Seguridad en Redes Locales de Datos. Contribución a la Seguridad de Redes 802.3 / Ethernet Extendidas.

TESIS DOCTORAL

Universitat Politècnica de Catalunya
Septiembre-1996

Autor: Francisco Recacha Castro
Director: Dr. José Luis Melús Moreno

A mis padres.

Prólogo

La sociedad actual está viviendo una creciente demanda de servicios de seguridad de la información. Esta demanda ha dejado de ser exclusiva de los mundos militar y diplomático como ha venido ocurriendo tradicionalmente y ha aparecido en entornos cotidianos muy diversos como el comercial, el financiero, el médico, etc. Entre los factores causantes de este fenómeno cabe señalar los siguientes [FOR94]:

- el uso cada vez mayor de redes de computadores para procesar y transmitir información sensible (e.g. transacciones financieras) aumenta el "reclamo" hacia posibles atacantes.
- la implantación de los "Sistemas Abiertos" está provocando un crecimiento espectacular de la interconexión de redes de computadores, haciendo que cualquier sistema sea potencialmente accesible por una población de usuarios que crece explosivamente.
- el rápido avance de la tecnología es un arma de doble filo: permite por un lado el desarrollo de aplicaciones más avanzadas y a un coste menor, pero por otro lado hace más asequibles la realización de ataques contra estos mismos sistemas a un número mayor de posibles atacantes.

El estudio de técnicas automáticas de protección de la información se ha venido considerando dividida en dos disciplinas: Seguridad de Ordenadores y Seguridad de Comunicaciones. Como suele ocurrir en muchas clasificaciones categóricas, la frontera entre estas dos disciplinas no es ni mucho menos nítida. De forma simplista, la Seguridad de Comunicaciones estudiaría los problemas de seguridad que afectan a la información cuando ésta se encuentra en tránsito entre dos sistemas y la Seguridad de Ordenadores lo haría con los problemas de seguridad que afectan a la información mientras es procesada o almacenada en un ordenador.

La presente tesis se enmarca dentro del área de la Seguridad de Comunicaciones. Concretamente, se estudian aquí mecanismos de seguridad adecuados a los entornos de Red de Área Local, proponiéndose y evaluándose un conjunto de mecanismos de protección de las comunicaciones para redes locales extendidas de tipo 802.3 / Ethernet.

Históricamente, los orígenes de este trabajo se sitúan en 1991 con el lanzamiento de un proyecto de colaboración entre dos grupos de la Universidad Politécnica de Cataluña: el equipo responsable de la administración los servicios informáticos y el grupo de investigación en criptografía y seguridad en redes de comunicación del Dpto. de Matemática Aplicada y Telemática. El propósito de este proyecto era desarrollar un prototipo de seguridad para las comunicaciones sobre la red extendida de esta universidad, la UPCNET. A partir de entonces se desarrollaron dos versiones de laboratorio que han permitido evaluar en el laboratorio la viabilidad de los mecanismos de seguridad que se estudian en este trabajo, y demostrar su interés mediante la buena acogida en importantes foros internacionales que han tenido las publicaciones realizadas. El objetivo de la primera fase del proyecto fue el desarrollo de unos prototipos de *bridge* con funciones de cifrado para redes Ethernet/802.3 [REC92] [REC93]. Este prototipo de desarrolló básicamente utilizando tecnología *hardware* disponible en el mercado siendo diseñado específicamente todo el *software* del dispositivo (exceptuando unas tarjetas de cifrado de desarrollo propio, los restantes módulos de este *bridge* eran tarjetas estándares para bus VME). Una vez observado el interés que mostró la comunidad internacional por el trabajo, y evaluadas las relativamente buenas prestaciones conseguidas, se atacó en 1993 una segunda fase del proyecto. El propósito de esta segunda fase del proyecto fue doble: por un lado el

desarrollo de una arquitectura *hardware* específica para los *bridges* seguros que permitiera optimizar el comportamiento de los mismos; y por otro lado, el diseño de mecanismos apropiados de gestión de seguridad propios. En esta segunda fase, el proyecto siguió teniendo la buena acogida internacional que tuvo la primera, como queda avalado por [FOR93] [SOR93].

La presente tesis nace como un intento de formalizar desde un punto de vista "académico" los mecanismos de seguridad diseñados durante las dos fases de este proyecto, realizando al mismo tiempo un estudio comparativo, en cuanto a prestaciones, con otros mecanismos similares existentes. Las aportaciones originales de esta tesis, así como su ubicación en esta memoria, se enumeran como sigue:

- Diseño de una arquitectura de seguridad para las comunicaciones en redes locales extendidas Ethernet/802.3 (Capítulo 2).
- Diseño de un mecanismo de encadenamiento para cifrado en bloque que permite garantizar simultáneamente los servicios de confidencialidad e integridad de datos con un coste y nivel de seguridad ofrecidos similares a los de los mecanismos "convencionales" de cifrado en bloque que garantizan sólo confidencialidad o sólo integridad (Capítulo 3).
- Diseño de un protocolo de seguridad (con cuatro variedades) que permite garantizar la confidencialidad e integridad de las comunicaciones sobre redes locales extendidas Ethernet/802.3 (Capítulo 4).
- Evaluación del impacto sobre las prestaciones de la red al introducir los mecanismos de seguridad. Evaluación comparada de las cuatro variedades del protocolo de seguridad entre sí y con otros protocolos "afines" existentes (Capítulo 5).

La estructura general de esta memoria es como sigue. Los dos primeros capítulos sirven a modo de introducción del estado del arte en el área de seguridad de redes de datos y seguridad en redes locales respectivamente. En el capítulo 2, además se describen las características generales de los mecanismos de seguridad estudiados en este trabajo, dando una perspectiva "integradora" del conjunto de todos ellos en lo que podríamos denominar "arquitectura de seguridad". Los restantes capítulos se dedican a exponer los mecanismos de seguridad para redes locales propuestos en esta tesis, evaluación de prestaciones y comparación con otros trabajos en el área. Se supone aquí que el lector tiene unos conocimientos mínimos del estado actual de técnicas criptográficas, pudiéndose en todo caso consultar cualquiera de las numerosas referencias existentes en la bibliografía.

En el capítulo 1 se expone una panorámica general del problema de seguridad en redes de datos. En él se analizan, en el entorno de redes de comunicación de datos, los conceptos de análisis de riesgo, política de seguridad, amenazas, servicios y mecanismos de seguridad. Finalmente se presenta un estudio de arquitecturas de seguridad en cuanto a integración de mecanismos y servicios de seguridad en una arquitectura de red genérica. En el capítulo 2 se analiza la problemática de seguridad específica a las redes locales de datos así como los mecanismos de seguridad que pueden ser adecuados a este entorno. Se realiza un estudio cualitativo mediante el cual se concluye que es conveniente el desarrollo de mecanismos de seguridad para el nivel de enlace de datos de redes locales. Como repaso del estado del arte de este tipo de mecanismos se expone el estándar de seguridad SILS (*Standard for Interoperable LAN Security*) del IEEE (*Institute of Electrical and Electronics Engineers*) y su primera aplicación conocida: seguridad en redes SMDS (*Switched Multimegabit Data Service*). A continuación se expone un breve sumario de los trabajos encontrados en la bibliografía de seguridad en redes locales. Finalmente se describe el escenario de comunicaciones y las características generales de los mecanismos de seguridad propuestos en este trabajo (arquitectura de seguridad). Estos mecanismos se fundamentan en la encapsulación de las tramas estándares de nivel de enlace de datos en un nuevo protocolo de seguridad mediante la adición de nuevos campos y el encriptado de las nuevas tramas resultantes (a este protocolo lo denominamos "protocolo de sesión"). Este encapsulamiento de las tramas estándares en nuevas tramas protegidas es llevada a cabo por un conjunto de "*bridges* seguros" estratégicamente ubicados en la red. Esta ubicación en los puntos de interconexión entre áreas de la red de difícil acceso con las zonas fácilmente accesibles a potenciales atacantes, da a los mecanismos propuestos dos características muy atractivas: transparencia a las estaciones conectadas a la red; y compartición de los mecanismos de seguridad por múltiples estaciones.

En el capítulo 3 se propone y analiza un nuevo modo de encadenamiento para cifradores en bloque que permite obtener confidencialidad e integridad procesando una sola vez cada mensaje, denominado *IOBC* (*Input and Output Block Chainning*). Este modo de encadenamiento resulta de especial interés en la aplicación a seguridad en redes locales debido a los requisitos relativamente altos que en cuanto a velocidad se tienen.

En el capítulo 4 se presentan las cuatro versiones de que se compone el protocolo de sesión propuesto. Se describe detalladamente su operación y finalmente se apuntan las necesidades que en cuanto a administración de seguridad plantea este protocolo. Las cuatro versiones diseñadas corresponden a las cuatro combinaciones posibles a la hora de considerar por un lado dotar de protección sólo a los datos de los niveles superiores al de enlace o a las tramas completas de este nivel, y por otro lado fragmentar o no fragmentar las tramas excesivamente largas. El interés de estudiar las versiones sin fragmentación es fundamentalmente el de aislar en la evaluación de prestaciones el impacto que tienen los mecanismos de fragmentación sobre las prestaciones de la red .

En el capítulo 5 se evalúa el precio que se debe pagar, en cuanto a prestaciones de la red, por la introducción de los mecanismos de seguridad. Aunque el interés de este tipo de evaluaciones es obvio, es de señalar la escasez en la literatura especializada de este tipo de trabajos. Para realizar estas evaluaciones se han escogido los dos parámetros más interesantes en cuanto a medida de prestaciones en redes locales: el *throughput* y el retardo (la probabilidad de pérdidas aunque es un parámetro importante en otro tipo de redes, en el tipo aquí estudiado es notoriamente pequeña y por tanta de importancia secundaria). El impacto sobre las características de retardo (tiempo que tardan los datos en atravesar la red) puede ser especialmente crítico para aplicaciones que si bien no necesitan gran ancho de banda, requieren que la respuesta del sistema sea rápida (p.ej. protocolos de acceso remoto desde terminales). Por otro lado, el impacto sobre el *throughput* puede ser importante para aquellas aplicaciones que si bien no son muy sensibles al retardo, sí que requieren transferir grandes cantidades de información (p.ej. transferencia de ficheros).

Finalmente, en el capítulo 6 se sumarizan las conclusiones más relevantes de este trabajo y se describen las líneas de trabajo futuras que lo continuarán.

Antes de finalizar este prólogo debo reconocer que el desarrollo de esta tesis doctoral hubiera sido algo que hubiera estado fuera de mi alcance de no ser por el apoyo y colaboración de un número considerable de colegas. A todos ellos mi más sincero reconocimiento y gratitud. A J. L. Melús por su constante dirección y empuje durante estos años. A J. Forné y M. Soriano por su contribución al diseño de los mecanismos de seguridad aquí propuestos y junto a X. Simón, E. Pallarés, L. De La Cruz, Ll. Cedó y A. Cabrera por la contribución mediante sus Proyectos Final de Carrera a la realización de los prototipos que hicieron patente la viabilidad e interés de estos mecanismos. A F. Rico, C. Carpintero y J. Paradells por la aportación de algunas ideas iniciales que marcaron fuertemente el trabajo desarrollado. Finalmente, a E. Sanvicente por su desinteresada ayuda en la redacción del artículo [REC93], que constituye seguramente la publicación internacional de mayor relevancia de entre las realizadas a raíz de este trabajo.

Índice

1. Seguridad en Redes de Datos

1.1 Introducción

La sociedad actual ha venido experimentando durante las últimas décadas una significativa evolución marcada por el uso, cada día mayor, de los medios electrónicos de tratamiento de la información. El uso primero de ordenadores aislados y posteriormente de redes de ordenadores está permitiendo una actividad económica más eficaz y una mejora en la calidad de vida de las personas.

No obstante, el desarrollo de estas tecnologías facilita al mismo tiempo el acceso y manipulación ilegítimas de la información así como de los recursos utilizados para procesarla, almacenarla o transmitirla. Este hecho, sumado a la creciente dependencia por parte de organismos e individuos con respecto a estos medios de tratamiento automático de la información, y sumado también a que cada día son más frecuentes los usos "deshonestos", está provocando una importante sensibilización en los ámbitos científico, comercial, civil, político, etc. Como resultado a todo esto, está apareciendo una creciente demanda de servicios de seguridad que protejan al usuario de los riesgos que conlleva el utilizar estas nuevas tecnologías de la información.

A partir de la década de los ochenta el mundo de las comunicaciones de datos ha estado caracterizado por la implantación de los *Sistemas Abiertos*, apadrinados por el crecimiento mundial de la red Internet, la elaboración de la recomendación OSIRM (*Open Systems Interconnection Reference Model*) por ISO (*International Standards Organization*) [ISO84] y por la homogeneización de los productos ofrecidos por la industria de las comunicaciones de datos, gracias a una estandarización en el área [MIN91].

El uso de este tipo de redes ha puesto de manifiesto de forma especial el problema de seguridad de la información: si bien el propósito de los sistemas abiertos es facilitar el acceso a la información y a los recursos informáticos, implícitamente facilita también su acceso de forma ilícita [COT75] [RUS91] [MAD92] [REC94].

Paralelamente a la rápida proliferación de problemas de seguridad en los sistemas informáticos se han producido importantes avances en lo que se ha venido a denominar como "Criptografía Moderna": durante la segunda mitad de los setenta se produjeron numerosas y muy significativas aportaciones a las técnicas de cifrado [DIF76] [NBS77] [RIV77], provocando que desde entonces la comunidad investigadora venga manteniendo un esfuerzo creciente en este terreno [SIM92] [PFL89] [BRA80] [DEN82] [DAV89].

Si bien las aportaciones de la criptografía moderna ya permiten abordar eficazmente el desarrollo de técnicas de seguridad para sistemas informáticos, desafortunadamente las arquitecturas de las redes de telecomunicación que actualmente están en funcionamiento son en general inseguras [PUR93]. Esto se debe fundamentalmente a dos razones: la falta de sensibilización por parte de los diseñadores durante el desarrollo de estas arquitecturas y al relativamente tardío desarrollo de las técnicas criptográficas modernas.

No obstante, la situación actualmente está cambiando rápidamente, ya sea con el estudio de mecanismos de seguridad apropiados para las redes en funcionamiento, o bien con el desarrollo integrado de estos mecanismos de seguridad para las nuevas redes en desarrollo. Simultáneamente, la actividad estandarizadora necesaria para que estos estudios y desarrollos

se puedan llevar a la práctica se ha activado de forma espectacular [ISO88] [CCI] [DOD87] [EUR91] (entre muchos otros).

Cuando se contempla el problema de seguridad de una organización el primer paso a realizar es un análisis exhaustivo de riesgos con respecto a posibles amenazas a las que se enfrenta el sistema, ya sean accidentales o intencionadas. Este análisis puede ser tanto cuantitativo como cualitativo [MAD92]. En un análisis cuantitativo, se entiende por *riesgo* básicamente el coste que tendría un determinado accidente o ataque, ponderado por la probabilidad de que éste se produzca. De esta manera, el análisis de riesgos cuantitativo permite obtener criterios "numéricos" que determinan si una inversión en unos u otros mecanismos de protección es rentable. Sin embargo, en muchos casos este análisis cuantitativo no es factible dada la dificultad de obtener números que midan de forma realista tanto los costes como las probabilidades de las amenazas. Cuando un análisis cuantitativo no es posible, la alternativa es realizar un análisis cualitativo que permita decidir, de alguna manera, contra qué amenazas se debería proteger al sistema. Las metodologías utilizadas en estos análisis cualitativos son muy diversas: discusión entre los miembros del equipo de análisis de riesgos; tramitación de cuestionarios a los usuarios del sistema preguntándoles qué amenazas creen que se deberían solventar; utilización de sistemas expertos; etc. Actualmente es posible encontrar en el mercado paquetes *software* cuya función es realizar, o guiar, análisis de riesgos cuantitativos y/o cualitativos.

El objetivo de un análisis de riesgos es establecer qué elementos *software* y *hardware*, así como qué procedimientos requiere un sistema para ser seguro. A estos elementos y procedimientos se les denomina con el nombre genérico de *mecanismos de seguridad*. El estudio de los mecanismos necesarios para contrarrestar las amenazas a las que se puede enfrentar un sistema informático se clasifican en diversas disciplinas [FOR94]:

- Seguridad de ordenadores.
- Seguridad de las comunicaciones.
- Seguridad física.
- Seguridad de personal (control del personal, sensibilización del mismo, etc.).
- Seguridad administrativa (procedimientos de control de software, de auditoria, etc.).
- Seguridad de soportes (papel, cintas magnéticas, discos, etc.).
- Seguridad de emanaciones.
- Seguridad en el control del ciclo de vida (ver un ejemplo más adelante).

Es necesario señalar que las técnicas de seguridad no sólo son un medio de prevención frente a posibles pérdidas, sino que gracias a ellas (basadas fundamentalmente en el uso de criptografía) es posible crear nuevos servicios de telecomunicación donde la seguridad actúa como una fuente de ingresos o de reducción de costes [GAN94]. Entre estos servicios se puede citar: realización de negocios a distancia gracias al uso de técnicas de firma digital; distribución (tarifada) bajo demanda de películas, música, etc.; agilización de transferencias de fondos al utilizar redes de comunicación de datos; ...

En cierto sentido, la seguridad en un sistema es comparable a una cadena: su fortaleza es la del eslabón más débil. Así, desde el punto de vista global, el estudio de implantación de seguridad debe contemplar todos los posibles aspectos. El objetivo de la presente tesis es el de estudiar y proponer contribuciones particulares en el mundo de la seguridad en redes locales de ordenadores. Un estudio total de este problema conllevaría un análisis de todo el posible abanico de amenazas y de mecanismos de seguridad en el ámbito de las redes locales. No obstante, hay que reconocer que los objetivos de este trabajo no son tan ambiciosos, sino que se limitan al análisis y proposición de soluciones en el área de la seguridad de las comunicaciones sobre redes locales exclusivamente. Por ejemplo, no nos ocuparemos de si las radiaciones electromagnéticas de los equipos pueden ser captadas y analizadas por un posible oponente (seguridad de emanaciones), ni tampoco de qué procedimientos de evaluación o certificación se pueden utilizar para garantizar al usuario el nivel de seguridad del sistema o subsistemas (seguridad en el control de ciclo de vida), tampoco de cómo garantizar a los usuarios la confidencialidad de sus ficheros guardados en una máquina con sistema operativo multiusuario (seguridad de computadores), etc. En definitiva, el objetivo de este trabajo es estudio de mecanismos de seguridad que protejan el intercambio de información entre ordenadores conectados a una misma red local.

1.2 Amenazas, Servicios y Mecanismos de Seguridad

En este apartado se realiza una introducción de los conceptos básicos de seguridad que se utilizan repetidamente en toda la memoria. Como ocurre en la mayoría de disciplinas jóvenes, las definiciones de términos básicos han venido dependiendo de los autores en el área de seguridad. Afortunadamente, a partir de la aparición del documento [ISO88], donde entre otras cosas se definen una serie de conceptos elementales, parece existir mayor unanimidad en las acepciones de estos términos. La exposición que se hace en este apartado de los conceptos de amenazas, servicios y mecanismos de seguridad está basada fundamentalmente en el citado documento.

Antes de pasar a exponer los conceptos de amenazas, servicios y mecanismos de seguridad, se hace conveniente explicitar qué se entiende por "sistema seguro". O dicho de otra forma, qué características debe reunir un sistema para ser considerado seguro. Tradicionalmente, se ha considerado que un sistema seguro ha de reunir una o varias de las tres características siguientes [RUS91]:

- *Confidencialidad.* Un sistema es confidencial si no permite acceder a una información determinada a nadie que no esté autorizado.
- *Integridad.* Un sistema es integro si puede garantizar que los datos que contiene no se han creado, manipulado o destruido ya sea de manera accidental o malintencionada.
- *Disponibilidad.* Un sistema es seguro desde el punto de vista de la disponibilidad si puede garantizar que sus usuarios legítimos podrán tener acceso a la información, o recursos, bajo condiciones (consideradas) normales.

Actualmente, se reconoce un cuarto componente en la seguridad de sistemas informáticos [FOR94]:

- *Uso legítimo.* Un sistema seguro que garantice un uso legítimo no permite su utilización ya sea por usuarios no autorizados o de maneras no autorizadas.

Hay que observar que la medida en que un sistema ha de satisfacer cada uno de los cuatro componentes es algo que depende de los criterios de la organización propietaria del sistema. Estos criterios vendrán dados por la legislación existente, por los resultados de una análisis de riesgos, existencia de tarificación en los servicios del sistema, por estrategias internas a la organización, etc. Este conjunto de criterios de cómo debe ser y cómo se debe gestionar la seguridad constituye lo que se denomina *política de seguridad* de la organización.

1.2.1 Amenazas

Entendemos como amenaza cualquier posible evento que intencionadamente o accidentalmente podría poner en peligro la seguridad del sistema en alguna de sus cuatro componentes antes definidas. Aunque hoy por hoy las pérdidas de información accidentales continúan siendo las amenazas más importantes (e.g. avería en un disco duro), el crecimiento de las amenazas intencionadas es espectacular [ADA92]. Las amenazas intencionadas[1] en redes de comunicación de datos se pueden clasificar en dos categorías:

- Ataques pasivos.
- Ataques activos.

[1] Las amenazas accidentales, aunque importantes, no son tratadas en este trabajo. Tampoco son tenidas en cuenta las amenazas debidas a virus informáticos ya que, aunque también muy importantes, su estudio se enmarca en la disciplina de la seguridad de computadores.

Ataques pasivos son aquellos en los cuales el atacante únicamente escucha, mientras que ataques activos son aquellos en los que el atacante altera las comunicaciones, interviniendo activamente en ellas.

En un ataque pasivo, el atacante únicamente observa los mensajes intercambiados entre *entes dialogantes*[2] de la red, sin interferir para nada en la comunicación. En general, los ataques pasivos son difíciles de detectar, aunque se puede imposibilitar al atacante la extracción de información útil. Los propósitos del atacante pasivo pueden ser diversos, entre ellos a destacar:

- Intercepción de datos, esto es, la observación de los datos transferidos a los que el atacante no tiene acceso legítimo.
- Análisis de tráfico que permita conocer la naturaleza de la comunicación entre dos usuarios. Los parámetros de mayor interés son identificadores de entes dialogantes, longitud de mensajes, frecuencia, ausencia/presencia, etc. También en este grupo se pueden recoger los canales encubiertos, a través de los cuales un usuario del sistema puede pasar información al exterior encubriéndola en algún parámetro de la comunicación (e.g. codificando la información en cuestión mediante la longitud de los mensajes que emite).

En los ataques activos el atacante puede modificar, borrar, retrasar, reordenar, duplicar e insertar mensajes en la red. Los ataques activos son, en general, difíciles de evitar aunque por poderse detectar, sus efectos se pueden prevenir. Para estudiar las contramedidas a estos ataques es conveniente clasificarlos según el siguiente criterio [PUR93]:

- Degradación fraudulenta de servicio (*Service Denial*). En este tipo de ataque el intruso intenta impedir que los entes dialogantes puedan comunicarse. El ataque puede consistir en la destrucción o retardo de mensajes, en la inserción de mensajes espúreos con el fin de congestionar a alguna entidad de la red, etc.
- Suplantación de identidad (*Masquerade*). En este tipo de ataques el intruso intenta hacerse pasar por una entidad diferente. Una suplantación se puede llevar a cabo, capturando una secuencia de autenticación (por ejemplo la secuencia de *login* y *password* en una sesión remota) y repitiéndola después. El caso más frecuente de suplantación se da con usuarios autorizados, pero con privilegios restringidos, que intentan adquirir privilegios mayores.
- Modificación de Mensajes. El propósito de este tipo de ataques es cambiarlos datos transmitidos.
- Reactuación (*Play-Back*). Este ataque se lleva a cabo cuando uno o varios mensajes son capturados y repetidos total o parcialmente para producir un efecto no autorizado.
- Repudiación. Una de las partes niega haber participado en una comunicación, o bien niega el contenido de la misma (e.g. un usuario ordena a su banco realizar una transferencia desde una de sus cuentas y posteriormente lo niega).

Otras amenazas que cabe mencionar:

- Puerta trasera (*Trapdoor*). Se crea una puerta trasera al modificar alguna entidad del sistema para permitir al atacante llevar a cabo una acción no autorizada. Por ejemplo, la entidad de validación de *passwords* puede ser modificada para permitir al intruso acceder a algún recurso de la red. Este ataque suele producirse en sistemas que permiten el acceso sin utilizar mecanismos de autenticación.

---

[2] La exposición que se realiza hay que entenderla en un entorno de arquitectura de red como OSI, donde el camino lógico que siguen los datos en una comunicación es un secuenciamiento de entidades abstractas (capas o niveles) a través del cual los programas de aplicación (nivel más alto) se comunican. Entendemos como entes dialogantes o interlocutores de una comunicación a dos entidades (usuarios, procesos, etc.), en el mismo nivel de la arquitectura, pero cada uno situado en uno de los sistemas dialogantes, y que se comunican entre sí mediante un *peer-to-peer* protocol. (Para una exposición más detallada ver [TAN89]). También se está utilizando el término "mensaje" para designar a lo que en el entorno OSI se denomina *Protocol Data Unit*.

- Caballos de Troya. Son entidades que introducidas "legítimamente" en el sistema provocan efectos no autorizados pero deliberadamente planeados.
- Encaminamiento incorrecto. El atacante cambia la información de control de los mensajes para reencaminarlos hacia otros destinos.

1.2.2 Servicios de Seguridad

En el entorno de seguridad en comunicación de datos, se entiende como *servicio de seguridad* a las funcionalidades de un sistema que actúan como contramedida a una cierta amenaza. Este concepto fue introducido en [ISO88] estableciendo cinco grupos genéricos de servicios de seguridad (para más detalle consultar por ejemplo [ABR87] y [BRA87], o el mismo documento de ISO):

- Autentificación.
- Control de acceso.
- Confidencialidad de información.
- Integridad de datos.
- No repudiación.

Los servicios de *autentificación* permiten la verificación de la identidad de una entidad comunicante remota. Existen dos tipos de autentificación: autentificación de la identidad de los comunicantes y autentificación de la fuente de un mensaje. La segunda modalidad difiere de la primera en que la entidad generadora del mensaje no tiene por qué intervenir directamente en la comunicación. Hay que observar que los restantes servicios de seguridad se apoyan de una u otra forma en este servicio (por ejemplo, en el servicio de control de acceso es necesario identificar de forma segura al solicitante para así permitirle operar sólo con los privilegios autorizados).

Los servicios de *control de acceso* previenen la utilización no autorizada de la red (ya sea de los mismos servicios de comunicaciones o del acceso remoto a información o recursos). Para ello se identifica al ente que quiere acceder al recurso, y a continuación se controla que sólo acceda con los privilegios autorizados. Dado que el acceso autorizado incluye lectura/escritura de información y acceso a los recursos, este servicio contribuye al soporte de algunos servicios de confidencialidad y de integridad.

Los servicios de *confidencialidad* protegen la información contra escuchas no autorizadas. El servicio puede prestarse tanto para evitar la escucha de datos como el análisis de tráfico (i.e. protección contra ataques pasivos). Con respecto a los servicios de confidencialidad de datos, se pueden considerar tres modalidades diferentes dependiendo de la granularidad de los datos protegidos: confidencialidad orientada a enlace, confidencialidad no orientada a enlace y confidencialidad selectiva de campos. En un servicio de confidencialidad orientado a enlace se protegen todas los mensajes que componen una conexión entre dos entidades comunicantes. En un servicio de confidencialidad no orientado a enlace, cada mensaje recibe protección por separado. Finalmente, en el servicio de confidencialidad selectiva cada mensaje recibe protección exclusivamente en algunos de sus campos (e.g. direcciones de origen y destino).

Los servicios de *integridad de datos* permiten garantizar la integridad de los datos entregados en destino. Es decir, garantizar que los datos se reciben tal cual fueron emitidos por la fuente y que no se han insertado datos adicionales, que no se han borrado, modificado, reordenado ni replicado. Similarmente al caso de la confidencialidad, se consideran tres modalidades de servicios de integridad en sistemas de comunicación de datos: integridad de conexión, integridad no orientada a conexión e integridad selectiva de campos. En el tercer caso, integridad selectiva de campos, se garantiza la integridad de sólo algunos campos de los mensajes (ya sean aislados o de todos los que forman una conexión). En el segundo, integridad no orientada a conexión, se garantiza para cada mensaje por separado que no ha sido modificado o insertado. En el caso de integridad orientada a conexión se garantiza que en toda la secuencia de mensajes de una conexión no se han insertado o modificado datos. Observar que en principio no se garantiza la duplicación o borrado de datos. Si deseamos garantizar que un determinado mensaje no es réplica de otro anterior, o que no ha sido borrado, es necesario

utilizar un servicio de integridad orientado a conexión. Pero aún así queda la posibilidad de que el atacante intente replicar toda una conexión completa.

Hay que señalar que para muchos autores los conceptos de autenticidad e integridad han sido sinónimos hasta la aparición del modelo de seguridad de ISO [ISO88] para comunicación en sistemas abiertos, en el cual se separan y distinguen claramente ambos conceptos[3].

Los servicios de *no repudiación* difieren de todos los demás en tanto que no protegen a los comunicantes de las acciones de terceros, sino que los protegen entre sí en entornos de desconfianza mutua. Este servicio tiene dos modalidades: no repudiación con prueba de remitente y no repudiación con prueba de entrega a destinatario. En el primer caso la prueba protege al destinatario de que el remitente niegue haberle enviado el mensaje (o el contenido de éste). En el segundo caso se protege al remitente frente a la negación del receptor de haber recibido el mensaje (o contenido de éste).

1.2.3 Mecanismos de Seguridad

En el anterior subapartado se exponen las funciones que debe prestar un sistema de comunicaciones de datos para ser considerado seguro. Estas funciones son lo que se ha definido como "servicios de seguridad". Los *mecanismos de seguridad* según la definición de [ISO88] constituyen las tecnologías de base de que dispone el ingeniero para realizar estos servicios de seguridad. Es decir, el conjunto de elementos *software, hardware,* procedimientos de operación y de gestión, que por separado o combinados permiten garantizar la presencia de uno o más servicios de seguridad, y por ende, garantizar la protección del sistema frente a las amenazas contra las que se le desea cubrir. Los servicios de seguridad a considerar en el entorno de redes de comunicación de datos son los siguientes:

- Cifrado.
- Firma digital.
- Control de acceso.
- Integridad de datos.
- Intercambio de autentificación.
- Tráfico de relleno.
- Control de enrutamiento.
- Notarización.

El *cifrado* es el mecanismo de seguridad seguramente más importante por ser aplicable a la realización de mayoría de servicios de seguridad. Los algoritmos de cifrado pueden ser reversibles o no reversibles. Entre los reversibles encontramos dos clases de algoritmos: los simétricos, en los cuales el conocimiento de la clave de cifrado implica el conocimiento de la de descifrado y viceversa; y los asimétricos en los que el conocimiento de la clave de cifrado no implica el conocimiento de la de descifrado, o viceversa. Los algoritmos de cifrado no reversibles pueden utilizar o no una clave. En el caso de utilizar clave, ésta puede ser pública o secreta. Dada la abundante bibliografía existente respecto al tema de criptografía (véase por ejemplo [BRA90], [DAV89], [DEN82], [FOR94], [PFL89], [PUR93], [SIM92], [STA95], etc.), en esta memoria omitimos deliberadamente un análisis detallado de las técnicas de cifrado existentes en la fecha. Es de señalar que los mecanismos de cifrado necesitan mecanismos auxiliares de gestión de claves (salvo en el caso de algoritmos no inversibles).

La *firma digital* es otro mecanismo criptográfico que permite al receptor de un mensaje comprobar la integridad de los datos así como la identidad del remitente. Si esta comprobación es correcta, el receptor podrá demostrar la procedencia y contenido de tal mensaje. La firma de un mensaje se realiza añadiendo a éste un código de comprobación (firma) que depende tanto del mensaje como de la identidad del emisor. Para ello, el emisor debe utilizar información secreta que sólo él conoce. A su vez, el receptor utiliza para

[3] Sin embargo, hay que reconocer que la elección de terminos puede que no fuese muy acertada, pues tanto en inglés como en castellano algo es auténtico cuando no ha sido falsificado o manipulado. No obstante, en todo este decumento se utiliza el término *autentificación* en el sentido de "identificación segura" y el de *integridad* en el sentido de "no falsedad".

comprobar la autenticidad de la firma una información pública asociada al emisor a partir de la cual no es posible deducir la secreta. Este mecanismo normalmente utiliza mecanismos de cifrado basados en algoritmos de clave pública y permite realizar servicios tales como autentificación y no repudiación.

Los mecanismos de *control de acceso* utilizan la autentificación de identidad de una entidad (o bien sus credenciales en caso de accesos anónimos) para permitirle operar con un recurso según los privilegios que tenga autorizados. Con estos mecanismos se puede garantizar que únicamente los usuarios autorizados tengan acceso a servicios o recursos protegidos. Es importante distinguir que todo servicio de control de acceso estará compuesto básicamente por dos bloques funcionales: el primero se encarga de la identificación y de la obtención de los privilegios asociados; el segundo se encargará de supervisar y controlar que el acceso se haga efectivamente sólo con estos privilegios. El control de acceso se fundamenta en la identificación auténtica: cuando un usuario quiere acceder a un recurso debe dar al sistema alguna información que le permita a éste establecer la autenticidad de su origen. Las técnicas existentes se pueden clasificar en criptográficas (autenticación criptográfica) y no criptográficas (*passwords*, llaves, tarjetas magnéticas, huellas digitales, firma, etc.). Obviamente estos mecanismos tienen como objetivo la realización de los servicios de seguridad de idéntico nombre.

Los mecanismos para garantizar la *integridad de datos* en un entorno de red se pueden clasificar en dos grupos. Los primeros garantizan la integridad de los datos de un sólo mensaje (no orientados a enlace). Para ello, se suele añadir un código de redundancia (*checksum*) que o bien es calculado por medios criptográficos[4] o bien por medios convencionales y posteriormente cifrado[5]. Este código permite posteriormente que el receptor del mensaje compruebe su integridad. Los mecanismos del segundo grupo (orientados a enlace) garantizan la integridad de toda una secuencia de mensajes evitando la destrucción, inserción, repetición, etc. de cualquier mensaje de la secuencia. Para ello, se añade un código que además del *checksum* del mensaje incorpore una marca de secuencia o tiempo, o bien se utiliza algún encadenamiento criptográfico entre mensajes [GON93]. De esta forma se garantiza a la entidad receptora que los mensajes recibidos a lo largo de la conexión corresponden exactamente a los que generó el emisor.

Los mecanismos de *intercambio de autenticidad* permiten corroborar que las entidades que intervienen en una comunicación son las que dicen ser. Como ya se ha comentado en el párrafo de mecanismos de control de acceso, las técnicas de autentificación existentes se pueden clasificar en criptográficas (autenticación criptográfica) y no criptográficas (*passwords*, llaves, tarjetas magnéticas, huellas digitales, firma, etc.). Un importante requisito de todo mecanismo o protocolo de autentificación es que esté protegido contra ataques de réplica.

Los mecanismos de *relleno de tráfico* generan mensajes espúreos y/o rellenan los mensajes transmitidos. De esta forma se anulan las amenazas de análisis de tráfico. Para que estas medidas sean efectivas se deben utilizar junto a la confidencialidad de datos.

Los mecanismos de *control de enrutamiento* garantizan que las rutas utilizadas por los mensajes a través de la red son "correctas". Permiten cambiar la ruta de una comunicación si en la anterior se han detectado posibles ataques. Además si se asocian etiquetas de grado de sensibilidad a los datos (confidencial, *top secret*, ...) entonces este tipo de mecanismos enrutarán los datos únicamente a través de subredes con nivel de seguridad apropiado.

Los mecanismos de *notarización* permiten demostrar ante terceros propiedades de los datos intercambiados entre dos o más interlocutores, tales como la integridad, origen, destinatario, instante y contenido. Para ello se dispone de una tercera entidad, denominada notario, que es de confianza para todas las partes y que dispone de suficiente información para realizar las comprobaciones.

---

[4] A este código se le suele denominar MAC (del inglés *Message Authentication Code*). La utilización del término *autentificación* se debe a razones históricas: como se ha comentado anteriormente, algunos autores incluyen en el concepto de *autenticidad* el de *integridad*. A fin de evitar confusiones, en bibliografía reciente es frecuente encontrar sustituido este término por el de "ICV" (del inglés *Integrity Check Value*), también por el de "*checksum* criptográfico" y otros.

[5] A este código se le suele denominar MDC (del inglés *Manipulation Detection Code*).

1.3 Arquitecturas de seguridad

Como es bien sabido, el diseño de cualquier red de comunicación de ordenadores presenta, en general, el denominador común de estar organizado como una pila de capas o niveles. Cada uno de estos niveles tiene como función el ofrecer ciertos servicios al nivel inmediatamente superior (principalmente la transmisión y recepción de datos de este nivel superior), "escondiendo" los detalles de cómo son realizados [TAN89]. Este principio de diseño permite que el problema de comunicar aplicaciones distribuidas sobre una red, que en principio es de complejidad considerable, se puede separar en problemas más elementales y por tanto de más fácil solución.

Figura 1.1: Organización en niveles de la arquitectura de red.

En principio, por encima del nivel superior, el de las aplicaciones, encontramos a los usuarios (ya sean humanos o simplemente procesos) y en el nivel inferior los medios físicos de transmisión (cables metálicos, fibra óptica, radioenlaces, etc.). En los niveles intermedios encontramos distribuidas funciones diversas necesarias para el acceso distribuido sobre la red tanto de información como de recursos (corrección de errores, enrutamiento, control de flujo, etc.). En cada máquina conectada a la red (ya sea una estación terminal o un nodo de interconexión), las funcionalidades del nivel n dialogan con las del nivel n en otras máquinas a fin de desarrollar los servicios que tengan especificados, según se muestra en la Figura 1.1. Las reglas de este diálogo (i.e. vocabulario, semántica y gramática [HOL91]) constituyen el protocolo de nivel n de esta red. La especificación para una red determinada de su torre de protocolos completa es lo que se denomina *arquitectura de red*. Actualmente existen multitud de arquitecturas de red especificadas y en funcionamiento, siendo seguramente la arquitectura TCP/IP de Internet [COM88] la más extendida entre ellas.

En 1984, ISO publicó un modelo de referencia de protocolos estándares para la interconexión de sistemas abiertos, formado por siete niveles y conocido por *modelo OSI* (*Open Systems Interconnection*) presentado esquemáticamente en la Figura 1.2. Independientemente de que esta arquitectura de protocolos no sea la más utilizada en la actualidad, el modelo de referencia propuesto se ha mostrado muy útil a la hora de discutir tópicos diversos en el área de redes de ordenadores. Esto se debe a que la mayoría de los elementos de otras arquitecturas se corresponden de una u otra forma con elementos del modelo OSI. Se pueden encontrar estudios detallados de esta arquitectura, por ejemplo, en [TAN89], [BLA91].

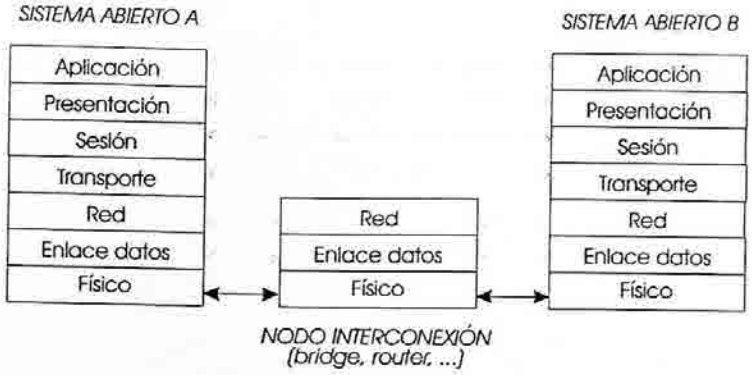


Figura 1.2: Modelo de referencia de siete niveles de OSI (incluyendo un posible nodo de interconexión entre nodos extremos)

En este apartado se hace un recorrido por el problema de arquitecturas de seguridad en redes de datos. Por *arquitectura de seguridad* se entiende el modelo de integración de mecanismos y servicios de seguridad en los diferentes niveles de una determinada torre de protocolos. Se comparan los tres modelos de arquitecturas de seguridad más interesantes que se pueden encontrar en la bibliografía: el tradicional modelo de cifrado de enlace *versus* cifrado extremo a extremo; la arquitectura de seguridad propuesta para OSI en [ISO88]; y finalmente un modelo de cuatro niveles muy original e interesante propuesto en [FOR93]. En todo momento, por las razones ya citadas, se utilizará OSI como referencia de arquitectura de red. El estudio de estos modelos de arquitectura de seguridad nos permitirá en los siguientes capítulos analizar mejor las ventajas e inconvenientes de ubicar mecanismos de seguridad en uno u otro nivel de la arquitectura de red.

