada. Un nodo receptor descifra el encabezado interno utilizando una función 1307 de descifrado para extraer la parte de sincronización privada. Este paso es necesario solo si la vida útil de la sincronización privada almacenada en el búfer ha expirado. (Si la sincronización privada actualmente almacenada en el búfer sigue siendo válida, simplemente se extrae de la memoria y se “agrega” (que podría ser un hash inverso) a la sincronización pública, como se muestra en el paso 1308.) Las partes de sincronización pública y descifrada privada se combinan en la función 1308 para generar la sincronización combinada 1309. La sincronización combinada (1309) se alimenta al RNG (1310) y se compara con el par de direcciones IP (1311) para validar o rechazar el paquete.

Una consideración importante en esta arquitectura es el concepto de “futuro” y “pasado” en lo que respecta a los valores de sincronización pública. Aunque los valores de sincronización, en sí mismos, deben ser aleatorios para evitar ataques de suplantación de identidad, puede ser importante que el receptor pueda identificar rápidamente un valor de sincronización que ya se haya enviado, incluso si el paquete que contiene ese valor de sincronización nunca fue realmente recibido por el receptor. Una solución es introducir un número de marca de tiempo o secuencia en la parte de sincronización pública, que podría extraerse, verificarse y descartarse rápidamente, validando así la parte de sincronización pública.

En una realización, los paquetes se pueden verificar comparando el par de IP de origen/destino generado por el campo de sincronización con el par que aparece en el encabezado del paquete. Si (1) coinciden, (2) la marca de tiempo es

válida y (3) el temporizador de hombre muerto ha expirado, se produce una resincronización; de lo contrario, el paquete es rechazado. Si hay suficiente potencia de procesamiento disponible, el temporizador de hombre muerto y las tablas de sincronización se pueden evitar por completo, y el receptor simplemente volvería a sincronizar (por ejemplo, validar) en cada paquete.

El esquema anterior puede requerir matemática de enteros grandes (por ejemplo, 160 bits), lo que puede afectar su implementación. Sin estos registros de enteros grandes, el rendimiento del procesamiento se vería afectado, lo que podría afectar la seguridad desde el punto de vista de la denegación de servicio. Sin embargo, a medida que las funciones de procesamiento matemático de enteros grandes se vuelvan más frecuentes, los costes de implementar dicha función se reducirán.

D. Otros esquemas de sincronización

Como se explicó anteriormente, si se pierden W o más paquetes consecutivos entre un transmisor y un receptor en una VPN (donde W es el tamaño de la ventana), la ventana del receptor no se habrá actualizado y el transmisor estará transmitiendo paquetes que no están en la ventana del receptor. El transmisor y el receptor no recuperarán la sincronización hasta que quizás los pares aleatorios en la ventana se repitan por casualidad. Por lo tanto, es necesario mantener un remitente y un receptor sincronizados siempre que sea posible y restablecer la sincronización siempre que se pierda.

Se puede utilizar un esquema de “punto de control” para recuperar la sincronización entre un remitente y un receptor que se ha quedado sin sincronización. En este esquema, se utiliza un mensaje de punto de control que comprende un par de direcciones IP aleatorias para comunicar información de sincronización. En una realización, se usan dos mensajes para comunicar información de sincronización entre un remitente y un receptor:

1. SYNC_REQ es un mensaje utilizado por el remitente para indicar que desea sincronizar; y

2. SYNC_ACK es un mensaje utilizado por el receptor para informar al transmisor que se ha sincronizado.

De acuerdo con una variación de este enfoque, tanto el transmisor como el receptor mantienen tres puntos de control (ver FIG. 14):

1. En el transmisor, ckpt_o (“punto de control antiguo”) es el par IP que se utilizó para reenviar el último paquete SYNC_REQ al receptor. En el receptor, ckpt_o (“punto de control antiguo”) es el par de IP que recibe paquetes SYNC_REQ repetidos del transmisor.

2. En el transmisor, ckpt_n (“punto de control nuevo”) es el par de IP que se utilizará para enviar el siguiente paquete SYNC_REQ al receptor. En el receptor, ckpt_n (“punto de control nuevo”) es el par de IP que recibe un nuevo paquete SYNC_REQ del transmisor y que hace que la ventana del receptor se vuelva a alinear, ckpt_o se establezca en ckpt_n, se genere un nuevo ckpt_n y se generará un nuevo ckpt_r.

3. En el transmisor, ckpt_r es el par de IP que se utilizará para enviar el siguiente paquete SYNC_ACK al receptor. En el receptor, ckpt_r es el par de IP que recibe un nuevo paquete SYNC_ACK del transmisor y que genera un nuevo ckpt_n. Dado que SYNC_ACK se transmite desde el ISP del receptor al ISP del transmisor, el remitente ckpt_r se refiere a la ckpt_r del receptor y el receptor ckpt_r se refiere a la ckpt_r del transmisor (véase la figura 14).

Cuando un transmisor inicia la sincronización, el par de IP que usará para transmitir el siguiente paquete de datos se establece en un valor predeterminado y cuando un receptor recibe por primera vez un SYNC_REQ, la ventana del receptor se actualiza para centrarse en el próximo par de IP del transmisor. Este es el mecanismo principal para la sincronización del punto de control.

La sincronización puede iniciarse mediante un contador de paquetes (por ejemplo, después de cada N paquetes transmitidos, iniciar una sincronización) o mediante un temporizador (cada S segundos, iniciar una sincronización) o una combinación de ambos. Ver FIG. 15. Desde la perspectiva del transmisor, esta técnica funciona de la siguiente manera: (1) Cada transmisor transmite periódicamente un mensaje de “solicitud de sincronización” al receptor para asegurarse de que esté sincronizado. (2) Si el receptor todavía está sincronizado, envía un mensaje de “confirmación de sincronización”. (Si esto funciona, no es necesaria ninguna otra acción). (3) Si no se ha recibido una “confirmación de sincronización” dentro de un período de tiempo, el transmisor retransmite la solicitud de sincronización nuevamente. Si el transmisor alcanza el siguiente punto de control sin recibir una respuesta de “confirmación de sincronización”, la sincronización se interrumpe y el transmisor debe dejar de transmitir. El transmisor continuará enviando sync_reqs hasta que reciba un sync_ack, momento en el cual se restablecerá la transmisión.

Desde la perspectiva del receptor, el esquema funciona de la siguiente manera: (1) cuando recibe una solicitud de “solicitud de sincronización” del transmisor, avanza su ventana a la siguiente posición de punto de control (incluso omitiendo pares si es necesario), y envía un mensaje “sync ack” al transmisor. Si la sincronización nunca se perdió,

entonces el “avance” realmente avanza al siguiente par de direcciones disponibles en la tabla (es decir, avance normal).

Si un intruso intercepta los mensajes de “solicitud de sincronización” e intenta interferir con la comunicación enviando mensajes nuevos, se ignorará si se ha establecido la sincronización o realmente ayudará a restablecer la sincronización.

Una ventana se vuelve a alinear cada vez que se produce una resincronización. Esta realineación implica actualizar la ventana del receptor para abarcar los pares de direcciones utilizados por el paquete transmitido inmediatamente después de la transmisión del paquete SYNC_REQ. Normalmente, el transmisor y el receptor están sincronizados entre sí. Sin embargo, cuando ocurren eventos de red, la ventana del receptor puede tener que avanzar muchos pasos durante la resincronización. En este caso, es deseable avanzar la ventana sin tener que pasar secuencialmente por los números aleatorios que intervienen. (Esta característica también es deseable para el enfoque de sincronización automática discutido anteriormente).

E. Generador de números aleatorios con capacidad de avance

Un método atractivo para generar direcciones saltadas aleatoriamente es utilizar generadores de números aleatorios idénticos en el transmisor y el receptor y avanzarlos a medida que se transmiten y reciben paquetes. Hay muchos algoritmos de generación de números aleatorios que podrían usarse. Cada uno tiene fortalezas y debilidades para las aplicaciones de salto de direcciones.

Los generadores de números aleatorios congruenciales lineales (LCR) son generadores de números aleatorios rápidos, simples y bien caracterizados que se pueden hacer para saltar adelante n pasos de manera eficiente. Un LCR genera números aleatorios X_1, X_2, X_3 ... X_k comenzando con la semilla X_0 utilizando una recurrencia

$$X_i=(aX_{i-1} + b) \bmod c, \qquad (1)$$

donde a, b y c definen un LCR particular. Otra expresión para X_i,

$$X_i=((a^i(X_0+b)-b)/(a-1)) \bmod c \qquad (2)$$

habilita la capacidad de avance. El factor a' puede crecer mucho incluso para personas modestas si no se libera. Por lo tanto, algunas propiedades especiales de la operación de módulo se pueden utilizar para controlar el tamaño y el tiempo de procesamiento requerido para calcular (2). (2) se puede reescribir como:

$$X_i=(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c. \qquad (3)$$

Se puede demostrar que:

$$(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c =$$

$$((a^i \bmod((a-1)c)(X_0(a-1)+b)-b)/(a-1)) \bmod c \quad (4).$$

$(X_0(a-1)+b)$ se puede almacenar como $(X_0(a-1)+b) \bmod c$, b como b mod c y calcular a' mod ((a-1)c) (esto requiere pasos 0(log(i))).

Una implementación práctica de este algoritmo saltaría una distancia fija, n, entre sincronizaciones; Esto equivale a sincronizar cada n paquetes. La ventana comenzaría n pares de IP desde el inicio de la ventana anterior. Usando X_j^{w2}, el número aleatorio en el j-ésimo punto de control, como X_0 y n como i, un nodo puede almacenar a^n mode(a-1) c) una vez por LCR y establece

$$X_{j+1}^w=X_{n(j+1)}=((a^n \bmod((a-1)c) (X_j^w(a-1)+b)-b)/(a-1))\bmod c, \qquad (5)$$

para generar el número aleatorio para el $j+1^{ra}$ sincronización. Utilizando esta construcción, un nodo podría saltar una distancia arbitraria (pero fija) entre sincronizaciones en una cantidad de tiempo constante (independiente de n).

Los generadores de números pseudoaleatorios, en general, y los LCR, en particular, eventualmente repetirán sus ciclos. Esta repetición puede presentar vulnerabilidad en el esquema de salto de IP. Un adversario simplemente tendría que esperar una repetición para predecir secuencias futuras. Una forma de hacer frente a esta vulnerabilidad es crear un generador de números aleatorios con un ciclo largo conocido. Una secuencia aleatoria puede ser reemplazada por un nuevo generador de números aleatorios antes de que se repita. Los LCR se pueden construir con ciclos largos conocidos. Esto no es cierto actualmente para muchos generadores de números aleatorios.

Los generadores de números aleatorios pueden ser criptográficamente inseguros. Un adversario puede derivar los parámetros RNG al examinar la salida o parte de la salida. Esto es cierto para los LCG. Esta vulnerabilidad puede

mitigarse incorporando un cifrador, diseñado para codificar la salida como parte del generador de números aleatorios. El generador de números aleatorios evita que un adversario realice un ataque, por ejemplo, un ataque de texto sin formato conocido, contra el cifrador.