1.3.1 Modelo de medidas orientadas a enlace *versus* extremo a extremo

Tradicionalmente[6] se han considerado básicamente dos formas de integrar de seguridad en redes de comunicaciones [VOY85], [STA95]. La primera protege cada enlace de que interviene en la comunicación de forma independiente (*Link Measures*). En el segundo caso cada mensaje es protegido durante todo su trayecto en la red desde el origen hasta el destino (*End-to-End Measures*). Considerando la arquitectura de red OSI, las protecciones orientadas a enlace se deben situar en los niveles inferiores de la arquitectura (nivel físico o nivel de enlace de datos). Por contra, las protecciones extremo a extremo se situarían en los niveles superiores de la red. En este subapartado se analiza la ubicación de los mecanismos de seguridad desde el punto de vista de ambas alternativas.

1.3.1.1 Medidas orientadas a enlace

Un mecanismo orientado a enlace presta seguridad a la información transportada sobre el enlace físico que une dos nodos vecinos, independientemente del emisor y destinatario finales de la misma (véase Figura 1.3). Cada enlace corresponde a una conexión en el nivel de enlace de datos del modelo de referencia OSI. Un enlace puede ser un circuito telefónico, un radioenlace de microondas, un cable coaxial en una red local Ethernet, etc. Este tipo de enlaces no presentan normalmente ningún tipo de protección física y por tanto suelen ser fáciles de "pinchar" para llevar a cabo cualquier tipo de ataque.

[6] Este modelo proviene de la literatura criptográfica, siendo los terminos originales *cifrado de enlace* y *cifrado extremo a extremo*.

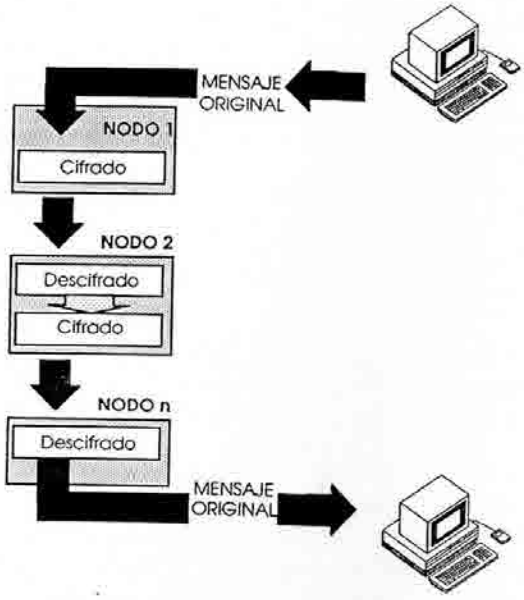


Figura 1.3: Cifrado de enlace.

En las redes que utilizan medidas orientadas a enlace se cifran los datos independientemente en cada enlace susceptible a "pinchazos", utilizando una clave diferente en cada uno de ellos. De esta forma si la seguridad en un enlace se ve comprometida no tiene por qué afectar la seguridad del resto de enlaces. Puesto que los mensajes no necesitan ser procesados en el enlace, se puede cifrar la información relativa a origen y destino, pudiendo realizar el servicio de confidencialidad de tráfico.

Un gran atractivo de este tipo de protección es que es realizable de forma independiente de los sistemas conectados a la red y además, estos últimos pueden seguir operando de la misma forma que lo hacían antes de introducir el servicio. El principal inconveniente es que al aplicar el cifrado sólo en los enlaces físicos, toda la información, sensible o no, es procesada en claro por los nodos de interconexión de la red. Por tanto, se requiere que los nodos de interconexión sean de confianza.

1.3.1.2 Medidas orientadas extremo a extremo

En las medidas orientadas extremo a extremo, la información es protegida antes de entrar en la red y no es desprotegida hasta salir de ella al llegar a la máquina destinataria final (ver Figura 1.4). Este método garantiza la seguridad de las comunicaciones independientemente de la confiablilidad de los nodos intermedios que intervienen en la red ya que ninguno de ellos tiene acceso a la información en modo desprotegido.

El número de servicios de seguridad que potencialmente pueden dar las técnicas extremo a extremo es mucho mayor al que pueden prestar las orientadas a enlace por estar los mecanismos de seguridad más "cercanos" a las aplicaciones. Esto implica que en función de la naturaleza de la información sea más fácil seleccionar unos u otros mecanismos[7]. No obstante, la integración de servicios en los niveles superiores de la red puede presentar importantes inconvenientes (ver 1.3.3).

[7] Por ejemplo, en el nivel de enlace de la red existe, en general, una gran multiplexación de información procedente de muchas aplicaciones sin, por lo que es difícil dar servicios "especializados" que dependan de la procedencia de la misma.

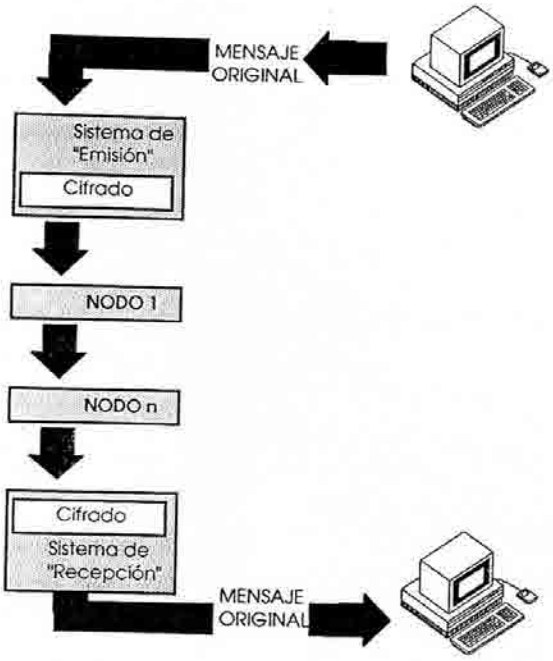


Figura 1.4: Cifrado extremo a extremo.

Entre las medidas de seguridad extremo a extremo hay que remarcar aquellas integradas en el nivel de aplicación. Por ser independientes de la arquitectura de red, permiten su realización aún cuando esta arquitectura no contemple el problema de seguridad. Pero si los tipos de máquinas donde se implementan estos mecanismos es muy diverso, entonces nos encontramos con el inconveniente de un diseño independiente para cada una de ellas, con el siguiente incremento de coste.

Tradicionalmente se denominaba cifrado extremo a extremo a cualquier cifrado realizado por encima del nivel físico de comunicaciones. En el contexto OSI esta definición es demasiado ambigua pues no especifica en qué nivel de la arquitectura se deben situar los mecanismos de seguridad. Según [ISO88], la ubicación de este tipo de protecciones puede realizarse en los niveles de red, transporte, y superiores. Sin embargo, esto no hace más que poner de relieve la excesiva simplicidad del modelo de dos alternativas. Por ejemplo, sigue siendo ambiguo el caso de ubicación de mecanismos de seguridad en el nivel 3, el de red. Puesto que las entidades de este nivel que intervienen en una comunicación no sólo se encuentran en los nodos extremos de la misma sino también en los *routers* intermedios, resulta entonces paradójico que los mecanismos de seguridad que se integren en ellas se califiquen como de extremo a extremo.

Las dos opciones presentadas no difieren únicamente en sus características de realización sino también en la naturaleza de los servicios de seguridad que permiten prestar. El cifrado en enlace tiene el inconveniente de que los datos son procesados en claro por los procesadores de comunicación de la red. Por otro lado, el cifrado extremo a extremo es más robusto, desde el punto de vista de las aplicaciones, ya que los datos no están accesibles en claro en los nodos intermedios. Pero esta alternativa tiene a su vez importantes inconvenientes, entre ellos a destacar: la complejidad de los sistemas de gestión de claves es en general es mayor puesto que el número de claves gestionar es mucho mayor (tener en mente que los recursos de la red, como los enlaces, son en general unos pocos recursos compartidos por muchas estaciones) [FOR94]; la información de los niveles inferiores a aquél donde se integran los mecanismos viaja en claro por la red, dejando sin cubrir amenazas que pueden ser importantes [STA95].

1.3.2 Arquitectura de seguridad de ISO

En el estándar ISO 7498-2 [ISO88] además de presentar conceptos tales como amenazas, servicios y mecanismos de seguridad relevantes en redes de datos, se presenta una matriz donde para cada uno de los servicios de seguridad considerados se identifica en qué niveles de la arquitectura OSI es posible ubicarlo (ver Tabla 1.1). Los argumentos empleados para realizar estas asignaciones han sido muy discutidos [FOR94], [MAD92]. No obstante, éste es seguramente el primer documento en la bibliografía donde se estudia de forma pormenorizada la integración de seguridad en arquitecturas de redes de ordenadores.

Nivel OSI	*Confidencialidad*	*Autentificación*	*Integridad*	*No repudiación*	*Control de acceso*
Aplicación	Sí	Sí	Sí	Sí	Sí
Presentación	Sí	Sí	-	-	-
Sesión	-	-	-	-	-
Transporte	Sí	Sí	Sí	-	Sí
Red	Sí	Sí	Sí	-	Sí
Enlace de datos	Sí	-	-	-	-
Físico	Sí	-	-	-	-

Tabla 1.1: Ubicaciones posibles para los servicios de seguridad en las capas OSI (según ISO 7498-2)

Este estándar no especifica cómo integrar exactamente los servicios en la arquitectura, es decir no especifica los protocolos OSI con seguridad "incorporada", sino que se limita a identificar en qué niveles parece viable ubicar los servicios. Jugando con las palabras, se puede decir que este documento no especifica protocolos estándares sino que es un estándar para la especificación de protocolos estándares. Es digno de comentar que además de identificar en qué niveles es posible integrar cada servicio, el documento enuncia una serie de recomendaciones acerca de la ubicación final de los mecanismos de seguridad. Por ejemplo, es bastante "curioso" que se desaconseje la ubicación de los mecanismos de confidencialidad en el nivel de enlace[8]. Como se ve en el capítulo 2, el primer protocolo estándar para seguridad en redes locales [IEE91] ubica precisamente el servicio de confidencialidad en el nivel 2. Realmente, algunos de los argumentos de esta recomendación carecen de peso como se reconoce en [ABR87]. Además, como se ve también el capítulo 2, los argumentos a favor de ubicar determinados servicios de seguridad precisamente en el nivel de enlace son muy importantes en el caso de redes de área local.

1.3.3 Modelo de arquitectura de seguridad de cuatro niveles

Según Ford [FOR94], el análisis realizado por ISO en [ISO88] es excesivamente detallado dando lugar a una arquitectura de seguridad redundante donde algunos servicios necesitan ser realizados simultáneamente en más de un nivel de la red y por tanto el resultado es notoriamente ineficiente. Este autor argumenta que, desde un punto de vista pragmático[9], el estudio de seguridad no requiere discernir entre las tres capas altas del modelo OSI, ni tampoco entre las dos inferiores. Esta idea le lleva a proponer un modelo de seguridad de cuatro niveles más simple que el de OSI, consiguiendo un modelo más eficiente y aplicable, sin llegar a la excesiva simplificación del modelo de enlace *versus* extremo a extremo. Además, este modelo tiene el atractivo adicional de presentar una correspondencia casi inmediata con la arquitectura de comunicaciones TCP/IP donde existen básicamente cuatro niveles: aplicación (procesos usuarios de los recursos de comunicación); transporte (protocolos TCP y UDP);

[8] De hecho, como refleja uno de los autores del estándar en el artículo [TAR85], antes de la publicación del mismo ni siquiera se consideraba la posibilidad de ubicar servicios de confidencialidad en el nivel de enlace.

[9] Este punto de vista "pragmático" consiste, según el autor, en que los protocolos de comunicación que intervienen en una arquitectura de red se pueden clasificar en cuatro grupos que corresponderían con los cuatro niveles de seguridad analizados en este apartado.

internet (protocolo IP); y de interfase, compuesto por los niveles de subred, enlace y físico (e.g. protocolo X.25 en redes públicas conmutadas, redes locales, líneas punto a punto, etc.).

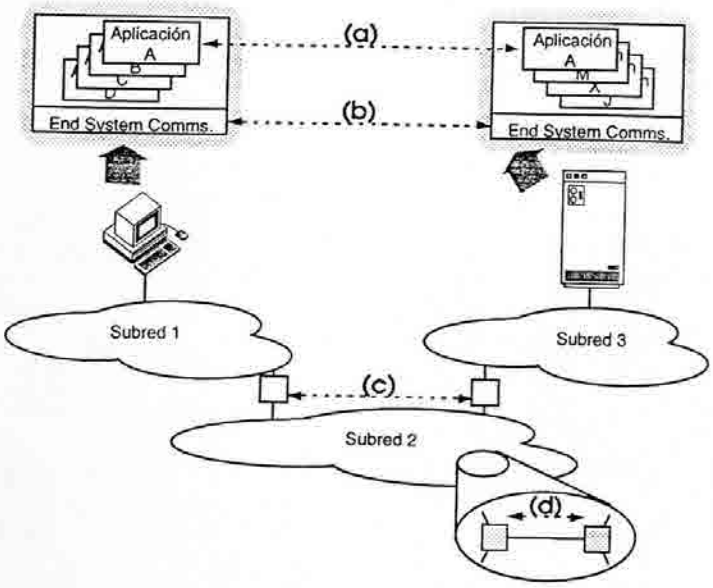


Figura 1.5: Ubicación de servicios de seguridad en una red de comunicación de datos

El modelo de Ford clasifica los mecanismos de seguridad de una red de comunicación de datos según su ubicación en uno u otro de los cuatro niveles que aparecen en la Figura 1.5:

a) Nivel de aplicación: elementos de seguridad que están integrados en las aplicaciones o bien que dependen directamente de las mismas.
b) Nivel de extremos (*end system comms.*): elementos de seguridad que integrados en las máquinas extremo (terminales, estaciones de trabajo, servidores de disco, etc.) dependen directamente de las entidades de comunicación extremo a extremo.
c) Nivel de subred: elementos de seguridad que están integrados en los nodos y protocolos de interconexión de redes, a fin de proveer protección sobre subredes que son más susceptibles de ataques que otras partes de la red.
d) Nivel de enlace: elementos de seguridad internos a una subred, cuyo propósito es proteger enlaces determinados que son más susceptibles de ataques que otros en esta subred.

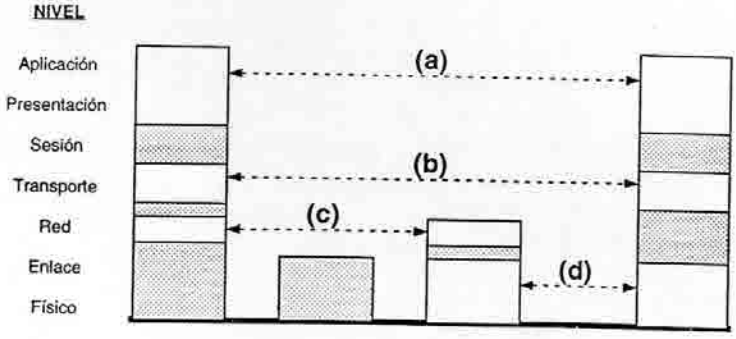


Figura 1.6: Correspondencia entre el modelo de cuatro niveles de seguridad con la arquitectura OSI.

En la Figura 1.6 se muestra la ubicación aproximada que les correspondería a estos cuatro niveles de seguridad en la arquitectura OSI.

Este modelo de arquitectura de seguridad no pretende recomendar qué servicios son convenientes en un entorno de comunicaciones de datos, ni mucho menos entra en detalles de cómo se deberían realizar exactamente los mecanismos correspondientes (i.e. los protocolos de seguridad). Este modelo simplemente intenta analizar las ventajas e inconvenientes de la ubicación de los mecanismos de seguridad en uno u otro nivel de la arquitectura de red en función de los requisitos de los servicios que se deseen desarrollar. Antes de realizar un servicio de seguridad en uno u otro de los niveles considerados hay que tener en cuenta los requisitos de este servicio. Simplificadamente, se pueden clasificar las características de estos requisitos en los siguientes grupos (interrelacionados entre sí):

- Mezcla de tráfico: En principio, cuanto menor sea el nivel considerado tendremos una mayor multiplexación de información procedente de diferentes aplicaciones o usuarios. Si la política de seguridad a seguir deja en manos de los usuarios o aplicaciones la especificación del tipo de protección que requieren los datos, entonces la realización y gestión de los mecanismos de seguridad será tanto más compleja cuanto menor sea el nivel donde se ubiquen. Por otro lado, si la política de seguridad especifica una misma protección para todos los datos independientemente de su naturaleza, entonces es más conveniente la integración de los mecanismos de seguridad en los niveles inferiores.
- Conocimiento de la ruta: Cuanto menor sea el nivel de comunicaciones considerado, mayor conocimiento se tendrá de la ruta y enlaces que seguirán los datos sobre la red. Si la necesidad de seguridad en la red viene dada por la presencia de algunas subredes o enlaces que no son de confianza, entonces será más eficiente la ubicación de mecanismos únicamente en los nodos de interconexión de subredes o en los enlaces que lo requieran. Supongamos que no se tiene confianza en algunos elementos de un subconjunto de la red, ya sean enlaces, nodos de interconexión, etc. En líneas generales, este subconjunto estará formado por enlaces, subredes o bien toda la red. Dependiendo de si se da el primer caso, el segundo o el tercero, será conveniente considerar mecanismos de clase (d) en el primero, de clase (c) en el segundo y de clase (b) o (a) en el tercero.
- Número de puntos de protección: Como se comenta en el primer punto, cuanto menor es el nivel considerado se tiene mayor multiplexación de datos procedentes de diferentes aplicaciones. Esto significa que el número de elementos de seguridad que deberá utilizar será mayor cuanto mayor sea el nivel donde se integran (suponiendo constante la cantidad de datos a proteger). Así, el coste total de instalación y de gestión será en principio mayor cuanto mayor sea el nivel considerado. Además, un factor extraordinariamente importante a tener en cuenta es que si la red donde se van a integrar los mecanismos de seguridad se encuentra ya en operación, la cantidad de elementos *software* y/o *hardware* de la misma que deberán ser sustituidos o modificados[10] será menor cuanto menor sea el nivel donde se integren estos mecanismos, minimizando de esta forma el coste y las "molestias" de la instalación de seguridad [JOH95].
- Protección de las cabeceras de los protocolos de comunicación: Un mecanismo de seguridad que esté ubicado en el nivel *n* de la arquitectura de red no puede proteger la información de control de los protocolos de los niveles inferiores, ya que como máximo puede proteger su propia información y la de los niveles superiores. Si se desea que los protocolos de comunicaciones sean seguros es aconsejable ubicar los mecanismos de seguridad en los niveles inferiores.
- Dependencia con las entidades de origen y destino: Determinados servicios dependen fuertemente de la identidad de las entidades emisora y/o receptora (e.g. autentificación de origen de datos). En este caso es conveniente ubicar los mecanismos de seguridad en el nivel en el que se encuentren estas entidades.

[10] Incluso puede ser imposible integrar los mecanismos de seguridad en algunos niveles. Por ejemplo, supongamos que el *software* de aplicaciones que utiliza la organización está compuesto por paquetes comprados en el mercado (no por *software* "hecho a medida"). En este caso, que resulta ser seguramente el más frecuente, es imposible modificar las aplicaciones para integrar en ellas la seguridad. Por otro lado, la construcción "a medida" de nuevas aplicaciones integrando la seguridad está fuera del alcance de la mayoría de organizaciones pequeñas o medianas.

2. Estado del arte en seguridad de redes locales

2.1 Seguridad en nivel de enlace de redes locales

Podemos definir una LAN como un conjunto de mecanismos físicos y lógicos (entidades de los niveles 1 y 2 de la arquitectura OSI) que permiten la comunicación de datos a velocidades relativamente altas sobre un canal de comunicaciones de tipo *broadcast* donde cada nodo puede escuchar transmisiones de las que no es el destinatario [KIR89][1].

En esta descripción se han de apuntar dos factores como los más relevantes en cuanto a seguridad se refiere: por un lado al entender como red local al conjunto de mecanismos de comunicación de niveles físico y lógico, el problema de seguridad queda delimitado a la inclusión de mecanismos de seguridad en alguno (o ambos) de estos niveles; por otro lado el carácter *broadcast* de los canales utilizados viene a potenciar la gravedad del problema en este tipo de redes.

Teniendo en cuenta que el documento ISO 7498-2 considera conveniente el ubicar mecanismos sólo de confidencialidad en los dos niveles inferiores de la arquitectura de red, y que además desaconseja que estos mecanismos se ubiquen aquí a favor de hacerlo en los niveles superiores, puede parecer injustificado el dedicar esfuerzo alguno al estudio de mecanismos de seguridad precisamente para los niveles inferiores. Sin embargo, existen diversos factores que permiten afirmar que estas recomendaciones, realizadas en principio para una arquitectura OSI "pura", no son aplicables al pie de la letra en el caso de redes con arquitecturas diferentes a este modelo, como es el caso de las comunicaciones sobre redes locales [PAR90], [GIR91], [SCH89]. Hay que tener en cuenta que las recomendaciones de ISO se realizaron considerando arquitecturas de red con todas las funcionalidades de los siete niveles del modelo OSI, y con el propósito último de establecer una guía para el desarrollo de protocolos estándares OSI donde se recojan servicios de seguridad. En el caso de las redes locales existen una serie de razones que invalidan los argumentos de ISO. Entre estas razones cabe destacar:

- En general, la mayor parte de los datos que viajan sobre una red local corresponden a datos intercambiados entre estaciones finales ubicadas en la misma red local. Por tanto en muchas aplicaciones es innecesario utilizar una arquitectura de red que realice todas las funciones especificadas en el modelo OSI (e.g. son innecesarias en estos casos las funcionalidades de enrutamiento pues todas las estaciones están conectadas, en principio, a un mismo medio físico). En la práctica esto se plasma en que en muchas redes locales encontramos familias de protocolos en las que no está presente alguna o

[1] Observar que en esta definición no se asume nada con respecto a la extensión geográfica de este tipo de redes. Aunque en la definición tradicional de red local se supone que esta no tiene una cobertura más allá de unos pocos quilómetros, actualmente el uso de enlaces de larga distancia y alta velocidad está permitiendo la construción de redes locales extendidas cuyas subredes distan unas de otras hasta miles de quilómetros.

ninguna de las funcionalidades de los niveles de red y transporte de OSI. Siendo imposible por tanto la realización de los servicios de seguridad tal como ISO los especifica para estos niveles. Un ejemplo claro de esta situación, entre otros, es la familia de protocolos LAT [MAL89], en la cual el nivel de aplicación está ubicado inmediatamente por encima del subnivel MAC (*Medium Access Control* [STA94a]) del nivel de enlace de datos.

- Teniendo en cuenta la anterior consideración, es obvio que aún queda la posibilidad de ubicar los servicios de seguridad en el nivel de aplicación. Sin embargo esta opción requiere precisamente que las aplicaciones se desarrollen teniendo como requisito funcional precisamente que incorporen mecanismos de seguridad. Si bien esto es posible para nuevas aplicaciones[2], el hecho es que la mayor parte del *software* que podemos encontrar funcionando actualmente en redes locales no tiene en cuenta ningún tipo de seguridad. Esto significa que la incorporación de mecanismos de seguridad exclusivamente en el nivel de aplicación implicaría en muchos casos un coste inviable de renovación del *software* [JOH95].
- La situación más usual en una red local es que se utilicen simultáneamente diferentes familias de protocolos. Algunas permiten el acceso distribuido a recursos tales como discos duros e impresoras, otras permiten el funcionamiento distribuido de bases de datos, monitorización y control de procesos industriales, otras permiten el acceso desde o hacia redes externas interconectadas mediante *routers*, *gateways*, etc. Aún a pesar de esta diversidad de familias de protocolos coexistiendo sobre una misma red local, existe un nivel común a todas ellas: el de enlace de datos. Ante esta heterogeneidad se puede concluir que la ubicación de mecanismos de seguridad en el nivel de enlace de una red local permite una única solución compartida por todas las arquitecturas de protocolos presentes, e independiente de ellas [GAS89].
- De la definición que se ha dado de red local se puede extraer que la comunicación de datos sobre este tipo de redes es especialmente vulnerable a ciertas amenazas [PAR90]. En la Tabla 2.1 se resumen las amenazas a las que se enfrenta una red local, las causas de estas y los servicios de seguridad necesarios para contrarrestarlas. El hecho de que el control de acceso al medio esté distribuido entre las estaciones y que las transmisiones sean de tipo *broadcast* permite que cualquier estación con acceso al canal pueda tanto escuchar todos los datos que circulan por la red como insertar nuevos datos. Teniendo en cuenta que en muchos casos la distribución geográfica de la red posibilita que un posible atacante tenga fácil acceso al medio físico[3], las ya importantes amenazas debidas a la naturaleza *broadcast* del medio adquieren una importancia muy considerable. Aunque las soluciones a estos problemas se pudieran dar por encima del nivel de enlace, es interesante realizar servicios de seguridad en los niveles inferiores, entre otras, por la siguiente razón: si bien la utilización de redes locales permite una distribución más eficiente tanto de información como de recursos dentro de una organización, en algunas configuraciones (ver nota al pie 3) la utilización de esta tecnología puede crear vulnerabilidades más importantes que el beneficio aportado. Por tanto, es deseable que las funciones de comunicación de la red local (i.e. los niveles físico y de enlace de datos) ya incorporen de por sí los mecanismos que contrarresten estas amenazas. De esta forma, por ejemplo, la utilización de una red local inalámbrica no supondrá más amenazas a la seguridad que la utilización de una red cableada y circunscrita al centro de cálculo de la organización. Que la utilización de una tecnología determinada no suponga la aparición de nuevas vulnerabilidades a la organización que decide usarla, es una característica muy deseable en cualquier tecnología de la información. En caso contrario, existe el riesgo de que a medida que los fraudes informáticos, escuchas electrónicas, etc. sean cada día más frecuentes, el mercado se muestre reacio a aceptar a aquellas tecnologías que demuestran ser especialmente vulnerables [EUR95].

---

[2] De hecho muchos paquetes de *software* que aparecen hoy en día en el mercado ya empiezan a incorporar mecanismos diversos de seguridad.

[3] Por ejemplo: cableados compartidos en edificios de oficinas; cobertura de la red en un campus universitario o en un parque industrial de forma que algunos segmentos de cableado son fácilmente accesibles desde la vía pública; redes locales inalámbricas; redes locales distantes interconectadas mediante líneas digitales de larga distancia y alta velocidad, permitiendo que el conjunto se comporte virtualmente como una única red local; etc.

Atributo LAN	Vulnerabilidad	Amenaza	Servicios convenientes
Transmisión	Cualquier estación puede transmitir utilizando cualquier dirección.	Suplantación de identidad. Utilización no autorizada del recurso.	Autentificación del origen de datos. Control de acceso.
Recepción	Cualquier estación puede escuchar cualquier transmisión.	Escucha no autorizada. Modificación de datos. Reactuación.	Integridad y confidencialidad de datos. Autentificación de origen de datos.
Espacio direccionamiento	No existe control de la dirección que utiliza cada estación.	Suplantación de identidad. Utilización no autorizada del recurso.	Autentificación del origen de datos. Control de acceso.
Dispersión Geográfica	Facilidad de acceso físico al canal de transmisión.	Todas	Integridad/ confidencialidad. Autentificación de origen de datos. Control de acceso.

Tabla 2.1: Amenazas y servicios de seguridad específicos a redes locales.

En las redes locales actuales el tráfico se puede dividir burdamente en dos subconjuntos. En el primero encontraríamos los datos que utilizan la red local como puente hacia otras redes, y por tanto que son transmitidos por torres de protocolos "similares" a OSI. En este caso la incorporación de seguridad puede seguir directrices similares a las marcadas por ISO. El segundo subconjunto sería el de los datos transmitidos por aplicaciones totalmente locales (tráfico puramente local). Para este segundo caso la incorporación de mecanismos de seguridad por encima del nivel de enlace puede ser o inviable o excesivamente costosa, por lo que es interesante el estudio de mecanismos de seguridad ubicables en el nivel de enlace de datos. Adicionalmente, esta ubicación de mecanismos permite en el primer caso la compartición por diversas familias de protocolos con la consiguiente optimización de costes, y además permite eliminar las posibles vulnerabilidades que el uso de tecnologías de comunicación sobre redes locales pudiera conllevar.

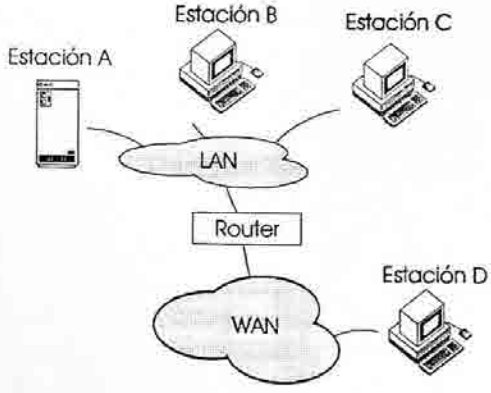


Figura 2.1: Ámbito de protección de mecanismos en nivel de enlace en redes locales.

No obstante, hay que reconocer que la integración de servicios de seguridad en el nivel de enlace de datos no es una panacea a todos los problemas de seguridad. Por ejemplo, es muy difícil la realización de servicios de seguridad que estén estrechamente ligados a la aplicación generadora de los datos (e.g. no repudiación de contratos electrónicos, autentificación de usuarios humanos, etc.). Además, hay que tener muy en cuenta el ámbito geográfico donde los servicios son garantizados. Dado que estamos hablando de mecanismos internos a la red local, los servicios son sólo garantizables durante el transito de los datos sobre la red local. Así pues, el tráfico local podrá tener garantizada una transferencia segura. Ahora bien, aquellos datos que se encuentren sólo temporalmente en tránsito sobre la red local (proceden o se dirigen a

otras redes conectadas a la primera), sólo tendrán garantías de transferencia segura sobre ésta, correspondiendo a mecanismos en niveles superiores el garantizar su seguridad más allá de la red local. Esta situación se ilustra en la Figura 2.1. Supongamos que en la situación de esta figura, una aplicación en la estación B accede a un fichero guardado en el servidor de ficheros "estación A", entonces dado que la transferencia es totalmente interna se podrá llegar a garantizar la seguridad de la misma con mecanismos en el nivel de enlace. Ahora bien si una aplicación en la estación C se comunica con otra en la estación D (accesible a través de la conexión de la LAN a una red de área extensa), entonces los mecanismos de seguridad en nivel de enlace no pueden proteger la comunicación más allá de los dominios de la red local. Esto se debe a que las entidades de nivel 2 de la estación C no pueden interaccionar con entidades de nivel 2 situadas en estaciones más allá del *router*. Si se requiere proteger las comunicaciones "exteriores" entonces son necesarios mecanismos en niveles superiores de la arquitectura de red, o bien protecciones en los enlaces de la red externa, tal y como se discute en el apartado de arquitecturas de seguridad del primer capítulo.

Desde un punto de vista menos restrictivo que el de [ISO88], los servicios de seguridad realizables en el nivel de enlace de datos son:

- Confidencialidad de datos. Se puede prestar confidencialidad a los datos de todos los niveles superiores cuando son transmitidos sobre la red local.
- Integridad de datos. Se puede garantizar la integridad a todos los datos de niveles superiores cuando son transmitidos sobre la red local.
- Autenticidad de origen de datos. Se puede garantizar la autenticidad de las direcciones origen de nivel 2 en los paquetes de datos.
- Control de acceso. Se pueden realizar mecanismos de control de acceso de forma que cada estación sólo acepte comunicaciones con otras estaciones autorizadas [GAS89], o bien se limite el acceso de algunas estaciones a determinadas zonas de la red [FUL93].

Este segundo capítulo está dedicado a realizar un recorrido por el estado del arte en lo que se refiere a los mecanismos que permiten realizar estos servicios de seguridad. La principal contribución a tener en cuenta en el área es la sin duda el estándar del IEEE de seguridad en redes locales [IEE89], [IEE91], [IEE92].

2.2 El estándar IEEE 802.10

2.2.1 Objetivos y características del estándar

En la línea de cubrir la laguna existente en cuanto a estandarización en el área de seguridad de redes locales de datos, en 1988 el IEEE organizó un grupo de trabajo con el objetivo de estudiar y especificar mecanismos de seguridad para las comunicaciones en redes de área local y metropolitanas "IEEE 802"[4] [KIR90], [FOR94] (en la Tabla 2.2 se detallan las diferentes redes locales especificadas por los proyectos IEEE 802). Los objetivos últimos de esta actividad estandarizadora es conseguir la especificación de mecanismos de seguridad interoperables entre sí, que puedan ser producidos por cualquier fabricante y así conseguir una disminución de los costes de este tipo de mecanismos (SILS, de *Standard for Interoperable LAN Security*). Actualmente ya han aparecido parte de los documentos (algunos en estado de "borrador") que conformarán el conjunto definitivo de estándares [IEE89] [IEE91] [IEE92], aunque todavía queda un camino considerable hasta que los primeros productos sean ofrecidos por la industria.

[4] Las tecnologías 802.3, 802.4 y 802.5, junto a Ethernet (compatible con 802.3) constituyen actualmente en gran medida el parque mundial de redes locales) [FRE92].

Proyecto IEEE	Título
802.3	CSMA/CD ("Ethernet")
802.4	Token Bus
802.5	Token Ring
802.6	DQDB (Metropolitan Area Networks)

Tabla 2.2: Redes locales 802.

Los servicios de seguridad contemplados son los de confidencialidad, integridad y autenticidad del origen de datos, así como el control de acceso a los servicios de comunicación prestados por estas redes locales. Los servicios de confidencialidad e integridad no son orientados a enlace. Es decir, la confidencialidad e integridad se garantizan de forma independiente para cada unidad de datos por separado. El estándar está organizado en cuatro partes (cada uno con categoría de estándar independiente[5]):

- El Modelo (IEEE 802.10a): Describe la base arquitectural de los mecanismos 802.10 y establece las relaciones entre las otras tres partes.
- Intercambio Seguro de Datos (SDE, *Secure Data Exchange*), (IEEE 802.10b): Define un protocolo de seguridad para proteger los datos transferidos entre estaciones en la misma red local.
- Gestión de Claves (IEEE 802.10c): Define la gestión de las claves requeridas por las entidades SDE.
- Gestión de Sistema/Seguridad (IEEE 802.10d): Define el soporte requerido a la gestión de red por parte del protocolo SDE.

La arquitectura 802 comprende dos subniveles. El subnivel MAC de control de acceso al medio (*Medium Access Control*), que se corresponde con la parte baja del nivel de enlace de datos y con parte del nivel físico de la arquitectura OSI. El subnivel LLC de enlace lógico de datos (*Logical Link Control*) formaría con el subnivel MAC el nivel 2 de la arquitectura OSI. El subnivel MAC presta esencialmente servicios de envío y recepción no fiable de tramas sobre el medio, mientras que el subnivel LLC puede prestar opcionalmente diferentes tipos de servicio: desde la transmisión no fiable de tramas hasta el establecimiento de conexiones fiables para el intercambio de datos entre dos estaciones en la misma red.

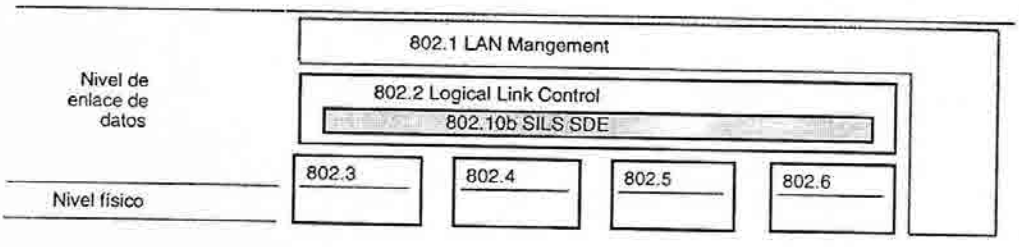


Figura 2.2: Arquitectura IEEE 802.

El protocolo SILS SDE se sitúa entre los subniveles MAC y LLC en la arquitectura 802, y su presencia es transparente a ambos subniveles (ver Figura 2.2). El propósito de esta transparencia es que la inserción de los mecanismos de seguridad SILS en una estación sea tan sencillo como cambiar su tarjeta controladora de red y que además, junto a la filosofía de gestión de este protocolo, se permita la coexistencia de estaciones con y sin protocolo SDE: una estación que soporte el protocolo SDE podrá comunicarse de forma segura con otras que también lo soporten, pero también será capaz de comunicarse con cualquier otra estación en la red local (sin protecciones, claro está). De esta forma el coste de instalar seguridad SILS en una

[5] Esto tiene el propósito de que los fabricantes puedan ofrecer productos que se acojan independientemente a: gestión de claves SILS (802.10c); intercambio seguro de datos SILS (802.10.b); o gestión de sistema/seguridad SILS (802.10.d).

red local es básicamente el coste de cambiar las tarjetas controladoras de red sólo en aquellas estaciones que puedan intercambiar información "sensible" sobre la red. De manera que el propietario de la red no necesite enfrentarse al considerable desembolso que significaría instalar tarjetas con seguridad en todas sus estaciones. La gestión de claves y de otros parámetros de los mecanismos de seguridad (*gestión de seguridad*) se realiza de forma independiente al funcionamiento del protocolo SDE. Cada estación donde se incorpora el subnivel SDE, posee también entidades de gestión de claves y de seguridad que se encargan de negociar con otras estaciones los parámetros de seguridad que se utilizarán en las comunicaciones entre ellas. Los valores negociados se depositan en una base de datos local a la estación (esta base, denominada SMIB, de *Security Management Information Base*, forma parte de la base de datos local MIB de información de gestión de la torre de protocolos, de *Management Information Base*). De esta manera, la entidad SDE simplemente debe obtener desde la base SMIB los valores de "configuración" con los que ha de operar.

2.2.2 El protocolo de seguridad SDE

En la estación emisora, la entidad SDE recoge la unidad de datos que le entrega el subnivel LLC para transmitir (en terminología OSI sería una SDE SDU, de SDE *Service Data Unit*), la procesa para prestarle los servicios de seguridad que correspondan y la unidad de datos resultante, una SDE PDU (*SDE Protocol Data Unit*), es entregada al subnivel MAC para que la transmita hacia la estación receptora, donde la entidad SDE correspondiente realizará el proceso inverso antes de entregar estos datos al subnivel LLC.

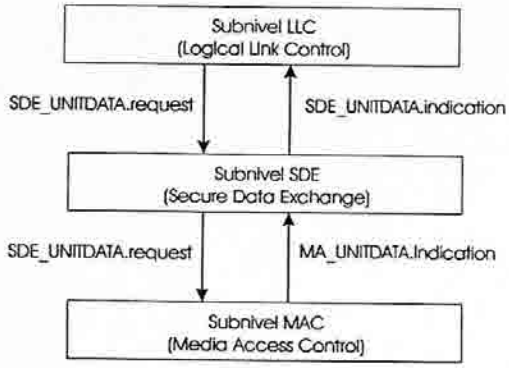


Figura 2.3: Las primitivas de la interfase del subnivel SDE son idénticas a la del subnivel MAC

Hay que señalar que el subnivel SDE se inserta entre los subniveles MAC y LLC de forma transparente a ellos. Es decir, la interfase que presenta el subnivel SDE al subnivel LLC es la misma que presentaría el subnivel MAC en caso de no utilizarse seguridad. Análogamente, el subnivel SDE utiliza la interfase superior del subnivel MAC. En la Figura 2.3 se ilustran estas interfases definidas con sus primitivas correspondientes.

En la Figura 2.4 se puede observar el proceso de encapsulamiento de la SDE PDU a partir de la SDE SDU. Hay que señalar que la protección por parte de la entidad SDE de la unidad de datos sólo se lleva a cabo si existe abierta con la estación destinataria lo que se conoce en el argot 802.10 como "asociación de seguridad".