F. Ejemplo de generador de números aleatorios

Considere un RNG donde a = 31, b=4 y c=15. Para este caso, la ecuación (1) se convierte en:

$$X_i=(31\ X_{i-1} + 4)\text{mod } 15. \quad (6)$$

Si uno establece X_0=1, la ecuación (6) producirá la secuencia 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. Esto secuencia se repetirá indefinidamente. Para un salto por delante de 3 números en esta secuencia a"= 31^3= 29791, c∗(a-1) = 15∗30 = 450 y a^n mod ((a-1) c) = 31^3 mod (15∗30) = 29791mod(450) = 91. La ecuación (5) se convierte en:

$$((91\ X_i30+4)-4)/30)\text{mod } 15 \quad (7).$$

La Tabla 1 muestra los cálculos de salto hacia adelante de (7). Los cálculos comienzan en 5 y saltan 3.

Tabla 1

I	X_i	(X_i30+4)	91 (X_i30+4)-4	((91 (X_i30+4)-4)/30	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

G. Filtro de paquetes rápido

Las VPN de salto de dirección deben determinar rápidamente si un paquete tiene un encabezado válido y, por lo tanto, requiere un procesamiento adicional o si tiene un encabezado no válido (un paquete hostil) y debe rechazarse de inmediato. Dichas determinaciones rápidas se denominarán "filtrado rápido de paquetes". Esta capacidad protege la VPN de los ataques de un adversario que transmite paquetes hostiles en el receptor a una velocidad alta con la esperanza de saturar el procesador del receptor (un llamado ataque de "denegación de servicio"). El filtrado rápido de paquetes es una característica importante para implementar VPN en medios compartidos como Ethernet.

Suponiendo que todos los participantes en una VPN compartan un bloque de direcciones "A" no asignado, una posibilidad es utilizar un bloque experimental "A" que nunca se asignará a ninguna máquina que no sea salto de direcciones en el medio compartido. Los bloques "A" tienen una dirección de 24 bits que se puede saltar en lugar de los 8 bits en los bloques "C". En este caso, un bloque de salto será el bloque "A". El uso del bloque experimental "A" es una opción probable en un Ethernet porque:

1. Las direcciones no tienen validez fuera de Ethernet y no serán enrutadas a un destino externo válido por una puerta de enlace.

2. Hay 2^{24} (~16 millones) direcciones que se pueden saltar dentro de cada bloque "A". Esto produce >280 billones de pares de direcciones posibles, por lo que es muy poco probable que un adversario adivine una dirección válida. También proporciona una probabilidad aceptablemente baja de colisión entre VPN separadas (todas las VPN en un medio compartido generan independientemente pares de direcciones aleatorias del mismo bloque "A").

3. Los paquetes no serán recibidos por alguien en Ethernet que no esté en una VPN (a menos que la máquina esté en modo promiscuo) minimizando el impacto en los ordenadores que no son VPN.

El ejemplo de Ethernet se usará para describir una implementación del filtrado rápido de paquetes. El algoritmo ideal examinaría rápidamente un encabezado de paquete, determinaría si el paquete es hostil y rechazaría cualquier paquete hostil o determinaría qué par de IP activo coincide con el encabezado del paquete. El problema es un problema clásico de memoria asociativa. Se han desarrollado una variedad de técnicas para resolver este problema (hashing, árbol binario, etc.). Cada uno de estos enfoques tiene sus fortalezas y debilidades. Por ejemplo, se puede hacer que las tablas hash funcionen bastante rápido en un sentido estadístico, pero en ocasiones pueden degenerar en un algoritmo mucho más lento. Esta lentitud puede persistir por un período de tiempo. Dado que existe la necesidad de descartar paquetes hostiles rápidamente en todo momento, el hash sería inaceptable.

H. Algoritmo de vector de presencia

Un vector de presencia es un vector de bits de longitud 2^n que puede indexarse con n números de bits (cada uno de los cuales varía de 0 a 2^n-1). Se puede indicar la presencia de k n números de bit (no necesariamente únicos), estableciendo los bits en el vector de presencia indexados por cada número a 1. De lo contrario, los bits en el vector de presencia son 0. Un número bits n de, x, es uno de los k números si y solo si el $x^{ésimo}$ bit del vector de presencia es 1. Se puede implementar un filtro de paquetes rápido indexando el vector de presencia y buscando un 1, que se denominará “prueba”.

Por ejemplo, supongamos que uno quisiera representar el número 135 utilizando un vector de presencia. Se establecería el bit 135 del vector. En consecuencia, se podría determinar rápidamente si una dirección de 135 era válida al revisar un solo bit: el bit 135. Los vectores de presencia podrían crearse de antemano correspondientes a las entradas de la tabla para las direcciones IP. En efecto, las direcciones entrantes se pueden utilizar como índices en un vector largo, lo que hace que las comparaciones sean muy rápidas. A medida que cada RNG genera una nueva dirección, el vector de presencia se actualiza para reflejar la información. A medida que se mueve la ventana, el vector de presencia se actualiza a cero direcciones que ya no son válidas.

Existe una compensación entre la eficiencia de la prueba y la cantidad de memoria requerida para almacenar los vectores de presencia. Por ejemplo, si uno usara los 48 bits de direcciones de salto como índice, el vector de presencia tendría que ser de 35 terabytes. Claramente, esto es demasiado grande para fines prácticos. En cambio, los 48 bits se pueden dividir en varios campos más pequeños. Por ejemplo, uno podría subdividir los 48 bits en cuatro campos de 12 bits (ver Figura 16). Esto reduce el requisito de almacenamiento a 2048 bytes a expensas de tener que procesar ocasionalmente un paquete hostil. En efecto, en lugar de un vector de presencia larga, las porciones de dirección descompuestas deben coincidir con los cuatro vectores de presencia más cortos antes de que se permita el procesamiento adicional. (Si la primera parte de la porción de dirección no coincide con el primer vector de presencia, no hay necesidad de verificar los tres vectores de presencia restantes).

Un vector presencia tendrá un 1 en el $y^{ésimo}$ bit si y sólo si una o más direcciones con un campo correspondiente de y están activos. Una dirección está activa solo si cada vector de presencia indexado por el subcampo apropiado de la dirección es 1.

Considere una ventana de 32 direcciones activas y 3 puntos de control. Un paquete hostil será rechazado por la indexación de un vector de presencia más del 99% de las veces. Un paquete hostil será rechazado por la indexación de los 4 vectores de presencia más del 99.9999995% de las veces. En promedio, los paquetes hostiles serán rechazados en menos de 1.02 operaciones de índice de vector de presencia.
El pequeño porcentaje de paquetes hostiles que pasan el filtro de paquetes rápido se rechazará cuando no se encuentren pares coincidentes en la ventana activa o sean puntos de control activos. Los paquetes hostiles que coinciden casualmente con un encabezado serán rechazados cuando el software VPN intente descifrar el encabezado. Sin embargo, estos casos serán extremadamente raros. Hay muchas otras formas en que este método se puede configurar para arbitrar las compensaciones de espacio/velocidad.

REIVINDICACIONES

1. Un método para un primer nodo (801) para establecer una sesión con un segundo nodo (811), el método se realiza en el primer nodo (801), en el que el método comprende:

enviar, desde una primera dirección para el primer nodo (801) a una primera dirección para el segundo nodo (811), una solicitud (821) para iniciar la sesión; y

recibir, en la primera dirección para el primer nodo (801) desde la primera dirección para el segundo nodo (811), un primer mensaje (822) de confirmación,

caracterizado porque:

el primer mensaje de confirmación incluye un bloqueo de transmisión que el primer nodo (801) se utiliza para comunicarse con el segundo nodo (811) durante la sesión, en el que el bloque de salto de transmisión comprende: un bloque de direcciones IP; y un algoritmo y una semilla de aleatorización para seleccionar, del bloque de direcciones IP, una dirección de origen y una dirección de destino para utilizar en la transmisión del siguiente mensaje;

y además caracterizado porque el método comprende, además:

durante la sesión, comunicarse con el segundo nodo (811) utilizando una segunda dirección para el primer nodo (801) y una segunda dirección para el segundo nodo (811), en el que las segundas direcciones son especificadas por el bloque de salto de transmisión.

2. El método de la reivindicación 1, en el que el segundo nodo (811) es un enrutador.

3. El método de la reivindicación 1 o la reivindicación 2, en el que el primer nodo (801) es un ordenador de cliente.

4. El método de la reivindicación 1 o la reivindicación 2, en el que el primer nodo (801) es un enrutador.

5. El método de cualquiera de las reivindicaciones anteriores, en el que el método comprende además enviar un segundo mensaje (823) de confirmación desde la primera dirección para el primer nodo (801) a la primera dirección para el segundo nodo (811).

6. El método de la reivindicación 5, que comprende además enviar, desde una tercera dirección para el primer nodo (801) a una tercera dirección para el segundo nodo (811), un tercer mensaje (824) de confirmación para la sesión.

7. El método de cualquiera de las reivindicaciones anteriores, en el que el primer mensaje (822) de confirmación incluye además un bloque de salto de recepción que el primer nodo (801) debe utilizar para comunicarse con el segundo nodo (811) durante la sesión, en el que el bloque de salto de recepción comprende:

un segundo bloque de direcciones IP; y

un algoritmo y una semilla de aleatorización para seleccionar, del segundo bloque de direcciones IP, una dirección de origen y una dirección de destino para utilizar en la recepción del siguiente mensaje.

8. El método de la reivindicación 1, en el que la solicitud para iniciar la sesión incluye una credencial de autenticación para el primer nodo (801).

9. El método de la reivindicación 1, que comprende además recibir el primer mensaje de confirmación del segundo nodo (811) cuando el segundo nodo valida la solicitud para iniciar la sesión.

10. El método de cualquiera de las reivindicaciones anteriores, en el que la sesión es una sesión segura.

11. Un primer nodo configurado para realizar el método de cualquiera de las reivindicaciones anteriores.

12. Un método para un segundo nodo (811) para establecer una sesión con un primer nodo (801), el método se realiza en el segundo nodo (811), en el que el método comprende:

recibir, en una primera dirección para el segundo nodo (811) desde una primera dirección para el primer nodo (801), una solicitud (821) para iniciar la sesión; y

enviar, desde la primera dirección para el segundo nodo (811) a la primera dirección para el primer nodo (801), un primer mensaje (822) de confirmación,

caracterizado porque:

el primer mensaje de confirmación incluye un bloqueo de salto de transmisión que el primer nodo (801) se utiliza para comunicarse con el segundo nodo (811) durante la sesión, en el que el bloque de salto de transmisión comprende: un bloque de direcciones IP; y un algoritmo y una semilla de aleatorización para seleccionar, del bloque de direcciones IP, una dirección de origen y una dirección de destino para utilizar en la transmisión del siguiente mensaje;

y además caracterizado porque el método comprende, además:

durante la sesión, comunicarse con el primer nodo (801) utilizando una segunda dirección para el primer nodo (801) y una segunda dirección para el segundo nodo (811), en el que las segundas direcciones son especificadas por el bloque de salto de transmisión.

13. El método de la reivindicación 12, en el que el primer mensaje (822) de confirmación incluye además un bloque de salto de recepción que el primer nodo (801) debe utilizar para comunicarse con el segundo nodo (811) durante la sesión, en el que el bloque de salto de recepción comprende:

un segundo bloque de direcciones IP; y

un algoritmo y una semilla de aleatorización para seleccionar, del segundo bloque de direcciones IP, una dirección de origen y

una dirección de destino para utilizar en la recepción del siguiente mensaje.

14. Un segundo nodo configurado para realizar el método de la reivindicación 12 o la reivindicación 13.

FIG. 1

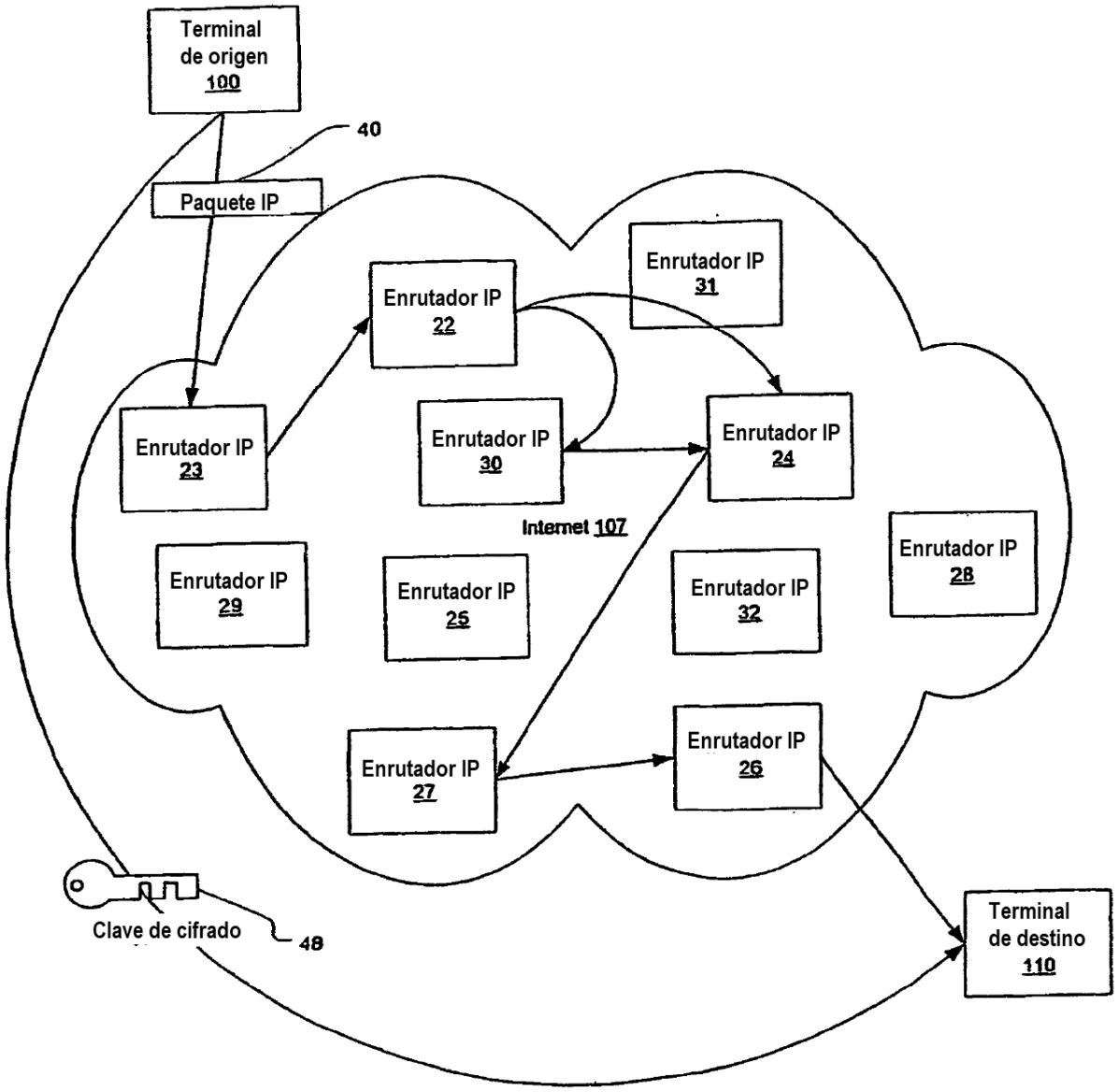

Terminal de origen 100
40
Paquete IP
Enrutador IP 22
Enrutador IP 31
Enrutador IP 23
Enrutador IP 30
Enrutador IP 24
Internet 107
Enrutador IP 29
Enrutador IP 25
Enrutador IP 32
Enrutador IP 28
Enrutador IP 27
Enrutador IP 26
Clave de cifrado
48
Terminal de destino 110

FIG. 2

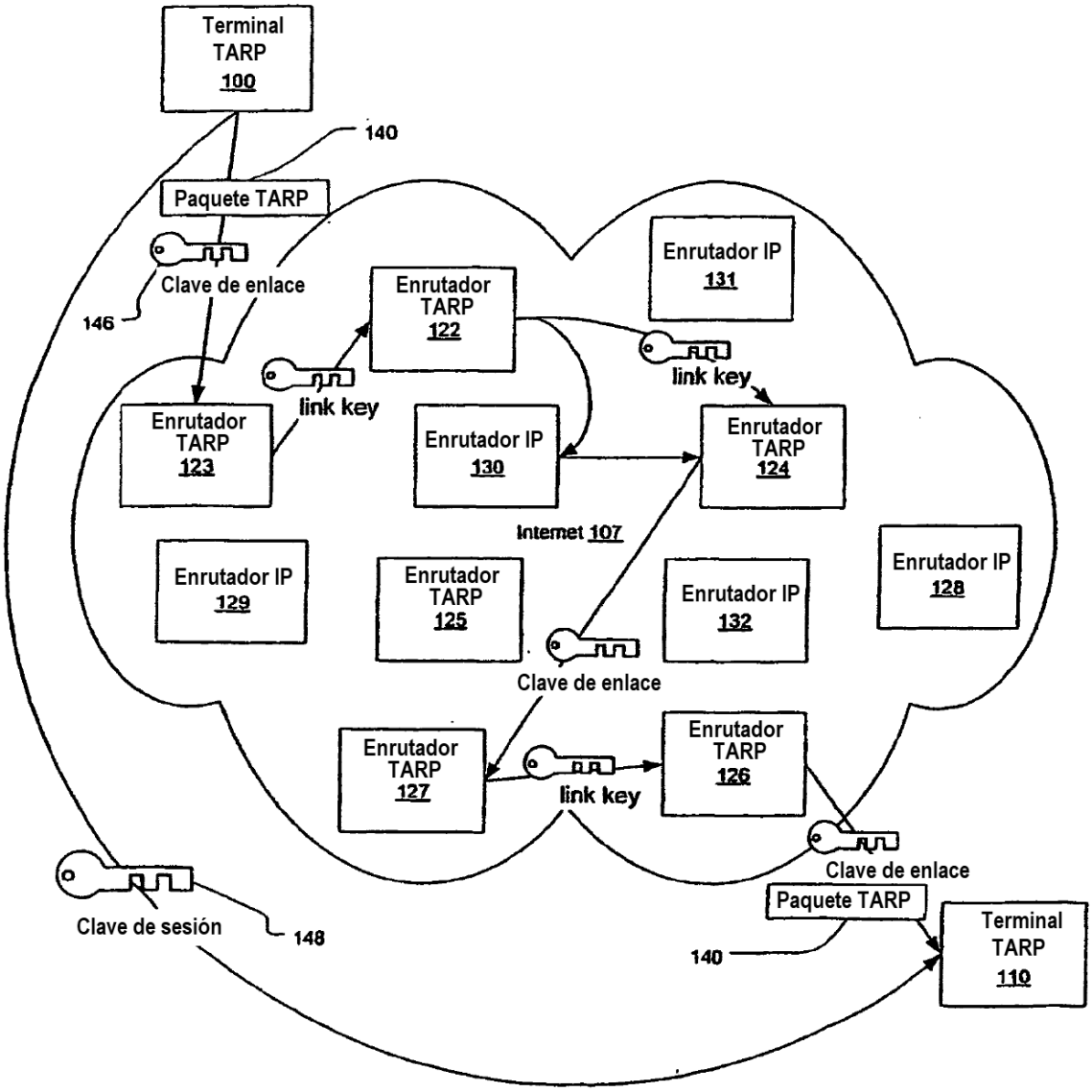

Terminal TARP 100
140
Paquete TARP
Clave de enlace
146
Enrutador TARP 122
Enrutador IP 131
link key
link key
Enrutador TARP 123
Enrutador IP 130
Enrutador TARP 124
Internet 107
Enrutador IP 129
Enrutador TARP 125
Enrutador IP 132
Enrutador IP 128
Clave de enlace
Enrutador TARP 127
link key
Enrutador TARP 126
Clave de enlace
Clave de sesión
148
Paquete TARP
140
Terminal TARP 110