Hay que señalar que todos los campos SDE son opcionales y su utilización dependerá de los parámetros de configuración de la asociación de seguridad con la estación destinataria. Pudiéndonos encontrar desde una SDE PDU idéntica a la LLC PDU original, en caso de tener desactivados los mecanismos de seguridad (esto es, no hay establecida una asociación de seguridad entre las estaciones origen y destino), hasta una SDE PDU con todos los campos mostrados en la Figura 2.4. Por otro lado, la recomendación 802.10 no especifica qué algoritmos criptográficos se han de utilizar para cifrar o para generar el código de comprobación de integridad (ICV, *Integrity Check Value*), siendo estos una opción a determinar

por el propietario de la red y los fabricantes de tarjetas de red. En la Figura 2.5 se detalla el formato de las tramas SDE con todos los campos posibles que puede incorporar (exceptuando campos de fragmentación, según se indica más adelante)

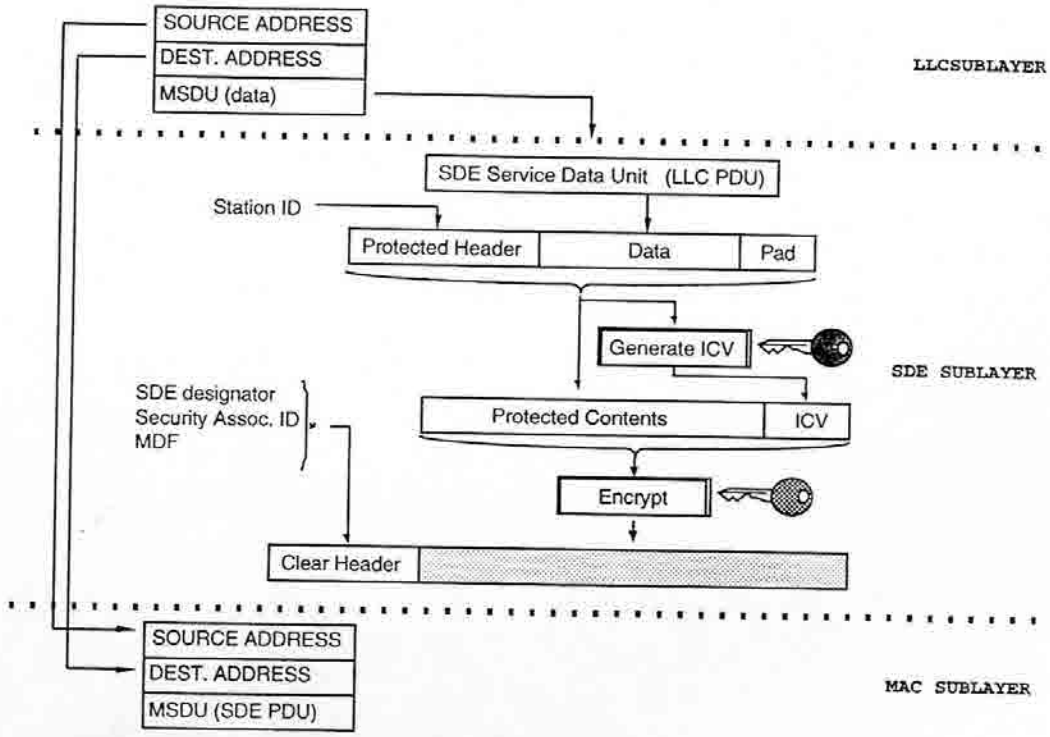


Figura 2.4: Encapsulamiento de una unidad de datos mediante el protocolo SILS SDE.

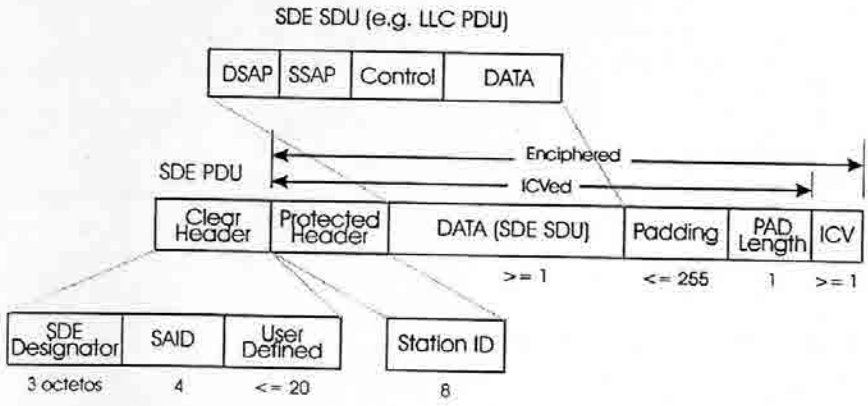


Figura 2.5: Formato de la trama SDE.

2.2.2.1 Fragmentación

Un importante problema asociado a este tipo de mecanismos de seguridad "instalables" en redes que se encuentran en operación se debe a que el subnivel MAC acepta unidades de datos con una longitud máxima (que suele fijarse por razones de equidad: esto es, evitar que ninguna estación acapare el medio). Dado que el subnivel SDE expande la unidad de datos que recibe del subnivel LLC, debería ser posible indicarle al subnivel LLC que la longitud

máxima de la unidad de datos ha disminuido. Desafortunadamente, en la mayoría de casos es imposible acortar la máxima unidad de datos que maneja la entidad superior debido a que en el momento de su desarrollo no se preveía que tal cosa pudiera ser necesaria. En este caso (seguramente el más frecuente), la única alternativa posible es obvia: cuando una SDE SDU sea demasiado larga para ser transmitida en una sola SDE PDU, deberá fragmentarse en el origen y ser recompuesta en destino por las entidades SDE correspondientes (ver Figura 2.6). Evidentemente esta nueva funcionalidad aumenta la complejidad del sistema, pudiendo tener esto un importante impacto en las prestaciones del mismo.

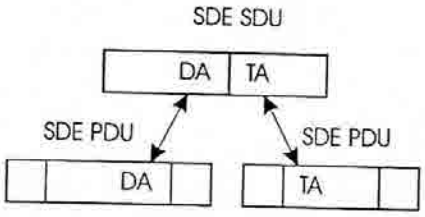


Figura 2.6: Fragmentación.

En uno de los apéndices del documento [IEE91] se describe un posible protocolo de fragmentación y reensamblaje. No obstante, este protocolo sólo tiene carácter de recomendación, es decir, no se considera parte del estándar SDE. En grandes rasgos, el funcionamiento de este protocolo es como sigue. Cuando la entidad de gestión establece una asociación de seguridad con alguna otra estación, negocia entre otros parámetros si esta asociación permitirá la fragmentación de tramas. En caso positivo, todas las tramas SDE de esta asociación de seguridad incorporan un nuevo campo, denominado *Flags*[6], a continuación del campo *Station ID*. Cuando la entidad SDE recibe una unidad de datos del nivel superior correspondiente a esta asociación y si es excesivamente larga para ser transmitida sobre una única trama, la divide en dos fragmentos que son procesados y transmitidos por separado hacia la entidad SDE receptora, la cual se encargará de su reensamblaje. A cada uno de los fragmentos se le añade un nuevo campo, denominado *Fragment Identifier*, que tiene el mismo valor para los dos fragmentos asociados (y sólo para ellos, incrementándose su valor en una unidad en cada operación de fragmentación). Esto permite a la entidad SDE receptora identificar inequívocamente los dos fragmentos que forman la unidad de datos que deberá entregar a su nivel superior.

El funcionamiento de este protocolo de fragmentación ha sido ilustrado en la Figura 2.7. Observar que si se desea que la identificación de los fragmentos sea absolutamente inequívoca, dada la longitud finita del campo *Fragment ID* (4 octetos), es conveniente que las claves utilizadas por la asociación de seguridad sean renovadas antes de utilizar de nuevo un mismo valor para este campo. Finalmente, se ha de señalar que después de pasar un cierto intervalo de tiempo desde que se recibió el primer fragmento, si la entidad SDE no recibe el segundo entonces descarta al primero. El valor de este intervalo es uno de los parámetros negociados por la entidad de gestión durante el establecimiento de la asociación de seguridad.

[6] Este campo tiene longitud de un octeto, utilizándose únicamente los dos bits de menor peso para los *flags* indicados en la Figura 2.7).

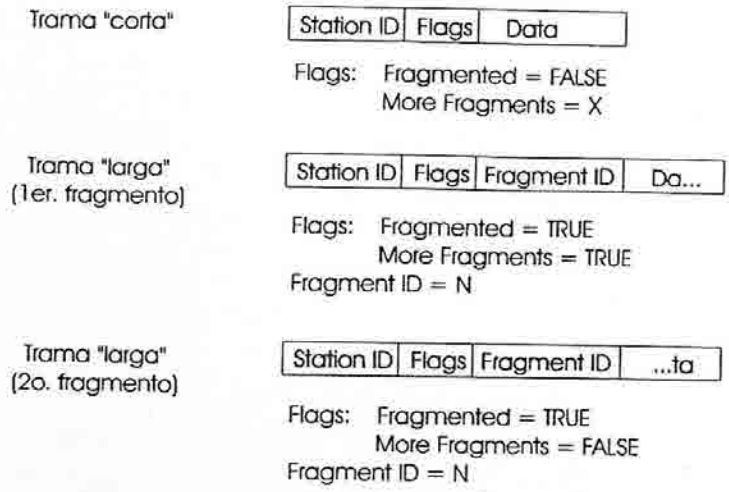


Figura 2.7: Tramas SDE de una asociación de seguridad con fragmentación.

2.2.3 Seguridad en redes SMDS

La única aplicación práctica del estándar IEEE 802.10 conocida en el momento de escribir este documento es el trabajo publicado en [FUL93]. Se trata de un estudio de incorporación de servicios de seguridad en redes SMDS (*Switched Multimegabit Data Service*) del que sus autores han construido un prototipo de demostración. El trabajo consta de dos partes fundamentalmente. La primera consiste en la integración del protocolo SDE en la torre de protocolos de comunicaciones utilizado en entornos SMDS. La segunda parte consiste a su vez en la construcción (mediante un diseño propio que no responde al estándar SILS) de un sistema de gestión de claves y de los mecanismos de seguridad.

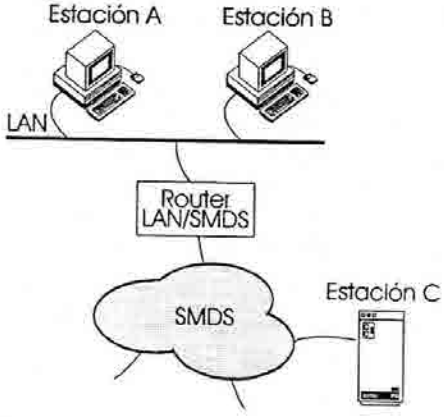


Figura 2.8: Modos de acceso a servicio SMDS.

SMDS es un servicio público de conmutación de paquetes no orientado a conexión, compatible con los servicios no orientados a conexión de las redes metropolitanas IEEE 802.6 (conocidas comúnmente por DQDB, de *Distributed Queue Dual Bus*). Estas redes presentan un especial interés debido a que actualmente los operadores públicos norteamericanos están ofreciendo servicios SMDS a velocidades de 1,544 y 44,736 Mbps (denominados enlaces DS1 y DS2, respectivamente) para interconectar redes locales distantes entre sí[7]. Este servicio tiene

[7] Por ejemplo, dos redes locales de una compañía que tenga sedes ubicadas en las costas este y oeste del país.

como objetivo actuar a modo de extensor transparente de redes locales, en el sentido de que permite obtener prestaciones similares a las de una LAN pero sobre áreas geográficas mucho mayores de las que son posibles usando únicamente tecnologías LAN.

Los requisitos básicos de seguridad de este tipo de redes son impuestos por dos factores a tener en cuenta. El primero es que la dispersión geográfica de los enlaces utilizados hace a estas redes especialmente vulnerables a todo tipo de ataques. El segundo factor es que los servicios que se prestan están sometidos a tarificación, haciendo esto que sea conveniente la incorporación de mecanismos que controlen el acceso a los servicios.

Una estación puede acceder a los servicios SMDS de dos maneras diferentes (véase la Figura 2.8):

- Directamente. En este caso la estación debe ser capaz de "dialogar" según el protocolo de acceso a la red, denominado SIP (*SMDS Interface Protocol*). Este protocolo se sitúa inmediatamente por encima del nivel físico y define la interfase SNI (*Subscriber Network Interface*) entre el usuario y la red.
- Indirectamente a través de un *router* que interconecta la red local en que se encuentra la estación con la red SMDS.

Desde el punto de vista de los equipos terminales SMDS (i.e. *routers* SMDS o estaciones terminales directamente conectadas a la red) los enlaces SMDS se comportan como una red local más, de forma que la arquitectura de red, una vez integradas las entidades de seguridad SDE, es la que se refleja en la Figura 2.9. En esta figura se ilustran los dos posibles casos: estación final indirectamente conectada mediante un *router* SMDS (estación A); estación final directamente conectada a la red SMDS (estación C). En este trabajo es muy interesante la integración de los mecanismos de seguridad en los nodos de interconexión (*router* SMDS/LAN). De esta forma estos mecanismos son compartidos por todas las estaciones ubicadas en una red local cuyas comunicaciones internas se pueden considerar seguras, permitiendo al mismo tiempo que la presencia de estos mecanismos sea transparente a estas estaciones.

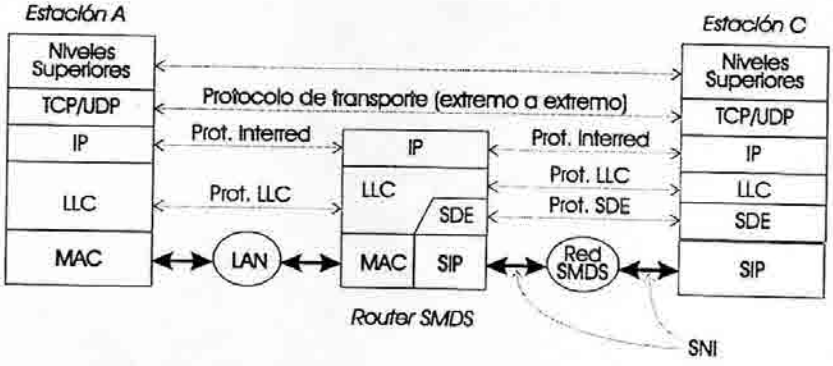


Figura 2.9: Arquitectura de seguridad SDE en el trabajo [FUL93]

2.3 Otros trabajos en seguridad de redes locales

Antes de exponer los mecanismos de seguridad para redes locales de datos que se proponen en este trabajo , se hace obligado realizar un repaso a los trabajos relacionados con este área que aparecen en la bibliografía especializada. A continuación se expone un sumario de las publicaciones encontradas más interesantes o afines a las ideas propuestas en este trabajo.

2.3.1 [AGN] *"Secrecy and Privacy in a Local Area Network Environments"*

Analiza de forma cualitativa (y algo superflua) la problemática de seguridad específica a redes locales. Lo más interesante de este trabajo es la propuesta de un sistema de asignación de claves que aparentemente consigue aumentar el número efectivo de claves utilizadas sin que lo haga el número real. La idea es como sigue. Para cada estación, a cada entidad local ubicada en el nivel *n* de la arquitectura de comunicaciones se le asigna una clave. Cuando se transmite un mensaje a una estación remota, al "descender" este mensaje por la torre de protocolos de la estación local, en cada nivel, la entidad que procesa el mensaje lo cifra con la clave de la entidad correspondiente en la estación remota. De esta forma, aparentemente el espacio de claves posibles (efectivas) dada una estación remota es el producto del número de entidades que tenga en cada nivel de la arquitectura de comunicaciones. No obstante, el número real de claves que se utilizan es la suma de todas las entidades en la red. Sin embargo, el autor no considera en su exposición un resultado bien conocido en el mundo criptográfico: el uso de cifrado doble (esto es cifrar dos veces un mensaje con dos claves diferentes) no requiere mucho más esfuerzo criptoanalítico que romper un cifrado simple (mediante el ataque conocido como *meet-in-the-middle* [DIF77])[8].

2.3.2 [SHI82] *"Security in Local Area Networks"*

De entre todas las referencias consultadas, ésta es la primera donde se menciona que la incorporación de mecanismos de seguridad en una red local puede suponer un coste en cuanto a las prestaciones de la misma. Las características que según el autor se deben mantener al introducir mecanismos de seguridad son: transparencia (el usuario no necesita saber nada de cómo funciona la red local); altas prestaciones (los datos se deben transferir entre máquinas en tiempos similares a los que requieren para ser "movidos" internamente en una máquina); interoperabilidad (se debe garantizar la comunicación entre subsistemas de diferentes fabricantes). Por lo demás, el artículo analiza la utilización de tarjetas controladoras de red (NIU, *Network Interface Unit*) confiables que garantizan que los datos clasificados en varios niveles de seguridad no puedan fluir hacia destinos no autorizados (en principio, el sistema no hace uso de mecanismos de cifrado). Esta idea está sacada del trabajo que compañeros del autor en MITRE Corp. publican en [SID82]. Además estudia la posibilidad de utilizar mecanismos de cifrado opcionales, y propone un protocolo de gestión de claves mediante un centro de distribución.

2.3.3 [SID82] "*A Multilevel Secure Local Area Network*"

En este trabajo se presenta un sistema de seguridad basado en los controladores de red confiables mencionados en el punto anterior (TIUs, *Trusted Interface Units*). La seguridad del sistema se basa en la confiabilidad de estos controladores con respecto a que no permitan la transferencia de datos clasificados en un cierto nivel de sensibilidad hacia sistemas clasificados en niveles inferiores. Para que el sistema sea seguro es necesaria la premisa que no puedan haber estaciones "anómalas" conectadas a la red. En caso contrario, los autores señalan la necesidad de utilizar "mecanismos de cifrado convencional" en algunos tramos de cableado, obviando el problema. No se menciona si se ha realizado algún prototipo del sistema propuesto, ni tampoco se analiza el impacto sobre las prestaciones de la red.

---

[8] Supongamos, por ejemplo, una red compuesta por dos estaciones que utilizan una arquitectura de comunicaciones de dos niveles (por tanto, que usa cifrado doble), y cada estación tiene 3 entidades de comunicaciones en cada nivel. Según el resultado de este autor, utilizando únicamente 2x(3+3)=12 claves se tiene un espacio efectivo de 2x3x3=18 claves. Sin embargo, utilizando el método *meet-in-the-middle*, el esfuerzo a desarrollar por un atacante que quiera romper todos los posibles enlaces de la red, es únicamente del mismo orden que si se utilizasen 12 claves con cifrado simple, quedando por tanto el "resultado" central de esta publicación en entredicho.

2.3.4 [HER88] *"Developing Ethernet Enhanded-Security System"*

En este interesante artículo se describe un sistema de seguridad para redes Ethernet desarrollado por *Digital Equipment Corporation* y denominado EESS (*Ethernet Enhanded-Security System*). El sistema está compuesto por controladores de red seguros, denominados DESNC (*Digital Ethernet Secure Network Controllers*), y un paquete de software para equipos VAX que realiza las funciones de centro de distribución de claves, denominado KDC (*Key Distribution Center*).

Cada controlador dispone de cinco puertos, a cuatro de los cuales se puede conectar una estación diferente y el quinto se conecta a la red Ethernet). Los servicios de seguridad desarrollados se basan en el cifrado de las tramas intercambiadas por dos estaciones cuyo acceso a red es a través de dos controladores seguros diferentes. El algoritmo de cifrado utilizado es el estándar DES (mediante encadenamiento CBC, *Cipher Block Chaining*). La existencia de este mecanismo de cifrado es totalmente transparente a las estaciones conectadas a la red: el cifrado de las tramas se realiza en el nivel de enlace de datos de forma que las estaciones pueden utilizar cualquier familia de protocolos en los niveles superiores (e.g. DECnet, TCP/IP, etc.). En la Figura 2.10 se ilustra el formato de las tramas de nivel de enlace una vez cifradas.

Destination Address		6 octetos
Source Address		6 octetos
IEEE 802 Header		10 octetos
Message Type		2 octetos
Encryption Identifier		2 octetos
Original Header		10 octetos
Secuence Number	*	4 octetos
Message Type Copy	*	2 octetos
Original Header	*	
Original Data Field	*	
Padding	*	0-7 octetos
MDC	*	2 octetos
Ethernet FCS		4 octetos

* Campos sometidos a encriptado

Figura 2.10: Formato de la trama cifrada en el sistema EESS.

El KDC tiene fundamentalmente asignadas dos funciones: la distribución de claves de cifrado y la administración de la política de control de acceso. El esquema de distribución de claves empleado es una variación del protocolo de Needham-Schroeder [NEE78].

El sistema posibilita tres tipos de comunicaciones. El primer caso se da con la comunicación entre dos estaciones conectadas a un mismo DESNC (p.ej. estaciones *B* y *C* de la Figura 2.11). Las tramas de una comunicación de este tipo no llegan a salir a la red y por tanto no requieren del mecanismo de cifrado. El segundo caso es el de la comunicación entre dos estaciones conectadas a DESNCs diferentes (p.ej. estaciones *A* y *B* de la Figura 2.11). Las tramas de una comunicación de este tipo se protegen con el protocolo de seguridad de la Figura 2.10. La clave de cifrado empleada se selecciona en este caso en función de las direcciones de las estaciones comunicantes. Finalmente, las comunicaciones del tercer caso se dan entre una estación directamente conectada a la red Ethernet y otra estación cuyo acceso a la red es mediante un DESNC (p.ej. estaciones *A* y *D* de la Figura 2.11). Estas comunicaciones sólo son posibles si previamente han sido habilitadas por el KDC.

Las funciones del KDC son básicamente determinar si una determinada comunicación en la que interviene una estación conectada a un DESNC está habilitada y establecer en caso necesario la clave de cifrado. Cuando un DESNC recibe de una estación *X* una trama dirigida a otra estación *Y* que no está conecta a él mismo, interroga al KDC acerca de: si esta comunicación es "legal" (i.e. está habilitada); y si hay que activar los mecanismos de cifrado para la misma. El KDC mantiene una base de datos sobre las comunicaciones permitidas, gestionada por el administrador del sistema según la política de seguridad de la organización.

Si la comunicación entre X e Y está permitida, se lo comunicará al DESNC de la estación X. Si además esta comunicación se debe proteger (i.e. la estación Y está conectada a otro DESNC), el KDC le comunicará la clave de sesión a emplear. A partir de este momento el DESNC de X se pone de acuerdo con el de Y acerca de los parámetros de configuración de lo que se denomina "asociación entre X e Y". La realización de todo el proceso descrito se lleva a cabo mediante una adaptación del protocolo de Needham-Schroeder. Una vez establecida la asociación de seguridad entre X e Y, sus DESNCs pueden cifrar y descifrar las tramas intercambiadas por estas estaciones. El tiempo de vida de una asociación está limitado, por lo que poco antes de que éste expire, se lleva a cabo de nuevo todo el proceso de negociación de parámetros de una nueva asociación.

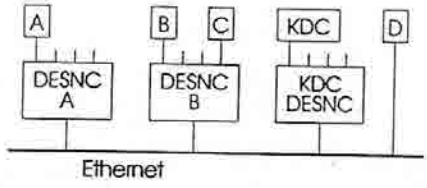


Figura 2.11: Ejemplo de configuración del sistema EESS

El servicio de confidencialidad de datos se consigue mediante el cifrado del campo de datos original. A su vez, el servicio de integridad se obtiene generando un MDC (*Manipulation Detection Code*, véase 1.2.3) mediante el cálculo de un CRC estándar de 16 bits a partir del campo de datos original y el cual es posteriormente cifrado. El servicio de autentificación de las estaciones comunicantes se establece en la fase de negociación de la clave de cifrado y se garantiza a lo largo de la comunicación mediante el mecanismo de integridad (el cual sólo da por buenas las tramas que han sido cifradas con la clave apropiada). Además la utilización de un número de secuencia previene los ataques de reactuación (y los de reflexión: el atacante captura una trama cifrada, intercambia las direcciones origen y destino y la reenvía de nuevo hacia la estación emisora). Este número de secuencia también hace que la inicialización del campo a cifrar sea en cada ocasión distinto, de forma que dos tramas cuyos campos de datos originales sean idénticos producen tramas cifradas completamente diferentes.

El funcionamiento transparente del protocolo de seguridad se basa, como se ha dicho, en la incorporación de nuevos campos a las tramas originales y en el cifrado de parte de las mismas. Esta incorporación de nuevos campos significa un aumento en la longitud de las tramas. Esto puede ser problemático en el caso de que la estación entregue al controlador una trama Ethernet de longitud máxima (el sistema de seguridad es transparente para las estaciones). En este caso el encapsulamiento del protocolo de seguridad da lugar a una trama de longitud mayor que la máxima permitida. Este inconveniente es un factor común en los protocolos de seguridad transparentes, como son el estándar 802.10 y la solución propuesta en esta tesis. La solución propuesta por los autores del EESS es la misma que se adopta en los otros dos casos mencionados: la entidad de seguridad emisora fragmenta la trama larga en dos tramas las cuales son reensambladas en la trama original por la entidad de seguridad receptora. Aunque los autores reconocen que esta fragmentación afecta a las prestaciones del sistema, no realizan ningún tipo de estimación de la importancia de esta degradación.

Este trabajo presenta un interés particular ya que presenta algunas características similares a las de algunos de los mecanismos de seguridad que se proponen en esta tesis. Esta afinidad nos ha permitido realizar estudios comparativos de las características y comportamiento de ambos diseños. No obstante, es de señalar que los autores de este sistema de seguridad no exponen ninguna evaluación de las prestaciones del mismo, que además algunos detalles del diseño no parecen suficientemente justificados (p. ej. el tamaño del código de detección de manipulaciones, 16 bits, es demasiado corto si lo comparamos con el "inmenso" contador de secuencia de 32 bits que se emplea) y que algunos aspectos son claramente mejorables (p.ej. durante todo el proceso de establecimiento de asociación de seguridad el sistema tiene en espera a una trama cuyo tiempo de latencia puede llegar a ser problemático).

2.3.5 [LAM89] *"Architectural Considerations for LAN Security Protocols"*

En este trabajo se estudia la problemática de seguridad específica en redes locales. Además lleva a cabo un análisis cualitativo de los pros y contras de ubicar los mecanismos de seguridad en, y entre, los niveles de la arquitectura de comunicaciones en una red local (i.e. subniveles de control de acceso al medio, MAC, y de control enlace lógico, LLC). La conclusión más interesante a la que llega es que la integración de mecanismos de seguridad dentro de un subnivel determinado es indeseable en general, pues una solución de este tipo requeriría la modificación de estándares de uso ya extendido. De esto se deduce que la ubicación idónea sea entre los subniveles MAC y LLC. Hay que señalar que el autor de este artículo es miembro del comité responsable de la elaboración del estándar IEEE 802.10.

2.3.6 [BAR89] *"The Impact of Security Service Selection for LANs"*

Este trabajo, en la misma línea que [LAM89] argumenta por qué en el estándar 802.10 se ubican los mecanismos de seguridad entre los subniveles MAC y LLC.

2.3.7 [SCH89] *"Secure Relays: An Alternative Approach to LANSEC"*

En este interesante artículo se estudia la posibilidad de incorporar mecanismos de seguridad en *bridges* utilizados para interconectar varias redes locales. Presenta esta solución como una solución de bajo coste con prestaciones iguales o mejores que la integración de mecanismos de seguridad en el nivel de enlace de datos de las estaciones.

2.3.8 [HOU89] *"Encapsulation Security Protocol Design for Local Area Networks"*

En este artículo se expone un protocolo de encapsulación de seguridad para redes locales. Este protocolo realiza los servicios de confidencialidad, integridad y autenticidad de origen de datos para la información mediante cifrado DES en modo CBC. El servicio de confidencialidad se garantiza mediante el cifrado del campo de datos de las tramas de nivel 2. El servicio de integridad se consigue a su vez mediante la adición de un código de autentificación de mensaje (según el estándar ANSI X9.19 [ANS86]) que es cifrado conjuntamente con los datos. Dado que se emplea un algoritmo de cifrado simétrico la autenticidad del origen de datos se garantiza conjuntamente con el servicio de integridad (sólo las tramas que den una comprobación satisfactoria del MAC pueden proceder del remitente con el que se comparte el secreto de la clave). En la Figura 2.12 se muestra el formato de las tramas una vez encapsulado el protocolo de seguridad. Finalmente, mencionar que el sistema de gestión de claves propuesto se basa en la utilización de un servidor de claves basado en el esquema de Needham-Schroeder.

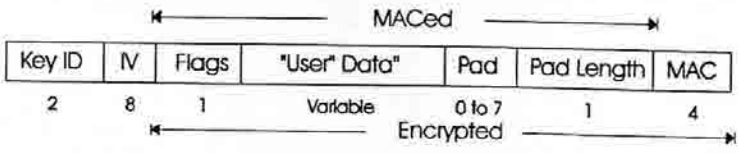


Figura 2.12: Formato de trama del protocolo propuesto en [HOU89].

2.3.9 [SIR94] *"A Secure Medium Access Control Protocol: Security versus Performances"*

En este artículo se presenta un mecanismo de control de acceso seguro para redes locales cuya función es evitar "fugas" de información a través de cualquier canal encubierto. Aparte del interés de este tipo de mecanismos para aplicación en entornos con fuertes requisitos de seguridad (p. ej. militar), resulta muy interesante y original el estudio de prestaciones que realizan los autores. De este estudio realizado mediante simulación, se muestran gráficas comparativas entre el comportamiento del protocolo propuesto y un CSMA-CD clásico en lo que respecta a tiempo de transferencia de las tramas y número medio de colisiones.

2.4 Seguridad *versus* prestaciones

Todo mecanismo de seguridad cuando es incorporado a una red debe aportar nuevos servicios, pero además, idealmente debería no degradar el comportamiento del sistema en ninguno de sus aspectos [SHI82]. Obviamente, la realidad dista de ser así. Por ejemplo, la utilización de algoritmos de cifrado en bloque requiere generalmente la utilización de rellenos que redondean la longitud del mensaje original a una que sea múltiplo del tamaño del bloque del algoritmo utilizado. El uso de estos rellenos supone un incremento del tráfico ofrecido a la red que puede llegar a afectar notoriamente las prestaciones de la misma.

Dadas las características de las redes locales en cuanto a

- retardo (muy pequeño, como mucho algunas decenas de milisegundos),
- ancho de banda útil (medio/alto, alguna decena de megabits por segundo (Mbps) en LANs tradicionales o centenar de Mbps en el caso de redes metropolitanas y algunas redes locales de reciente aparición),
- probabilidad de corrupción, pérdida o duplicación de tramas (prácticamente nulas),

y dado que muchas aplicaciones hacen uso (explícito o implícito[9]) de estas "buenas" prestaciones, es evidente que en toda integración de mecanismos de seguridad se debe cuidar con esmero cómo se van a ver alteradas estas (y otras) características.

Aunque actualmente existen multitud de propuestas de mecanismos de seguridad en el entorno de redes de comunicación de datos, es de reseñar la notoria escasez de bibliografía sobre evaluación de prestaciones [ZOR94], ya sea mediante modelación analítica, simulación o medidas de laboratorio sobre prototipos [JAI91]. En la mayoría de casos, los esfuerzos de los autores se circunscriben a el diseño de protocolos de seguridad, la verificación de los mismos (tanto en el sentido de su robustez como en el sentido "tradicional" de validación y verificación de protocolos), y su integración en la arquitectura de comunicaciones de la red. En cuanto a la evaluación de mecanismos de seguridad desde el punto de vista de las prestaciones, en la bibliografía sólo se pueden encontrar algunos pocos casos en los que después de implementar un demostrador del sistema propuesto (e.g. [FUL93] y [HER88]), se verifica que "desde el punto de vista del usuario" (subjetivo) no se aprecia degradación alguna en las prestaciones de la red [LAN89]. Dos interesantes trabajos, por lo excepcional, son [ZOR94] y [SIR94]. En el primero, el autor realiza un análisis genérico mediante teoría de colas del compromiso que puede haber entre seguridad y prestaciones en un sistema de comunicaciones. En el segundo se presenta un protocolo seguro de control de acceso al medio para redes locales alternativo a los "tradicionales" (i.e. Ethernet, Token Ring, etc.) y cuya

[9] Por ejemplo, el protocolo LAT [MAL89] esta diseñado específicamente para funcionar sobre redes locales Ethernet y hace uso explícito de las características de retardo [RIC90], pues exige que desde que una estación transmite un paquete hasta que recibe el reconocimiento del mismo no ha de transcurrir un tiempo mayor a unas pocas decenas de milisegundos. En el grupo de aplicaciones que hacen uso implícito de estas buenas prestaciones podríamos situar los servidores de disco, pues permiten que el usuario acceda a ficheros que físicamente están almacenados en discos remotos de la misma manera que lo hace con ficheros almacenados en su propia estación (i.e. utilizando los mismos procedimientos y obteniendo aproximadamente la misma velocidad de acceso).

principal característica es la anulación de cualquier posible canal encubierto en la red. Los autores de este trabajo realizan una evaluación por simulación de su mecanismo de control de acceso al medio, comparando los resultados obtenidos con los de mecanismos tradicionales.

2.5 Arquitectura de seguridad propuesta

2.5.1 Punto de partida de esta tesis

Como se discute en el apartado 2.1, existen diversos factores que hacen interesante el estudio y desarrollo de mecanismos de seguridad en el nivel de enlace de redes locales. El objetivo fundamental de estos mecanismos es proteger la transmisión de datos entre aplicaciones locales, y en segundo término proteger el tránsito por la red local de datos procedentes o encaminados hacia aplicaciones remotas ubicadas en estaciones de otras redes. Las principales razones que hacen interesante la ubicación de mecanismos de seguridad en el nivel de enlace son la dificultad de integrarlos en niveles superiores y la conveniencia de que las tecnologías de comunicación sobre redes locales no expongan a sus usuarios a vulnerabilidades indeseables[10]. La dificultad de integración en niveles superiores al de enlace se debe principalmente a tres factores:

- inexistencia de algunos de los niveles de la arquitectura OSI en muchas arquitecturas de comunicación sobre red local (en un entorno de red local puede bastar con las primitivas de comunicación que realiza el nivel de enlace);
- dificultad de renovar el parque del *software* de aplicaciones por nuevos paquetes que integren seguridad (ya sea por elevado coste o por inexistencia de *software* "seguro");
- y finalmente, a la diversidad de familias de protocolos de nivel superior al de enlace que conviven usualmente en las redes locales actuales.

El objetivo de esta tesis es precisamente el diseño, y evaluación, de mecanismos de seguridad apropiados para ser ubicados en el nivel de enlace de redes locales. Evidentemente, toda tarea de creación en ingeniería requiere un estudio previo del estado del arte a fin de decidir si está justificado el llevar a cabo un nuevo diseño, o por el contrario la existencia de soluciones suficientemente buenas desaconseja dedicar esfuerzo en el desarrollo de una nueva. En los anteriores apartados de este capítulose se ha realizado precisamente un recorrido del estado del arte actual en lo que respecta a seguridad en redes locales. Entre todos los trabajos analizados cabe destacar dos como los más afines a nuestras propuestas: el estándar de seguridad para redes locales IEEE 802.10 [IEE89] [IEE91] [IEE92], y el sistema EESS desarrollado por Digital Equipment Corporation [HER88].

En cuanto al sistema EESS presentado por Digital Equipment Corporation hay que decir que se trata de un estudio llevado a cabo por esta empresa con el fin de desarrollar un producto a fabricar y distribuir, en principio, de manera exclusiva. El hecho de que este producto sólo pueda ser ofrecido por un fabricante, junto a las limitaciones que la legislación en USA pone a la exportación de productos de seguridad, limita fuertemente la potencial importancia que pudiera tener en el mercado de seguridad para redes locales. Además de las limitaciones de este trabajo desde el punto de vista comercial, existen también razones técnicas que hacen que el sistema EESS no deje cerrado el campo de investigación sobre seguridad en redes locales. El diseño de este sistema, publicado en [HER89], presenta muchos aspectos aparentemente mejorables, como son:

- la gestión de claves (p. ej. el intervalo de establecimiento de una asociación de seguridad es notoriamente problemático);

[10] Según este punto de vista, los servicios de seguridad en los niveles inferiores de una red local se pueden considerar como un valor añadido a los servicios de transmisión de datos que tradicionalmente han venido ofreciendo las tecnologías LAN.

- algunas funciones de los controladores DESNC son redundantes con las funciones de los *bridges* que puedan haber en la red;
- el dimensionamiento de algunos parámetros de los mecanismos de seguridad es un tanto arbitrario (p. ej. se utilizan tan sólo 16 bits para la comprobación de integridad de las tramas cuando por otro lado se utiliza el doble para un contador de secuencia como protección frente a posibles reactuaciones);
- el problema de fragmentación de tramas no recibe suficiente atención;
- etc.

Además, también es notoria la falta de una evaluación del comportamiento del sistema que hubiera permitido un diseño más optimizado, así como una estimación de la bondad del mismo, posibilitando una comparación con otros diseños. Sin embargo, hay que reconocer que el balance entre "virtudes" y "carencias" de este trabajo es sin duda positivo ya que se trata del primer desarrollo de mecanismos de seguridad en el nivel de enlace de redes locales de que se tiene referencia en la literatura. Esto hace que ese trabajo sea muy valioso como punto de referencia en el diseño de este tipo de mecanismos de seguridad.

Por otro lado, parece natural ponerse en duda la necesidad de diseñar y evaluar nuevos mecanismos de seguridad cuando un organismo internacional como el IEEE ya está desarrollando el estándar 802.10 para seguridad en redes locales 802. Sin embargo, a pesar de esta actividad estandarizadora, existen diversos factores que justifican el estudio de nuevos mecanismos de seguridad para nivel de enlace que se realiza en esta tesis. Uno de ellos es de carácter cronológico: en el momento de iniciar este trabajo el desarrollo del estándar se encontraba todavía en una fase muy preliminar y, por tanto, muy dispuesta a "aprender" de las experiencias que la comunidad investigadora realizase en este área. Esto queda reflejado en la buena acogida que el trabajo a tenido en foros internacionales, como plasman entre otras, la publicación del artículo [REC93] en un *Journal* del mismo IEEE y la presentación del trabajo [SOR93] en el congreso *"1st ACM Conference on Computer and Communications Security"* organizado conjuntamente por el IEEE y el prestigioso ACM (*Association for Computer Machinery*). Otro factor que también justifica el desarrollo de este trabajo frente al desarrollo del estándar 802.10 son las características diferenciales entre los mecanismos propuestos en ambos trabajos. A grandes rasgos, el estándar está principalmente enfocado a la incorporación de mecanismos de seguridad en el nivel de enlace de las estaciones finales conectadas a red, mientras que los mecanismos que aquí se proponen se integran en el nivel de enlace de algunos nodos de interconexión (que denominamos *bridges* seguros). Además, el estándar se limita a describir pormenorizadamente los elementos de seguridad sin entrar, como es natural en un estándar, en una evaluación de prestaciones. Por ello, la evaluación de la incidencia que pueda tener la introducción de mecanismos de seguridad en las prestaciones de la red local se puede considerar cómo una contribución original y complementaria al desarrollo del estándar.

2.5.2 Escenario de comunicaciones considerado

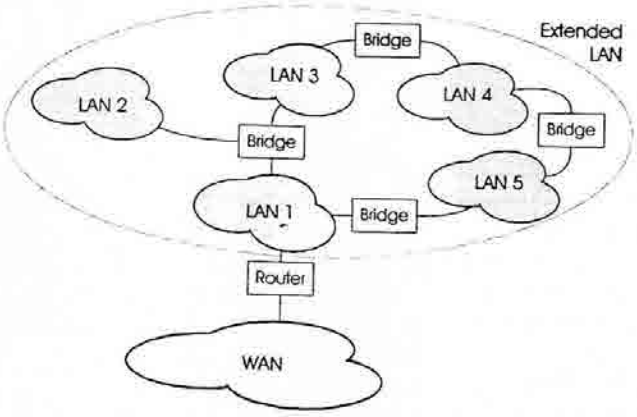


Figura 2.13: Configuración típica de una red local

En la Figura 2.13 se muestra una configuración genérica de una red local. Como se ilustra en esta figura, es usual que la red local de una organización esté formada por un conjunto de subredes locales interconectadas mediante *bridges*, formando lo que se conoce como red local "extendida"[11]. Un *bridge* es un dispositivo que permite que dos estaciones conectadas a redes locales distintas se puedan comunicar como si ambas estuvieran conectadas a la misma red local, construyendo de manera transparente una extensión de las redes locales originales.

En la Figura 2.13 se observa la posibilidad de que existan conexiones al "mundo exterior" mediante conexión a una (o más) red de área extendida. Dado que en este trabajo sólo consideramos comunicaciones de ámbito local ignoraremos la posible existencia de interconexión con redes WAN. Dicho de otra manera, sólo se consideran las comunicaciones entre las entidades de nivel de enlace dejando a un lado las comunicaciones entre entidades de niveles superiores de la arquitectura de red. Desde este punto de vista cualquier *router* o *gateway* que conecte nuestra red local extendida a una red WAN puede ser considerado como una estación más, "absorbiendo" en su interior todas las estaciones de la red WAN a la que da acceso, sin que ello signifique una limitación en el tipo de redes locales consideradas.

Las razones para utilizar redes locales extendidas mediante interconexión con *bridges* son muy diversas, entre ellas se pueden destacar las siguientes [STA83] [HAW84] [SCH89]:

- *Eficiencia.* Un *bridge* aísla los tráficos locales de cada red conectada a uno de sus puertos, de forma que sólo deja "pasar" aquellas tramas cuyas direcciones origen y destino corresponden a estaciones ubicadas en subredes diferentes. Si la mayor parte del tráfico en una subred corresponde a tráfico local (como debe ocurrir en una red bien planificada), la capacidad total del conjunto será mayor a la que se obtendría en caso de conectar todas las estaciones a una única red local. A esta función se le suele denominar "filtrado de tráfico".
- *Interconexión de LANs diferentes.* La evolución más frecuente de la red local de una organización ha sido la siguiente. Cada departamento instala su propia red local, decidiendo únicamente por criterios internos el tipo de tecnología a instalar. Esto lleva a que cuando se detecta la conveniencia de interconectar a toda la organización, resulta que es imposible hacerlo simplemente "uniendo" los cables, o mediante repetidores, debido a que se han utilizado diferentes medios físicos (cableados mediante par trenzado, cable coaxial, etc.) o bien se utilizaron redes locales totalmente diferentes (802.3 / Ethernet, 802.4, 802.5, etc.). En esta segunda situación sólo es posible

[11] A partir de este punto, utilizaremos el término "red local", o simplemente "red", para denotar a toda una red local extendida y el término "subred local", o "subred departamental", para denotar una de las redes locales que componen la red extendida.

interconectar toda la red usando *bridges*, *routers* o *gateways* (ver Figura 2.14), dependiendo de hasta qué nivel hayan diferencias en las torres de protocolos utilizados. En general, la alternativa de los *bridges* es más atractiva que la de *routers* o *gateways*, debido a que su mayor sencillez permite obtener mejores prestaciones a costes menores [STA83], [HAJ91]. Por otro lado, el uso de repetidores está además limitado por los factores que se discuten en el siguiente punto.

- *Incremento de cobertura y número de estaciones.* La cobertura geográfica de una red local y el número de estaciones que se le pueden conectar son parámetros acotados en la mayoría de tecnologías LAN (debido principalmente al retardo [STA94b], a la atenuación de propagación y a limitaciones eléctricas en las redes con colisión). El uso de repetidores permite solventar los problemas de atenuación y limitación eléctrica de la señal, pero no el de retardo de propagación. Puesto que un *bridge* es un dispositivo de tipo *store-and-forward*, desde el punto de vista "físico" de cada una de las redes locales que conecta el dispositivo, éste se comporta como una estación más. De esta manera las limitaciones en cuanto cobertura geográfica y número máximo de nodos sólo son aplicables a cada una de las subredes por separado, consiguiéndose una ampliación de ambos parámetros. Esta característica permite incluso interconectar de forma transparente LANs distantes entre sí mediante lo que se conoce como *bridges* "remotos" (también denominados *half-bridges*): a cada una de las redes se conecta una "mitad" del *bridge*, y ambas mitades se conectan entre sí mediante un enlace remoto que hace las veces del *bus* interno en un *bridge* "normal" (este enlace puede ser una línea telefónica dedicada, un radioenlace, o incluso toda una red de comunicación de datos[12]). Evidentemente, a cambio del incremento en la cobertura geográfica y del número máximo de estaciones hay que pagar como precio el retardo debido a que, al tratarse de un dispositivo *store-and-forward*, cada trama debe ser transmitida de nuevo en cada *bridge* que cruza en su camino[13].
- *Fiabilidad.* Es bien conocido que las redes locales son muy susceptibles frente a "pequeños" fallos: si el cable se rompe en un sólo punto o una estación empieza a comportarse anómalamente es muy fácil que la red entera deje de funcionar[14]. La construcción de la red local de una organización como red local extendida permite aislar el alcance de estos fallos al interior de las subredes donde se produzcan, permitiendo que el resto puedan continuar funcionando normalmente.
- *Seguridad.* La función de filtrado de tráfico en un *bridge* no sólo permite aumentar la utilización de la red, sino que también sirve como mecanismo elemental de confidencialidad y control de acceso limitando que estaciones pueden realizar comunicaciones más allá de la subred a la que están conectadas.

Hoy en día existen básicamente dos tipos estándares de *bridges*, los transparentes y los *source routing* [TAN89]. En las redes donde se utiliza la filosofía *source routing*, cada estación conectada a una subred recibe una dirección que le es exclusiva dentro de este ámbito. A su vez, cada subred y cada *bridge* reciben también un identificador numérico. De esta forma, cuando una estación quiere enviar una trama, además de indicar la dirección de la estación destinataria añade un campo indicando la secuencia de redes y *bridges* por los que la trama deberá pasar para llegar hasta la subred destino.

---

[12] Por ejemplo, con este tipo de *bridges* se pueden interconectar, de manera transparente, dos redes locales similares a través de una tercera incompatible con ellas [VAR90]. En este caso las tramas originales son literalmente transportadas en el campo de datos de tramas con formato adecuado a la red intermedia. Este tipo de *bridges* suelen recibir el nombre de "*bridges* de encapsulamiento" (o incluso "*tunnelling bridges*" debido al que el conjunto se comporta como un "túnel" que une las dos redes locales distantes.

[13] De hecho, este retardo debido a la necesidad de recibir toda la trama en el bridge antes de procesarla limita en algunos casos el número de estos dispositivos que se pueden ubicar en el camino entre dos estaciones cualesquiera. Por ejemplo, si en una red extendida se utiliza el protocolo LAT [MAL89], el fabricante limita este número máximo a 10 *bridges*.

[14] Los fallos eléctricos como el mencionado, pueden ser aislados simplemente utilizando repetidores "inteligentes". No obstante, esta solución ya no es válida para fallos "lógicos" (p. ej. una estación transmite continuamente).

Por otro lado, en las redes donde se utiliza interconexión "transparente", cada estación recibe una dirección que le es exclusiva (ya sea en un ámbito "universal" o solamente en el de la red extendida). Cuando una estación quiere enviar una trama, solamente debe añadir la dirección destino y la transmite, sin "preocuparse" de qué camino sobre la red deberá seguir, siendo los *bridges* intermedios los que se encargarán de todo el trabajo de enrutamiento. Para que esto sea posible, por cada trama recibida en uno de sus puertos el *bridge* transparente se "apunta" en una tabla asociada a este puerto la dirección origen de la misma (a este procedimiento se le conoce como algoritmo de aprendizaje de Baran). Al cabo de un cierto periodo de aprendizaje, el *bridge* conocerá la ubicación de casi todas las estaciones con respecto a sus puertos. De esta forma cuando el *bridge* recibe una trama, analiza la dirección destino de la misma para saber en qué puerto está accesible la estación destinataria y la retransmite hacia éste. En caso de que la estación destino esté en el mismo puerto que la remitente, la trama en cuestión será simplemente descartada. Finalmente, en el caso de que desconozca la ubicación de la estación destinataria se retransmite la trama hacia todos los puertos (exceptuando, claro está, aquél por el que se recibió). Hay que observar que para que este procedimiento sea correcto es necesario que no existan caminos cerrados ("bucles") sobre la topología de la red.

En rasgos generales, los *source routing bridges* son mucho más sencillos de realizar que los *bridges* transparentes. No obstante presentan el inconveniente de que todas las estaciones deben conocer bien la topología de la red a la que están conectadas, requiriendo de su colaboración, y además cada trama ve incrementada su longitud con la información de enrutamiento. Tradicionalmente, los bridges transparentes se han venido utilizando en entornos 802.3 / Ethernet, mientras que los de tipo *source routing* lo han sido en redes 802.5 (Token Ring). No obstante, actualmente está en marcha la estandarización de un nuevo tipo de *bridge* que intenta combinar las virtudes de los dos anteriores y permitir la coexistencia de ambas filosofías en un mismo entorno de red local extendida: el *transparent source routing bridge* [LAT92] [NET91].

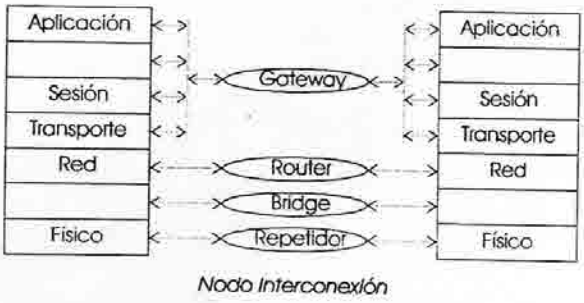


Figura 2.14: Repetidores, *bridges*, *routers* y *gateways* en el marco OSI

Hasta aquí se ha considerado una red local general, donde cada una de las subredes utiliza una tecnología de control de acceso al medio diferente (i.e. redes Ethernet, Token Ring, Apple Talk, FDDI, etc.)[15]. También en un caso general, es posible encontrar bucles en la topología de la red extendida. Esto se debe a la posible utilización de más *bridges* de los que son estrictamente necesarios para conectar totalmente la red (i.e. mediante una topología en árbol en la cual dados dos estaciones cualesquiera, existe exactamente una camino que las conecta)[16]. No obstante, a efectos de simplificación del problema, en este trabajo se suponen ciertas restricciones tanto con respecto a las tecnologías LAN utilizadas por las subredes como con

[15] En [KÜM87] se encontrará una comparación entre las tecnologías más utilizadas en redes locales.

[16] La creación de caminos redundantes con *bridges* adicionales tiene el objetivo de independizar el sistema frente a fallos en un sólo *bridge* y/o de dotar de multiplicidad de caminos que palien posibles problemas de sobrecarga. En el caso de *bridges* transparentes, éstos usualmente tienen la capacidad "autoconfigurarse" de forma que se eliminan los bucles que en caso contrario darían lugar a lo que se conoce como "efecto Chernobil". Para ello, durante una fase de inicialización todos los *bridges* se ponen de acuerdo entre sí (mediante un protocolo denominado "*spanning tree*") para que algunos de ellos se desconecten y queden a la espera de activarse en caso de que una parte de la red quede aislada por alguna anomalía.

respecto a la topología de la red extendida (ver Figura 2.15). La principal característica de este tipo de redes es la interconexión de las redes departamentales a través de otra subred sobre la cual en principio sólo circula el tráfico interdepartamental (denominada *backbone* en la literatura especializada). Esta estructura tiene la ventaja de permitir un diseño modular de la red extendida permitiendo, por ejemplo, cambiar fácilmente la subred *backbone* por otra de mayores prestaciones cuando éste está suponiendo un "cuello de botella" en el sistema (p.ej., debido a un crecimiento sostenido en el tráfico que soporta[17]). Este tipo de topología es el utilizado normalmente para dar cobertura a todo un campus universitario, un campus industrial[18], en los sistemas de cableado en edificios, etc..

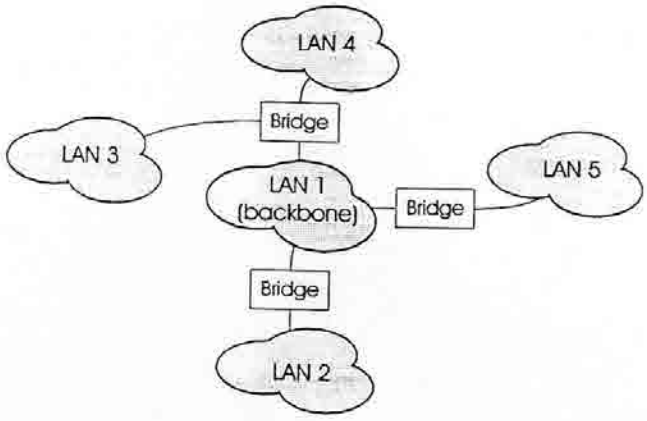


Figura 2.15: Red local extendida articulada alrededor de una subred *backbone*.

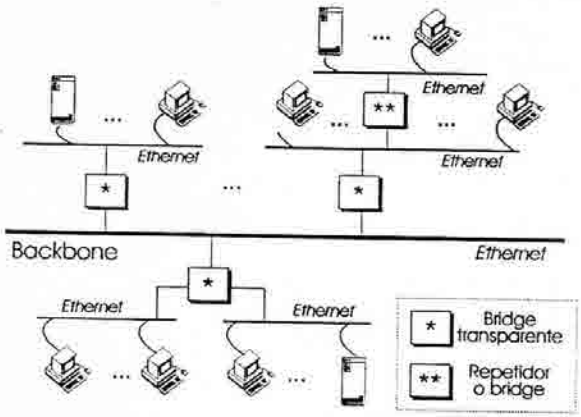


Figura 2.16: red local extendida de referencia.

En la Figura 2.16 se muestra la configuración específica que constituye el escenario de comunicaciones de referencia para el diseño los mecanismos de seguridad propuestos. Como se puede observar, se trata de un caso particular de red extendida con *backbone*, en la cual no existen bucles y todas sus subredes son segmentos Ethernet / 802.3[19]. Dentro de las líneas de

[17] Por ejemplo, esta es la experiencia sufrida en la red local extendida de la Universidad Politécnica de Cataluña, la cual inicialmente utilizó como *backbone* un segmento Ethernet, pasando despues a utilizar un anillo FDDI, y más recientemente un *backbone* ATM.

[18] Por este motivo, este tipo de redes extendidas son denominadas en ocasiones "redes de tipo campus".

[19] Hay que señalar además, que la discusión sobre los mecanismos de seguridad que aquí se hace únicamente considera *bridges* con dos puertos. No obstante, su generalización al caso de múltiples

continuación de este trabajo, se considera la ampliación de los mecanismos de seguridad propuestos a configuraciones más generales. No obstante, hay que señalar que el escenario considerado constituye seguramente uno de los más frecuentes en el parque actual de redes locales extendidas.

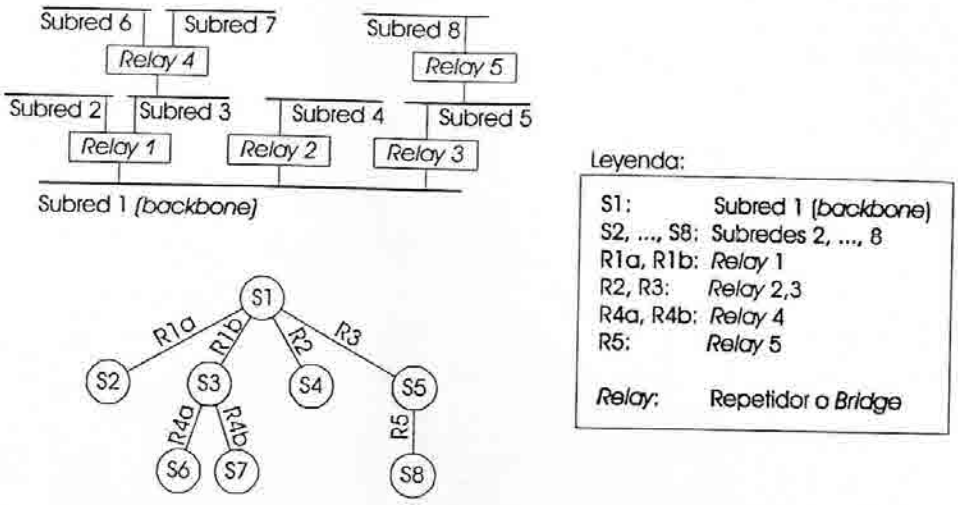


Figura 2.17: Red extendida sin ciclos y su árbol equivalente

En la Figura 2.17 se muestra un modelo más formal de la topología de esta red local de referencia. Dado que esta red de referencia no presenta bucles, permite ser modelada mediante un grafo sin caminos cerrados, esto es, por un árbol como el de la Figura 2.17. En este árbol cada subred es simbolizada mediante un nodo y la interconexión de dos subredes (por repetidor o un *bridge*) es simbolizada mediante una rama que une los dos nodos correspondientes. En este modelo, se ha tomado como nodo raíz a la subred *backbone*[20]. Hay que observar también que en este modelo cada repetidor o *bridge* con *N* puertos queda simbolizado por *N-1* ramas que interconectan a un determinado nodo con *N-1* de sus descendientes. A partir de este punto, a efectos de simplificar la exposición, se considerarán únicamente *bridges* con dos puertos.

2.5.3 Arquitectura de seguridad propuesta para redes Ethernet / 802.3 extendidas

La ubicación en el nivel de enlace de datos de los mecanismos de seguridad a desarrollar es una premisa de este trabajo. Una cuestión paralela que se debe resolver es la ubicación en la arquitectura física de la red. Esto es, decidir en qué puntos de la red local es conveniente integrar los nuevos mecanismos de seguridad. Para analizar esta cuestión es conveniente tener en cuenta el esquema de la Figura 2.18. En esta figura se muestran las tres alternativas básicas en cuanto a una posible localización física de los mecanismos de seguridad (puntos A, B y C).

puertos es trivial. La principal razón de haber realizado esta simplificación ha sido simplemente permitir una realización más "cómoda" de los prototipos de laboratorio con los que se ha experimentado.

[20] Observar que el modelo es igualmente válido para una red local extendida en la que no hay ninguna subred con las funciones de *backbone*, siempre que no hayan ciclos. En este caso podríamos elegir arbitrariamente cualquier subred como nodo raíz.

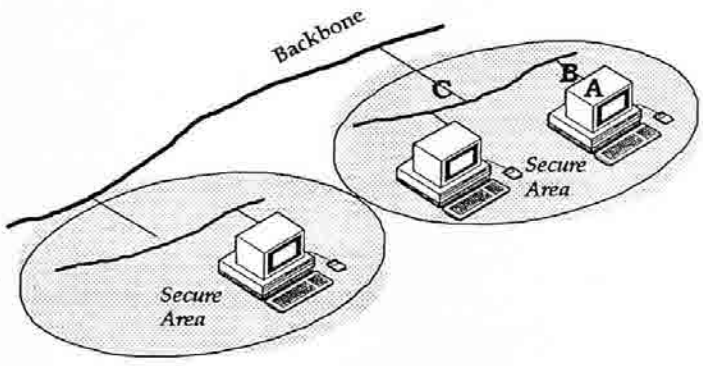


Figura 2.18: Alternativas de ubicación física de mecanismos de seguridad en una red local

En la alternativa A (integración en las estaciones[21]), es necesario una realización independiente para cada modelo de estación que se tenga en la red. Esto significa que un hipotético fabricante de este tipo de mecanismos de seguridad deberá desarrollar todo un catálogo de versiones diferentes, una para cada modelo de estación en el mercado (y por tanto deberá ampliar este catálogo cada vez que aparezca una estación de arquitectura nueva). De esta manera, el coste de "puesta en mercado" de esta alternativa es proporcional al número de modelos diferentes de estación. Además, el número de dispositivos a instalar vendrá dado por el número de estaciones cuyas comunicaciones se desee proteger. Finalmente, hay que tener en cuenta que en esta alternativa la operación y administración de los mecanismos de seguridad puede suponer un impacto importante en el funcionamiento de la estación.

En la alternativa B (integración en la interfase entre la estación y la red) el diseño de los dispositivos que realicen las funciones de seguridad es independiente de la arquitectura de las estaciones[22]. Esta alternativa será pues más atractiva desde el punto de vista del fabricante. No obstante, dado que el número de dispositivos a instalar viene dado por el número de estaciones a proteger, igual que en la opción A, el coste para el propietario de la red será similar a esta primera opción. Esta opción tiene la virtud adicional de que los mecanismos de seguridad son totalmente externos a las estaciones, sin que tener necesidad de añadir a éstas nuevas funciones que pudieran afectar su comportamiento.

Finalmente, la alternativa C (integración en el punto de interconexión de las subredes departamentales al *backbone*) presenta las mismas virtudes que la opción B frente a la A. Esto es, un único diseño de los dispositivos de seguridad y operación independiente de las estaciones. Pero esta tercera opción presenta además la virtud de con un solo dispositivo de seguridad se protegen simultáneamente las comunicaciones de todas las estaciones ubicadas en una subred departamental, reduciendo de manera muy significativa el coste de instalación (ahora vendrá dado por el número de subredes y no por el de estaciones).

Los mecanismos de seguridad propuestos en este trabajo corresponden, en principio, a la alternativa C. Esto es, las funciones de protección de las comunicaciones de nivel de enlace son llevadas a cabo por dispositivos ubicados físicamente en la interconexión de las subredes departamentales con el *backbone*. Pudiera parecer en principio que el área geográfica dónde esta alternativa permite proteger las comunicaciones es muy limitada (y rígida). En la Figura 2.19 se muestra una situación más genérica que la de la Figura 2.18 pero que es equivalente a ésta. Aquí se ha descompuesto la red extendida en dos áreas: la primera estaría constituida por el *backbone* de la red o bien por todo un subconjunto de la red sobre el que se desean proteger las comunicaciones, y al que denominaremos "núcleo inseguro"; la segunda estaría formada por todas las subredes periféricas dónde hay estaciones cuyas comunicaciones sobre el núcleo se desea proteger. A estas subredes periféricas las denominaremos "subredes protegidas". Desde esta perspectiva la ubicación de los dispositivos de seguridad puede ser el punto de

[21] El estándar 802.10 está pensado principalmente para esta alternativa. No obstante, el estándar apunta (aunque vagamente) la posibilidad de ser integrados en *bridges*, según la alternativa C.

[22] El sistema EESS para redes Ethernet [HER88] se puede clasificar en esta alternativa, ya que los dispositivos DESNC en los que se basa son controladores de comunicaciones seguras a los que se pueden conectar hasta 4 estaciones.

conexión de una sola estación al resto de la red[23], o el de un centro de cálculo, el de todo un edificio del campus al *backbone*, etc.

Figura 2.19: Núcleo Inseguro y Subredes Protegidas

Desde el punto de vista del árbol topológico presentado en la Figura 2.17, el lugar donde ubicaremos los mecanismos de seguridad serán algunas de las ramas de este árbol. De forma que cada una de las subredes protegidas de la Figura 2.19 se corresponderá con todo un subárbol que cuelgue de una rama donde se ha ubicado un dispositivo de protección. A su vez el núcleo inseguro de la red se corresponderá con el resultado de "extraer" del árbol original todas las subredes protegidas. En la Figura 2.20 se ilustra esta idea. Observar que en este modelo se supone que sólo existe un núcleo inseguro en la red. Es decir el subconjunto de la red donde se pueden producir ataques a las comunicaciones es conexo. Esto implica que en una subred protegida sólo hay un dispositivo de seguridad, ubicado precisamente en su conexión con el núcleo inseguro.

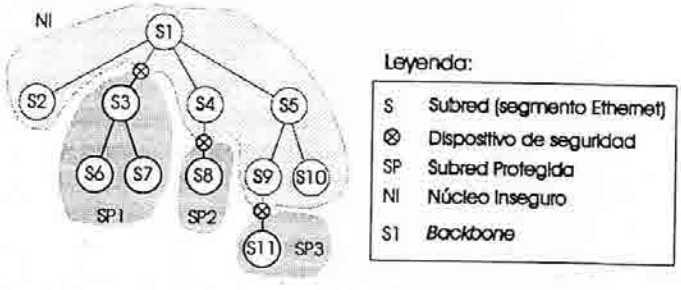


Figura 2.20

Llegados a este punto ya se puede empezar a pensar qué debe hacer el dispositivo de seguridad. Para ello se debe partir de las premisas en cuanto a ubicación lógica y física en la arquitectura de la red: este dispositivo desarrollará sus funciones en el nivel de enlace y tendrá una ubicación física en la red según la alternativa C presentada. Un tercer requisito muy deseable es que los mecanismos deben ser "fácilmente" incorporables a redes ya en funcionamiento ("fácil" = transparente + coste bajo):

- que la operación de estos dispositivos sea funcionalmente transparente tanto para las estaciones (de forma que no sea necesario retocar el *software* en las estaciones) como para los usuarios (de manera que éstos no tengan que modificar sus hábitos de trabajo).
- que se puedan tener comunicaciones desprotegidas, de forma que esto posibilite una instalación gradual de los mecanismos de seguridad sin exigir grandes inversiones iniciales. En principio, se debe posibilitar la comunicación entre estaciones protegidas, entre estaciones desprotegidas, y entre estaciones protegidas con no protegidas.

[23] En este caso la situación se correspondería con la alternativa B presentada.

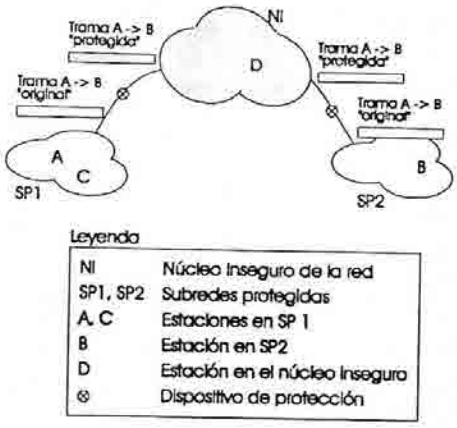


Figura 2.21: Principio de operación de los mecanismos de seguridad

Nuestro objetivo es garantizar servicios de seguridad a las comunicaciones sobre el núcleo inseguro de la red, y esto se debe hacer de forma transparente a las estaciones finales. Bajo esta premisa, la operación de estos mecanismos debe basarse en la intercepción de las unidades de datos (tramas) tal cual son enviadas por las estaciones de la subred protegida, someterlas a algún procedimiento de protección (p.ej. cifrado) y retransmitirlas de nuevo hacia el núcleo inseguro de la red. Análogamente, cuando la trama llega hasta la subred protegida donde se encuentra la estación destinataria es sometida al proceso inverso por el dispositivo de seguridad que hace de "puerta" a esta subred. Esta filosofía de funcionamiento se muestra a grandes rasgos en la Figura 2.21. Una trama que va desde una estación *A* a otra *B* tiene el siguiente formato [MET76], [SCH87], [IEE85] (ver Figura 2.22):

$A \rightarrow B$: { *DA, SA, P/L, DF, CRC* }
donde

DA (6 octetos): Dirección de la estación destinataria *B*. Dado que el canal es de tipo *broadcast*, cada estación examina este campo para determinar si la trama va dirigida a ella, caso en que la deberá "capturar". Si el bit menos significativo del primer octeto de este campo[24] es 1, entonces la dirección corresponde a todo un grupo de estaciones (direccionamiento *multicast*). Si todos los bits de este campo son 1, entonces la trama va dirigida a todas las estaciones de la red (direccionamiento *broadcast*). En este trabajo sólo se considera la protección de comunicaciones punto a punto.

SA(6 octetos): Dirección de la estación remitente *A*.

P/L (2 octetos): Si las entidades de nivel de enlace de las estaciones *A* y *B* utilizan el estándar "de facto" Ethernet este campo indica el protocolo de nivel superior que encapsula el campo de datos. Si estas entidades siguen el estándar 802.3 entonces indica la longitud en octetos del campo de datos[25].

DF (entre 46 y 1500 octetos): Campo de longitud variable donde se transportan los datos que la entidad de nivel superior solicitó que se transmitieran.

CRC (4 octetos): *Checksum* que permite detectar con probabilidad muy alta la aparición de algún bit erróneo durante la transmisión de la trama. Para su cálculo y comprobación se utiliza el polinomio:

---

[24] El orden de transmisión es tal que para cada octeto que forma la trama de la Figura 2.22 se transmite primero el bit menos significativo. Esto significa que si el primer bit (en transmisión) de una trama es 1, entonces está destinada a un grupo de estaciones en vez de a una sóla.

[25] En una misma red local pueden coexistir estaciones Ethernet y estaciones 802.3, ya que exceptuando este campo ambas normas son totalmente compatibles. Dada una determinada trama, se puede determinar a qué formato corresponde simplemente evaluando el valór numérico de su campo *P/L*: dado que la longitud máxima del campo de datos es de 1500 octetos, valores superiores a esta cantidad denotan tramas con formato Ethernet.

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

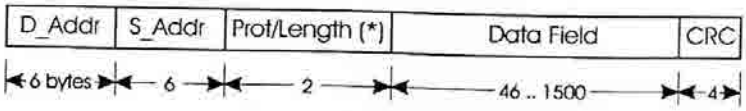


Figura 2.22: Formato unidad de datos del nivel de enlace en redes locales Ethernet / 802.3

Sea $S = \{ SP_i \} \cup NI$ una red local extendida como la de la Figura 2.21, donde SP_i son las subredes protegidas y NI el núcleo inseguro. En el punto de interconexión de SP_i con NI podemos encontrar los siguientes tipos de tramas en función de la ubicación de las estaciones emisora y destinataria (se ha señalado con las letras 'L'y 'R' el hecho de si la trama llega por uno u otro puerto del dispositivo de protección: 'L' denota al puerto local conectado a la subred protegida local, y 'R' al puerto remoto conectado al núcleo inseguro):

(L) $A \rightarrow B$: El origen está en la subred protegida local y el destino está en una subred protegida remota. Esta trama puede ser protegida antes de su "viaje" sobre el núcleo inseguro (ver Figura 2.21).

(R) $B \rightarrow A$: El origen está en una subred protegida remota y el destino en la subred protegida local. Esta trama puede estar protegida, en cuyo caso el dispositivo de protección deberá recomponer la trama original antes de retransmitirla (ver Figura 2.21).

(L) $A \rightarrow D$: El origen está en la subred local protegida y el destino está en el núcleo inseguro. Si esta trama se sometiera al procedimiento de protección no prodría recuperarse la trama original antes de ser entregada a la estación destino. Por tanto este tipo de tramas no puede protegerse.

(R) $D \rightarrow A$: El origen está en el núcleo inseguro y el destino en la subred protegida local. Está trama llega sin proteger y por tanto deberá ser retransmitida "tal cual" al puerto local.

(L) $A \rightarrow C$: Tanto origen como destino están en la misma subred protegida. Esta trama no debería "salir" al backbone.

(R) $D \rightarrow B$: Tanto origen como destino son externos a la subred protegida local. Por tanto esta trama puede ser ignorada y ni tan siquiera ser retransmitida al interior de la subred.

Se observa que el procesamiento selectivo que el dispositivo de seguridad da a cada trama en función de sus direcciones origen y destino es muy similar al que realiza un *bridge* transparente (al igual que éste, para cada trama recibida decide si debe retransmitirla o no en función de la ubicación física en la red de la estación destinataria). Esto, junto a que en el punto de interconexión de la subred local con el núcleo inseguro encontraremos frecuentemente un *bridge* transparente, hace aconsejable la integración de las funciones de ambos dispositivos en uno sólo[26]. Así, el dispositivo de seguridad además de proteger las tramas que lo requieran se encargará de el encaminamiento de todas las tramas que "lleguen a sus manos", según el funcionamiento de un *bridge* transparente. Al dispositivo resultante lo denominamos, de aquí en adelante, "*bridge* seguro".

Desde el punto de vista de este *bridge* seguro, según la localización en la red de las estaciones origen y destino de una trama podemos considerar básicamente los tres siguientes tipos de comunicación:

[26] En caso de no integrar ambos dispositivos, tendríamos una redundancia de funcionalidades que impactaría por un lado en el coste de la interconexión, y por otro lado en el retardo adicional que sufrirían las tramas al recibir dos veces el mismo procesamiento al cruzar de una subred a otra. Ambos inconvenientes se pueden paliar con la integración de los mecanismos de seguridad y de "*bridging*" en un sólo dispositivo.

- *Tipo Ia.* Comunicación entre estaciones "internas" a la subred protegida (p.ej. $A \leftrightarrow C$). El *bridge* seguro deberá realizar un "filtrado" de las tramas de una comunicación de este tipo. Es decir, no debería retransmitir estas tramas hacia el núcleo inseguro, donde un posible intruso podría tener acceso a la información que transportan.
- *Tipo Ib.* Comunicación entre estaciones "externas" a la subred protegida. Desde el punto de vista de seguridad es indiferente que se haga con estas tramas. No obstante, dadas las funciones de *bridge* transparente, filtrará estas tramas no retransmitiéndolas hacia la subred protegida.
- *Tipo II.* Comunicación entre una estación en la subred protegida y otra en el núcleo inseguro (p.ej. $A \leftrightarrow D$). Dado que la estación en el núcleo inseguro no dispone de un dispositivo de seguridad en su acceso a la red, las tramas de este tipo de comunicación simplemente serán retransmitidas al puerto contrario por el que lleguen sin ser sometidas a los mecanismos de protección.
- *Tipo III.* Comunicación entre estaciones en subredes protegidas diferentes (p.ej. $A \leftrightarrow B$). En este caso ambas estaciones acceden al núcleo inseguro a través de dispositivos de seguridad. Por tanto estas tramas pueden viajar "protegidas" sobre el núcleo inseguro reconstruyéndose la trama original antes de ser retransmitida a la subred protegida donde se encuentra la estación destinataria.

Independientemente de cómo se realice la protección de las tramas que lo requieran, cada *bridge* seguro debe conocer las direcciones de las estaciones ubicadas en su subred protegida y en las de todos los *bridges* seguros que se encuentren en la red. La cuestión que se plantea ahora es cómo llega esta información a conocimiento de cada *bridge* seguro. La solución que proponemos se basa en la distribución de esta información desde un "centro de administración". En este centro de administración, el administrador de la red mantiene una base de datos con las listas de estaciones conectadas a cada subred protegida. El contenido de esta base de datos deberá ser distribuido por medios seguros a todos los *bridges* seguros presentes en la red. Dado que en general el conjunto de estaciones conectadas a una subred protegida puede ser muy grande, y que además puede ser muy "dinámico" (i.e. se conectan/desconectan estaciones con relativa frecuencia), exigir al administrador de la red que mantenga en todo momento unos listados exhaustivos de todas las estaciones en las subredes protegidas puede significar un "engorro" considerable. Para aliviar este problema, es conveniente que el administrador tenga la opción de registrar para cada subred protegida únicamente aquellas estaciones cuyas comunicaciones considera que se debe proteger. Esta solución es claramente apropiada para entornos "cotidianos" donde el número de estaciones que manipulan información sensible es relativamente reducido con respecto a su número total.

En la Figura 2.23 se muestra el escenario final de seguridad propuesto para redes locales 802.3 / Ethernet extendidas. En este escenario tanto las subredes protegidas como el núcleo inseguro son modelados cada uno como un único segmento de red local, ya que desde el punto de vista de los *bridges* seguros es indiferente de cuantos segmentos se compone el núcleo inseguro (o cada una de las subredes protegidas).

En resumen, nuestro dispositivo de seguridad será un *bridge* en cuyos mecanismos de filtrado y retransmisión de tramas se integrarán los nuevos mecanismos de seguridad. Por sencillez en la exposición, suponemos que este *bridge* seguro sólo dispone de dos puertos (esto es, que sólo conecta una subred protegida al núcleo inseguro de la red). Este *bridge* dispondrá de una lista de las estaciones protegidas en su subred protegida, y de otra lista análoga para cada uno de los restantes *bridges* seguros existentes[27]. Además, como cualquier *bridge* transparente, "aprenderá" mediante el algoritmo de Baran la ubicación en la red de las estaciones "desprotegidas" (es decir, para cada una de ellas si está accesible en su puerto local o en su puerto remoto), lo que le permitirá llevar a cabo además de sus funciones de seguridad las funciones de filtrado y encaminamiento tradicionales en este tipo de *bridges*.

[27] La generalización al caso de un *bridge* con más de dos puertos es trivial. Bastará con que la lista de estaciones locales (protegidas o no) esté divida en una sublista para cada subrel protegida conectada al *bridge*.

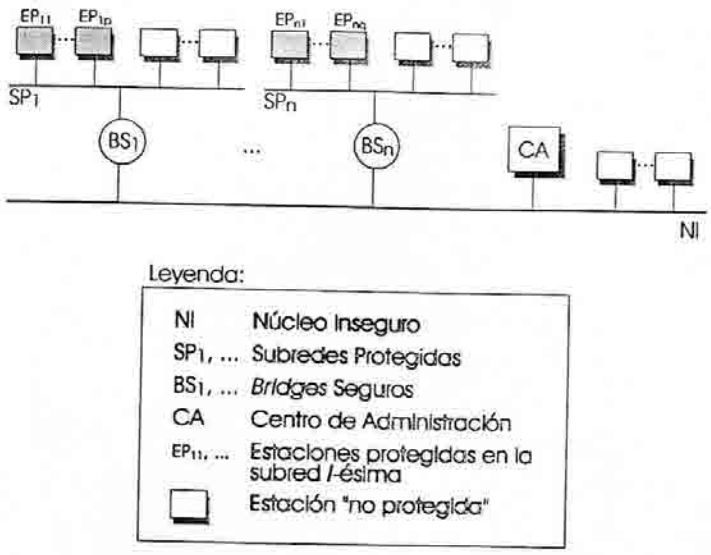


Figura 2.23: Escenario de seguridad

En la Figura 2.24 se muestra la arquitectura lógica de este *bridge* seguro. Esto es, la ubicación de los mecanismos de seguridad y de filtrado/encaminamiento de tramas por encima del subnivel de control de acceso al medio de la red local. En esta misma figura se muestra esquemáticamente la información que deberá mantener para realizar estas funciones de protección y filtrado/encaminamiento de tramas.

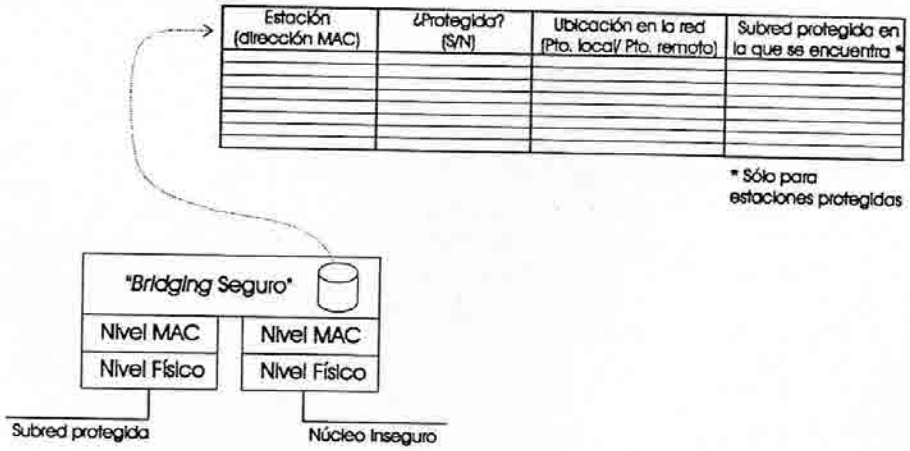


Figura 2.24: Arquitectura del *bridge* seguro en el que se basan los mecanismos de seguridad considerados..

2.5.4 Características generales de los protocolos de sesión.

Sean A y B dos estaciones protegidas en subredes protegidas diferentes ($A \in SP1$ y $B \in SP2$). Supongamos que A envía a B una trama T = { *DA, SA, P/L, DF, CRC* }. Dado el escenario de seguridad presentado en el anterior apartado, y tal como se muestra en la Figura 2.25, la trama original T llegará sobre *SP1* hasta el *bridge* seguro *BS1*. A raíz de esto, *BS1* deberá hacer llegar a

BS2, por medios "seguros" información que le permita recomponer la trama original *T* para que *BS2* pueda transmitirla hacia la estación destinataria *B* en *SP2*.

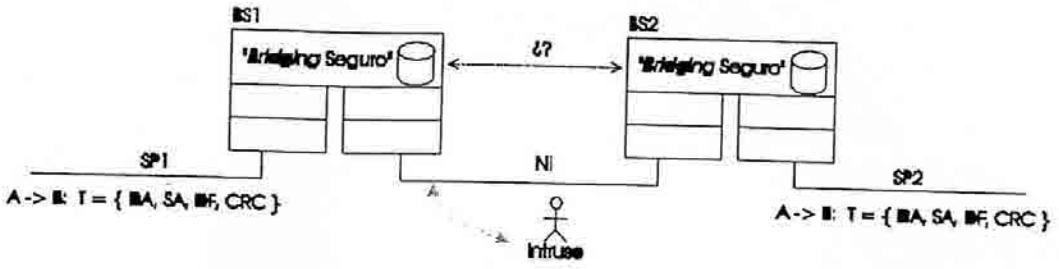


Figura 2.25

La filosofía de funcionamiento de los mecanismos de seguridad que se van a presentar es similar a la de los *tunnelling bridges* utilizados para la interconectar redes locales distintas mediante una tercera red local incompatible [VAR90] (ver nota al pie 16). Si bien en el escenario de comunicaciones que consideramos todas las redes locales que componen la red local extendida utilizan la misma tecnología LAN (i.e. redes Ethernet/802.3), desde el punto de vista de seguridad podríamos considerar que el protocolo de nivel de enlace en el núcleo inseguro de la red es incompatible con el de las subredes protegidas. Por tanto, será necesario modificar el protocolo de nivel enlace en el núcleo inseguro para aquellas tramas que correspondan a una comunicación entre estaciones protegidas. A este nuevo protocolo seguro de nivel de enlace lo denominaremos "protocolo de sesión".