FIG. 3a

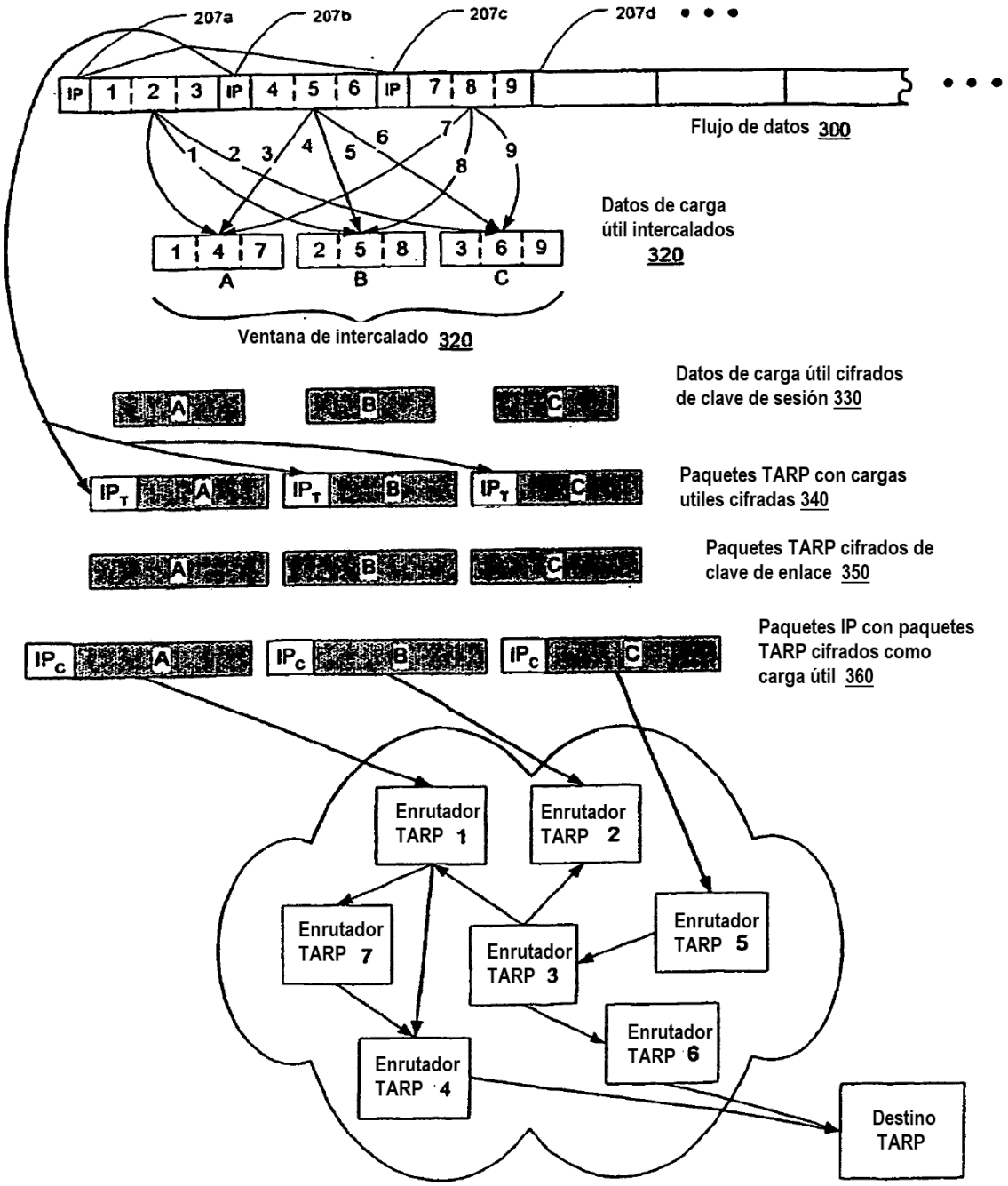
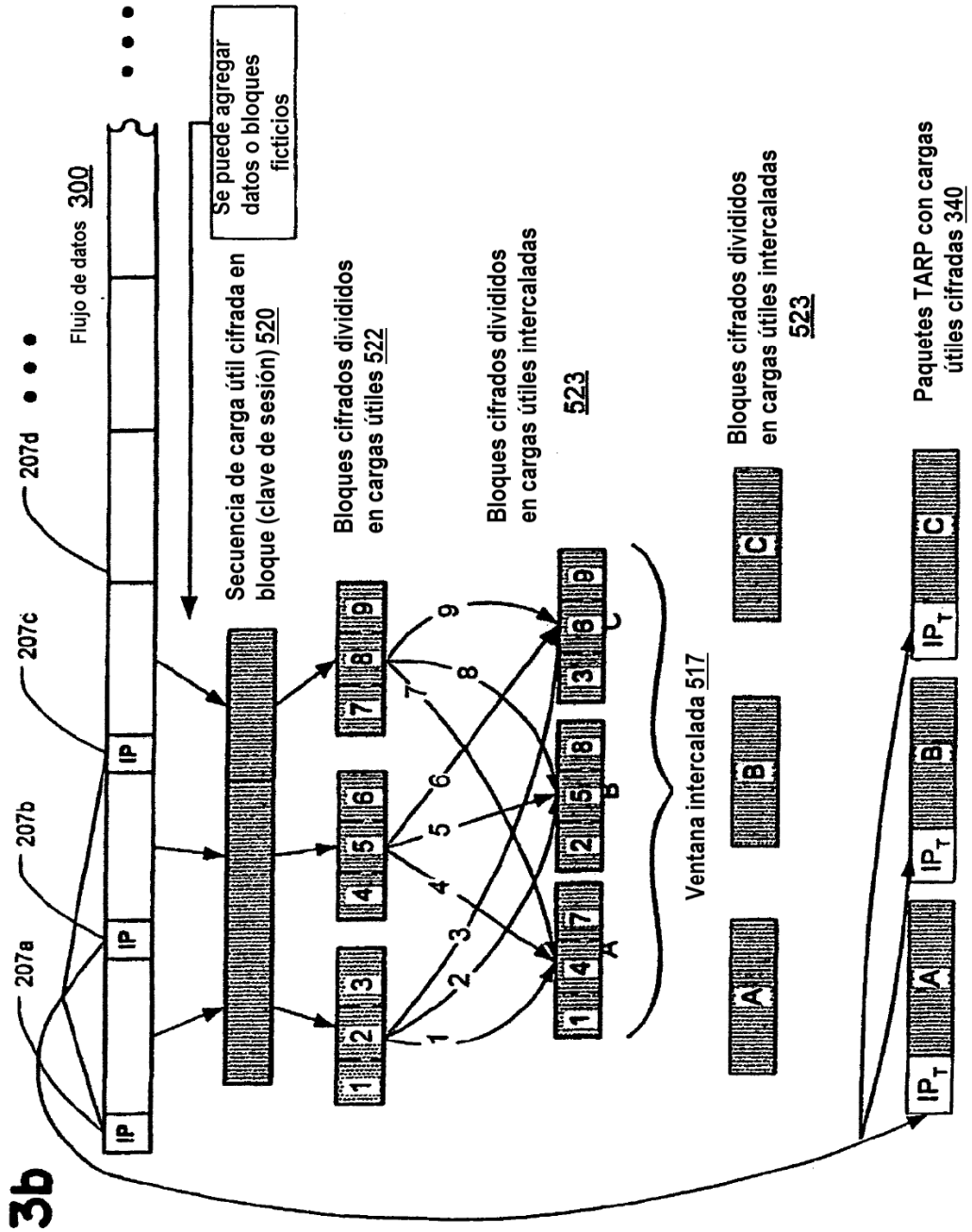

207a
207b
207c
207d
IP
1 2 3
IP
4 5 6
IP
7 8 9
Flujo de datos 300
1 2 3 4 5 6 7 8 9
1 4 7
2 5 8
3 6 9
A
B
C
Datos de carga útil intercalados 320
Ventana de intercalado 320
Datos de carga útil cifrados de clave de sesión 330
Paquetes TARP con cargas utiles cifradas 340
Paquetes TARP cifrados de clave de enlace 350
Paquetes IP con paquetes TARP cifrados como carga útil 360
Enrutador TARP 1
Enrutador TARP 2
Enrutador TARP 7
Enrutador TARP 3
Enrutador TARP 5
Enrutador TARP 6
Enrutador TARP 4
Destino TARP

FIG. 3b


Flujo de datos 300
207a
207b
207c
207d
IP
Secuencia de carga útil cifrada en bloque (clave de sesión) 520
Se puede agregar datos o bloques ficticios
Bloques cifrados divididos en cargas útiles 522
Bloques cifrados divididos en cargas útiles intercaladas 523
Ventana intercalada 517
Bloques cifrados divididos en cargas útiles intercaladas 523
Paquetes TARP con cargas útiles cifradas 340
IPT
A
B
C

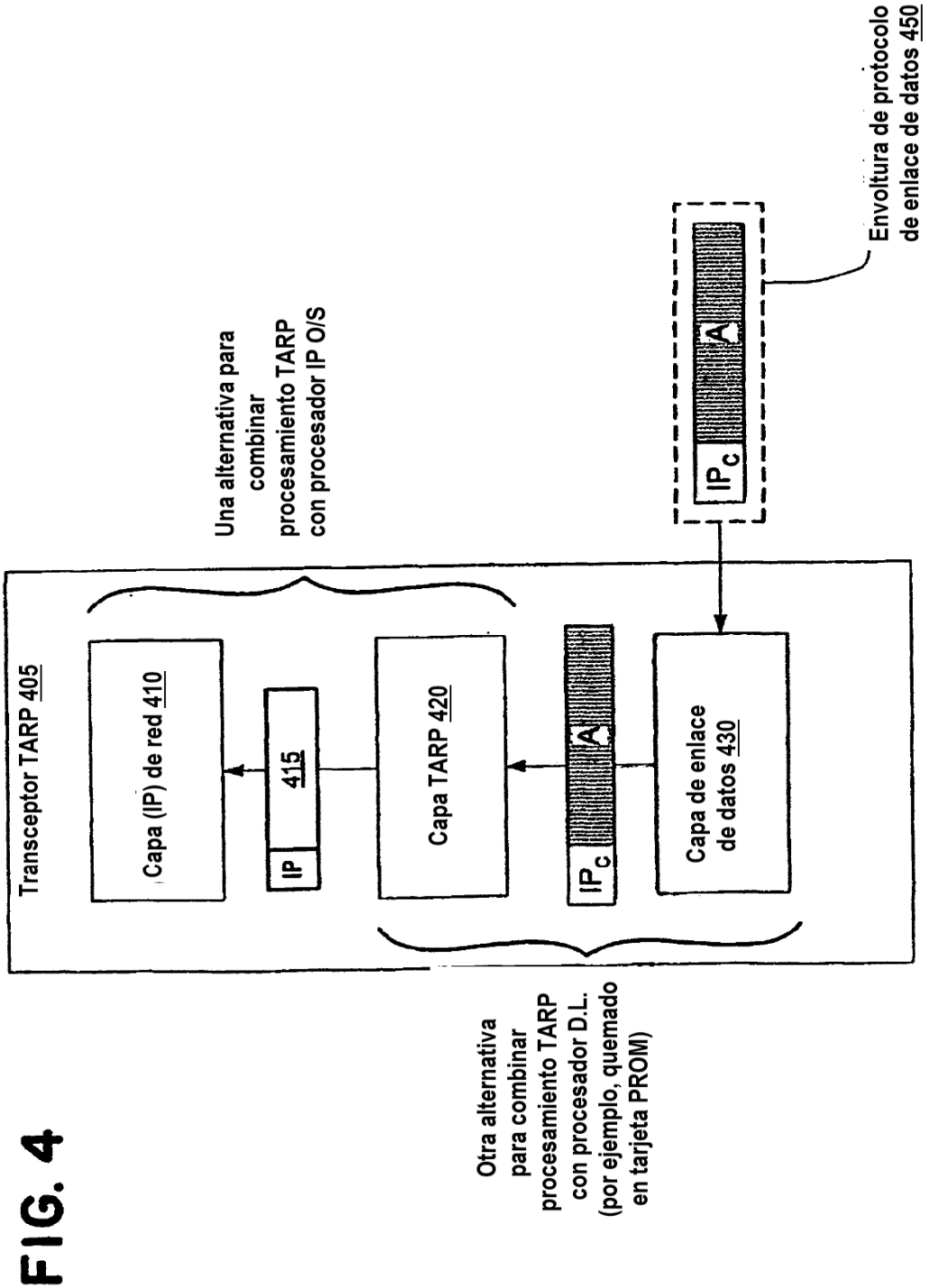

Transceptor TARP 405
Capa (IP) de red 410
IP
415
Capa TARP 420
A
IPc
Capa de enlace de datos 430
Una alternativa para combinar procesamiento TARP con procesador IP O/S
Otra alternativa para combinar procesamiento TARP con procesador D.L. (por ejemplo, quemado en tarjeta PROM)
Envoltura de protocolo de enlace de datos 450