Desde un punto de vista "tradicional" de la comunicación de datos, este protocolo de sesión puede ser "fiable" o no "fiable". Es decir, con garantía o sín garantía de que la trama llegue al *bridge* destinatario. El protocolo de nivel de control de acceso al medio en redes Ethernet/802.3 es un protocolo no fiable: la trama se puede perder en el camino debido a que el transmisor desiste de enviarla después de haber intentado transmitirla un número determinado de veces (15 en concreto) y se ha producido colisión en todos los intentos; también puede descartarse en el receptor debido a que el mecanismo de detección de errores detecta la presencia de algún bit erróneo en la trama[28], etc.. En un entorno de red local es responsabilidad de los niveles superiores al de control de acceso al medio (en las estaciones extremas) el garantizar la fiabilidad de las comunicaciones. Ante esta situación, plantear como requisito que el protocolo de sesión a diseñar sea fiable frente a pérdidas "accidentales" es innecesario: garantizar que una trama llegue hasta el *bridge* destinatario exigiría una complejidad en la realización y no mejoraría sustancialmente la probabilidad de que esta trama llegase hasta la estación destinataria (la cual de por sí ya es muy alta).

Dadas las consideraciones del anterior párrafo, basta en principio con que el el protocolo de sesión sea un protocolo de "encapsulamiento". Es decir, el cuando un *bridge* seguro recibe una trama (a proteger) desde su puerto local, simplemente la encapsula en un nuevo formato seguro y la envía sobre el puerto remoto hacia el *bridge* seguro destinatario, sin preocuparse para nada de si llega hasta éste o no. En el ejemplo de la Figura 2.25, cuando *BS1* recibe la trama original *T* la transforma en una nueva trama *T'* "robusta" frente a posibles ataques que pueda sufrir en el núcleo inseguro y la transmite sobre éste. A su vez, cuando *BS2* recibe la trama *T'*, recompone a partir de ésta la trama original *T* y la retransmite hacia la estación *B* en *SP2*.

[28] La razón de que el subnivel MAC de redes locales no se integren mecanismos que garanticen la fiabilidad del enlace es que la probabilidad de que una trama se "pierda" es considerablemente baja en la práctica (por los propios protocolos de acceso y por las características de los medios físicos empleados). Añadir estos mecanismos significaría generalmente un coste en complejidad y en prestaciones no compesable por la mejora en fiabilidad obtenida.

3 Mecanismos de cifrado

3.1 Introducción

En 1983 la organización ANSI (*American National Standards Institute*) estandarizó cuatro modos de operación para el algoritmo simétrico en bloque DES (*Data Encryption Standard*) [ANS83]. Posteriormente, la organización ISO ha estandarizado estos mismos cuatro modos de operación para cualquier tipo de cifrado en bloque [ISO89]. Estos cuatro modos de utilización de los cifradores en bloque son:

- ECB (*Electronic Code Book*). Constituye el modo nativo. Cada bloque de criptograma se obtiene directamente de la aplicación del operador de encriptado a su correspondiente bloque de mensaje en claro. Presenta el inconveniente de que dada una clave de cifrado dada, entonces bloques de mensaje idénticos producen bloques de criptograma idénticos. Por ello, se suele considerar inadecuado para algunos cifrados en bloque de tamaño relativamente pequeño (p.ej. DES "simple"). No obstante, se considera adecuado para cifrar mensajes aleatorios (i.e. en los que no se dan estructuras predeterminadas cuyo conocimiento pudiera facilitar la labor del atacante), siendo una de sus aplicaciones más comunes el cifrado de claves. Se debe observar que, en este modo de cifrado, la corrupción o alteración de un bit en el criptograma provoca que en el proceso de descifrado cada uno de los bits obtenidos en el bloque correspondiente sean erróneos con una probabilidad de ½ (esta propiedad debe ser satisfecha por todo operador de cifrado en bloque que se considere "bueno").
- CBC (*Cipher Block Chainning*). Cada bloque del criptograma es sumado (OR-exclusiva) con el siguiente bloque de mensaje en claro antes de someterlo al operador de encriptado. Presenta la virtud de que cada bloque de criptograma no sólo depende de su correspondiente bloque de mensaje en claro, sino además de todos los anteriores. Esta característica lo hace interesante como mecanismo de cifrado para ocultar patrones repetitivos en los mensajes y también como mecanismo de generación de códigos de autentificación [ANS86]. Los inconvenientes más relevantes son, por un lado, la necesidad de utilizar un vector de inicialización, y por otro, que la corrupción de un sólo bit en el criptograma provoca al descifrar que todo el bloque correspondiente y el siguiente sean erróneos.
- CFB (*Cipher Feed-Back*). En cada paso se aplica el operador de cifrado a un registro de desplazamiento, del resultado obtenido se seleccionan p bits que se suman a p bits de mensaje en claro. Para realizar el siguiente paso, estos p bits de criptograma son realimentados hacia el registro de desplazamiento en la entrada al operador. La principal ventaja de este modo de operación es que permite utilizar un algoritmo en bloque como si se tratase de un algoritmo en flujo (el valor de p es arbitrario). Además, igual que en el modo CBC, bloques de mensaje en claro idénticos no producen bloques de criptograma idénticos. Si b es la anchura, en bits, de los bloques procesados por el operador de cifrado, entonces un bit erróneo en el criptograma provoca en la operación de descifrado $p + b$ ($\leq 2b$, puesto que $p \leq b$) bits erróneos (suponiendo b múltiplo de p, en

caso contrario $p+\lceil b/p \rceil \times p$). Además se debe observar que de cada b bits que procesa el operador de cifrado, sólo se cifran p bits de mensaje, desaprovechándose "ancho de banda" del operador (por cada bit procesado, sólo se envían p/b).

- OFB (*Output Feed-Back*). Es similar al anterior modo. A diferencia de éste, en vez de realimentar bits de criptograma, en cada operación de cifrado se toman todos los bits a la salida del operador en la anterior operación. De cada nuevo bloque obtenido a la salida del operador se seleccionan p bits sumándose a p bits de mensaje en claro. Esto permite que este modo de operación se comporte exactamente como un cifrador en flujo síncrono, esto es, el operador de cifrado se utiliza como un generador de secuencia seudoaleatoria que se suma, de p en p bits, al mensaje en claro. De esta manera, al no utilizarse ni el mensaje ni el criptograma en la generación de la secuencia, un bit erróneo en el criptograma sólo produce un bit erróneo en la operación de descifrado. Hay que observar que tanto en el modo CFB como en el OFB, el operador utilizado tanto para cifrar como para descifrar son el mismo (y con la misma clave).

Además de estos cuatro modos de encadenamiento estándares para cifradores en bloque, se pueden encontrar en la literatura otros modos no estandarizados [PAS88], [JAN87], como son el PBC (*Plaintext Block Chainning*), el PFB (*Plaintext Feed Back*) y el OFBNLF (*Output Feedback with Non Linear Function*), por citar algunos.

Proponemos aquí un mecanismo de cifrado, al que denominamos IOBC (de *Input and Output Block Chainning*), que permite realizar simultáneamente los servicios de confidencialidad e integridad sin necesidad de procesar dos veces el mensaje en claro. Usualmente (p.ej. en el estándar 802.10 y en el sistema de seguridad EESS) los mecanismos de confidencialidad e integridad se realizan de manera independiente exigiendo procesar dos veces el mensaje antes de ser transmitido. En algunos sistemas (p.ej. en el estándar 802.10) ambos servicios son seleccionables "por separado". De esta forma si un servicio no está realmente justificado se puede prescindir de él, minimizando el tiempo requerido para procesar cada trama[1]. El precio que se paga a cambio es evidentemente un ligero aumento en la complejidad de gestión en los mecanismos de seguridad. En el resto de sistemas donde ambos servicios son independientes no existe esta "opcionalidad", pero a cambio la administración es más sencilla.

Los protocolos de seguridad estudiados en este trabajo utilizan un único mecanismo de cifrado para garantizar simultáneamente los servicios de confidencialidad e integridad. En este mecanismo cada bit del mensaje a proteger es procesado una sola vez por el *bridge* seguro origen garantizando que es recuperado de manera confidencial e íntegra por el *bridge* seguro destino. Este mecanismo de cifrado se fundamenta en el uso de un cifrado en bloque encadenado sobre mensajes de longitud limitada.

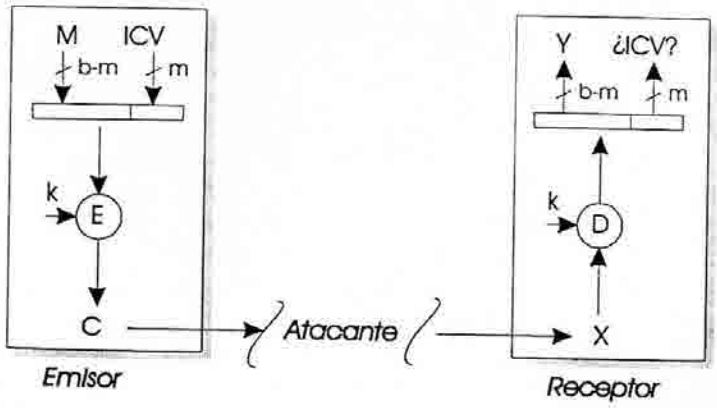


Figura 3.1: Mecanismo de confidencialidad e integridad con un cifrador de bloques en modo ECB

[1] No obstante, hay que señalar que el modo IOBC presentado aquí es integrable en los protocolos 802.10 de una manera bastante "natural" (siendo esto una ventaja más).

En la Figura 3.1 se muestra un mecanismo "convencional" que cifrando una sola vez cada bit del mensaje *M*, consigue garantizar tanto su confidencialidad como su integridad. Sin entrar en detalles, el principal inconveniente de este mecanismo es que requiere que a cada bloque de mensaje en claro se le añada un patrón de bits predeterminado *ICV* (que hace las funciones de código de comprobación de integridad, pero cuyo valor es independiente del mensaje). La principal ventaja del modo de encadenamiento que proponemos, es que permite una realización análoga de los servicios de confidencialidad e integridad para mensajes de más de un bloque de longitud, añadiendo únicamente un código al final del mensaje y que puede se puede escoger de manera independiente del contenido del mensaje.

Existen algunos precedentes de modos de encadenamiento para cifradores en bloque cuyo objetivo es precisamente el que aquí se busca. Entre ellos cabe citar el modo PCBC utilizado en la versión 4 del sistema de autentificación Kerberos y el modo PBC descrito en [JAN87] y que fue utilizado en las etapas iniciales de este trabajo.

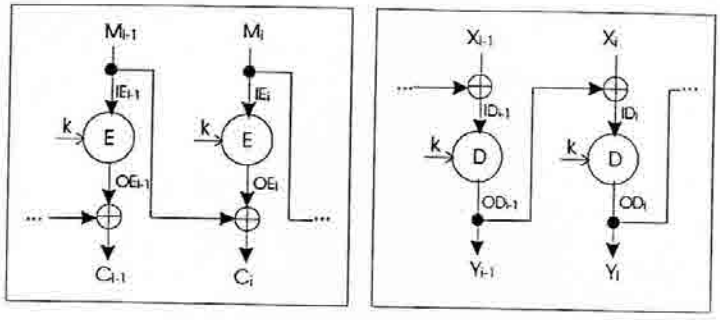


Figura 3.2: Cifrado y descifrado según modo PBC

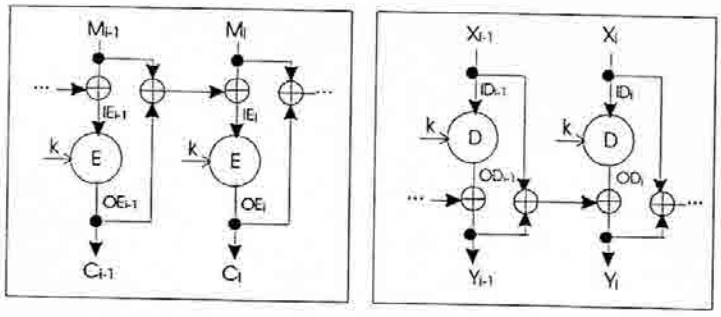


Figura 3.3: Cifrado y descifrado según el modo PCBC

En la Figura 3.2 se muestra el modo de encadenamiento PBC. Dado que cada bloque de mensaje en claro recuperado por el receptor depende de todos los anteriores, este modo presenta una deseable de propagación de errores que permite realizar el mecanismo de integridad: aparentemente, si el atacante modifica un bloque de criptograma el bloque en claro correspondiente que recuperará el receptor será (en principio) erróneo y este error se propagará indefinidamente a través de los siguientes pasos hasta el último bloque. Supongamos que el receptor ha acordado previamente con el emisor que el último bloque en claro ha de tener un determinado valor *ICV* (*Integrity Check Value*). Si al descifrar un criptograma no se obtiene en el último bloque el valor *ICV* entonces se puede afirmar, con total seguridad, que este criptograma no ha sido generado por el emisor legítimo, o que ha sido modificado durante su transferencia.

Desafortunadamente, si dos bloques en claro M_i y M_j coinciden y el atacante lo sabe, entonces puede reemplazar el bloque de criptograma C_{i+1} por C_{j+1} (y lo mismo con todos los posteriores). Observar que en este caso el receptor no podrá detectar la alteración del criptograma, ya que el bloque de mensaje en claro que recupera en el paso $(j+1)$ es

precisamente M_{j+1}, y análogamente con todos los posteriores (incluyendo el último donde "viaja" *ICV*).

En la Figura 3.3 se muestra el modo de encadenamiento PCBC [MEY82] utilizado en la versión 4 de Kerberos. Este mecanismo de cifrado no ha sido utilizado en versiones posteriores del sistema Kerberos por ser vulnerable frente a algunos ataques a la integridad de los datos [STA95]. El mecanismo de integridad utilizado es análogo al descrito para el modo PBC: el emisor pone en el último bloque en claro un valor *ICV* preestablecido y el receptor a su vez sólo da por buenos los mensajes para los que recupera este código. No obstante, el método es vulnerable, por ejemplo, a la reordenación de bloques de criptograma (esto es, intercambiar dos bloques cualesquiera C_i y C_j) y a la inserción de pares de bloques espúreos idénticos (entre los bloques C_i, C_{i+1} y C_j, C_{j+1} se inserta un bloque X de valor arbitrario).

3.2 Encadenamientos cruzados

Se analizan aquí cuatro modos de encadenamiento muy similares y que tienen características interesantes para una realización "simultánea" de los mecanismos de confidencialidad e integridad. En la Figura 3.4 se intenta reflejar que estos cuatro modos son las combinaciones posibles que se pueden escoger a la hora de encadenar de manera cruzada desde la entrada a la salida del operador de cifrado, y viceversa. El modo de encadenamiento IOBC, descrito en el siguiente apartado, se construirá como una variación de la cuarta modalidad de encadenamiento cruzado que aquí se describe.

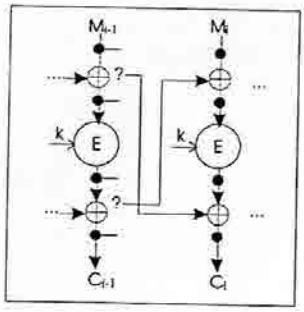


Figura 3.4: Posibilidades de encadenamiento "cruzado"

3.2.1 Encadenamiento cruzado de texto en claro y criptograma

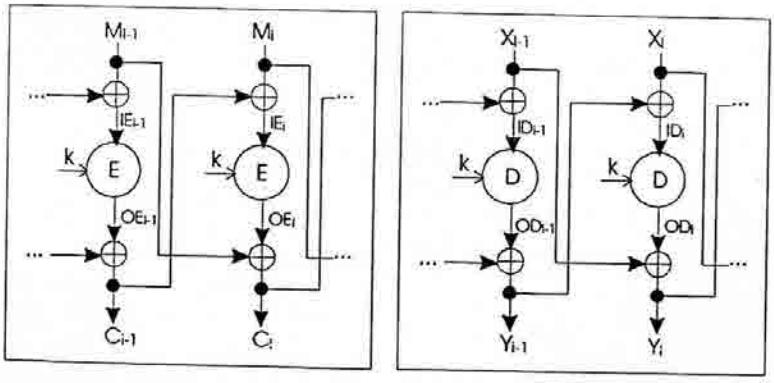


Figura 3.5: Cifrado y descifrado según la 1ª alternativa

En la Figura 3.5 se muestra la primera alternativa considerada. En ella, la entrada al operador de cifrado en cada paso es la suma x-or del bloque de mensaje correspondiente con el bloque de criptograma del paso anterior. La salida del operador se suma a su vez con el bloque de mensaje en claro anterior para determinar el bloque de criptograma actual. Un estudio detallado de este encadenamiento permite concluir que presenta buenas características de confidencialidad. No obstante, en cuanto a integridad el método presenta una serie de vulnerabilidades que desaconsejan su utilización. Entre ellas cabe destacar el siguiente ejemplo. Supongamos que el emisor quiere enviar al receptor un mensaje en claro como el siguiente, en el que se da una racha de bloques en claro idénticos (y el atacante lo sabe)

$$<\ldots, M_2, M_3 = M, M_4 = M, M_5 = M, M_6 = M, M_7 = M, M_8, M_9, \ldots>,$$

y que el correspondiente criptograma es

$$<\ldots, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, \ldots>.$$

Ahora supongamos que el atacante intercepta este criptograma y lo sustituye (por ejemplo) por el siguiente

$$<\ldots, C_2, C_3, C_6, C_7 \oplus C_3 \oplus C_5, C_4, C_7, C_8, C_9, \ldots>.$$

Entonces el receptor recuperará hasta el tercer paso el mensaje en claro que envió el emisor. Pero en el cuarto, obtendrá

$$Y_4 = X_3 \oplus D_k\{Y_3 \oplus X_4\} = C_3 \oplus D_k\{M \oplus C_6\}.$$

Puesto que $M_6 = M = C_5 \oplus D_k\{M \oplus C_6\}$, entonces $D_k\{M \oplus C_6\} = M_6 \oplus C_5$, e

$$Y_4 = C_3 \oplus C_5 \oplus M_6 = C_3 \oplus C_5 \oplus M.$$

A su vez, en los pasos quinto, sexto y séptimo

$$Y_5 = X_4 \oplus D_k\{Y_4 \oplus X_5\} = C_6 \oplus D_k\{(C_3 \oplus C_5 \oplus M) \oplus (C_7 \oplus C_3 \oplus C_5)\} = C_6 \oplus D_k\{M \oplus C_7\}$$
$$= M_7 = M$$

$$Y_6 = X_5 \oplus D_k\{Y_5 \oplus X_6\} = (C_7 \oplus C_3 \oplus C_5) \oplus D_k\{M \oplus C_4\} = C_7 \oplus C_5 \oplus M_4 = C_7 \oplus C_5 \oplus M$$

$$Y_7 = X_6 \oplus D_k\{Y_6 \oplus X_7\} = C_4 \oplus D_k\{(C_7 \oplus C_5 \oplus M) \oplus C_7\} = C_4 \oplus D_k\{M \oplus C_5\} = M_5 = M.$$

Se puede observar que en el séptimo paso, con $X_7 = C_7$, el receptor recupera $Y_7 = M_7$, de manera que atacante puede replicar a partir de aquí todos los bloques de criptograma en claro C_8, C_9, ... y el receptor no podrá detectar la presencia de los bloques falsos 4°, 5° y 6°.

3.2.2 Encadenamiento de salida y texto en claro

En la Figura 3.6 se muestra la segunda posibilidad en cuanto a encadenamiento cruzado. Ahora cada bloque de criptograma sigue calculándose como suma x-or del bloque de mensaje precedente y la salida de la operación de cifrado. Pero el operador de cifrado se aplica ahora a la suma x-or del bloque de mensaje actual y la salida del operador del paso anterior (en lugar del bloque de criptograma precedente como se hacía en la primera alternativa). No obstante este modo de encadenamiento es vulnerable a ataques muy similares a los del primero.

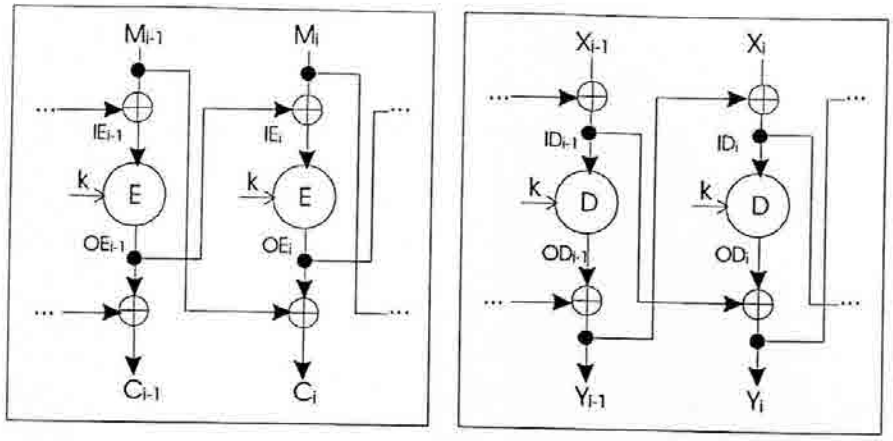


Figura 3.6: Cifrado y descifrado según la 2ª alternativa

A continuación se expone uno de los ataques a los que es vulnerable este método de encadenamiento. Supongamos que el emisor quiere enviar al receptor un mensaje en claro como el siguiente

$$<\ldots, M_2, M_3 = M, M_4 = M, M_5 = M, M_6 = M, M_7 = M, M_8, M_9, \ldots>,$$

y que el correspondiente criptograma es

$$<\ldots, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, \ldots>.$$

Ahora supongamos que el atacante intercepta este criptograma y lo sustituye (por ejemplo) por el siguiente

$$<\ldots, C_2, C_3, C_6, C_7 \oplus C_3 \oplus C_5, C_8, C_9, \ldots>.$$

Mediante un análisis análogo al del primer caso es fácil deducir que el mensaje en claro que recuperará el receptor será:

$$<\ldots, M_2, M, M \oplus C_3 \oplus C_5, M, M_8, M_9, \ldots>.$$

De manera que modificaciones de este tipo en bloques intermedios no quedan "reflejadas" en los últimos bloques, haciendo desaconsejable el uso de este encadenamiento.

3.2.3 Encadenamiento de entrada y texto cifrado

En la Figura 3.7 se muestra la tercera alternativa de encadenamiento cruzado. Según esta nueva modalidad de encadenamiento, en cada paso la entrada al operador de cifrado se determina como la suma x-or del bloque de criptograma precedente y el bloque de mensaje en claro actual. A su vez, el bloque de criptograma se toma como la suma del resultado de la operación de cifrado y la entrada a este operador en el paso precedente. Desafortunadamente este encadenamiento es vulnerable a ataques similares a los de los anteriormente estudiados.

A continuación se expone uno de los ataques a los que es vulnerable este tercer modo. Supongamos que el emisor quiere enviar al receptor el mensaje en claro $<\ldots, M_2, M_3, M_4, M_5, M_6, M_7, \ldots>$, que el atacante conoce los bloques en claro M_3 y M_4, y que el correspondiente criptograma es $<\ldots, C_2, C_3, C_4, C_5, C_6, C_7, \ldots>$.

Ahora supongamos que el atacante intercepta este criptograma y lo sustituye (por ejemplo) por el siguiente

$<\ldots, C_2, C_3, C_5 \oplus M_3 \oplus C_2 \oplus M_4 \oplus C_3, C_6, C_7, \ldots>$.

Tal y como se ilustra en la Figura 3.8, para el quinto bloque del criptograma falso, C_6, el receptor obtiene a la salida del operador de descifrado el valor IE_6, precisamente el mismo valor que correspondía al sexto paso en el criptograma auténtico. De ahí que el atacante pueda replicar a partir de este punto el criptograma original sin que el receptor tenga posibilidad de detectar las modificaciones. Una vez más, este y los restantes ataques a los que es vulnerable el método hacen desaconsejable su utilización.

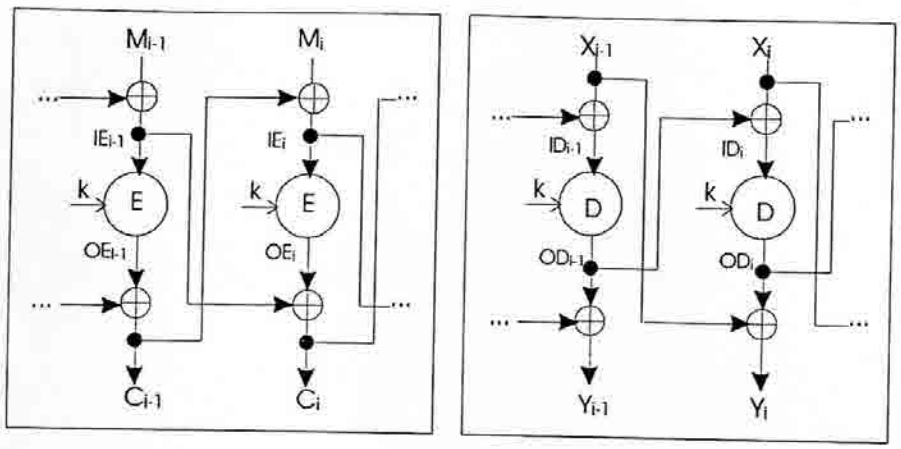
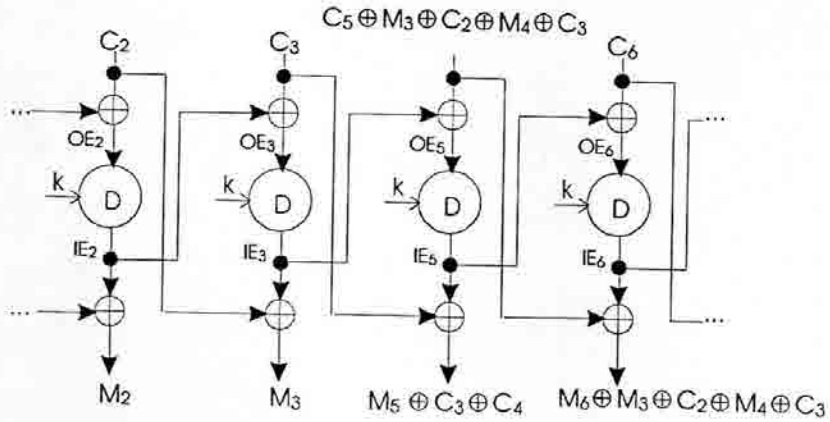


Figura 3.7: Cifrado y descifrado según la 3ª modalidad



NOTA:

$$\left.\begin{array}{l} IE_3 = M_3 \oplus C_2 \\ IE_4 = M_4 \oplus C_3 \\ OE_5 = C_5 \oplus IE_4 \end{array}\right\} \Rightarrow$$

$$\begin{aligned} ID_4 &= X_3 \oplus OD_3 = \\ &= (C_5 \oplus M_3 \oplus C_2 \oplus M_4 \oplus C_3) \oplus IE_3 = \\ &= (C_5 \oplus \cancel{M_3} \oplus \cancel{C_2} \oplus M_4 \oplus C_3) \oplus \cancel{M_3} \oplus \cancel{C_2} = \\ &= C_5 \oplus M_4 \oplus C_3 = C_5 \oplus IE_4 = OE_5 \end{aligned}$$

Figura 3.8: Modificación de un criptograma no detectable mediante la 3ª modalidad

3.2.4 Encadenamiento cruzado de la entrada y la salida

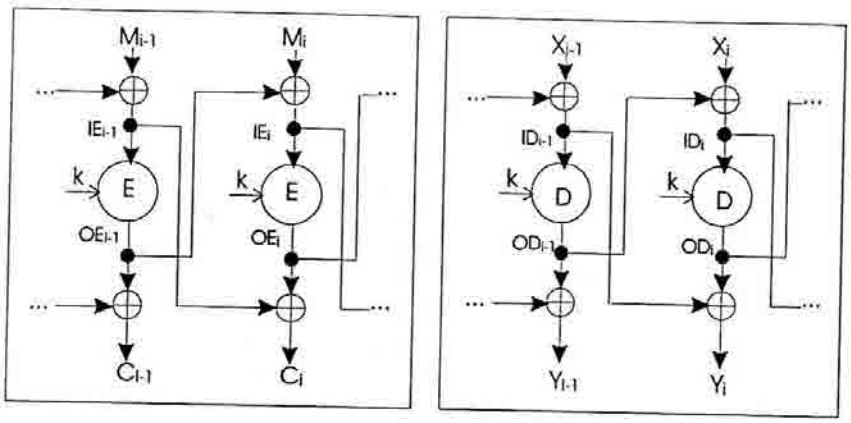


Figura 3.9: Cifrado y descifrado según la 4ª alternativa

En este cuarto modo de encadenamiento cruzado, en cada paso de la operación de cifrado la entrada al operador viene dada por la suma x-or del bloque de mensaje correspondiente y la salida del operador en el paso precedente. A su vez, cada bloque de criptograma viene dado por la suma x-or de la salida del operador de cifrado y la entrada en el paso precedente. Es decir

$$C_1 = IV_b \oplus E_k\{M_1 \oplus IV_a\};\quad C_i = IE_{i-1} \oplus E_k\{M_i \oplus OE_{i-1}\};\quad i = 2, ..., N; \tag{3.1}$$

Siendo IE_j y OE_j los vectores a la entrada y salida, respectivamente, del operador de cifrado en el paso j-ésimo (según se indica en la Figura 3.9), e IV_a y IV_b dos vectores de inicialización.

Análogamente para la operación de descifrado:

$$Y_1 = IV_a \oplus D_k\{X_1 \oplus IV_b\};\quad Y_i = ID_{i-1} \oplus D_k\{X_i \oplus OD_{i-1}\};\quad i = 2, ..., N; \tag{3.2}$$

Siendo ID_j y OD_j los vectores a la entrada y salida, respectivamente, del operador de descifrado en el paso j-ésimo (según se indica en la Figura 3.9).

De la observación de las ecuaciones (3.1) podemos escribir el sistema de ecuaciones

$$\left.\begin{aligned} C_i &= IE_{i-1} \oplus OE_i \\ M_i &= OE_{i-1} \oplus IE_i \end{aligned}\right\};\ i = 1, ..., N;\ \text{donde}\quad IE_0 = IV_b\ \text{ e }\ OE_0 = IV_a \tag{3.3}$$

Suponiendo conocidos todos los bloques de mensaje en claro, las ecuaciones (3.3) constituyen un sistema de ecuaciones[2] con un total de $2N$ ecuaciones y $2N+2$ incógnitas (IE_0, ..., IE_N) y (OE_0, ..., OE_N). En este sistema, si se supone que tanto IV_a (OE_0) como IV_b (IE_0) son secretos, entonces no hay manera de despejar ninguna de las incógnitas aún cuando se conozcan todos los bloques de texto en claro[3]. Además, si se toman los vectores de inicialización aleatoriamente, entonces todas estas incógnitas se comportan como variables aleatorias uniformes (suponiendo que el operador de cifrado es ideal y que se desconoce la

---

[2] Z_{2^n} con el operador suma "x-or bit a bit" y el producto por { 0, 1 } constituye un espacio lineal binario donde se puede aplicar el álgebra usual de los espacios lineales.

[3] Realmente son dos sistemas totalmente independientes (suponiendo $IV_a \neq IV_b$) cada uno de N ecuaciones y $N+1$ incógnitas. La confidencialidad de cada uno de los vectores de inicialización garantiza la lo "resolubilidad" de uno de los sistemas. En los anteriores modos de encadenamiento también se pueden plantear sistemas de ecuaciones análogos a (3.3). No obstante, en aquellos casos uno o ambos son resolubles si se conocen los bloques de mensaje en claro (en el primero se puede determinar tanto IE_i como OE_i en cualquier paso i; en el segundo las salidas OE_i y en el tercero las entradas IE_i).

clave k). De manera que no se va a poder "adivinar" el valor de ninguna de ellas con una probabilidad de acierto mayor a 2^{-b} (siendo b el tamaño de bloque).

Esta característica resulta muy interesante, entre otras cosas porque va a permitir garantizar un muy buen nivel de confidencialidad. Supongamos que un posible intruso conoce todos los bloques de texto en claro cifrados con una determinada clave k excepto un determinado M_i cuyo valor quiere descubrir. En esta situación, el atacante tiene que añadir una incógnita más al sistema (3.3), de manera que la confidencialidad de M_i queda totalmente garantizada. Además, dado que no se le permite al atacante tener acceso a los pares claro-cifrado (IE,OE), en principio, el utilizar la misma clave k para cifrar más o menos cantidad de mensaje no va a aportar ninguna información sobre la misma al atacante[4].

Además del buen comportamiento desde el punto de vista de confidencialidad, este modo de encadenamiento también es muy atractivo para realizar un mecanismo de integridad. Excepto el caso que se estudia a continuación, en general, si el atacante sustituye cualquier C_i por otro valor X, entonces la entrada y salida del operador de descifrado tomarán valores desconocidos (e "incontrolables" en el caso de bloques a la salida, ODs) produciéndose una propagación de errores que evolucionará de manera imprevisible en cada uno de los siguientes bloques. De manera que el receptor no recuperará un vector ICV (previamente acordado con el emisor) en el último de los bloques, detectando de esta manera la no integridad del criptograma en cuestión.

No obstante, existe una excepción al anterior razonamiento. El atacante puede utilizar las ecuaciones (3.3) para realizar una modificación inteligente del criptograma que no puede ser detectada por el mecanismo descrito[5]. Supongamos que el atacante conoce absolutamente todo el texto en claro que envía el emisor. Entonces, puede sumar entre sí las ecuaciones (3.3) para obtener a la derecha de la ecuación resultante $IE_{i-1} \oplus OE_j$ (con $i, j = 1, ..., N$, e $i \neq j$). Por ejemplo, suponiendo que $j > i$,

$$\begin{aligned} X_i &= C_i \oplus M_{i+1} \oplus C_{i+2} \oplus \ldots \oplus M_{j-1} \oplus C_j = \\ &= (IE_{i-1} \oplus OE_i) \oplus (OE_i \oplus IE_{i+1}) \oplus (IE_{i+1} \oplus OE_{i+2}) \oplus \ldots (OE_{j-2} \oplus IE_{j-1}) \oplus (IE_{j-1} \oplus OE_j) = \\ &= IE_{i-1} \oplus OE_j \end{aligned}$$

Análogamente, para el caso $j < i$,

$$\begin{aligned} X_i &= M_{j+1} \oplus C_{j+2} \oplus M_{j+3} \oplus \ldots \oplus C_{i-2} \oplus M_{i-1} = \\ &= (OE_j \oplus IE_{j+1}) \oplus (IE_{j+1} \oplus OE_{j+2}) \oplus (OE_{j+2} \oplus IE_{j+3}) \oplus \ldots (IE_{i-3} \oplus OE_{i-2}) \oplus (OE_{i-2} \oplus IE_{i-1}) = \\ &= OE_j \oplus IE_{i-1} \end{aligned}$$

De manera que si el atacante sustituye C_i por el valor X_i determinado por una de las sumas anteriores, construyendo el criptograma falso

$$<\ldots, C_{i-2}, C_{i-1}, X_i, C_{j+1}, C_{j+2}, \ldots>,$$

entonces, al descifrar el receptor obtendrá $ID_i = OE_j$ y $OD_i = IE_j$. De manera que a partir del siguiente bloque puede reproducir el criptograma original (bloques $C_{j+1}, C_{j+2}, \ldots, C_N$) sin que el mecanismo de integridad detecte la modificación[6].

Además, si se utilizan idénticos vectores de inicialización para cifrar diferentes mensajes, el atacante podrá realizar un ataque análogo combinado las ecuaciones (3.3) correspondientes a estos mensajes[7].

---

[4] Esto será así mientras no se repitan los vectores de inicialización de manera determinista.

[5] A diferencia de los anteriores modos de encadenamiento, el ataque que se describe aquí constituye la única vulnerabilidad observada en esta cuarta modalidad.

[6] Observar que la entrada al operador de descifrado en el paso i+1 será $ID_{i+1} = X_{i+1} \oplus OD_i = C_{j+1} \oplus OD_j$, y que por tanto $Y_{i+1} = OD_{i+1} \oplus ID_i = IE_{j+1} \oplus OE_j = M_{j+1}$, y análogamente en los siguientes pasos.

Se deduce del análisis de esta cuarta modalidad de encadenamiento cruzado que tampoco garantiza suficientemente la integridad de los datos. No obstante, el hecho de que sólo sea vulnerable a un ataque muy específico (el descrito), va a permitir que con una "ligera" modificación se obtenga un mecanismo de encadenamiento robusto frente a amenazas tanto a la confidencialidad como a la integridad.

3.3 Modificaciones a la cuarta modalidad

El inconveniente de la cuarta modalidad de encadenamiento cruzado presentada en el apartado 3.2 es la posibilidad que tiene el atacante de combinar linealmente las ecuaciones (3.3) a fin de modificar un criptograma generado por el emisor. Vamos a estudiar aquí el efecto que introducen algunas modificaciones "sencillas" en el cuarto modo de encadenamiento estudiado en 3.2.

Tal y como se ilustra en la Figura 3.10, la modificación que proponemos se basa en insertar un operador $T\{\}$ que actúe sobre los vectores de encadenamiento. Este operador además de robustecer el método deberá ser suficientemente sencillo como para que la complejidad del método siga siendo sensiblemente inferior a la de los mecanismos de confidencialidad e integridad convencionales que utilizan una "pasada" para generar el código de comprobación de integridad y otra para cifrar el mensaje.

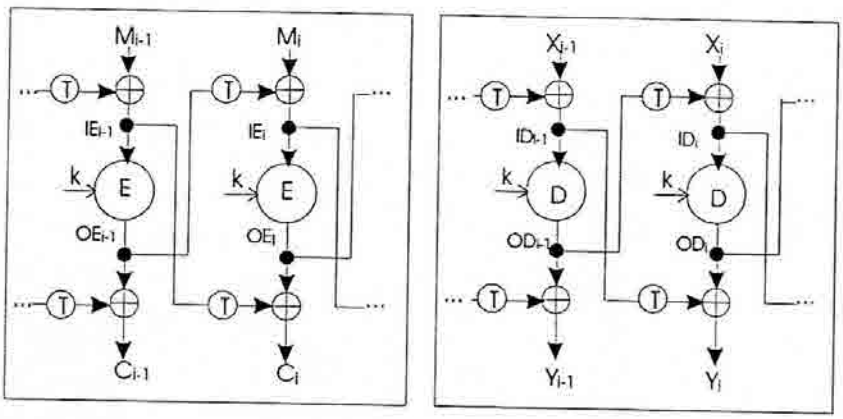


Figura 3.10

Las operaciones de cifrado y descifrado vienen dadas ahora por las ecuaciones

$$C_1 = IV_b \oplus E_k\{M_1 \oplus IV_a\};\quad C_i = T\{IE_{i-1}\} \oplus E_k\{M_i \oplus T\{OE_{i-1}\}\};\quad i = 2, ..., N; \tag{3.4}$$

$$Y_1 = IV_a \oplus D_k\{X_1 \oplus IV_b\};\quad Y_i = T\{ID_{i-1}\} \oplus D_k\{X_i \oplus T\{OD_{i-1}\}\};\quad i = 2, ..., N; \tag{3.5}$$

Análogamente a las ecuaciones (3.3), ahora podemos escribir

$$\left.\begin{aligned} C_i &= T\{IE_{i-1}\} \oplus OE_i \\ M_i &= T\{OE_{i-1}\} \oplus IE_i \end{aligned}\right\};\ i = 1, ..., N;\ \text{donde}\ \ T\{IE_0\} = IV_b\ \ \text{e}\ \ T\{OE_0\} = IV_a \tag{3.6}$$

El objetivo es diseñar el operador $T\{\}$ de manera que se siga garantizando la secreticidad de los *IE*s y *OE*s, y también de manera que si se sustituye cualquier C_i por cualquier otro valor, entonces el valor del ID_i correspondiente sea imprevisible. En particular, se deberá garantizar

[7] Independientemente de esta vulnerabilidad, es deseable que para cada mensaje en claro se utilicen vectores de inicialización diferentes. De esta manera se consigue que aunque dos mensajes en claro sean idénticos sus correspondientes criptogramas estén "incorrelados".

que incluso conociendo todos los bloques del texto en claro cifrados por el emisor, no se pueda determinar ningún valor $X_{ij} = T\{IE_{i-1}\} \oplus OE_j$ (para $i \neq j$ cualesquiera). En caso contrario, bastaría con sustituir el criptograma original por $<\ldots, C_{i-2}, C_{i-1}, X_{ij}, C_{j+1}, C_{j+2}, \ldots>$ para violar el mecanismo de integridad.

Una elección buena sería hacer $T\{\}$ = $E_{k'}\{\}$ (en principio con $k' \neq k$), pues los valores encadenados en el descifrado seguirían dependiendo de todos los bloques X e Y anteriores, serían imprevisibles y el atacante no tendría manera de combinar las ecuaciones (3.6) para obtener ningún valor $X_{ij} = E_{k'}\{IE_{i-1}\} \oplus OE_j$. No obstante este diseño requeriría cifrar un total de $3N$ bloques (y además requiere una segunda clave), de manera que la complejidad del cifrado será similar o superior a la de los mecanismos de confidencialidad e integridad convencionales.