FIG. 4

FIG. 5

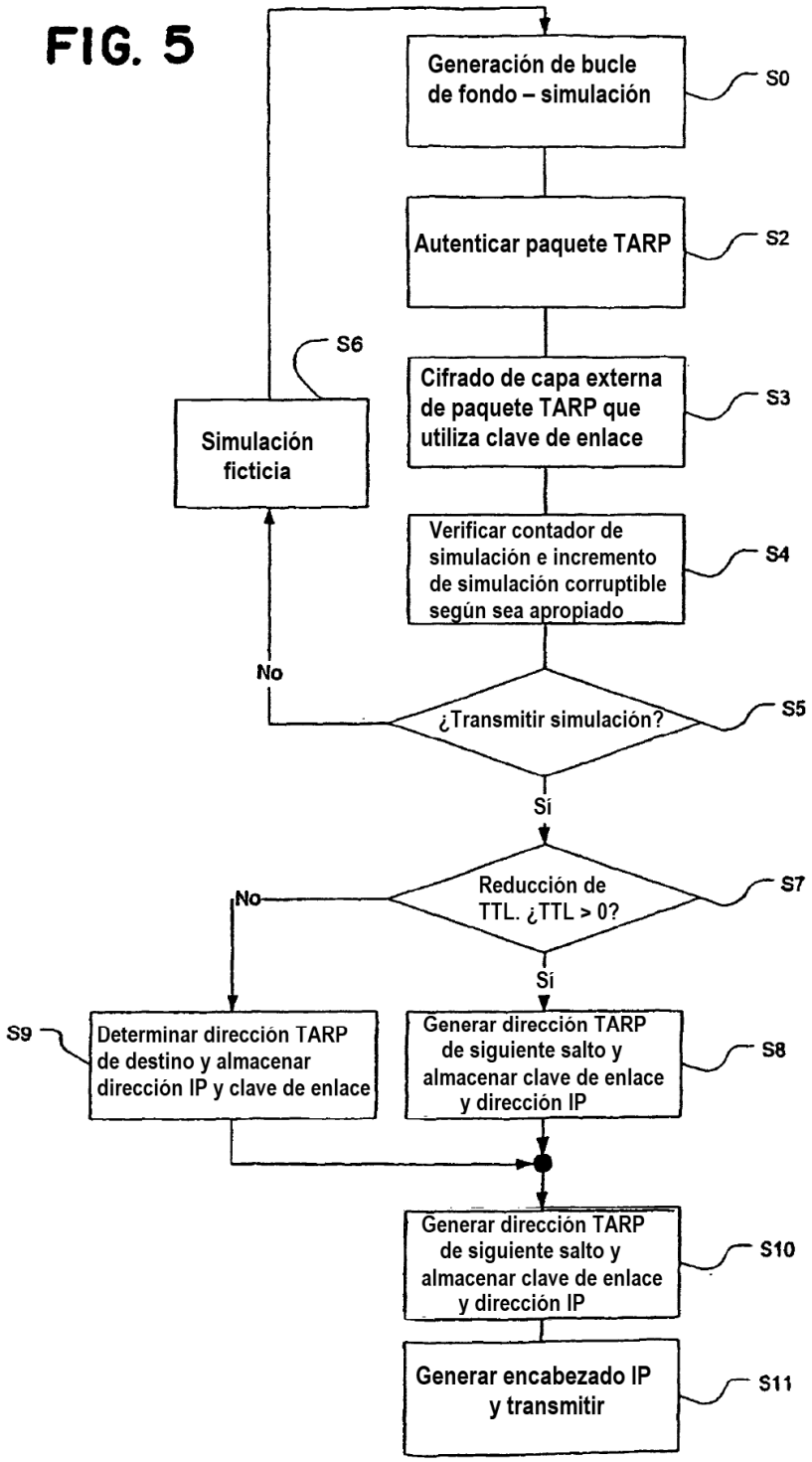
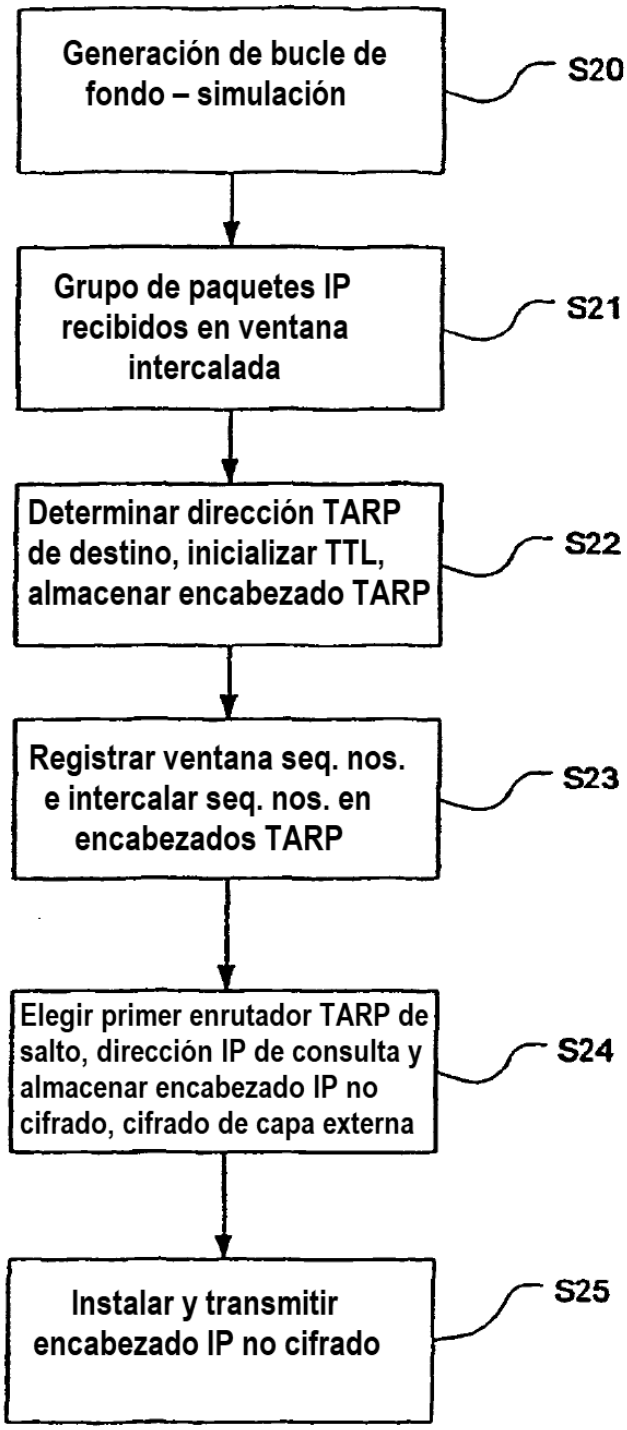

Generación de bucle de fondo – simulación
S0
Autenticar paquete TARP
S2
Cifrado de capa externa de paquete TARP que utiliza clave de enlace
S3
Verificar contador de simulación e incremento de simulación corruptible según sea apropiado
S4
¿Transmitir simulación?
S5
No
Sí
S6
Simulación ficticia
Reducción de TTL. ¿TTL > 0?
S7
No
Sí
Generar dirección TARP de siguiente salto y almacenar clave de enlace y dirección IP
S8
Determinar dirección TARP de destino y almacenar dirección IP y clave de enlace
S9
Generar dirección TARP de siguiente salto y almacenar clave de enlace y dirección IP
S10
Generar encabezado IP y transmitir
S11

FIG. 6


Generación de bucle de fondo – simulación
S20
Grupo de paquetes IP recibidos en ventana intercalada
S21
Determinar dirección TARP de destino, inicializar TTL, almacenar encabezado TARP
S22
Registrar ventana seq. nos. e intercalar seq. nos. en encabezados TARP
S23
Elegir primer enrutador TARP de salto, dirección IP de consulta y almacenar encabezado IP no cifrado, cifrado de capa externa
S24
Instalar y transmitir encabezado IP no cifrado
S25

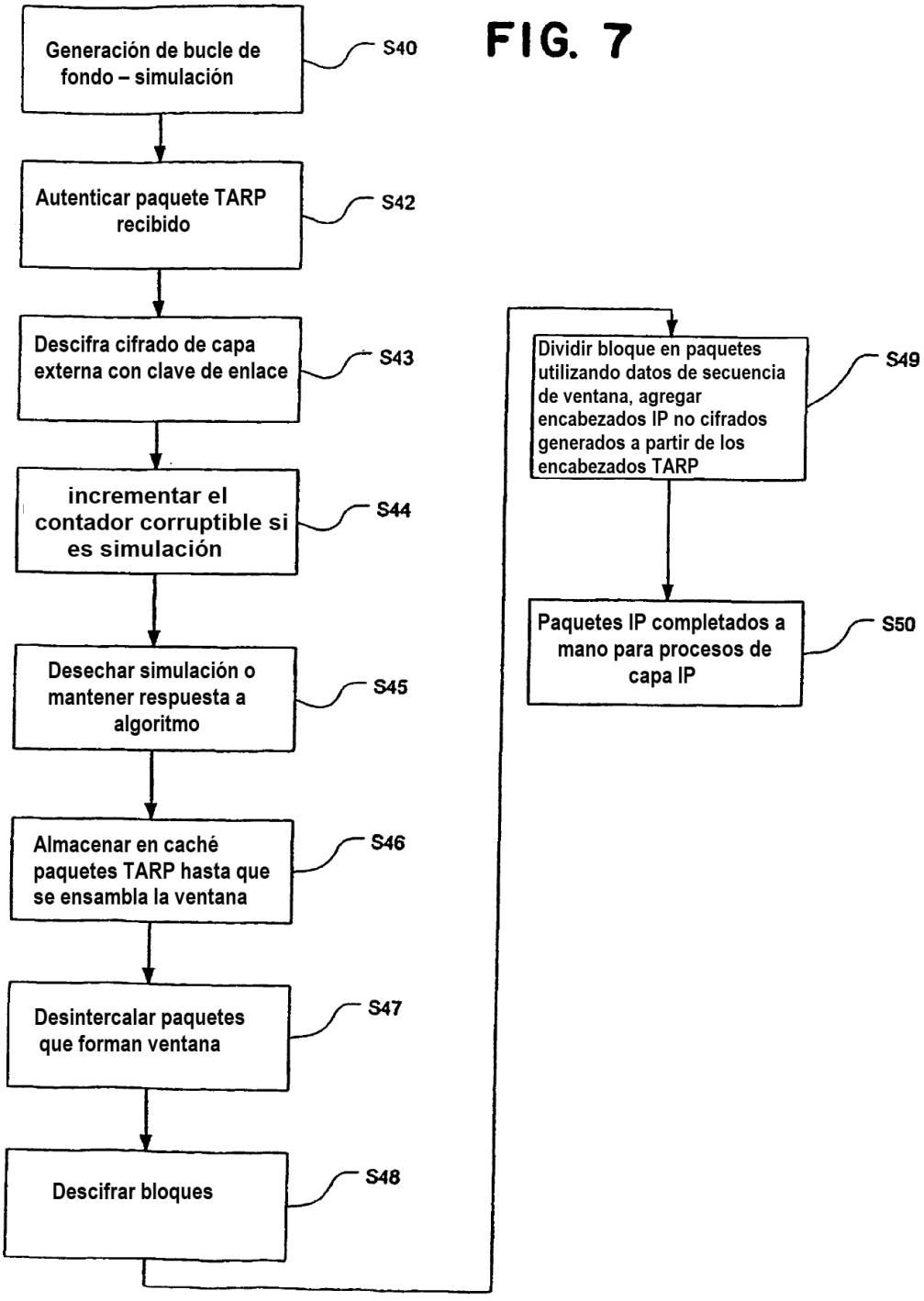

Generación de bucle de fondo – simulación
S40
Autenticar paquete TARP recibido
S42
Descifra cifrado de capa externa con clave de enlace
S43
incrementar el contador corruptible si es simulación
S44
Desechar simulación o mantener respuesta a algoritmo
S45
Almacenar en caché paquetes TARP hasta que se ensambla la ventana
S46
Desintercalar paquetes que forman ventana
S47
Descifrar bloques
S48
Dividir bloque en paquetes utilizando datos de secuencia de ventana, agregar encabezados IP no cifrados generados a partir de los encabezados TARP
S49
Paquetes IP completados a mano para procesos de capa IP
S50