Otra posibilidad mucho más simple podría ser utilizar como operador $T\{\}$ una rotación de bits. Sean $(W)_{>q}$ y $(W)_{<q}$ el resultado de rotar q posiciones a la derecha, e izquierda respectivamente, los bits de W. Entonces podríamos tomar, por ejemplo,

$$T(W) = (W)_{>1}.$$

Se puede observar que el operador rotación satisface las propiedades siguientes:

- $(W \oplus Z)_{>q} = (W)_{>q} \oplus (Z)_{>q}$
- $\left((W)_{>q}\right)_{<p} = \left((W)_{<p}\right)_{>q} = \begin{cases} (W)_{>(q-p)}; & \text{si } q \geq p \\ (W)_{<(p-q)}; & \text{si } p \geq q \end{cases}$
- $(W)_{>b} = (W)_{<b} = W$; donde b es el tamaño de W en bits o un múltiplo de éste.

Desafortunadamente, la utilización de rotaciones no va a ser una solución definitiva. Efectivamente, supongamos que $j > i$ (el caso opuesto es análogo), entonces el atacante puede calcular[8]

$$\begin{aligned}
&C_i = OE_i \oplus (IE_{i-1})_{>1} \\
&(M_{i+1})_{<1} = (IE_{i+1})_{<1} \oplus OE_i \\
&(C_{i+2})_{<2} = (OE_{i+2})_{<2} \oplus (IE_{i+1})_{<1} \\
&(M_{i+3})_{<3} = (IE_{i+3})_{<3} \oplus (OE_{i+2})_{<2} \\
&\ldots \\
&(M_{i+b-1})_{<(b-1)} = (IE_{i+b-1})_{<(b-1)} \oplus (OE_{i+b-2})_{<(b-2)} \\
&(C_{i+b})_{<b} = (OE_{i+b})_{<b} \oplus (IE_{i+b-1})_{<(b-1)}
\end{aligned}$$

Si ahora sumamos todas estas ecuaciones, obtenemos

$$\begin{aligned}
X_{i(i+b)} &= C_i \oplus (M_{i+1})_{<1} \oplus (C_{i+2})_{<2} \oplus (M_{i+3})_{<3} \oplus \ldots \oplus (M_{i+b-1})_{<(b-1)} \oplus (C_{i+b})_{<b} = \\
&= (IE_{i-1})_{>1} \oplus (OE_{i+b})_{<b} = (IE_{i-1})_{>1} \oplus OE_{i+b}
\end{aligned}$$

De manera que si el atacante sustituye un criptograma auténtico

$$<\ldots, C_{i-1}, C_i, C_{i+1}, \ldots, C_{i+b-1}, C_{i+b}, C_{i+b+1}, \ldots>,$$

por

$$<\ldots, C_{i-2}, C_{i-1}, X_{i(i+b)}, C_{i+b+1}, C_{i+b+2}, \ldots>,$$

[8] Se ha supuesto aquí que b es par. En cualquier caso, este procedimiento se puede aplicar a cualquier número de términos múltiplo de b, por lo que imponer que b sea impar no va a imposibilitar el ataque.

entonces el receptor obtendrá $ID_i = OE_{i+b}$ y $OD_i = IE_{i+b}$. De manera que a partir del siguiente bloque puede reproducir el criptograma original (bloques $C_{i+b+1}, C_{i+b+2}, \ldots, C_N$) sin que el mecanismo de integridad detecte la modificación.

Del anterior análisis se pueden extraer dos conclusiones:

- el operador rotación no es una buena elección para $T\{\}$, pues no permite contrarrestar suficientemente las amenazas a la integridad de los datos;
- no obstante, la utilización de este operador sobre los vectores de encadenamiento dificulta parcialmente los ataques.

La segunda conclusión obedece a que en el modo de encadenamiento original, el atacante tenía suficiente con combinar tres (o más) de las ecuaciones (3.3) para calcular el primer bloque falso de criptograma, sin embargo al introducir el operador de rotación ahora debe combinar b+1 ecuaciones (esto le significará, entre otras cosas, tener que conocer más bloques de mensaje en claro). Así pues parece que la introducción de las rotaciones aporta alguna propiedad que explotándola convenientemente puede llegar a robustecer definitivamente el método. El objetivo va a ser aumentar el número mínimo de ecuaciones que hay que combinar para llevar a cabo el ataque, de manera que se sobrepase la longitud máxima que pueden tener los criptogramas[9] (obviamente sin posibilitar nuevos ataques).

Una nueva variación a considerar consiste en aplicar el operador $T\{\}$ sólo a uno de los vectores encadenados, tal y como se muestra en la Figura 3.11.

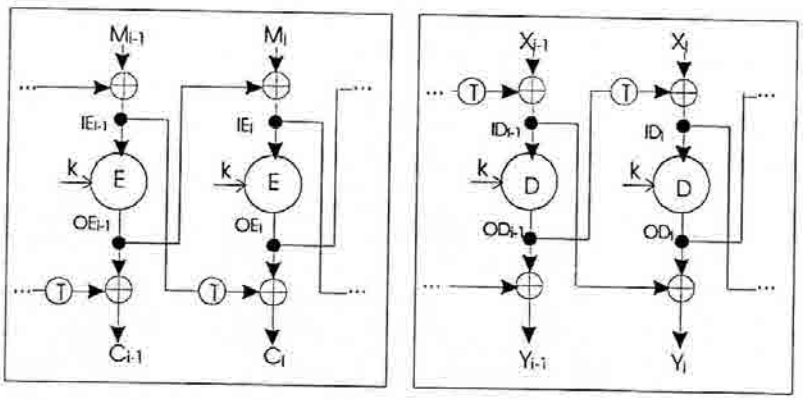


Figura 3.11

Si se toma $T\{\}$ como el operador rotación $\{.\}_{>1}$, es fácil deducir, mediante la modificación correspondiente de las ecuaciones (3.4) (3.5) y (3.6), que ahora se necesita combinar $2b$+1 ecuaciones análogas a las de (3.6) para determinar un X_{ij} tal que permita llevar a cabo el ataque descrito. Desafortunadamente, en el entorno que nos movemos las longitudes de los mensajes pueden ser bastante superiores a $2b$+1 bloques por lo que esta modificación no ofrece todavía suficientes garantías de integridad.

La característica del operador de rotación que viene a determinar el número de ecuaciones que se tienen que combinar para calcular el bloque falso X_{ij}, es lo que vamos a llamar el "periodo de inversión" del operador $T\{\}$. Definimos como periodo de inversión al menor entero m tal que

$$T^m\{X\} = X; \qquad \forall X .$$

En el caso del operador de rotación m es precisamente el tamaño del operando (en bits), esto es b. Dado que las permutaciones son operadores de naturaleza muy similar a las

[9] Se debe tener en cuenta que en las redes locales consideradas en este trabajo el tamaño de los paquetes de datos está acotado por un valor máximo.

rotaciones pero pueden presentar un periodo de inversión mucho mayor, pueden ser de gran interés en el problema que nos ocupa.

Una permutación $P\{\}$ definida sobre un conjunto A es una aplicación biyectiva de A en A. Si A tiene b elementos [VAR90], la permutación $P\{\}$ se suele representar

$$P=\begin{pmatrix} 1 & 2 & \dots & b \\ P\{1\} & P\{2\} & \dots & P\{b\} \end{pmatrix}$$

Si se aplica reiteradamente $P\{\}$ sobre un determinado elemento de A, necesariamente al cabo de unas cuantas iteraciones (como máximo b) se vuelve a obtener el mismo elemento. Cada una de las secuencias que determina la permutación de esta manera se denomina "ciclo". Dado un elemento x de A se denomina "grado o longitud del ciclo" que pasa sobre x, al menor número q tal que $P^q\{x\}=x$. Este ciclo se representa por la secuencia de elementos por los que pasa:

$$\left(x, P\{x\},\dots, P^{q-1}\{x\}\right).$$

Así pues, una permutación se puede representar también por el conjunto de ciclos que genera, por ejemplo:

$$P=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 4 & 1 & 2 & 6 \end{pmatrix}=(1\ \ 3\ \ 4)(2\ \ 5)(6)=(1\ \ 3\ \ 4)(2\ \ 5).$$

Se debe observar que cada ciclo define una subpermutación sobre A, en el sentido de que define una permutación sobre el subconjunto de los elementos por los que pasa.

1	2	3	4	5	6	7
2	4	1	3	7	5	6
4	3	2	1	6	7	5
3	1	4	2	5	6	7
1	2	3	4	7	5	6
2	4	1	3	6	7	5
4	3	2	1	5	6	7
3	1	4	2	7	5	6
1	2	3	4	6	7	5
2	4	1	3	5	6	7
4	3	2	1	7	5	6
3	1	4	2	6	7	5
1	2	3	4	5	6	7

Tabla 3.1: Permutación de 7 elementos (orden 12)

Dada una ordenación X de los elementos de A, entonces $P\{X\}$ define una nueva ordenación (aplicando la permutación a cada uno de los elementos de X). Si componemos reiteradamente $P\{\}$ sobre X, al cabo de un cierto número de veces m volveremos a obtener X. El mínimo número m para el que este suceso se produce se denomina "orden de la permutación". Es inmediato demostrar que el orden de una permutación es el mínimo común múltiplo de las longitudes de sus ciclos[10]. Es precisamente esta característica de las permutaciones la que vamos a explotar en nuestro problema.

En la Tabla 3.1 se muestran las diferentes ordenaciones que se obtienen a partir de la secuencia [1,2,...,7] aplicando reiteradamente la permutación

[10] Así por ejemplo, si la permutación tiene exactamente un ciclo, entonces el orden de la misma será precisamente el cardinal de A.

$$P=\begin{pmatrix}1 & 2 & 3 & 4 & 5 & 6 & 7\\ 2 & 4 & 1 & 3 & 7 & 5 & 6\end{pmatrix}=(1\ 2\ 4\ 3)(5\ 7\ 6).$$

Se observa en esta permutación que partiendo de dos subpermutaciones de ordenes m_1=4 y m_2=3 se obtiene un orden de 12, el cual es mayor que el tamaño del operando (7). Se debe observar también que los ordenes de estas subpermutaciones coinciden con el número de elementos a que afectan ya que cada una de ellas define únicamente un ciclo (ver nota al pie 10).

Si en el modo de encadenamiento ilustrado en la Figura 3.11 tomamos como operador de encadenamiento una permutación de bits diseñada "convenientemente", podemos conseguir que el número de ecuaciones que se han de combinar para llevar a cabo un ataque exceda en mucho la longitud máxima de los mensajes del sistema. Observar que lo que se ha definido como "periodo de inversión" del operador $T\{\}$ va a ser ni más ni menos que el orden de la permutación escogida.

El diseño de esta permutación $P\{\}$ ha de tener en cuenta los siguientes criterios:

- **Es deseable maximizar el orden de la permutación.**
  Es decir, maximizar el mínimo común múltiplo de las longitudes de sus ciclos. Habrá que buscar por tanto q enteros $m_1\ m_2\ ...\ m_q$, tales que su suma sea $m_1+m_2+...+m_q=b$ y que, en principio, su *mcm* sea máximo.
- **Es deseable maximizar la "robustez" del sistema.**
  Sea m' un entero que es múltiplo de las longitudes de p de estos ciclos, $m_1\ m_2\ ...\ m_p$ respectivamente (y de ningún otro ciclo de la permutación). Si R es un número aleatorio de b bits, después de aplicarle m' veces la permutación $P\{\}$, un total de $l=(m_1+m_2+...+m_p)$ de sus bits habrán vuelto a su posición original, de manera que la probabilidad[11] de que $P^{m'}\{R\}=R$ será $2^{-(b-l)}$. En el ejemplo de la Tabla 3.1, después de permutar 4, 8, 16 ... veces un número aleatorio de 7 bits, la probabilidad de que se produzca una inversión fortuita será de 2^{-3}, ya que los cuatro primeros bits coincidirán necesariamente y los restantes lo harán al azar.
  Esto significa que no es suficiente con obtener una permutación de orden elevado, sino que además habrá que garantizar que las probabilidades de inversión fortuita se mantengan por debajo de una cota para todas las subpermutaciones de orden pequeño[12].
- **Es deseable minimizar la complejidad de realización de la permutación.**
  El problema de diseño de la permutación $P\{\}$ se reduce al cálculo de los enteros $m_1\ m_2\ ...\ m_q$ según los anteriores criterios. Una vez determinada la longitud de cada ciclo, diseñar la subpermutación correspondiente es trivial. Por ejemplo, se puede realizar una partición de los b bits del operando en segmentos de $m_1\ m_2\ ...\ m_q$ bits consecutivos, de manera que cada ciclo determine una subpermutación sobre el segmento correspondiente. A su vez, cada una de estas subpermutaciones puede ser simplemente una rotación de una posición en los bits del segmento correspondiente[13]. Por tanto, minimizar la complejidad de realización bajo estas condiciones es equivalente a minimizar el número q de ciclos que forman la permutación $P\{\}$.

En definitiva, el diseño de la permutación $P\{\}$ se reduce a encontrar una secuencia de números enteros $m_1\ m_2\ ...\ m_q$, tal que su suma sea igual al tamaño de bloque, b, y que satisfaga

---

[11] En caso de producirse este suceso diremos que se ha dado una "inversión fortuita" de $P\{\}$.

[12] Esto es, que cualquier subpermutación con periodo de inversión menor o igual a la longitud máxima de los mensajes, no deberá afectar a un número elevado de bits. Cada una de estas subpermutaciones estará formada por uno de los posibles subconjuntos de los ciclos de la permutación original.

[13] Dado que los vectores sobre los que se aplica la permutación son números binarios aleatorios, es indiferente cómo se permuten los bits del operando ya que la probabilidad de que el resultado sea diferente del operando no viene dada por esta reordenación sino por el número de bits que conservan su posición original.

los criterios enunciados. De estos tres criterios, el más complejo de satisfacer es el segundo de ellos: para obtener un diseño óptimo según este criterio se deberían calcular todas las descomposiciones en sumas de b y para cada una de estas sumas tomar cada uno de los posibles subconjuntos de números que las forman, calcular su mínimo común múltiplo y confrontarlo con la diferencia entre la suma parcial que forman y b. De esta manera se podría escoger una permutación óptima desde el punto de vista del compromiso entre periodo de inversión y probabilidad de inversión fortuita.

Afortunadamente, otra alternativa de diseño mucho menos compleja va a ser mediante "prueba y error": se construye una permutación arbitraria que sea simple de realizar y se comprueba que satisfaga razonablemente el primer y segundo criterios. De esta manera, en el diseño se da prioridad al criterio de sencillez de realización, pero respetándose en cualquier caso una robustez del sistema que sea suficiente. En conclusión, el resultado será una permutación de mínimo coste de realización pero suficientemente segura.

Así por ejemplo, se puede tomar una permutación compuesta únicamente por dos ciclos de longitudes m_1 y m_2 (con $m_1+m_2=b$). Para minimizar la probabilidad de inversión fortuita hay que maximizar el mínimo de m_1 y m_2. Suponiendo que b viene predeterminado por el algoritmo de cifrado, entonces m_1 y m_2 deberán ser aproximadamente iguales. Por otro lado, para maximizar el periodo de inversión se pueden escoger m_1 y m_2 de manera que sean relativamente primos[14].

Utilizando este criterio de diseño se consigue que el permutador esté compuesto únicamente de dos registros de rotación de longitudes aproximadamente iguales, un periodo de inversión de longitud $b^2/4$, y una probabilidad de inversión fortuita de $2^{-(b/2)}$ (aproximadamente). En la Tabla 3.2 se muestran posibles valores m_1 y m_2 para los valores de b más usuales, junto a los correspondientes periodos de inversión (m) y probabilidades de inversión fortuita[15] (*PIF*).

b	m_1	m_2	m	*PIF*
64	31	33	1023	2^{-31}
128	63	65	4095	2^{-63}
256	127	129	16383	2^{-127}
512	255	257	65535	2^{-255}
768	383	385	147455	2^{-383}
1024	511	513	262143	2^{-511}

Tabla 3.2: Diseño de la permutación $P\{\}$ (dos ciclos) para los tamaños de bloque más usuales.

De los números presentados en la Tabla 3.2 se deduce que el criterio simplificado de diseño de $P\{\}$ es más que suficiente, siendo innecesario acudir a un análisis exhaustivo de todas las posibles permutaciones de b bits. Con este criterio se consigue que la permutación consista en dos simples rotaciones de un bit, el periodo de inversión mínimo sea de 1023 bloques para b=64 (el tamaño máximo de las unidades de datos en las redes locales que nos ocupan es de 1500 octetos) y que la probabilidad de inversión fortuita (mínima) sea de 2^{-31}.

Finalmente, antes de dar por acabado este apartado, es interesante plantearse la siguiente cuestión: ¿cómo ha de ser, en general, el operador de encadenamiento $T\{\}$, para que la confidencialidad y aleatoriedad de los vectores de inicialización, IV_a y IV_b, garanticen la confidencialidad y aleatoriedad de todos los valores *IE*s y *OE*s?

En un extremo, tenemos que si $T\{X\}=0\ \forall X$, entonces es inmediato determinar los valores de todos los *IE*s y *OE*s (excepto el de IE_1):

---

[14] Conseguir estas condiciones va a ser trivial gracias a que b es usualmente una potencia de dos. Si esto es cierto, entonces $b/2$ será par y podemos tomar $m_1=b/2-1$ y $m_2=m_1+2$. Si m_1 y m_2 sólo difieren en dos unidades, el único factor común que pueden tener es precisamente 2, pero por ser $b/2$ par, m_1 será impar. De donde se deduce los m_1 y m_2 construidos de esta forma serán primos entre sí y aproximadamente iguales a $b/2$.

[15] Se muestra la cota inferior de esta probabilidad, esto es, $p\left(P^{m_2}\{X\}=X\right)=2^{-m_1}$.

$$OE_i = C_i \; ; \quad i = 1, \ldots, N$$
$$IE_i = M_i \oplus OE_{i-1}; \quad i = 2, \ldots, N$$

En el otro extremo, si $T\{X\}=X$ entonces, como se muestra en 3.2.4, se tiene un sistema de ecuaciones con $2N$ ecuaciones y $2N+2$ incógnitas para el cual es inmediato demostrar que no tiene solución para ninguna de las incógnitas.

En general, si $T\{\}$ es una aplicación no biyectiva, entonces los valores que no pertenecen al conjunto imagen de $T\{\}$ se pueden descartar en el criptoanálisis. De manera que una vez se conoce C_i algunos valores de OE_i serán descartables. En conclusión, si $T\{\}$ no es biyectiva entonces se estará dando información a un posible intruso sobre los valores que pueden tener los *IE*s y *OE*s (ya sea en mayor o menor cuantía).

Pero si el operador $T\{\}$ es una aplicación biyectiva, supuesta la confidencialidad y aleatoriedad de IE_{i-1} , entonces el valor de $T\{IE_{i-1}\}$ será también secreto y "aleatorio" (en el sentido de que todos los posibles valores son equiprobables). De manera que si $T\{\}$ es biyectiva el atacante no podrá, a la vista de C_i, realizar ninguna suposición más probable que cualquier otra acerca de qué valor tiene OE_i. Observar que tanto la identidad, rotaciones y permutaciones constituyen aplicaciones biyectivas.

3.4 IOBC: un nuevo modo de encadenamiento

3.4.1 Especificación del modo IOBC

Llegados a este punto, estamos capacitados para especificar el modo de encadenamiento para cifradores en bloque que va a permitir garantizar con una sola pasada la confidencialidad e integridad de mensajes de longitud limitada: el modo IOBC.

Las operaciones de cifrado y descifrado vienen dadas en el modo IOBC por las siguientes ecuaciones

$$C_1 = IV_b \oplus E_k\{M_1 \oplus IV_a\}; \quad C_i = P\{IE_{i-1}\} \oplus E_k\{M_i \oplus OE_{i-1}\}; \quad i = 2, \ldots, N; \tag{3.7}$$

$$Y_1 = IV_a \oplus D_k\{X_1 \oplus IV_b\}; \quad Y_i = ID_{i-1} \oplus D_k\{X_i \oplus P\{OD_{i-1}\}\}; \quad i = 2, \ldots, N; \tag{3.8}$$

Análogamente a las ecuaciones (3.3), ahora podemos escribir

$$\left.\begin{array}{l} C_i = P\{IE_{i-1}\} \oplus OE_i \\ M_i = OE_{i-1} \oplus IE_i \end{array}\right\}; \; i = 1, \ldots, N; \text{ donde } P\{IE_0\} = IV_b \text{ e } OE_0 = IV_a \tag{3.9}$$

Donde la permutación $P\{\}$ viene dada por dos subrotaciones: una aplicada a los $b/2$-1 bits más significativos y otra a los $b/2$+1 menos significativos[16]. Esto es,

$$P\{X\} = \left\{[X]^{b/2-1}\right\}_{>1} \left| \left\{[X]_{b/2+1}\right\}_{>1}\right. ,$$

donde $\{x\}^y$ simboliza los y bits más significativos de x, $\{x\}_y$ los y bits menos significativos e $x \mid y$ la "yuxtaposición" de los bits de x e y (los de x en la parte más significativa).

[16] Supodremos a partir de este punto que b es potencia de dos.

Es inmediato demostrar que la operación de descifrado es efectivamente la inversa de la de cifrado. Por inducción, si $X_1=C_1$ entonces

$$Y_1 = IV_a \oplus D_k\{C_1 \oplus IV_b\} = IV_a \oplus D_k\{IV_b \oplus E_k\{M_1 \oplus IV_a\} \oplus IV_b\} = IV_a \oplus M_1 \oplus IV_a = M_1$$

y además,

$$ID_1 = X_1 \oplus IV_b = C_1 \oplus IV_b = IV_b \oplus E_k\{M_1 \oplus IV_a\} \oplus IV_b = E_k\{M_1 \oplus IV_a\} = OE_1$$
$$OD_1 = D_k\{ID_1\} = D_k\{OE_1\} = IE_1$$

Ahora, suponiendo que $ID_{i-1}=OE_{i-1}$, $OD_{i-1}=IE_{i-1}$ y que $X_i=C_i$, entonces

$$Y_i = OE_{i-1} \oplus D_k\{C_i \oplus P\{IE_{i-1}\}\} = OE_{i-1} \oplus D_k\{P\{IE_{i-1}\} \oplus E_k\{M_i \oplus OE_{i-1}\} \oplus P\{IE_{i-1}\}\} =$$
$$= OE_{i-1} \oplus D_k\{E_k\{M_i \oplus OE_{i-1}\}\} = OE_{i-1} \oplus M_i \oplus OE_{i-1} = M_i$$

De donde, $Y_i=M_i$, para $i=1, 2, \ldots, N$.

3.4.2 Evaluación de la confidencialidad garantizada

Para validar el modo IOBC evidentemente no es suficiente demostrar que la operación de descifrado es precisamente la inversa de la de cifrado, sino que además se debe comprobar que ofrece unas garantías razonables en cuanto confidencialidad e integridad.

En cuanto a confidencialidad, el modo IOBC conserva las buenas propiedades de la cuarta modalidad de encadenamiento estudiada en el apartado 3.2. A la vista de los resultados obtenidos en el apartado 3.3, dado que la permutación $P\{\}$ es una aplicación biyectiva vamos a tener garantizada la confidencialidad de los vectores a la entrada y salida del operador $E_k\{\}$ en el proceso de cifrado. Veamos no obstante que, análogamente al caso del apartado 3.2, aunque el atacante conozca todos los bloques de mensaje en claro, no podrá conocer los valores de los vectores directamente a la entrada y salida del operador de cifrado (*IE*s y *OE*s). Según las ecuaciones (3.9), tenemos

$$C_1 = IV_b \oplus OE_1 \quad \text{y} \quad M_1 = IV_a \oplus IE_1$$

Suponiendo que los vectores de inicialización IV_a e IV_b son secretos, el atacante no va poder averiguar los valores IE_1 ni OE_1. Además, si los vectores de inicialización se escogen de manera aleatoria, el conocimiento de C_1 y M_1 no permitirá realizar una suposición de los valores de IE_1 y OE_1 que sea mejor que otra cualquiera. No obstante, tal vez se pueda determinar los valores de IE_1 y OE_1 a partir de las ecuaciones correspondientes al segundo bloque:

$$C_2 = P\{IE_1\} \oplus OE_2 \quad \text{y} \quad M_2 = OE_1 \oplus IE_2 .$$

Se debe recordar que $P\{\}$ es una aplicación biyectiva, fácil de invertir y pública, por lo que conocer IE_1 equivale a conocer $P\{IE_1\}$. Ahora bien, para determinar IE_1 y OE_1 mediante estas ecuaciones sería necesario conocer previamente IE_2 y OE_2. Para determinar a su vez IE_2 y OE_2 podemos iterar el proceso hasta llegar al último paso con el propósito de determinar IE_{N-1} y OE_{N-1} a partir las ecuaciones

$$C_N = P\{IE_{N-1}\} \oplus OE_N \quad \text{y} \quad M_N = OE_{N-1} \oplus IE_N ,$$

donde "chocamos" con el desconocimiento de IE_N y OE_N. En definitiva, si los vectores de inicialización utilizados son secretos y aleatorios, el atacante no va a poder determinar los valores *IE*s ni *OE*s aun cuando conozca, además de los bloques de criptograma, todos los bloques del mensaje en claro.

Ahora, suponiendo que el atacante quiere determinar el valor de un determinado bloque de mensaje en claro que desconoce, $M_{i,}$, y que conoce todos los demás, se va a tener que enfrentar otra vez al sistema de ecuaciones (3.9), pero ahora con una incógnita adicional. Por tanto, no va a poder determinar de ninguna manera el valor de este bloque de mensaje, garantizándose una total confidencialidad del mismo.

La confidencialidad de los valores a la entrada y salida del operador de cifrado no sólo es importante desde el punto de vista de la confidencialidad del texto en claro, sino como se muestra en 3.4.3, va a ser una propiedad fundamental para garantizar la integridad de los datos. Esta propiedad también va a aportar otras ventajas. Por ejemplo, dificulta el criptoanálisis del algoritmo de cifrado empleado, $E_k\{\}$: puesto que los vectores de entrada y de salida al mismo son secretos, no se le posibilita al atacante que vaya recogiendo pares $(X,E_k\{X\})$ con el propósito de descubrir la clave k.

3.4.3 Evaluación de la integridad garantizada

Un análisis de las garantías de integridad de los datos que ofrece el método requiere de un estudio cuidadoso de todo tipo de ataques. Los ataques contra la integridad de datos se pueden clasificar en tres grupos [FOR94]: creación, alteración y destrucción de los datos. Dentro de estas tres categorías distinguimos los siguientes casos:

- *Creación*[17]:
  - ◊ De un criptograma completo. Este ataque consistirá en que el intruso "inyecte" un criptograma espúreo con la esperanza de que el receptor lo dé por bueno.
  - ◊ Inserción de datos en un criptograma. En este ataque, el atacante inserta datos espúreos en un criptograma auténtico.
- *Destrucción:*
  - ◊ De un criptograma completo, o de una secuencia de estos. El atacante intercepta un criptograma, o una secuencia de estos, no permitiendo que llegue al receptor, y sin modificar el contenido de los restantes.
  - ◊ Borrado de parte de un criptograma, pero sin modificar la parte restante.
- *Alteración:*
  - ◊ Reordenación de una secuencia de criptogramas (sin modificar su contenido).
  - ◊ Reordenación de bloques de un criptograma. Análogo al anterior pero con los bloques de un criptograma.
  - ◊ Modificación de un criptograma auténtico. El atacante altera parcialmente, o totalmente, un criptograma auténtico usando para ello toda la información que pueda reunir bajo las condiciones más desfavorables[18]. Este es el caso de análisis más complejo y el que le ofrece más posibilidades al atacante.

Obviamente, para garantizar la integridad de los datos es necesario añadir al mensaje en claro alguna información "redundante" que le permita al receptor comprobar la integridad de los criptogramas que descifra. Dadas las propiedades que presenta el modo IOBC, no es necesario incluir ningún tipo de *checksum* calculado a partir del mensaje (sea criptográfico o no), sino que bastará con añadir en el último bloque de mensaje en claro una secuencia de c bits ("vector o secuencia *ICV*"), independiente del mensaje y acordada previamente entre el emisor y receptor.

---

[17] La inserción de criptogramas, o trozos de estos, de contenido "seleccionado" (no espúreo), se contempla en la modalidad de alteración de datos

[18] En principio, esto significa que el atacante desconoce únicamente la clave de cifrado k (i.e. conoce todos los bloques de mensaje en claro cifrados con la misma). A la confidencialidad de la clave se añade la de los vectores de inicialización.

3.4.3.1 Ataques por inserción (espúrea) y destrucción de datos

INSERCIÓN DE UN CRIPTOGRAMA ESPÚREO

Para identificar unos criptogramas de otros y marcarlos con un "sello" único, se puede añadir a cada criptograma un número de secuencia identificativo[19]. De esta manera, si el criptograma no va acompañado del número de secuencia esperado, el receptor puede desecharlo inmediatamente (este procedimiento permite detectar réplicas de criptogramas antiguos además de permitir una resincronización muy simple en el caso de que se pierdan accidentalmente criptogramas en el canal). Aun en el caso de que el atacante envíe un número de secuencia correcto, si el resto de bloques los escoge "al azar", la probabilidad de que el receptor dé por bueno el criptograma será la probabilidad de que $[Y_N]_c = ICV$. Puesto que no se impone ninguna condición sobre los bloques de criptograma al construirlo, entonces Y_N se comportará como una variable aleatoria uniforme. Por tanto, el criptograma creado de esta manera se dará por bueno con una probabilidad de 2^{-c}, siendo c el tamaño en bits de la secuencia *ICV*.

INSERCIÓN DE BLOQUES ESPÚREOS EN UN CRIPTOGRAMA

Es también inmediato que si el atacante inserta algún bloque espúreo en un criptograma auténtico la probabilidad de éxito será aproximadamente 2^{-c}.

DESTRUCCIÓN DE UN CRIPTOGRAMA, O SECUENCIA DE ESTOS

En este caso, el primer criptograma que llegue al receptor después de un borrado (o varios) tendrá un número de secuencia superior al esperado. De esta manera el receptor detectará con total seguridad la desaparición del mismo. El caso en el que el atacante además modifica la numeración de los criptogramas posteriores a uno interceptado se considera en la categoría de modificación de datos.

DESTRUCCIÓN PARCIAL DE UN CRIPTOGRAMA

Si el atacante borra el final del criptograma, la probabilidad de éxito vendrá dada por la probabilidad de que en el último bloque de mensaje recuperado por el receptor aparezca el código *ICV*. Si el código de comprobación de integridad se escoge aleatoriamente, esto se producirá con una probabilidad de 2^{-c}. Si seleccionamos el valor de *ICV* de manera que no aparezca en el mensaje en claro, tendremos absoluta certeza de que este tipo de ataque nunca prosperará. No obstante, esto requeriría procesar dos veces el mensaje en claro: una para seleccionar el valor de *ICV* y otra para cifrarlo; de manera que el procesamiento requerido sería idéntico al de los mecanismos convencionales. Por otro lado, seleccionando aleatoriamente el valor *ICV*, la probabilidad de que el ataque sea factible va a ser suficientemente baja (haciendo c suficientemente alto), aun cuando el atacante conozca todo el mensaje en claro y el valor de *ICV*.

Si los bloques borrados están en posiciones intermedias, entonces el caso es parecido al de inserción de datos en un criptograma. Sea el criptograma original

$$< NS, C_1, \ldots, C_i, C_{i+1}, \ldots, C_j, C_{j+1}, \ldots, C_N >.$$

[19] A partir de este número de secuencia se pueden determinar los vectores de inicialización (p.ej tomando. $IV_a = E_{k'}\{NS\}$ e $IV_b = E_{k'}\{IV_a\}$, consiguiéndose que sean secretos, aleatorios y que varíen mensaje a mensaje). Como se ve más adelante, es aconsejable que la clave k' sea diferente a la utilizada para cifrar los datos, k.

Si el atacante borra el segmento $< C_i, C_{i+1}, \ldots, C_{j-1} >$ entregando al receptor el criptograma falso

$$< NS, C_1, \ldots, C_{i-1}, C_j, C_{j+1}, \ldots, C_N >,$$

entonces el ataque prosperará si $OE_j=OE_i$, lo cual ocurrirá con una probabilidad 2^{-b}, o en caso contrario si a pesar de todo, por "casualidad" ocurre que $[Y_N]_c=ICV$, suceso que tiene una probabilidad de ocurrencia de 2^{-c}.

3.4.3.2 Ataques por alteración de datos

Incluimos en esta modalidad todos aquellos ataques en los que el atacante entrega al receptor criptogramas formados de manera "inteligente" a partir de un análisis de los criptogramas auténticos.

REORDENACIÓN DE CRIPTOGRAMAS

El simple intercambio de dos criptogramas puede ser detectado por el receptor mediante chequeo del número de secuencia. Aun en el caso de que el atacante intercambie también los números de secuencia, es fácil ver por las ecuaciones (3.8) que al no utilizar en la operación de descifrado los vectores de inicialización que corresponden, la probabilidad de que este ataque pase desapercibido es de 2^{-c} (coincidencia al azar de los c bits bajos de Y_N con los de ICV).

REORDENACIÓN DE LOS BLOQUES DE UN CRIPTOGRAMA

Es inmediato por las ecuaciones (3.7) y (3.8), que el intercambio de dos bloques arbitrarios de un criptograma provocará una propagación "descontrolada" de errores hasta los últimos bloques, haciendo que la probabilidad de que el ataque pase desapercibido es otra vez de 2^{-c}.

FALSIFICACIÓN DE CRIPTOGRAMAS A PARTIR DE INFORMACIÓN "AUTÉNTICA"

Dejando a un lado todos los ataques anteriores, que podríamos considerar un tanto triviales, para validar el modo IOBC se deben analizar las posibilidades que tiene el atacante de construir criptogramas falsos a partir de toda la información que pueda reunir bajo las condiciones más desfavorables.

Sea $\tilde{C}$ un criptograma construido por el atacante de manera "inteligente" para que el receptor lo dé por bueno:

$$\tilde{C} =< NS, X_1, X_2, \ldots, X_{\tilde{N}-1}, X_{\tilde{N}} >.$$

Para que $\tilde{C}$ pueda llegar a pasar el control de integridad, es condición necesaria (aunque no suficiente) que al descifrar cada uno de sus bloques, a la entrada del operador de descifrado ID_i se obtenga un valor OE'_j correspondiente a algún criptograma auténtico C' (que puede ser el mismo criptograma C que espera actualmente el receptor y que el atacante deberá interceptar). Efectivamente, puesto que los IDs y ODs son números binarios aleatorios secretos, si el atacante permite que algún ID tome un valor absolutamente al azar, se producirá en el procedimiento de descifrado una propagación totalmente incontrolable a través de las operaciones $D_k\{\}$, de manera que los c bits más bajos de $Y_{\tilde{N}}$ coincidirán eventualmente con el vector ICV que espera el receptor sólo con una probabilidad de 2^{-c}. Sin embargo, si el atacante de alguna manera consigue imponer que ID_i tome el valor de algún OE'_j, siempre tendrá el recurso de replicar a continuación los bloques del criptograma al que corresponde este OE'_j (esto es, $C'_{j+1}, C'_{j+2}, \ldots$), forzando a que el último bloque descifrado contenga el vector ICV' de este criptograma. Si $ICV'=ICV$, entonces ya habrá conseguido prosperar el ataque, si no

tendrá que realizar alguna operación adicional para conseguir que $[Y_{\tilde{N}}]_c=ICV$. En cualquier caso, imponiendo que ID_i tome un valor OE'_j, el atacante conseguiría eludir momentáneamente la propagación incontrolable de errores a través del operador $D_k()$.

Teniendo en cuenta estas consideraciones vamos a ver cómo falsificar un criptograma falso $\tilde{C}$. Sea $C=<NS,C_1,C_2,\ldots,C_N>$ el criptograma que espera el receptor y que ha sido interceptado por el atacante. Dejando aparte el número de secuencia, que debe coincidir necesariamente con el que espera el receptor, el primer bloque de $\tilde{C}$, X_1, puede construirse de tres maneras diferentes:

a) $X_1=C_1$. De manera que se respeta el criptograma auténtico por lo menos hasta el primer bloque.
b) $X_1=IV_b\oplus OE'_j$. De manera que $ID_1= OE'_j$, correspondiendo OE'_j a algún bloque de un criptograma C' generado previamente por el emisor legítimo (obviamente, puede ser el mismo C). Más adelante se estudia cómo puede intentar el atacante calcular este X_1.
c) X_1 es tal que el atacante no tiene "ninguna certeza" de que $ID_1= OE'_j$ para ningún OE'_j. Es equivalente a escoger X_1 al azar y dar por supuesto que por "casualidad" ID_1 será igual a un determinado OE'_j, lo cual se producirá con una probabilidad de 2^{-b}.

Obviamente, el tercer caso equivale a un fracaso en el intento de construcción del criptograma falso, pues será detectada su falsedad con una probabilidad de $1-2^{-c}$ (prácticamente con absoluta certeza si c es suficientemente grande). Por otro lado, en el primer caso se desplaza temporalmente el problema a la falsificación de los siguientes bloques. De manera que con el segundo bloque X_2 vamos a encontrarnos otra vez con las mismas posibilidades:

a) $X_2=C_2$. De manera que se respeta el criptograma auténtico por lo menos hasta el segundo bloque.
b) $X_2=P\{IE_1\}\oplus OE'_j$. De manera que $ID_2= OE'_j$, correspondiendo OE'_j a algún bloque de algún criptograma C' generado por el emisor legítimo.
c) X_2 es tal que el atacante no tiene "ninguna certeza" de que $ID_2= OE'_j$ para ningún OE'_j. Es equivalente a escoger X_2 al azar y dar por supuesto que por "casualidad" ID_2 será igual a un determinado OE'_j, lo cual se producirá con una probabilidad de 2^{-b}.

Y así sucesivamente hasta el último bloque de criptograma, en caso de ir replicando tal cual el criptograma auténtico C.

Por tanto, si el atacante desea construir su criptograma falso deberá ser capaz de superar con éxito el segundo caso por lo menos en una ocasión. Genéricamente, el problema consiste en calcular un X_i (para algún $i\in\{1,\ldots,\tilde{N}\}$) tal que $ID_i=OE'_j$, suponiendo $X_1=C_1, \ldots, X_{i-1}=C_{i-1}$. Si OE'_j corresponde al criptograma C, bastará con replicar a continuación de X_i los bloques $C_{j+1},\ldots,C_N$ para que el receptor dé por bueno el criptograma falso $\tilde{C}$. Si OE'_j corresponde a otro criptograma anterior C', entonces caben dos posibilidades: que el vector ICV' de C' sea el mismo que el de C, en cuyo caso bastará con replicar a continuación los bloques de C' a partir del $(j+1)$; o bien $ICV'\neq ICV$, en cuyo caso será necesario continuar con el ataque en alguno de los bloques posteriores a fin de conseguir $\left[Y_{\tilde{N}}\right]_c=\left[OD_{\tilde{N}}\oplus ID_{\tilde{N}-1}\right]_c=ICV$. Para ello, el atacante deberá o bien construir otro X_l $(l>i)$ para que $ID_l=OE_m$, correspondiendo OE_m a alguno de los bloques de C, o bien deberá construir análogamente los últimos dos bloques $X_{\tilde{N}-1}$ y $X_{\tilde{N}}$ de manera que en $Y_{\tilde{N}}$ se obtenga el vector de comprobación de integridad.

Sin profundizar más en el detalle de estos intrincados razonamientos, es muy importante tener en cuenta la condición necesaria ya enunciada que debe satisfacer el criptograma falso $\tilde{C}$: cada uno de sus bloques debe construirse de forma que a la entrada del operador de descifrado ID_i se tenga algún valor OE'_j correspondiente a un criptograma auténtico C'. Esto significa que si aseguramos la imposibilidad de que el atacante pueda determinar nigua suma

de la forma $P\{IE_{i-1}\}\oplus OE'_j$, entonces tendremos garantizada la integridad de los datos obtenidos por el receptor.

3.4.3.3 ¿Cómo calcular sumas de la forma $P\{IE_{i-1}\}\oplus OE'_j$?

Como se ha visto, las mejores estrategias que se pueden seguir para construir criptogramas falsos pasan por utilizar toda la información disponible a fin de construir sumas de la forma $P\{IE_{i-1}\}\oplus OE'_j$. Esta información está constituida por todos los bloques de los mensajes en claro que se hayan cifrado con la clave k, de sus correspondientes bloques de criptograma y de la relación que el modo IOBC impone sobre ellos. En definitiva, la información de que puede disponer el atacante son las ecuaciones (3.9), donde los bloques C_i y M_i de cualquier mensaje transmitido por el emisor son conocidos por el atacante.