FIG. 7

FIG. 8

ESTABLECIMIENTO DE SESIÓN SEGURA
Y SINCRONIZACIÓN

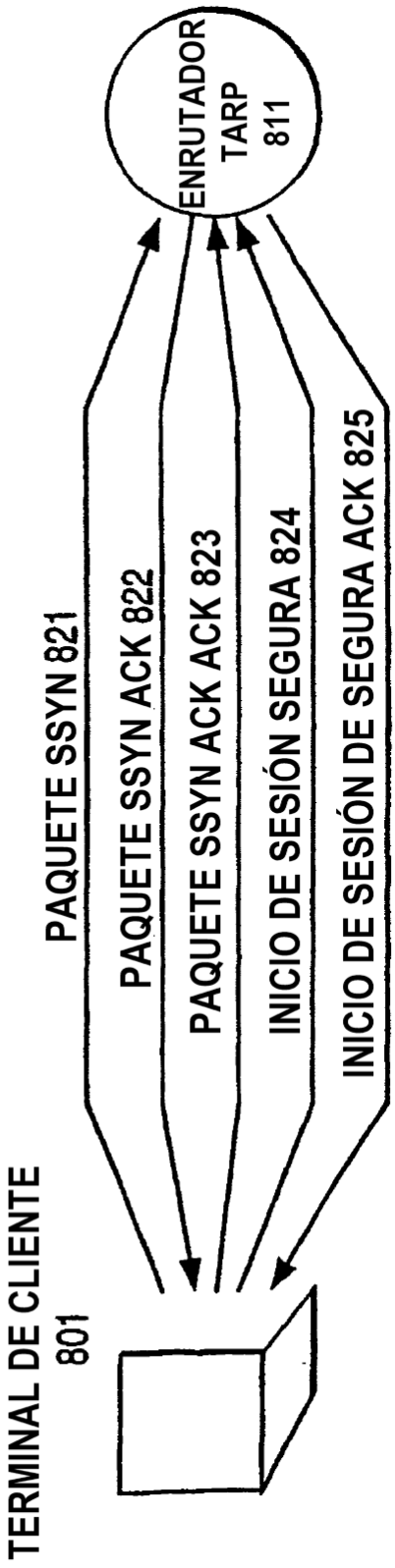


FIG. 9

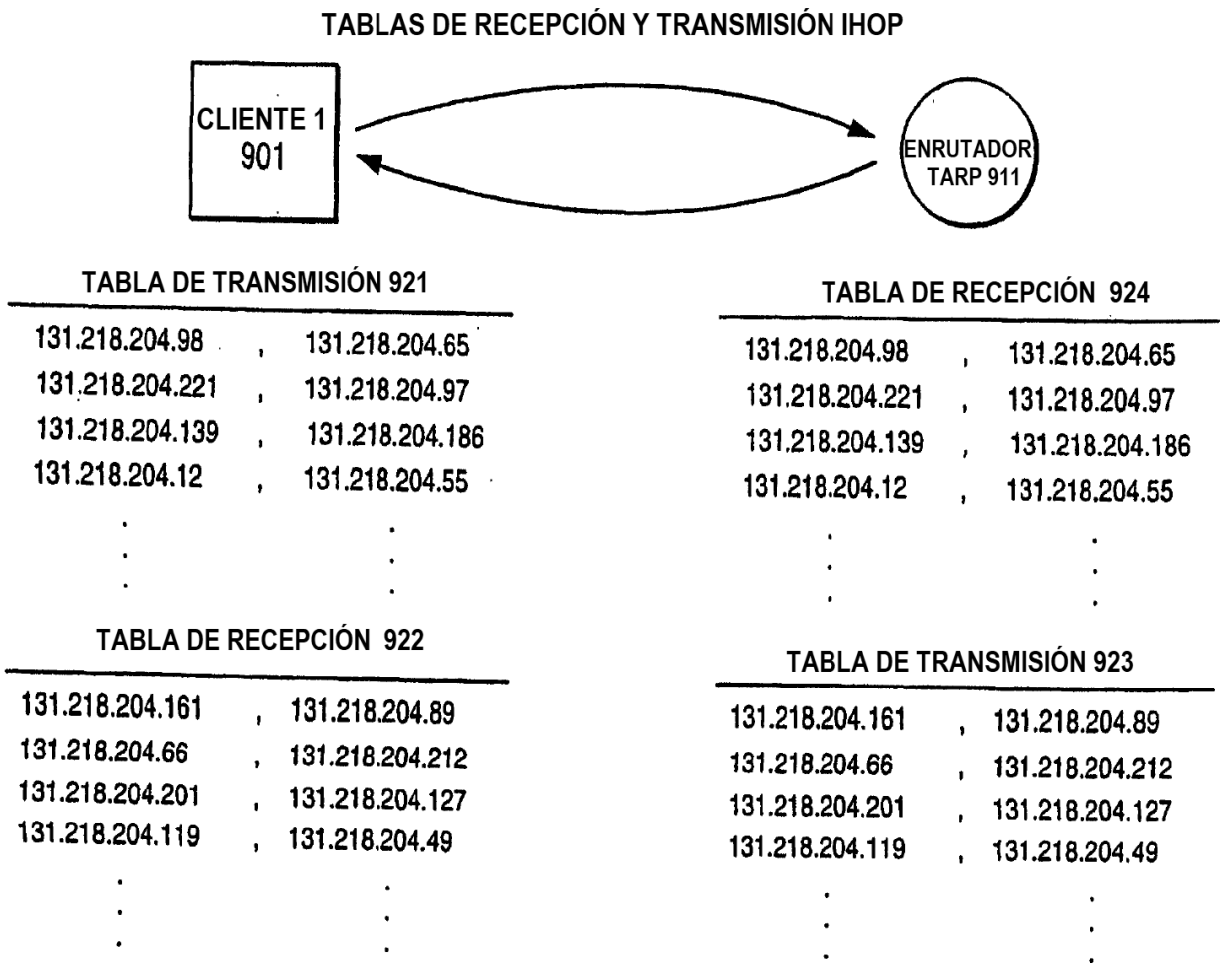

TABLAS DE RECEPCIÓN Y TRANSMISIÓN IHOP
CLIENTE 1
901
ENRUTADOR
TARP 911
TABLA DE TRANSMISIÓN 921
131.218.204.98 , 131.218.204.65
131.218.204.221 , 131.218.204.97
131.218.204.139 , 131.218.204.186
131.218.204.12 , 131.218.204.55
TABLA DE RECEPCIÓN 924
131.218.204.98 , 131.218.204.65
131.218.204.221 , 131.218.204.97
131.218.204.139 , 131.218.204.186
131.218.204.12 , 131.218.204.55
TABLA DE RECEPCIÓN 922
131.218.204.161 , 131.218.204.89
131.218.204.66 , 131.218.204.212
131.218.204.201 , 131.218.204.127
131.218.204.119 , 131.218.204.49
TABLA DE TRANSMISIÓN 923
131.218.204.161 , 131.218.204.89
131.218.204.66 , 131.218.204.212
131.218.204.201 , 131.218.204.127
131.218.204.119 , 131.218.204.49