Así pues, hemos sustituido el problema de integridad inicial por uno de confidencialidad: si se garantiza que el valor de estas sumas es confidencial se tendrá garantizada la integridad de los datos[20]. ¿Cómo se puede intentar el cálculo de estas sumas? Para obtener una respuesta, cabe distinguir dos casos diferentes:

- IE_{i-1} y OE'_j corresponden a criptogramas diferentes. En este caso para cada uno de los dos criptogramas implicados, C y C', tenemos un sistema de ecuaciones del tipo de (3.9). Si los vectores de inicialización empleados para ambos criptogramas son diferentes, entonces ambos sistemas son totalmente independientes. De manera que no habrá manera de despejar una de las incógnitas de uno de los sistemas exclusivamente en función de incógnitas del otro sistema. Por tanto, no se podrá determinar el valor de ninguna suma $P\{IE_{i-1}\}\oplus OE'_j$ siendo IE_{i-1} incógnita del sistema de C y OE'_j del de C'.

  En caso de que se utilizasen los mismos vectores de inicialización (esto es, $IV'_a=IV_a$ e $IV'_b=IV_b$), entonces se puede ver fácilmente que sí es posible el cálculo de estas sumas. Por ejemplo:

  $$X_2 = P\{M_1\}\oplus P\{M'_1\}\oplus C'_2 = P\{IV_a\}\oplus P\{IE_1\}\oplus P\{IV_a\}\oplus P\{IE'_1\}\oplus P\{IE'_1\}\oplus OE'_2 =$$
  $$= P\{IE_1\}\oplus OE'_2$$

- IE_{i-1} y OE'_j corresponden al mismo criptograma C (escribiremos OE_j). Estos vectores intervienen en las siguientes ecuaciones del sistema (3.9):

  $$IE_{i-1} \rightarrow M_{i-1} = OE_{i-2}\oplus IE_{i-1} \text{ y } C_i = P\{IE_{i-1}\}\oplus OE_i$$
  $$OE_j \rightarrow C_j = P\{IE_{j-1}\}\oplus OE_j \text{ y } M_{j+1} = OE_j\oplus IE_{j+1}$$

  Si $i<j$, para despejar $P\{IE_{i-1}\}$ en función[21] de algún OE_j se deben combinar las ecuaciones de C_i, M_{i+1}, C_{i+2}, ..., M_{j-1}, C_j:

---

[20] Es por esta razón que las buenas características de confidencialidad que presenta el modo IOBC van a permitir en segundo lugar garantizar la integridad de los datos.

[21] Como se ha dicho, el sistema de ecuaciones (3.9) está realmente constituido por dos sistemas de ecuaciones totalmente independientes entre sí. Cada uno de estos sistemas está constituido por N ecuaciones con $N+1$ incógnitas. El primero viene dado por los bloques de criptograma de índice impar y los de mensaje en claro de índice par. A su vez, el segundo viene dado por los bloques de criptograma de índice par y los de mensaje en claro de índice impar. Análogamente, en el primer sistema sólo intervienen los IEs de índice par y los OEs de índice impar, mientras que en el segundo es al revés. Esto implica que si i es par, entonces $P\{IE_{i-1}\}$ no se puede despejar en función de OE_j si j es impar (y viceversa también).

Además, esto implica que los dos vectores de inicialización deban ser diferentes. En caso contrario la situación sería análoga al caso en el que se utiliza el mismo vector de inicialización para criptogramas diferentes: si $IV_a=IV_b=IV$ se tiene, por ejemplo, que $P\{M_1\}\oplus P\{C_1\}\oplus P\{M_2\}\oplus C_2 = P\{IE_2\}\oplus OE_2$.

$$C_i = P\{IE_{i-1}\} \oplus OE_i$$
$$M_{i+1} = OE_i \oplus IE_{i+1}$$
$$C_{i+2} = P\{IE_{i+1}\} \oplus OE_{i+2}$$
...
$$M_{j-1} = OE_{j-2} \oplus IE_{j-1}$$
$$C_j = P\{IE_{j-1}\} \oplus OE_j$$

Como se puede comprobar fácilmente, estas ecuaciones permiten escribir[22]

$$P\{IE_{i-1}\} = C_i \oplus M_{i+1} \oplus P^{-1}\{C_{i+2}\} \oplus P^{-1}\{M_{i+3}\} \oplus P^{-2}\{C_{i+2}\} \oplus P^{-2}\{M_{i+3}\} \oplus \ldots$$
$$\ldots \oplus P^{-((j-i)/2-1)}\{M_{j-1}\} \oplus P^{-(j-i)/2}\{C_j\} \oplus P^{-(j-i)/2}\{OE_j\}$$

Análogamente podríamos despejar OE_j en función de $P\{IE_{i-1}\}$, llegando a una ecuación equivalente a la anterior (la misma transformada por el operador $P^{(j-i)/2}\{\ \}$).
Por otro lado, en el caso $i>j$, se deberán combinar las ecuaciones de M_{j+1}, C_{j+2}, M_{j+3}, ..., M_{j-1}, C_j . No obstante, nos ahorramos el análisis de este caso ya que conduce a las mismas conclusiones que el caso $i<j$.
De aquí se deduce que toda la información de que pueda disponer el atacante sólo le permitirá calcular alguna suma $P\{IE_{i-1}\} \oplus OE'_j$ en el caso de que pueda darse que $P^{-(j-i)/2}\{OE_j\} = OE_j$, para $i,j \in \{1,2,\ldots,N\}$ (con $|j-i|$ par). Es decir, que $(j-i)/2$ sea precisamente el periodo de inversión de la permutación $P\{\}$, o que se produzca una inversión fortuita con OE_j. De los resultados del apartado 3.3 se extrae que el periodo de inversión de la permutación $P\{\}$ utilizada en el modo IOBC vale $b^2/4-1$. Por tanto, si se desea imposibilitar los ataques a la integridad entonces es totalmente necesario garantizar que la longitud de los mensajes sea inferior al doble del periodo de inversión de $P\{\}$, es decir que

$$\max\left(\frac{j-i}{2}\right) = \frac{N-1}{2} < \frac{b^2}{4} - 1.$$

Por ejemplo, si b=64 bits, la longitud de los mensajes cifrados con este modo de encadenamiento se deberá limitar por debajo de 2047 bloques (≈16 Kbytes, lo cual es más que suficiente para aplicación en redes locales).
Limitar la longitud de los mensajes excluye la posibilidad de que el atacante pueda construir criptogramas falsos con absoluta seguridad de que no serán detectados. No obstante, la posibilidad de que se produzcan inversiones fortuitas va a acotar superiormente el nivel de integridad que el método garantiza. Efectivamente, la construcción de $P\{\}$ mediante dos ciclos de longitudes $m_1=b/2-1$ y $m_2=b/2+1$ respectivamente, hace que $P^{c_1 \cdot m_1}\{\ \}$ y $P^{c_1 \cdot m_2}\{\ \}$ (siendo c_1 y c_2 números naturales cualesquiera) inviertan "fortuitamente" con probabilidades de $2^{-(b-m_2)}$ y $2^{-(b-m_1)}$, respectivamente. Dado que $m_2>m_1$, entonces el atacante tendrá una probabilidad de $2^{-m_1} = 2^{-(b/2-1)}$ de poder determinar una suma $P\{IE_{i-1}\} \oplus OE'_j$ y por tanto que el ataque prospere "accidentalmente".

[22] En el apartado 3.4.3.4 se muestra que ésta es la única manera de combinar las ecuaciones anteriores consiguiéndo que se cancelen las incógnitas "ubicadas" entre $P\{IE_{i-1}\}$ y OE_j , de manera que sean únicamente estas dos las que aparezcan en la ecuación resultante.

3.4.3.4 Combinación y transformación de las ecuaciones

Sea $T\{\}$ un operador que satisface la propiedad distributiva sobre la suma x-or. Esto es, que

$$T\{X \oplus Y\} = T\{X\} \oplus T\{Y\}.$$

Y sea $T\{\}$ tal que admite operación inversa $T^{-1}\{\}$ (esto es, que $T\{\}$ es biyectiva), tal que

$$T^{-1}\{T\{X\}\} = T\{T^{-1}\{X\}\} = X.$$

Observar las permutaciones de bits, y en particular la permutación utilizada en el modo IOBC, satisfacen estas dos propiedades.

Supongamos ahora que tenemos un número binario Z de b bits definido como

$$Z = T\{X\} \oplus Y, \tag{3.10}$$

donde desconocemos el valor de los términos X e Y.

Vamos a ver que la única forma de transformar Z para obtener a la derecha de la ecuación (3.10) el término X (más otro término que sólo dependa de Y) es precisamente aplicando a operación $T^{-1}\{\}$. Es decir, si $T'\{\}$ es un operador tal que

$$T'\{Z\} = X \oplus T''\{Y\}, \tag{3.11}$$

entonces tanto $T'\{\}$ como $T''\{\}$ van a ser "fundamentalmente" el operador $T^{-1}\{\}$.

Dado que $T''\{\}$ debe ser independiente de X, la ecuación (3.11) se debe satisfacer independientemente del valor concreto que tenga X. Por ejemplo, se debe satisfacer para X=0. De donde se tiene que necesariamente $T''\{\}$=$T'\{\}$. Veamos ahora que si $T'\{\}$ es un operador tal que

$$T'\{Z\} = X \oplus T'\{Y\}, \tag{3.12}$$

entonces $T'\{\}$ tiene que ser la operación inversa $T^{-1}\{\}$ más (x-or) un término constante arbitrario. Si aplicamos el operador $T^{-1}\{\}$ a la ecuación (3.10), dadas las dos propiedades enunciadas tenemos que

$$T^{-1}\{Z\} = X \oplus T^{-1}\{Y\}. \tag{3.13}$$

Si sumamos (x-or) las ecuaciones (3.12) y (3.13), tenemos que para valores cualesquiera de X e Y, se satisface

$$T^{-1}\{Z\} \oplus T'\{Z\} = T^{-1}\{Y\} \oplus T'\{Y\}. \tag{3.14}$$

Observar que el hecho de que (3.14) se satisfaga para valores cualesquiera X e Y, se puede leer equivalentemente como que se debe satisfacer para valores cualesquiera de Z e Y (ya que el hecho de que $T\{\}$ sea biyectiva implica que al recorrer X todos los valores del conjunto $\{1,2,...,2^b\}$, $T\{X\}$ también lo hará). En particular (3.14) se debe satisfacer para Z (ó Y) igual a cero, de donde

$$T'\{X\} = T^{-1}\{X\} \oplus T'\{0\} \oplus T^{-1}\{0\} = T^{-1}\{X\} \oplus C; \tag{3.14}$$

siendo C una constante arbitraria (llamémosla de "indeterminación").

Este resultado va a implicar que para que el atacante pueda combinar las ecuaciones (3.9) con el propósito de obtener una suma $P\{IE_i\}+OE_j$ va a serle obligado tener que aplicar una o más veces el operador $P\{\}$, ó $P^{-1}\{\}$, sobre los bloques de texto en claro y de criptograma

(obviamente siempre puede añadir las constantes de "indeterminación", pero con ello no gana nada).

3.5 Conclusiones: longitud de mensaje, claves, vectores de inicialización y vector de comprobación de integridad

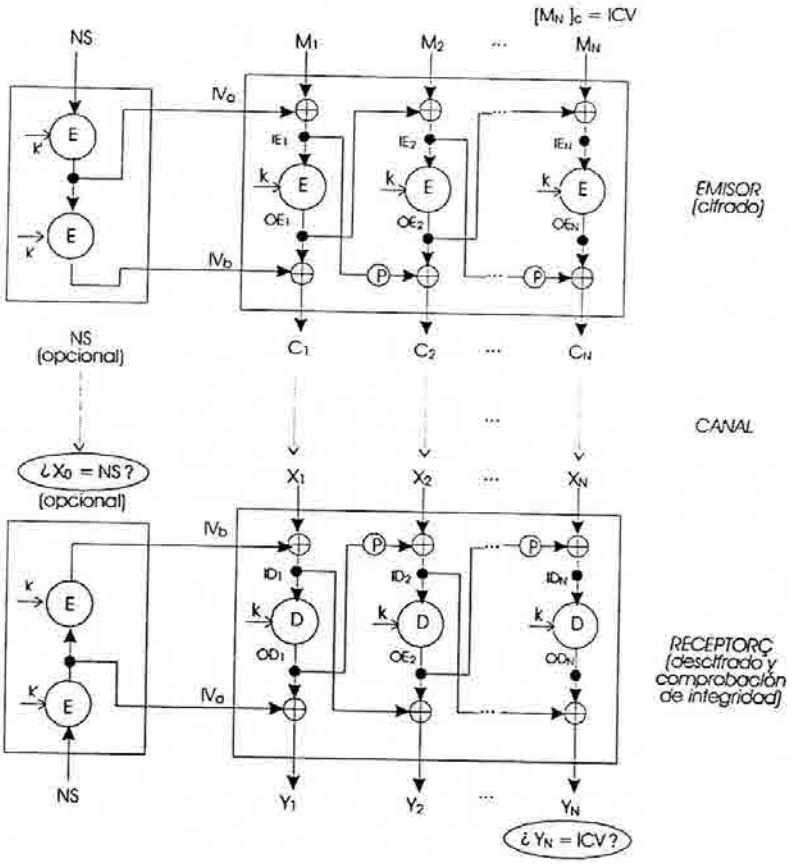


Figura 3.12: Modo de cifrado encadenado IOBC. Parámetros k, k', IV_a, IV_b e ICV.

Antes de dar por finalizado este capítulo se impone realizar un sumario de todas las conclusiones obtenidas con respecto a las condiciones que deben satisfacer los diferentes parámetros que utiliza el modo IOBC. En la Figura 3.12 se muestran la operación de cifrado, (emisor), y las de descifrado y comprobación de integridad (receptor). Se debe recordar que la permutación aplicada en el encadenamiento, $P\{\}$, consiste en dos rotaciones de un bit: la primera se aplica a los $b/2$-1 bits más altos y la segunda a los $b/2$+1 bits más bajos (b se supone par).

Se deben realizar dos puntualizaciones con respecto al contenido de la Figura 3.12. Por un lado señalar que no es necesario enviar el número de secuencia junto a cada criptograma, aunque sí conveniente[23]. Ello se debe a que de esta forma se permite al receptor que recupere el sincronismo con el emisor de manera automática en el caso de que desaparezca algún criptograma (accidental o malintencionadamente). Por otro lado, se debe señalar también que el procedimiento propuesto en esta figura para generar los vectores de inicialización IV_a e IV_b.

[23] No obstante, tampoco es necesario enviar todos los bits de este número de secuencia. En la práctica, bastará con enviar "unos cuantos" de los bits menos significativos a fin de que la probabilidad de desincronización, a consecuencia de una racha de criptogramas borrada accidentalmente por el canal, sea despreciable.

no es el único posible. Realmente, los únicos requisitos sobre estos vectores es que sean de contenido aleatorio[24], y que varíen también aleatoriamente de un criptograma a otro. No obstante, el procedimiento propuesto para generar estos vectores de inicialización resulta muy eficiente ya que hace innecesaria ninguna comunicación extra entre emisor y receptor (exceptuando la renovación de la clave k', como se ve más adelante).

N	IV_a, IV_b	ICV	k	k'
Debe ser menor que $b^2/2-1$ (16376 bytes si b=64) .	Aleatorios. Secretos. Diferentes entre sí. Deben variar mensaje a mensaje.	Aleatorio. Es conveniente que sea secreto (no necesario). Tamaño menor o igual a b/2.	Aleatoria. Secreta.	Aleatoria. Secreta. Debe de ser diferente para números de secuencia repetidos (renovar cada 2^b mensajes)

Tabla 3.3: Criterios de diseño de un sistema IOBC

En la Tabla 3.3 se enumeran las características que deben cumplir la longitud de los mensajes, N, los vectores de inicialización, IV_a e IV_b, la clave de cifrado de datos k, y la que denominaremos "clave de sesión", k'. Es de destacar la necesidad de "renovar" la clave de sesión k' cada vez que el número de secuencia sobrepase el valor máximo 2^b-1. En caso contrario, al repetirse un número de secuencia se repetirían los vectores de inicialización pudiéndose llevar a cabo el ataque descrito en 3.4.3.3. Por contra, si se garantiza que para dos números de secuencia idénticos no se utiliza la misma clave de sesión k', entonces este ataque será inviable. Se debe observar que si bien la clave de sesión es necesario renovarla cada (como máximo) 2^b mensajes cifrados, la clave de cifrado se puede utilizar, en principio, durante periodos mucho más largos. Ello se debe a la imposibilidad que tiene el atacante de recoger pares claro-cifrado asociados a la clave k (esto es, las entradas y salidas al operador de cifrado, *IE*s y *OE*s). Supongamos que cada criptograma procesado por el sistema tiene la longitud mínima de un bloque (medio para mensaje en claro y medio para el vector *ICV*). Utilizando las mismas claves para cifrar un total de 2^b-1 mensajes el emisor enviará un total de $b \cdot 2^b$ bits sobre el canal. Por ejemplo, con un tamaño de bloque de 64 bits (tamaño convencional más pequeño) y con una velocidad de transmisión sobre el canal de 1Tbit/s (todavía faltan unos cuantos años hasta que se disponga de sistemas de comunicación digital a estas velocidades), en el peor de los casos se podría usar el mismo sistema de claves durante 37 años! Esto significa que las necesidades de gestión de claves de este mecanismo de cifrado son prácticamente nulas: al instalar el sistema de seguridad entre el emisor y receptor se cargarán las claves k y k' y se podrán utilizar prácticamente de manera indefinida.

Se deben subrayar dos hechos muy importantes que han de ser tenidos en cuenta al utilizar el modo IOBC:

- El tamaño de los mensajes cifrados mediante este modo de encadenamiento debe ser inferior a dos veces el periodo de inversión de la permutación $P\{\}$ (exactamente,

  $(N-1)/2 < b^2/4-1$.

[24] Esto implica que sean independientes del mensaje en claro y del criptograma. Esta es la razón fundamental de que se utilice para su generación una clave k' diferente a la de cifrado de datos k. Por ejemplo, se puede comprobar de manera sencilla que tomando $k'=k$ y $IV_a=E_k\{NS\}$ e $IV_b=E_k\{IV_a\}$, entonces, entre otras cosas si $M_1=0$ entonces $C_1=0$ produciéndose una vulnerabilidad en la confidencialidad de datos.

- El nivel de integridad garantizado es tal que la probabilidad de que el mejor ataque tenga éxito es de

  $2^{-\min(b/2-1,c)}$.

  Por tanto, tal y como se señala en la Tabla 3.3, no tiene sentido utilizar vectores de comprobación de integridad de longitud superior a la mitad del tamaño de bloque.

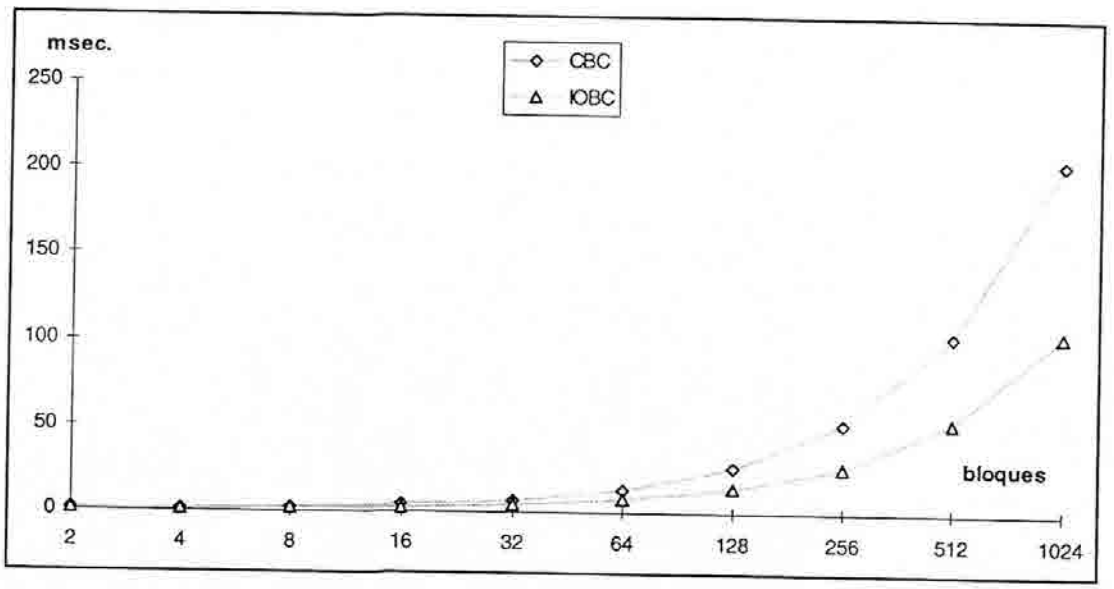


Figura 3.13: Tiempo requerido para dotar de confidencialidad e integridad a mensajes de diferentes tamaños, mediante los modos CBC (2 pasadas) e IOBC. Tiempos medidos para una realización *software* sobre una plataforma 486 DX4, usando el algoritmo DES.

En la Figura 3.13 se muestra una comparación entre los tiempos requeridos para cifrar y dotar de integridad a mensajes de diferentes longitudes mediante una realización *software* de los modos CBC e IOBC con el algoritmo DES. La longitud mostrada en el eje de abcisas corresponde a la del mensaje una vez se le ha añadido el código de comprobación de integridad (ocho octetos para el modo CBC y cuatro en el caso del IOBC). Para cada una de las longitudes mostradas (2, 4, 8, ..., 1024) se muestra el tiempo observado de procesamiento por mensaje, medido sobre una muestra de 1024 mensajes para cada longitud y utilizando claves, vectores de inicialización y números de secuencia diferentes en cada caso. De esta figura se extrae como conclusión respecto a velocidad de procesamiento que para mensajes muy cortos ambos métodos son equivalentes, pero que a medida que aumenta la longitud de éstos, en condiciones similares el modo IOBC duplica prácticamente la velocidad del modo CBC. Para el mismo experimento, en la Figura 3.14 se muestran el tiempo medio por bloque y la relación de velocidades para ambos modos de encadenamiento en función, una vez más, de la longitud de los mensajes procesados. La razón de que para mensajes muy pequeños el modo IOBC presente una complejidad similar o incluso superior a la del modo IOBC se debe a la necesidad de generar para cada mensaje los vectores de inicialización a partir del número de secuencia. Con el método propuesto para generar estos vectores, esto significa que con el modo IOBC se deben cifrar dos bloques extra además de los del propio mensaje (además de la relativamente pequeña sobrecarga que significa encadenar dos vectores en vez de uno y aplicar la permutación de encadenamiento a uno de ellos).

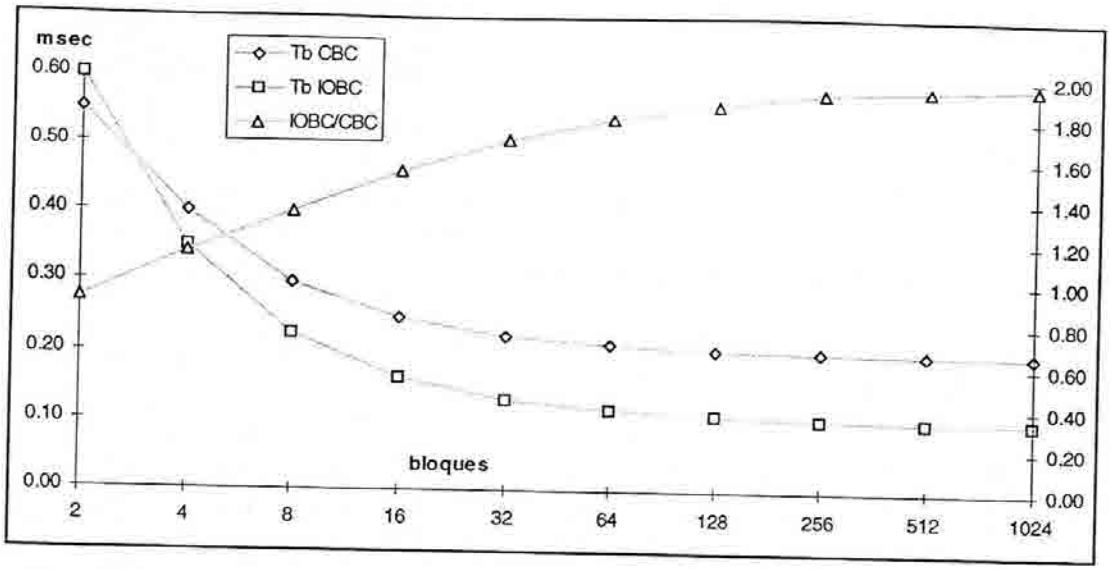


Figura 3.14: Tiempo de procesamiento por bloque (T_b) y relación de velocidad para los modos CBC e IOBC en el experimento de la Figura 3.13

Se concluye pues que el modo IOBC es realizable con un coste de complejidad aproximadamente la mitad que los mecanismos de confidencialidad e integridad basados en el estándar CBC. Por otro lado, dado que mediante el modo CBC se pueden generar códigos de comprobación de integridad de hasta un bloque, se podría pensar que el nivel de seguridad ofrecido por el modo IOBC es inferior ya que, como se ha argumentado, el tamaño máximo de código de integridad que permite utilizar este segundo modo es sólo de medio bloque. No obstante, en la práctica el modo CBC se utiliza normalmente para generar códigos de integridad de medio bloque [BELL94]. Por tanto, desde un punto de vista "práctico" el nivel de integridad garantizado por ambos modos de encadenamiento va a ser idéntico. En definitiva, el modo IOBC permite garantizar un nivel de integridad estándar a un coste aproximadamente la mitad que el de los mecanismos estándares basados en el modo CBC.

4. Protocolos de seguridad para redes 802.3 / Ethernet extendidas

4.1 Introducción

En la figura 2.23 se muestra el escenario de seguridad de referencia para el que se desean diseñar mecanismos de seguridad. En el anterior capítulo se ha presentado un mecanismo de encadenamiento de cifrado en bloque que permite garantizar tanto la confidencialidad como la integridad de los datos procesándolos tan solo una vez, por lo que su aplicación va a ser particularmente interesante en nuestro problema, donde las velocidades de transmisión son relativamente altas. En el actual capítulo vamos a presentar y analizar lo que ya en el capítulo 2 se ha definido como "protocolos de sesión".

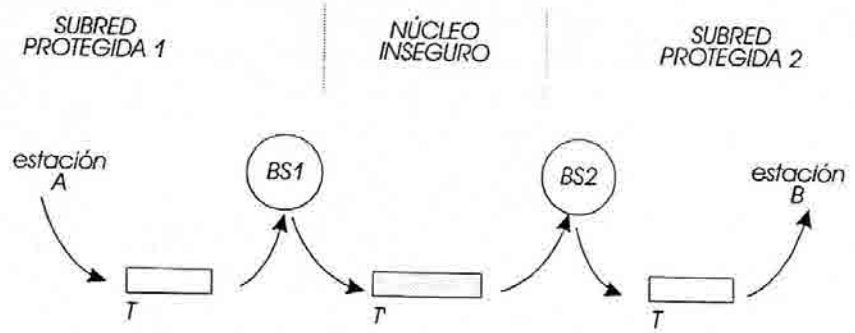


Figura 4.1

En la Figura 4.1 se esquematiza el principio de funcionamiento de estos protocolos de sesión[1]. Tal y como se apunta en apartado 2.5.4, se trata de protocolos que utilizan el principio de encapsulamiento. Cuando un *bridge* seguro *BS1* tiene que enviar una trama[2] *T* desde su subred protegida a otra (conectada al núcleo inseguro mediante otro *bridge* seguro *BS2*), la "encapsula" en una nueva *T'* según el formato definido por el protocolo de seguridad, y la transmite sobre el núcleo inseguro. A su vez, cuando *BS2* recibe la trama *T'* desde el núcleo inseguro deberá recomponer la original *T* y enviarla a su destinatario final en la subred protegida. Obviamente, para que *BS2* sea capaz de recomponer la trama original, toda la información que contiene *T* debe viajar en *T'*. Pero para ello, previamente deberá ser transformada de alguna manera tal que durante su trayecto se tengan garantizados unos determinados servicios de seguridad.

---

[1] Denominaremos *"Secure Bridge Protocol"* (*SBP*) a este conjunto de protocolos.

[2] Denominaremos trama a la unidad de datos del nivel de enlace.

Los protocolos que se proponen permiten garantizar los siguientes servicios de seguridad[3]:

- Confidencialidad e integridad del campo de datos de la trama *T*. En este caso se garantiza la confidencialidad e integridad de los datos de niveles superiores al de enlace.
- Confidencialidad e integridad de toda la información contenida en la trama *T*. En este caso, al proteger también la cabecera de *T* además de los datos de niveles superiores se garantizará el anonimato de las estaciones comunicantes así como la identidad de las mismas (autenticación de origen y destino de *T*).

Los protocolos que aquí se estudian son una evolución de las versiones publicadas en [REC93] y [SOR93]. Obviamente, la eficiencia es un requisito a tener en cuenta en la realización de estos protocolos de seguridad: la degradación de las prestaciones de la red local debería ser inapreciable. A fin de obtener la mencionada eficiencia, garantizando un buen nivel de seguridad, las modificaciones introducidas con respecto a los protocolos originales hacen uso de las buenas propiedades que presenta el modo de encadenamiento IOBC. Es importante señalar, que estos protocolos de sesión se han diseñado dando por supuesto que el operador de cifrado utiliza un tamaño de bloque de 64 bits (por ejemplo, DES e IDEA son dos de los algoritmos más empleados en la actualidad y ambos manipulan bloques de este tamaño).

4.2 Protocolo de sesión

Se describen aquí cuatro variaciones de un protocolo de sesión que permitirá encapsular las tramas a proteger en nuevas tramas "seguras" para ser reconstruidas en destino. Dado que cada una de estas variaciones presenta sus pros y contras, han sido integradas en único protocolo que puede operar opcionalmente en cualquiera de las cuatro versiones.

	CON FRAGMENTACIÓN	SIN FRAGMENTACIÓN
SÓLO CAMPO DE DATOS	v1	v3
CABECERA Y CAMPO DE DATOS.	v2	v4°

Tabla 4.1: Diferencias entre las cuatro modalidades del protocolo de sesión

Tal como se ilustra en la Tabla 4.1, las cuatro versiones del protocolo de sesión nacen de las cuatro posibles combinaciones que se tienen a la hora de incorporar o no:

a) protección a la cabecera de las tramas (campos *DA*, *SA* y *P/L* de la figura 2.22);
b) mecanismo de fragmentación y reensamblaje para tramas que excedan la longitud máxima permitida por la red;

Así pues, en las versiones v1 y v3 únicamente se garantizarán servicios de seguridad a los datos de niveles superiores mientras que en las versiones v2 y v4 se garantizan servicios de seguridad también a los datos de nivel de enlace (i.e. cabecera de las tramas). Si bien parece que la segunda opción es la más interesante, pues garantizará entre otras cosas la autenticidad de las estaciones finales, se verá que el precio a pagar justifica el considerar también la primera opción. El no considerar protección a las cabeceras de nivel de enlace no permitirá garantizar un servicio como el de autentificación de estaciones finales, pero a cambio permitirá una realización más eficiente.

Por otro lado, se puede intuir claramente que el fragmentar una trama larga en dos, enviarlas por separado sobre el núcleo inseguro de la red y reensamblarlas en el *bridge* seguro destinatario, va a introducir necesariamente un retardo importante en el transporte de esta trama sobre la red[4]. Por tanto, va a ser muy atractivo eludir esta fragmentación siempre que

[3] Se debe observar que dado que el mecanismo de cifrado mediante el cual se van a desarrollar estos servicios es el modo IOBC, confidencialidad e integridad van a ser servicios inseparables.

[4] Además va a añadir una complejidad nada despreciable en la realización de los *bridges* seguros: si hay fragmentación, éstos deberán almacenar temporalmente todos los fragmentos incompletos que han recibido en espera de sus "segundas mitades" (y dado que la red no es fiable, se necesitará de algún mecanismo que transcurrido un tiempo prudencial elimine los fragmentos "antiguos"). Como

sea posible. Existen dos alternativas para eludir esta fragmentación. La primera es reconfigurando el *software* de comunicaciones de las estaciones protegidas a fin de acortar la longitud máxima de las tramas que transmiten. De manera que las tramas seguras no sobrepasen en ningún caso la longitud máxima permitida por la red. Si bien la anterior opción es la más recomendable, no deja de ser un tanto engorrosa y en muchos casos incluso inviable (véase 2.2.2.1). La segunda alternativa consiste en tomarse la "licencia" de sobrepasar excepcionalmente la longitud máxima especificada por la red[5]. Si el diseño del protocolo de seguridad se realiza de manera eficiente, el aumento de tamaño que supone el proteger una trama será reducido, por lo que no debería afectar significativamente a las prestaciones de la red. El principal inconveniente de esta opción es la incompatibilidad con la posible presencia de dispositivos de monitorización de tráfico: si no hay manera de reconfigurar a estos dispositivos para indicarles que la longitud máxima ha variado, entonces se tendrán multitud de "alarmas" disparándose continuamente. Dado que en muchos casos no va a ser posible eludir la fragmentación, se hace imperativo la incorporación de algún mecanismo para llevarla a cabo.

4.2.1 Modalidades[6] v1 y v3

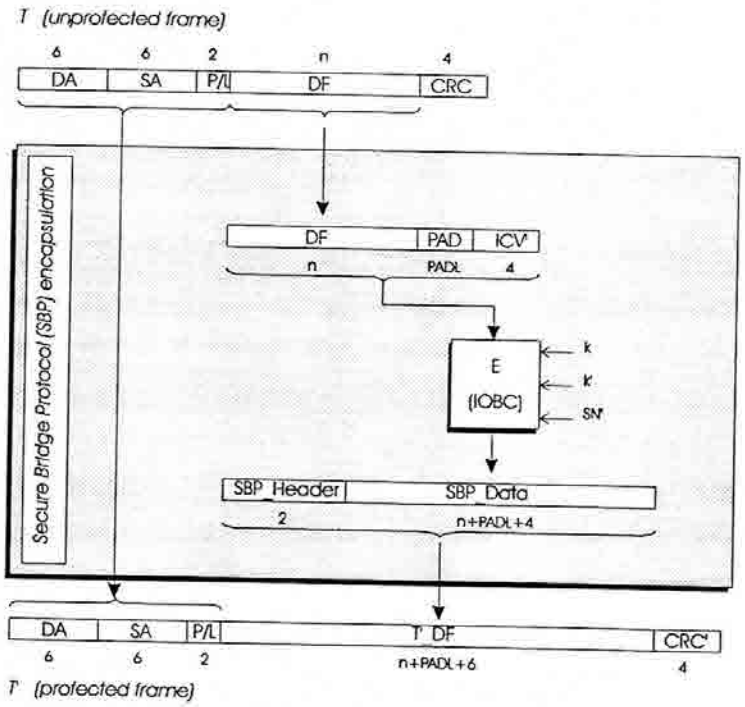


Figura 4.2: Encapsulamiento del protocolo de sesión según las versiones v1 y v3 (v1 sólo tramas cortas).

La característica común más relevante de las modalidades v1 y v3 del protocolo de sesión es que la protección se limita al campo de datos de la trama original. Esto se refleja en la Figura 4.2 en el hecho de que la cabecera de la trama *T* se "transcribe" sin modificaciones en la trama *T'*. Por otro lado, la característica diferencial entre v1 y v3 es el tratamiento que dan a las

consecuencia la realización de los protocolos v3 y v4 podrá ser comparativamente mucho más sencilla que la de los protocolos v1 y v2.

[5] Esto puede considerarse equivalente a modificar al alza la longitud máxima permitida.

[6] En la Tabla 4.2 se da la definición de los símbolos y acrónimos empleados en la descripción del protocolo de sesión.

tramas "excesivamente" largas, esto es a aquellas tramas T cuyo campo de datos tiene una longitud $n > n^a_{MAX}$ (ver más adelante).

En la Figura 4.3 se ilustra el procedimiento de fragmentación al que somete el protocolo v1 a las tramas excesivamente largas. Tanto v1 y v3 provocan un aumento en la longitud de T' con respecto a la de T que va entre 6 y 13 octetos, dependiendo de la cantidad de relleno que sea necesaria (para obtener una longitud a la entrada del operador de cifrado que sea múltiplo de 8 octetos). A partir de esto, es inmediato comprobar que si la longitud del campo de datos de la trama original T es superior a 1492 octetos[7] (n^a_{MAX}), entonces es necesario llevar a cabo el procedimiento de fragmentación.

k	Clave de cifrado de datos.
k'	Clave para el cálculo de los vectores de inicialización del modo de encadenamiento IOBC.
SN	Número de secuencia. Corresponde al número de tramas que el *bridge* ha procesado con la clave k'.
SN'	Número de secuencia ampliado con el bit de dirección D a fin de evitar ataques por "reflexión". Junto a k' determina los vectores de inicialización.
	$SN' = D \mid [SN]_{b-1}$.
TSN	(*Transmited Sequence Number*) Los 8 bits menos significativos de SN.
	$TSN = [SN]_8$.
PAD	Campo de relleno.
$PADL$	Longitud del campo de relleno $(0 \div 7)$.
ICV	Vector de comprobación de integridad.
ICV'	Vector de comprobación de integridad modificado.
SBP_Data	Campo de datos del protocolo *SBP*.
	$SBP\_Data = E^{IOBC}_{k,k',SN'}\{DF,PAD,ICV'\}$
SBP_Header	Cabecera del protocolo *SBP*.
	$SBP\_Header = \{Flags,TSN\}$
D	Bit de dirección.
PV	Identificador de versión de protocolo $(0 \div 3)$.
$Flags$	Campo de flags de control.
	$Flags = \{ PV, F, FN, PADDL, D \}$
F	Bit indicador de fragmentación.
FN	Bit indicador de número de fragmento.
$DSBA$	Dirección lógica del *bridge* seguro destinatario.
$SSBA$	Dirección lógica del *bridge* seguro origen.
SPB	Identificador Ethernet reservado para el protocolo *SPB*.

Tabla 4.2: Lista de símbolos y acrónimos

Se puede observar que la división de los datos en los dos fragmentos se realiza con posterioridad a la operación de cifrado. Alternativamente a este procedimiento, se podría

[7] 1492 octetos de datos más 4 del vector de comprobación de integridad dan un número de octetos múltiplo de 8 (1496 = 8·187) a la entrada del operador de cifrado. Al añadir los dos octetos de *SBP_Header*, la longitud del nuevo campo de datos, *T'_DF*, todavía se mantiene por debajo de los 1500 octetos.

partir el campo de datos de la trama original en dos segmentos, siendo cada uno de estos segmentos cifrado y transmitido de manera independiente. Este es el caso de los primeros protocolos diseñados en este trabajo [REC93] [SOR93] o del procedimiento de fragmentación recomendado por el estándar IEEE 802.10 [IEE92]. Desde un punto de vista de la arquitectura de comunicaciones, en la primera alternativa el mecanismo de fragmentación se situaría por debajo del mecanismo de cifrado, mientras que en la segunda sería al revés. Si se tiene en cuenta que el cifrado de un mensaje requiere en cualquier caso de una fase de inicialización (por ejemplo, acceder a la base de datos de seguridad, inicializar claves, etc.), entonces se hacen evidentes las ventajas de la primera alternativa frente a la segunda: incluso con tramas fragmentadas sólo será necesario inicializar una vez el mecanismo de cifrado, simplificándose de esta manera el procesamiento de las tramas.