FIG. 10

REDUNDANCIA FÍSICA DE ENLACE

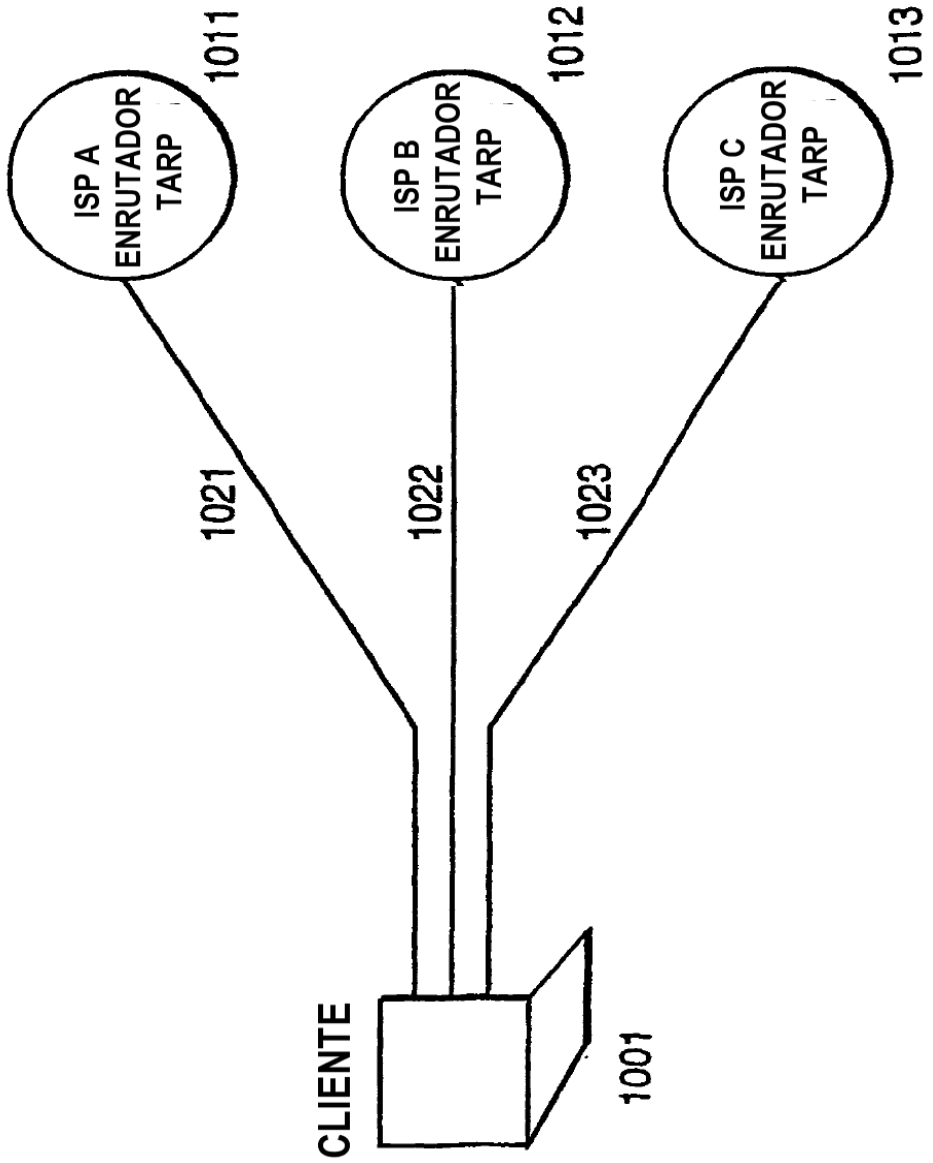


FIG. 11

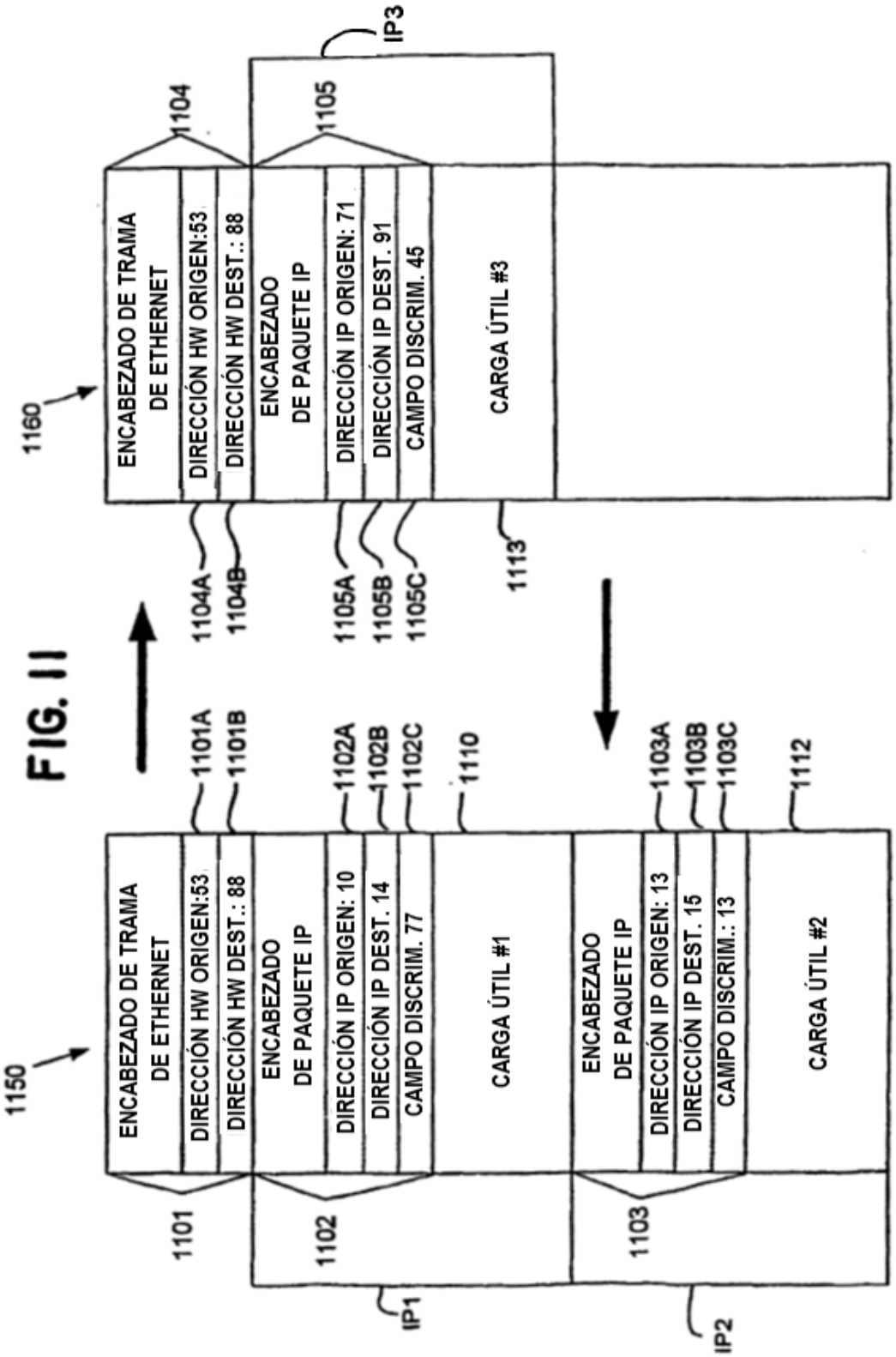

1150
ENCABEZADO DE TRAMA DE ETHERNET
DIRECCIÓN HW ORIGEN:53
DIRECCIÓN HW DEST.: 88
ENCABEZADO DE PAQUETE IP
DIRECCIÓN IP ORIGEN: 10
DIRECCIÓN IP DEST. 14
CAMPO DISCRIM. 77
CARGA ÚTIL #1
ENCABEZADO DE PAQUETE IP
DIRECCIÓN IP ORIGEN: 13
DIRECCIÓN IP DEST. 15
CAMPO DISCRIM.: 13
CARGA ÚTIL #2
1101
1102
1103
IP1
IP2
1101A
1101B
1102A
1102B
1102C
1110
1103A
1103B
1103C
1112
1160
ENCABEZADO DE TRAMA DE ETHERNET
DIRECCIÓN HW ORIGEN:53
DIRECCIÓN HW DEST.: 88
ENCABEZADO DE PAQUETE IP
DIRECCIÓN IP ORIGEN: 71
DIRECCIÓN IP DEST. 91
CAMPO DISCRIM. 45
CARGA ÚTIL #3
1104
1105
IP3
1104A
1104B
1105A
1105B
1105C
1113

FIG. 12A

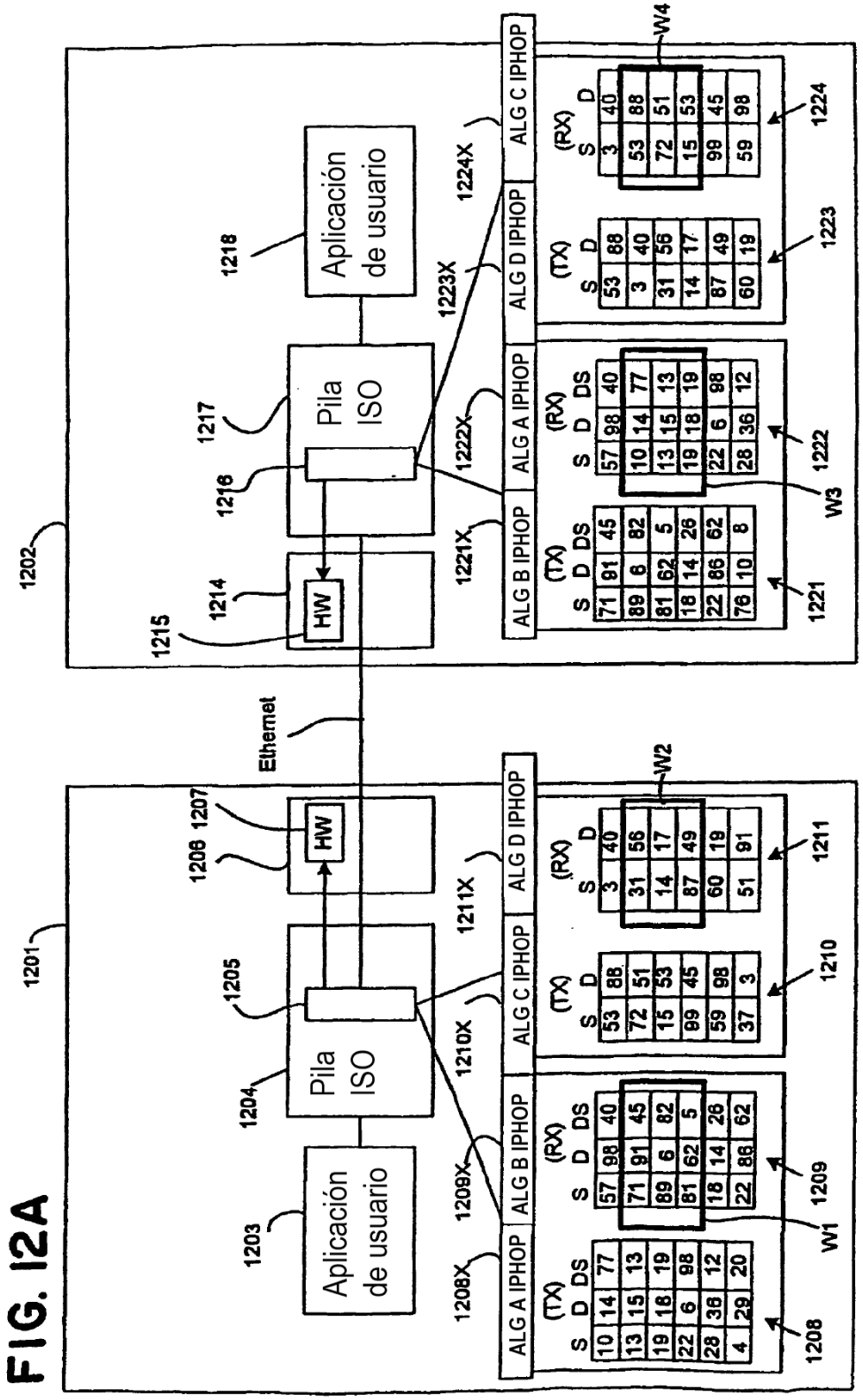

1201
1202
1203
Aplicación de usuario
1204
Pila ISO
1205
1206
1207
HW
1208
1208X
ALG A IPHOP
ALG B IPHOP
ALG C IPHOP
ALG D IPHOP
(TX)
(RX)
S D DS
1209
1209X
W1
1210
1210X
1211
1211X
W2
Ethernet
1214
1215
1216
1217
1218
1221
1221X
1222
1222X
W3
1223
1223X
1224
1224X
W4

Figura 12B

Modo o realización	Direcciones de hardware	Direcciones IP	Valores de campo de discriminador
1. Promiscuo	Igual para todos los nodos o completamente aleatorio	Se puede variar en sincronización	Se puede variar en sincronización
2. Promiscuo por VPN	Fijo para cada VPN	Se puede variar en sincronización	Se puede variar en sincronización
3. Salto de hardware	Se puede variar en sincronización	Se puede variar en sincronización	Se puede variar en sincronización

FIG. 13

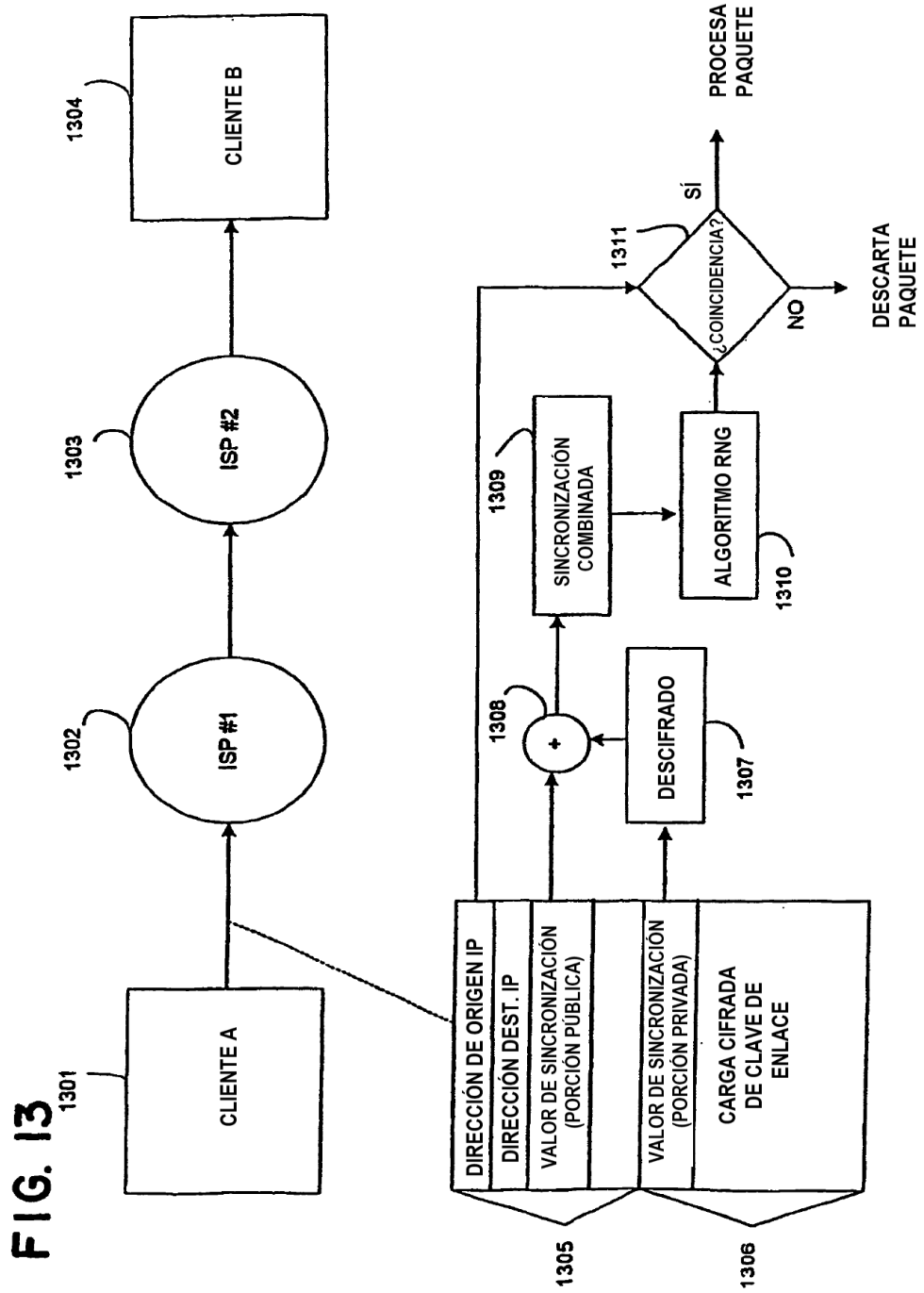

1301
CLIENTE A
1302
ISP #1
1303
ISP #2
1304
CLIENTE B
DIRECCIÓN DE ORIGEN IP
DIRECCIÓN DEST. IP
VALOR DE SINCRONIZACIÓN (PORCIÓN PÚBLICA)
VALOR DE SINCRONIZACIÓN (PORCIÓN PRIVADA)
CARGA CIFRADA DE CLAVE DE ENLACE
1305
1306
1307
DESCIFRADO
1308
+
1309
SINCRONIZACIÓN COMBINADA
1310
ALGORITMO RNG
1311
¿COINCIDENCIA?
SÍ
PROCESA PAQUETE
NO
DESCARTA PAQUETE

FIG. 14

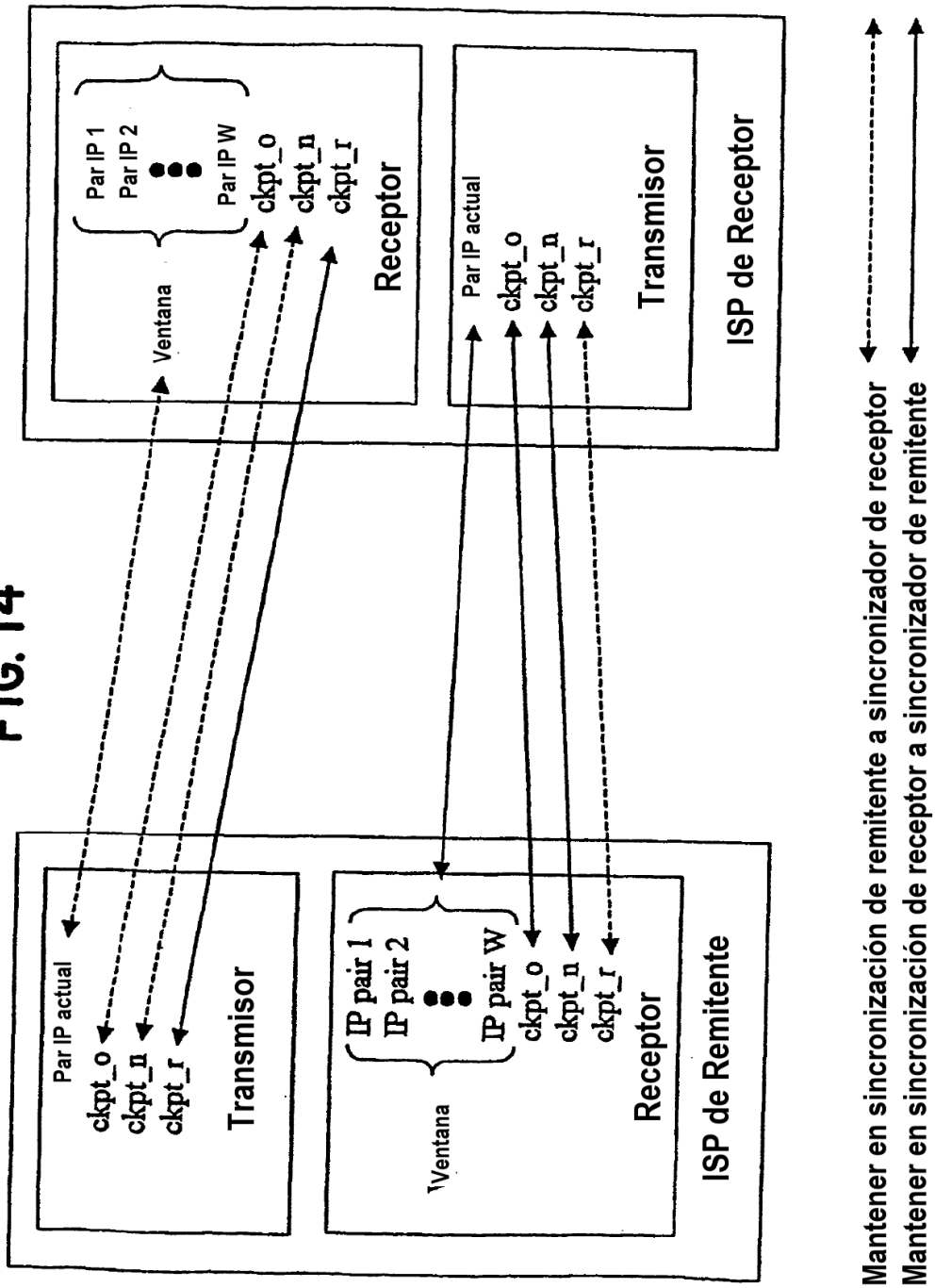

ISP de Remitente
Transmisor
Par IP actual
ckpt_o
ckpt_n
ckpt_r
Receptor
IP pair 1
IP pair 2
IP pair W
Ventana
ckpt_o
ckpt_n
ckpt_r
ISP de Receptor
Transmisor
Par IP actual
ckpt_o
ckpt_n
ckpt_r
Receptor
Par IP 1
Par IP 2
Par IP W
Ventana
ckpt_o
ckpt_n
ckpt_r
Mantener en sincronización de remitente a sincronizador de receptor
Mantener en sincronización de receptor a sincronizador de remitente

FIG. 15

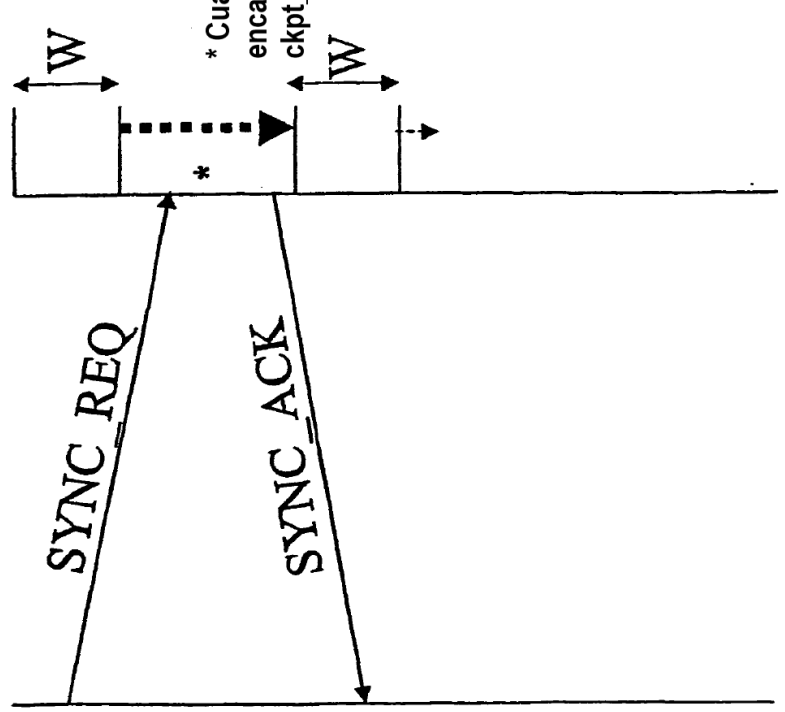

SYNC_REQ
SYNC_ACK
W
W
*
@
#
* Cuando SYNC_REQ llega con encabezado entrante = ckpt_n de receptor:
* Actualiza ventana
* Genera nuevo ckpt_n de par IP de punto de verificación en receptor
* Genera nuevo ckpt_r de par IP de punto de verificación en transmisor
* transmite SYNC_ACK utilizando nuevo ckpt_r de par IP de punto de verificación
@ Cuando se sincroniza comienza la trasmisión (retransmite periódicamente hasta ACKed) SYNC_REQ utilizando nuevo transmisor ckpt_n de par IP de punto de verificación de transmisor y generar ckpt_r de punto de verificación de respuesta de receptor nuevo
# Cuando SYNC_ACK llega con encabezado = ckpt_r: genera nuevo ckpt_r de par IP de punto de verificación en transmisor

FIG. 16

(Lan Ethernet – Bloques de dirección de dos A