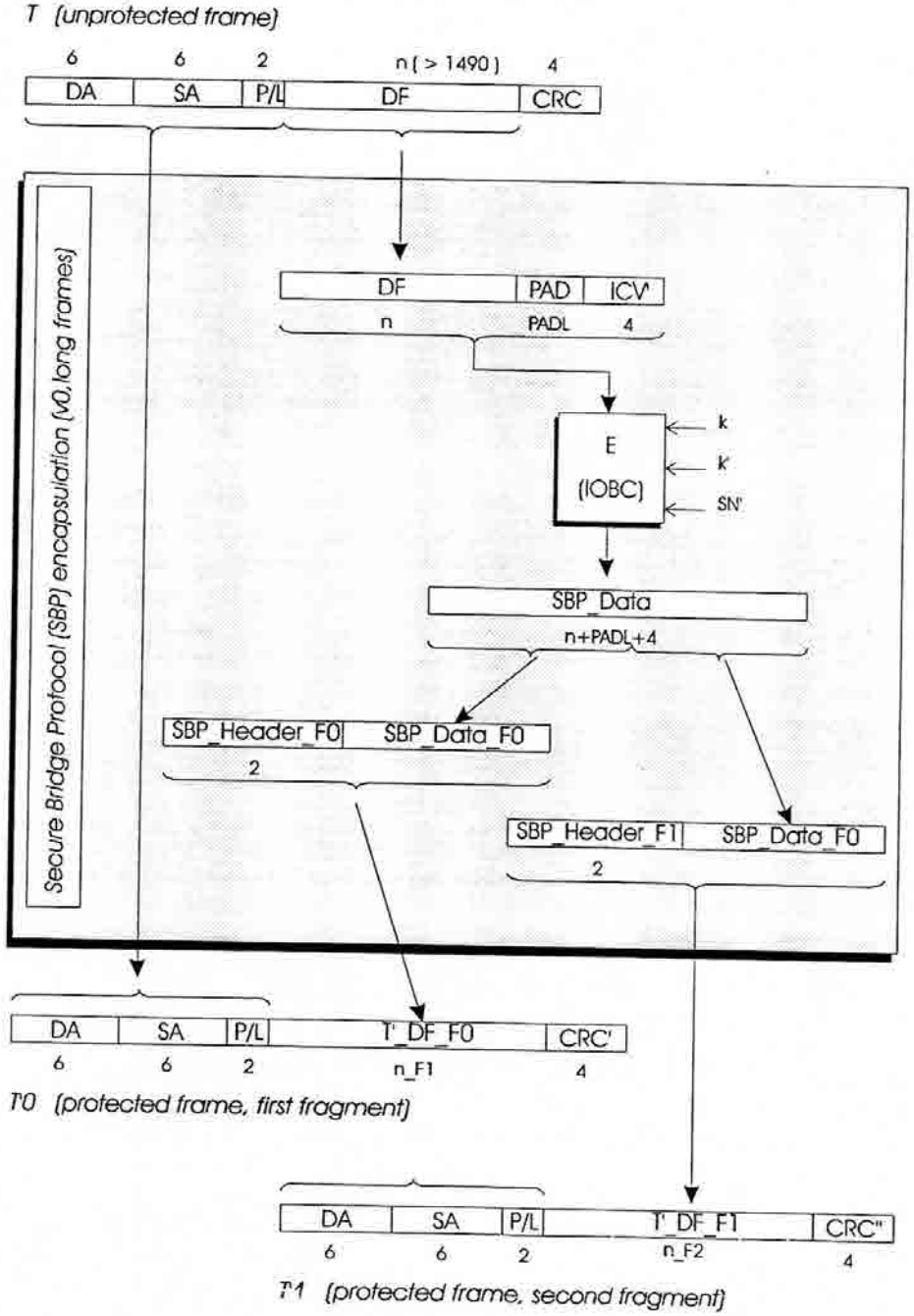


Figura 4.3: Fragmentación de tramas según el protocolo v1

4.2.2 Modalidades v2 y v4

La característica común más relevante de las modalidades v2 y v4 es que protegen toda la información de la trama original, incluyendo su cabecera (i.e. direcciones de las estaciones destino y origen y el campo *P/L*). Para permitir un correcto encaminamiento de la información, la cabecera de la trama protegida *T'* utiliza ahora las direcciones de los *bridges* seguros destino y origen (campos *DSBA* y *SSBA* respectivamente en la Figura 4.4) y un identificador reservado del protocolo de sesión (campo *SBP*). Tal como se indica en la Tabla 4.1, y análogamente al caso de las versiones v1 y v3, la característica diferencial entre v2 y v4 es el tratamiento que dan a las tramas largas (i.e. aquellas tramas *T* cuyo campo de datos tiene una longitud $n > n^{b}_{MAX}$.

El procedimiento de fragmentación llevado a cabo por el protocolo v2 en el caso de tramas excesivamente largas es idéntico al utilizado en el caso de v1: el campo de datos cifrados (*SBP_Data*) se divide en dos fragmentos, a cada uno de los cuales se le añade su correspondiente cabecera de seguridad, *SBP_Header*, y se envían sobre dos tramas hacia el *bridge* seguro destinatario (usando la cabecera { *DSBA, SSBA, SBP*} según se ve en la Figura 4.4). Dado que el incremento de longitud que producen v2 y v4 en las tramas está entre 20 y 27 octetos (dependiendo del relleno requerido), es inmediato comprobar que el protocolo v2 fragmentará las tramas *T* cuyo campo de datos exceda los 1478 octetos (n^{b}_{MAX}).

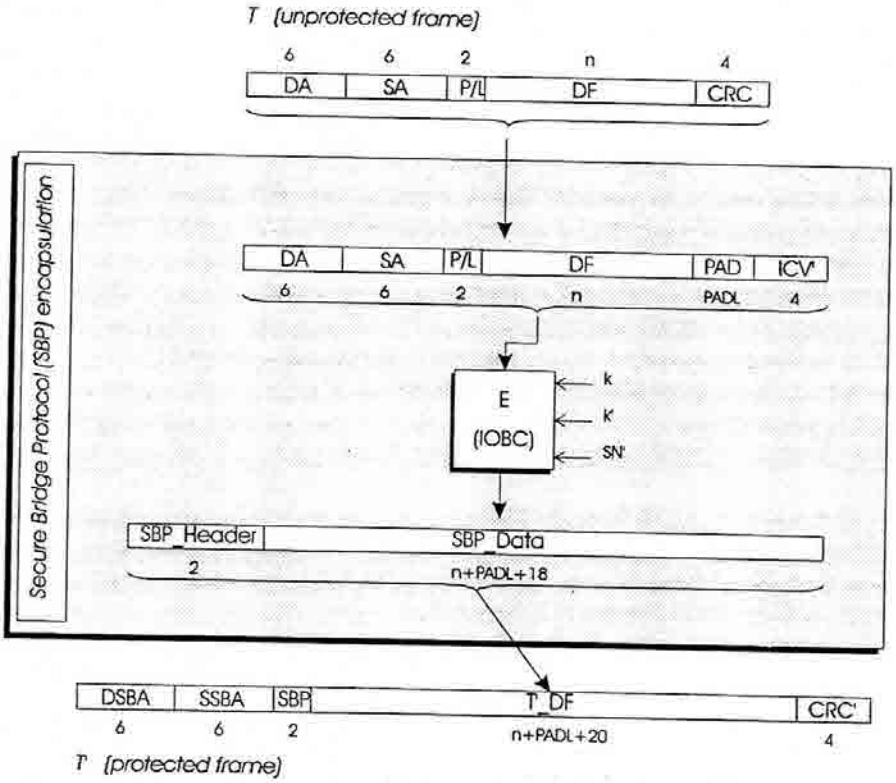


Figura 4.4: Encapsulamiento del protocolo de sesión según las versiones v2 y v4 (v2 sólo tramas cortas).

4.2.3 Obtención de parámetros y construcción de los campos del protocolo de sesión

4.2.3.1 Parámetros de cifrado *k*, *k'*, *SN'*, *PAD* e *ICV'*

El mecanismo de cifrado de datos requiere de los siguientes parámetros:

- la clave *k* para el cifrado de datos;
- la clave *k'* para generar los vectores de inicialización del modo IOBC;
- el número de secuencia *SN'* a partir del cual se generan estos vectores de inicialización;
- el campo de relleno *PAD*;
- el vector de comprobación de integridad *ICV'*.

La obtención de estos parámetros se realiza mediante consulta de una base de datos de seguridad (*SBP Data Base*), cuyo mantenimiento es responsabilidad de los mecanismos de administración de seguridad. En la Figura 4.5 se muestra cómo mediante consulta a esta base de datos, a partir del par de direcciones origen y destino de la trama original *T*, se obtienen, entre otros, los parámetros *k*, *k'*, *SN* e *ICV*. No obstante, se debe observar que los valores del número de secuencia *SN'* y del vector de comprobación de integridad *ICV'* empleados en el cifrado del campo de datos no corresponden directamente a los valores *ICV* y *SN* devueltos por la base de datos.

En primer lugar, el número de secuencia *SN'* (utilizado para generar los vectores de inicialización) se calcula como la extensión de *SN* con el bit de dirección *D* en la posición de mayor peso. Esto va a permitir que entre cada par de *bridges* seguros BS_1 y BS_2 se pueda utilizar el mismo par de claves *k* y *k'* en los dos sentidos posibles de la comunicación. Si se garantiza que el valor de *SN* no sobrepasa $2^{b-1}-1$, y a las dos posibles direcciones $BS_1 \rightarrow BS_2$ y $BS_2 \rightarrow BS_1$ se simbolizan con el bit *D*, entonces cada *bridge* utiliza un rango de contaje independiente, permitiendo que se puedan utilizar las mismas claves en ambos sentidos[8]. De esta manera se evitan engorrosos sistemas de mantenimiento distribuido de un único contador o tener que utilizar claves diferentes en ambos direcciones (reduciéndose a la mitad el número de claves que debe almacenar cada *bridge* con respecto a este último caso). Adicionalmente, dado que cada trama protegida tiene asociada una dirección en su transmisión y dado que se garantiza la integridad del bit *D*, este procedimiento imposibilita además los ataques por reflexión.

En segundo lugar, el vector de comprobación añadido al final de la trama *ICV'* se obtiene a su vez como suma x-or[9] del vector de integridad *ICV* obtenido en la base de datos de seguridad y de la cabecera del protocolo de seguridad *SBP_Header*[10]. El propósito de esta operación es garantizar la integridad del campo *PADL* que viaja en la cabecera *SBP_Header*. Como se ve más adelante, la modificación de cualquiera de los campos de esta cabecera no significa ninguna amenaza a la seguridad del protocolo, exceptuando en el caso del campo *PADL*: si no se autentifica este campo y se permite que un posible atacante lo pueda modificar a su voluntad, entonces el *bridge* receptor podría extraer una cantidad de relleno diferente a la que insertó el emisor. Una solución a esta amenaza podría ser añadir el campo indicador de longitud de relleno dentro del conjunto de datos cifrados. No obstante, esto requeriría expandir aun más la longitud de las tramas. Por contra, la solución que proponemos no produce expansión gracias a que aprovecha los bits disponibles en la cabecera *SBP_Header*, pero tiene el inconveniente de que esta viaja en claro. Afortunadamente, al usar como vector

---

[8] Se debe recordar del anterior capítulo que el modo de encadenamiento IOBC impone que mientras se estén utilizando unas determinadas claves *k* y *k'*, el valor del número de secuencia no se repita.

[9] Dado que *ICV* tiene 4 octetos y *SBP_Header* sólo 2, la suma se aplica sólo a los dos octetos bajos.

[10] Como se ve más adelante, en el caso de tramas fragmentadas las cabeceras *SBP_Header* de ambos fragmentos son ligeramente diferentes, pero para ambas se utiliza un único *ICV'*. En este caso se puede utilizar la cabecera de cualquiera de ellas en la suma con *ICV*.

de comprobación de integridad la suma de *SBP_Header* (y en particular PADL) con *ICV*, el *bridge* receptor puede comprobar la integridad de esta cabecera: si el atacante modifica uno sólo de sus bits, sustituyendo *SBP_Header* por otro *SBP_Header'* al descifrar los datos, el receptor no obtendrá en el último bloque $ICV \oplus SBP\_Header'$, detectando de esta manera la modificación. Se debe observar que este "truco" preserva las requisitos que en el anterior capítulo se imponían sobre el vector de integridad (i.e. si *ICV* es aleatorio, *ICV'* también lo será, y lo mismo en cuanto a su secreticidad).

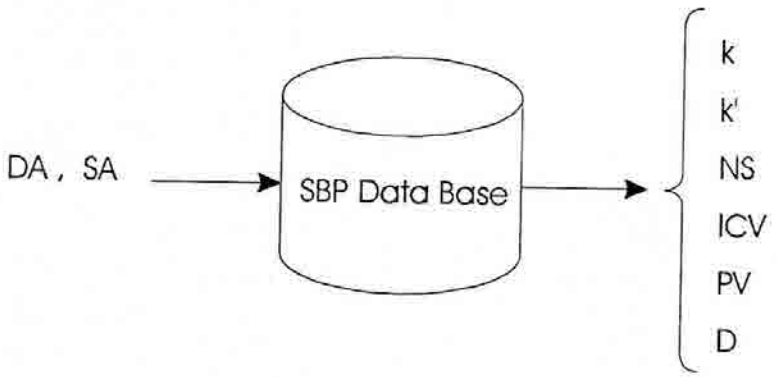


Figura 4.5: Obtención de los parámetros de seguridad

En cuanto al contenido del campo de relleno *PAD* cabe decir que su valor es indiferente: se puede rellenar tanto con ceros como con cualquier otro valor, no siendo necesario ni siquiera que los *bridges* seguros lo acuerden previamente. Esto se debe por un lado a que el *bridge* receptor simplemente descarta este relleno, y por otro lado, el modo de encadenamiento IOBC garantiza la confidencialidad e integridad de los datos incluso en el caso de que un posible atacante conozca parte del mensaje en claro.

4.2.3.2 Cabecera *SBP_Header*

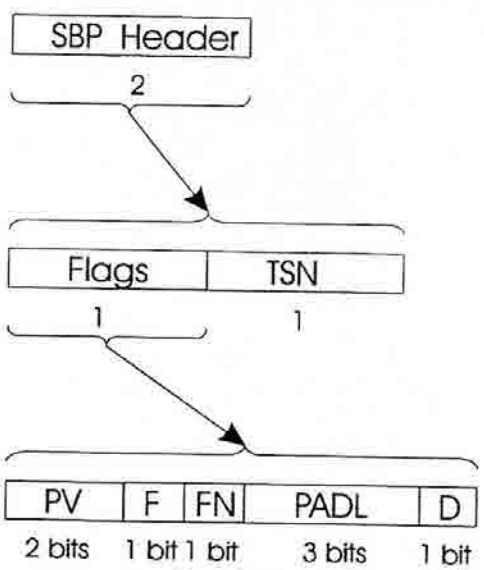


Figura 4.6: Construcción de la cabecera del protocolo *SBP*

En la Figura 4.6 se muestra que la cabecera *SBP_Header* está compuesta por dos campos: el primero, *Flags*, contiene una serie de elementos de control del protocolo; mientras que el segundo, *TSN*, recoge los ocho bits más bajos del contador de secuencia.

El propósito del campo *TSN* es permitir la resincronización automática del receptor en caso de que accidentalmente se pierda en el canal una ráfaga de hasta 255 tramas consecutivas. Suponiendo que el tamaño de la ráfaga más larga que se puede perder es precisamente 255, entonces el receptor[11] podrá calcular los 64 bits el número de secuencia *SN'* con que debe descifrar la trama. Si accidentalmente alguna vez desaparece una ráfaga más larga (o un atacante borra intencionadamente una secuencia larga de tramas), entonces el receptor no será capaz de recomponer el valor correcto de *SN'*. No obstante el mecanismo de integridad detectará este ataque, pudiéndose recuperar a través de otros medios el buen funcionamiento del sistema.

A su vez, el campo *Flags* está compuesto por los siguientes subcampos:

- *PV* (*Protocol Version*). Identificador de versión de protocolo que encapsula la trama *T'*. Su función es meramente informativa (es redundante pues el *bridge* destinatario puede determinar la versión de protocolo a partir de las direcciones en la cabecera de la trama).
- *F* (*Fragmented*). Este *flag* indica si *T'* encapsula una trama completa, o bien si corresponde a un fragmento (*F* activo en el segundo caso). Si se trata de un fragmento, el *bridge* seguro destinatario deberá recuperar el otro antes de proceder a desencapsular esta trama.
- *FN* (*Fragment Number*). En caso de que *F* esté activo, este bit indica si la trama en cuestión corresponde al primer (0) o segundo (1) fragmento.
- *PADL* (*Pad Length*). Estos tres bits indican la longitud del relleno que deberá extraerse en el procedimiento de desencapsulación.
- *D* (*Direction*). Este flag indica la dirección en que viaja la trama. Puesto que se utilizan las mismas claves en las dos direcciones posibles entre dos *bridges* seguros, este flag permite detectar el intercambio de las direcciones destino y origen en un ataque por reflexión. Observar que la presencia de este campo es puramente informativa (y redundante), pues se puede extraer de las direcciones origen y destino de la trama *T'*.

En definitiva, los campos *PV* y *D* son redundantes y por tanto el receptor sólo necesita estrictamente los campos *F*, *FN* y *PADL*. La modificación fraudulenta de los campos *F* y/o *FN* provocaría que el receptor descifre campos *SBP_Data* diferentes a los que envió el emisor, de manera que esta modificación sería detectada gracias al la integridad garantizada por el modo de encadenamiento IOBC. Sin embargo, si no se garantizase de alguna manera la integridad del campo *PADL*, entonces cualquier modificación fraudulenta del mismo provocaría que el receptor extrajera equivocadamente una cantidad de relleno diferente a la insertada por el emisor. De ahí la necesidad de utilizar algún mecanismo de integridad para este campo. Observar que el mecanismo propuesto ($ICV' = ICV \oplus SBP\_Header$), permite chequear la integridad de todos los elementos del campo *Flags*, por lo que el receptor, si lo "cree" conveniente, va a poder utilizar incluso *PV* y *D* con total "tranquilidad".

4.3 Administración de Seguridad

Los aspectos de gestión de la seguridad en una red de datos se pueden clasificar[12] en cuatro categorías [PUR93] [ISO88]:

- Gestión de seguridad del sistema. Se recoge aquí todo lo relacionado con la monitorización de eventos relacionados con la seguridad, su auditoría y el restablecimiento de los niveles de seguridad en caso de anomalías.

[11] Esto quiere decir que el receptor mantendrá internamente una "copia" del contador que utiliza el emisor.

[12] La cuarta categoría de mecanismos de administración de seguridad queda más allá del ámbito de interés de este trabajo por lo que no recibe más atención.

- Gestión de los servicios de seguridad. Se incluye en esta categoría la selección de servicios de seguridad (p.ej. confidencialidad de datos de aplicación, no repudiación, etc.), de los mecanismos que se utilizarán para realizarlos (p.ej. los algoritmos de cifrado), y en el caso de entornos multidominio, la negociación de servicios y mecanismos a aplicar en las interacciones con otros sistemas.
- Gestión de los mecanismos de seguridad. Este apartado contempla la gestión de todos los parámetros usados por los mecanismos de seguridad. Esto es, la construcción, negociación, distribución, actualización, etc. de información tal como las claves usadas por los algoritmos de cifrado.
- Seguridad de la gestión de red. El correcto funcionamiento de los mecanismos de gestión de los recursos y servicios de la red es un aspecto clave para garantizar la disponibilidad de los mismos. En un contexto de seguridad realista, no sólo son necesarias protecciones contra anomalías accidentales sino también contra las intencionadas.

Aunque en este trabajo no se profundiza en los aspectos de gestión del sistema de seguridad propuesto, parece conveniente apuntar qué debe ser gestionado y qué características generales deben satisfacer los mecanismos de gestión de seguridad.

De la descripción del sistema de seguridad propuesto (ver capítulo 2) se extrae que la centralización es la principal característica de los mecanismos de administración de seguridad de este sistema. Así, la arquitectura del subsistema de administración podría verse como un centro de administración "controlado" por el administrador de la red (sobre el cual recae la responsabilidad de la administrar la política de seguridad) que dispone de una serie de "agentes" distribuidos en cada uno de los *bridges* seguros. Según la anterior clasificación, las funciones de este sistema de seguridad se pueden ordenar de la siguiente manera:

- Gestión de seguridad.
  - ◊ Monitorización de eventos. Los agentes de seguridad recogerán información a lo largo de la operación del *bridge* seguro haciéndola llegar al centro de administración. Esta información hará referencia a eventos relacionados con la seguridad del sistema. Por ejemplo, estadísticas acerca de tramas que no han pasado el test de integridad, estadísticas acerca de tramas que han sufrido fragmentación, estadísticas generales acerca del tráfico total cursado por el *bridge* (con o sin protección), etc.
  - ◊ Auditoría. El centro de administración procesará la información de monitorización a fin de poder detectar posibles ataques, recomendar cambios en la configuración del sistema, etc.
  - ◊ Restablecimiento de seguridad. En condiciones anómalas puede ser necesario reinicializar total o parcialmente el sistema. Por ejemplo, cuando dos *bridges* pierdan la sincronización en el número de secuencia (ver 4.2.3.1) fortuitamente o a consecuencia de un ataque intencionado, será necesario que el centro de administración "colabore" con ellos a fin de realizar la resincronización.
- Gestión de los servicios de seguridad.
  - ◊ Servicios de confidencialidad e integridad. El sistema de seguridad propuesto únicamente protege tramas que viajan entre estaciones consideradas como "sensibles". Esto significa que para cada subred protegida es necesario que el administrador registre en el centro de administración la lista de direcciones físicas de las estaciones sensibles en esta subred. Esta información se hará llegar a los agentes de administración de los *bridges* seguros afectados durante el procedimiento de instalación de los mismos, cuando se produzcan cambios en estas listas y durante los procedimientos de restablecimiento que lo requieran. Los agentes a su vez mantendrán actualizada la base de datos local de seguridad a fin de que las entidades *SBP* puedan disponer de la información necesaria en su operación.
  - ◊ Servicio de control de acceso. Adicionalmente a los servicios de confidencialidad e integridad soportados mediante el protocolo *SBP*, los *bridges* seguros pueden prestar un mecanismo elemental de control de acceso a modo de *firewall*. Además de las listas de estaciones sensibles en cada estación, el administrador puede

establecer qué comunicaciones son posibles y de qué manera se deben llevar a cabo. Por ejemplo, puede decir que la estación *A* no puede establecer comunicaciones externas a su subred; que a la estación *B* sólo se le permitirán comunicaciones externas con otras estaciones para las que se pueda soportar el protocolo *SBP*; que la estación *D* sólo podrá mantener comunicaciones externas en modo inseguro con una determinada lista (enumerada) de estaciones; etc.

- Gestión de mecanismos de seguridad. Se contempla aquí la generación, distribución, mantenimiento, renovación y destrucción de todos los parámetros que requieren compartir cada par de *bridges* para poder llevar a cabo la encapsulación de tramas según el protocolo *SBP*. Entre estos parámetros hay que destacar: la clave de cifrado de datos, la de generación de vectores de inicialización y el código de comprobación de integridad. La gestión de estos parámetros (básicamente: negociación de su valor, distribución por medios seguros y renovación) constituiría lo que genéricamente se denomina "gestión de claves". Como se expone en el capítulo 3, el modo *IOBC* minimiza las necesidades de renovación de claves: incluso un procedimiento manual de distribución de claves podría llegarse a considerar adecuado. No obstante, dado que se necesita distribuir otras informaciones de manera segura desde el centro de administración hacia los *bridges* seguros, lo más razonable es automatizar también esta gestión de claves.

5. Evaluación del impacto sobre las prestaciones de la red.

5.1 Introducción

De entre todos los parámetros que pueden ser útiles para caracterizar las prestaciones de una red de datos cabe destacar dos en el caso de redes locales 802.3/Ethernet: la capacidad del sistema para transferir información por unidad de tiempo (*"throughput"*); y el tiempo que se requiere para que esta información atraviese la red desde el origen hasta el destino ("tiempo de respuesta") [HAW87]. Mientras algunas aplicaciones son muy sensibles al tiempo de respuesta de la red, otras requieren altas velocidades de transferencia[1].

La introducción del protocolo *SBP* en la operación del núcleo inseguro, va a afectar a ambos parámetros. El principal objetivo de este capítulo es determinar la cuantía en la que el *throughput* y el tiempo de respuesta son afectados por los mecanismos de seguridad. Por un lado, la expansión (y posible fragmentación) que sufren las tramas protegidas hace que el *throughput* máximo que pueden cursar los *bridges* seguros se vea reducido con respecto al de un *bridge* transparente convencional. Por otro lado, el incremento de funciones en los *bridges* seguros, así como la modificación de la carga ofrecida al núcleo inseguro (fruto de la expansión y fragmentación) va a implicar un mayor tiempo de respuesta para las tramas protegidas.

Las características que dificultan el estudio analítico de redes Ethernet (ver Anexo A) aún se complican más cuando se considera un sistema formado por varios segmentos de red interconectados entre sí mediante *bridges* (red local extendida). Esto hace que las únicas alternativas a la evaluación sean las medidas de campo y/o la simulación. En el caso de sistemas de nuevo diseño, un inconveniente de las medidas de campo frente a la simulación es la necesidad de disponer de prototipos mediante los cuales llevar a cabo la experimentación [JAI91]. En nuestro caso, este inconveniente tiene una importancia relativa ya que se podría haber dispuesto de los prototipos de *bridge* seguro desarrollados durante las primeras fases de este trabajo[2]. No obstante, las medidas de campo fueron descartadas como metodología de evaluación por las razones expuestas en las siguientes líneas.

Evidentemente, el comportamiento de una red como las que nos ocupan va a depender de factores tan dispares como el número de estaciones conectadas a los segmentos, su distribución física sobre los mismos, las características del tráfico generado por cada una de las estaciones, la topología de interconexión de estos segmentos, la velocidad de procesamiento en

---

[1] Un ejemplo del primer caso serían los protocolos de acceso de terminal en los que el *host* debe devolver un "eco" de cada tecla pulsada por el usuario para que apararezca en pantalla. Otro ejemplo del segundo caso serían los protocolos de transferencia de ficheros, no tan sensibles al retardo pero en los que es deseable que la operación sea "rápida".

[2] Se han desarrollado en el laboratorio dos versiones sucesivas de *bridges* seguros (dos unidades "prototipo" para cada versión) que utilizan protocolos de sesión diferentes al expuesto, pero en los que se podría haber integrado el nuevo protocolo mediante la correspondiente modificación *software* en los mismos.

los *bridges*, etc. Para realizar un estudio exhaustivo es necesario realizar experimentos en las más variadas condiciones de trabajo del sistema. Normalmente, las redes Ethernet/802.3 "reales" trabajan en condiciones de carga relativamente bajas [SMI91] [BOG88], siendo difícil tener la oportunidad de observar el sistema bajo niveles altos de tráfico. Por otro lado, parece obvio que provocar artificialmente situaciones de trabajo extremas va a provocar molestias a los usuarios del sistema, sin que además sea trivial asegurar que la situación "artificial" forzada en la que se han tomado las medidas corresponda a la que se daría en el sistema real en tales condiciones de trabajo.

5.2 Evaluación mediante simulación

Ante esta situación, para evaluar el impacto sobre las prestaciones de la red que pudieran provocar los mecanismos de seguridad, la alternativa más atractiva es la simulación. Disponiendo de un "buen" simulador es mucho más económico, sencillo y menos engorroso el experimentar con las más diversas condiciones de trabajo. Por otro lado, en situaciones de carga elevada "artificial" los resultados que se obtengan mediante simulación no tienen porqué ser notablemente diferentes a los que se obtendrían en un sistema real en condiciones forzadas similares. Debido a todos estos argumentos, se decidió realizar el análisis de prestaciones mediante simulación. Con este objetivo se construyó el simulador de redes Ethernet extendidas descrito en el Anexo A.

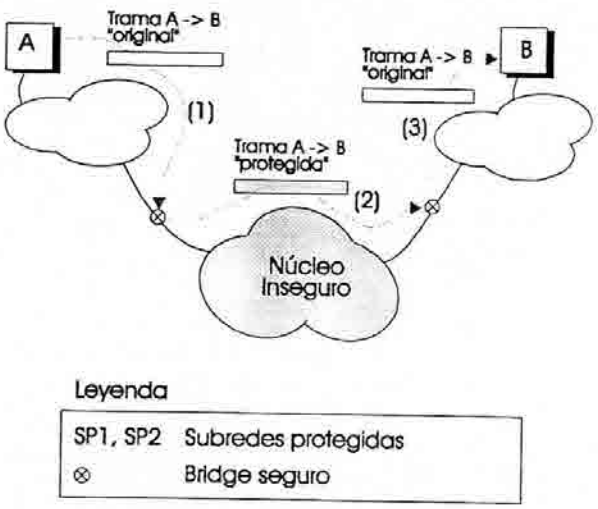


Figura 5.1: Trayecto sobre la red extendida de una trama protegida

En la Figura 5.1 se muestra el trayecto que sigue una trama protegida desde la estación emisora hasta la receptora. Este trayecto se puede descomponer en tres partes:

1) transmisión sobre la subred protegida de la estación origen
2) procesamiento en el *bridge* y retransmisión sobre núcleo inseguro
3) procesamiento en el *bridge* seguro final y retransmisión subred protegida de la estación destinataria

Dado que las tramas únicamente reciben protección en su trayecto sobre el núcleo inseguro, los efectos que puedan provocar los mecanismos de seguridad se pondrán de manifiesto fundamentalmente en esta parte de la red[3]. Este hecho se aprovechó a fin de simplificar el

[3] El trayecto sobre la subred protegida origen es completamente independiente de la existencia de los *bridges* seguros. En cuanto a la subred protegida destino, la única variación con respecto a un sistema con *bridges* transparentes convencionales será que la trama se intentará transmitir un poco más tarde, lo cual no tiene porqué variar apreciablemente el comportamiento de la red.

simulador construido. Según se describe en el anexo A, en las simulaciones realizadas se tomó como escenario de referencia una red extendida formada por cinco segmentos, cada uno con un número de estaciones entre 20 y 40, e interconectados entre sí mediante un sexto segmento que realiza las funciones de *backbone* "puro" (i.e. al que no se conecta directamente ninguna estación, únicamente los cinco *bridges* correspondientes a las cinco subredes respectivas). La simplificación de este simulador consiste en que únicamente se modela la transmisión de tramas dejando a un lado la recepción. Es decir, para cada trama generada por cada una de las estaciones se simulan los siguientes pasos:

- en la estación origen, algoritmo de acceso al medio y transmisión sobre el segmento correspondiente;
- examen de la trama por el *bridge* correspondiente a fin de determinar si debe retransmitirla al *backbone*;
- también en este bridge, en caso de que la trama se deba retransmitir:
  - ◊ si se trata de un *bridge* seguro se simulan los procedimientos de cifrado y encapsulamiento en una trama protegida (y fragmentación si procede);
  - ◊ algoritmo de acceso al medio y retransmisión sobre el *backbone*.

Como se puede apreciar, el trayecto que va desde el *backbone* hasta la estación destinataria no se contempla en este simulador, quedando modelado el tráfico externo de "entrada" a las subredes por el mismo tráfico generado por las estaciones locales. En definitiva, en las medidas de prestaciones efectuadas se realiza la "aproximación" de que la incorporación de *bridges* seguros en la conexión de las subredes al núcleo inseguro no modifica en absoluto el comportamiento de estas subredes. Es necesario señalar que el simulador no permite medir el tiempo requerido para desencapsular las tramas protegidas en el *bridge* seguro destinatario. No obstante, esto no supone ninguna limitación puesto que este tiempo va a venir dado exclusivamente en función de la longitud de las tramas pudiéndose calcular de manera determinista.

5.3 Modelos de *bridge* "convencional" y de *bridge* seguro

5.3.1 Modelo de *bridge* transparente "convencional"

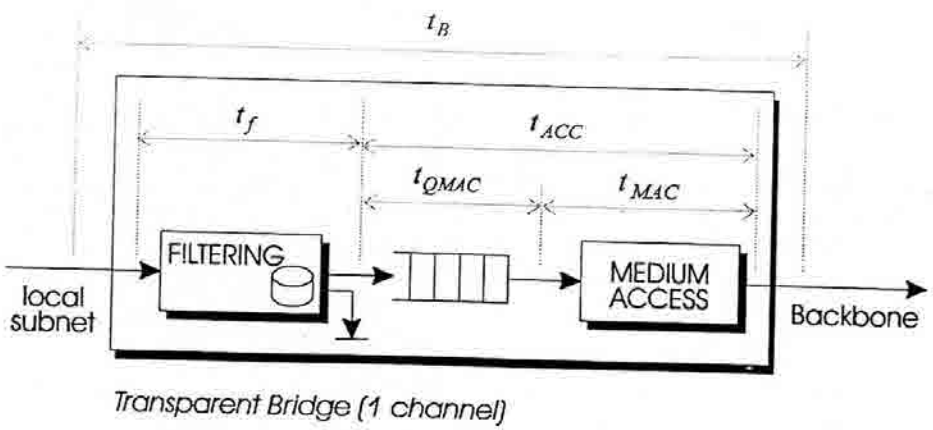


Figura 5.2: Modelo de simulación de los *bridges* transparentes

En la Figura 5.2 se muestra el modelo utilizado para simular *bridges* transparentes convencionales. El funcionamiento del mismo es como se describe a continuación:

- Cuando llega una trama a la subred local (i.e. la estación origen termina de transmitirla sobre el medio), el *bridge* "busca" la dirección de la estación destinataria en la tabla de ubicaciones. Si el destino está ubicado en la subred local[4], entonces esta trama es simplemente ignorada. En caso contrario, deberá ser retransmitida hacia el *backbone* de la red. El tiempo requerido para efectuar esta operación de filtrado se denota como t_f.

  Este tiempo dependerá de la técnica empleada para la realización de la tabla de direcciones y del procedimiento empleado para su exploración. En cualquier caso, el estado del arte actual permite conseguir tiempos de filtrado inferiores al mínimo tiempo que pueda haber entre dos recepciones consecutivas. De manera que cualquier dispositivo de este tipo que sale hoy en día al mercado tiene una capacidad de filtrado superior a las 14880 tramas por segundo que en el mejor de los casos se pueden cursar sobre un segmento Ethernet/802.3. A efectos de nuestro simulador se supuso que el tiempo requerido para filtrar una trama cualquiera es de 60 μseg.
- Una vez una trama a superado el procedimiento de filtrado, debe ser retransmitida sobre el *backbone*. Dependiendo de las condiciones de carga del *backbone*, esta trama se va a encontrar otras que llegaron y que todavía no han podido ser transmitidas. A fin de poder "absorber" transitorios de carga elevada es necesario disponer de una cola donde se almacenaran temporalmente las tramas que se vayan acumulando en espera de que les llegue su turno para ser enviadas al *backbone*. El tiempo de permanencia de cada trama en esta cola es denotado como t_{QMAC}. Aunque en la práctica la capacidad de estas colas debe ser necesariamente limitada, a efectos de las simulaciones realizadas se han supuesto colas de longitud infinita. En un caso real, la limitación en la capacidad de esta cola introducirá pérdidas de desbordamiento cuya importancia dependerá de esta capacidad y del tráfico presente en la red.
- Cuando a una trama le llega su turno para ser transmitida sobre el *backbone*, previamente deberá esperar según el algoritmo de contienda Ethernet hasta que pueda ser definitivamente transmitida. El tiempo que debe esperar la trama hasta haber "ganado" el acceso al medio más el requerido para transmitir todos sus bits se denota como t_{MAC}.

En la Figura 5.2 se define t_{ACC} al tiempo de permanencia de una trama en la cola de salida más el tiempo de "contienda" y el necesario para transmitirla. Por otro lado, se define el tiempo de respuesta del *bridge* t_B, como el tiempo desde la recepción de una trama hasta que se retransmite su último bit.

5.3.2 Modelo de *bridge* seguro *SBP*

En la Figura 5.3 se muestra el diagrama de bloques correspondiente a un canal del modelo de *bridge* seguro utilizado en las simulaciones. El diagrama correspondiente al otro canal sería análogo pero sustituyendo el bloque de encapsulación *SBP* por uno de desencapsulación. Como novedad frente al modelo de *bridge* transparente, se tiene que el bloque encargado de encapsular las tramas originales en nuevas tramas seguras según el protocolo *SBP*. Como este bloque debe procesar todos los bits de la trama original, el tiempo requerido para encapsular una trama T en otra segura T', va a venir dado por un término independiente de la longitud de T (i.e. carga de claves en el dispositivo de cifrado, construcción de los campos de la trama *SBP*, etc.) más otro que será aproximadamente[5] proporcional a la longitud de T.

---

[4] Esto también sería así en el caso de existir algún filtro de control de acceso que prohiba la "salida" al exterior de esta trama (ver apartado de gestión de seguridad del anterior capítulo).

[5] Dado que se añade relleno para obtener una longitud del campo cifrado que sea múltiplo de 8 octetos, realmente el tiempo de procesado será una función "escalera" con anchura del "peldaño" de ocho octetos.

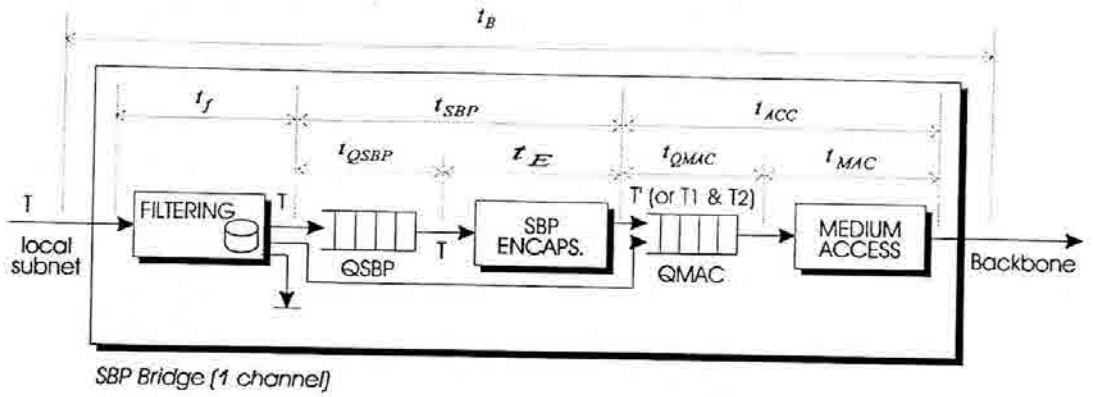


Figura 5.3: Modelo de simulación de los *bridges* seguros *SBP*

5.3.2.1 Consideraciones sobre la capacidad de la cola QSBP

Se muestra también en la Figura 5.3 la presencia de una cola de tramas a la entrada del encapsulador *SBP*. La función de esta cola es almacenar transitoriamente secuencias de tramas cortas recibidas durante el procesamiento de una trama larga. A fin de evitar posibles desbordamientos de esta cola, es necesario que el bloque de encapsulamiento *SBP* disponga de una velocidad de procesamiento por lo menos igual a la velocidad máxima a la que se pueden recibir tramas desde la subred local. El tiempo de encapsulamiento, t_E, viene dado en función de la longitud de trama, l. Si para tramas de una determinada longitud l, t_E es superior al necesario para transmitir una de estas tramas, $t_T(l)$, entonces se puede llegar a producir desbordamiento en esta cola. La razón es simple: si se recibe una ráfaga de tramas de esta longitud a velocidad máxima, el procesador no podrá seguir la cadencia de las llegadas, con el consiguiente crecimiento de la cola. Si esta cola es de capacidad limitada, como ha de ser en la práctica, entonces bastará una ráfaga suficientemente larga para que se pierda alguna trama.

Suponiendo que para cualquier longitud de trama $l \in [64,1518]$ se satisface

$$t_E(l) \le t_T(l) = \frac{20+l}{v_T}$$

donde:

- el término 20 en el numerador corresponde a los 8 octetos del preámbulo de sincronización que precede a cada trama y a los 12 octetos equivalentes de tiempo de guarda entre tramas;
- y v_T es la velocidad de transmisión (10 Mbits/seg.).

Entonces, la cola QSBP deberá tener una capacidad

$$Q_{MAX} = \frac{\max\{t_{SBP}(l)\}}{t_T(l_{min})} = \frac{t_{SBP}(1518)}{67.2\mu\text{seg}}$$

Por ejemplo, suponiendo el peor caso en el que todavía se puede garantizar el no desbordamiento de QSBP, se tendrá $t_{SBP}(1518)=t_T(1518)=1.23$ mseg. De donde, $Q_{MAX}=19$.

Obviamente, si $t_{SBP}(l)>t_T(l)$ para algún l_0, entonces el bloque *SBP* se puede congestionar temporalmente cuando reciba una secuencia de tramas de longitud l_0. En este caso, el dimensionamiento de la cola QSBP no puede tratarse de una manera determinista como se ha visto, sino que es necesario un estudio estadístico.

En cualquier caso, nosotros supondremos que nunca se produce desbordamiento de esta cola gracias a que $t_E(l) \leq t_T(l)$, lo cual es tecnológicamente sencillo de conseguir[6]. La razón es simple: con un procesador *SBP* "rápido" (i.e. que satisface lo anterior) se puede garantizar, en este bloque, una probabilidad de pérdidas nula simplemente dimensionando la cola a su entrada con una capacidad para tan sólo 19 tramas (y esto en el peor de los casos).

5.4 Impacto sobre el *throughput* máximo de los bridges

El *throughput* máximo que puede retransmitir un *bridge* transparente convencional viene dado por la capacidad del canal Ethernet. Al incorporar los protocolos de seguridad en un *bridge* de este tipo, las tramas sufren una expansión de longitud (y en algún caso también fragmentación). Esto va a significar que si desde la subred local se le ofrece una secuencia de tramas, el *bridge* no la podrá retransmitir a la misma velocidad si esta es excesivamente alta. Se evalúa aquí la reducción del *throughput* máximo que se produce al utilizar las cuatro versiones del protocolo *SBP* propuesto. El estudio se realiza para diferentes condiciones de carga del canal de salida (*backbone*): carga nula, baja, media y alta. Adicionalmente, se realiza una evaluación comparativa con los protocolos 802.10 SDE [IEE91] y EESS [HER88]. En estos últimos casos, la reducción de *throughput* estudiada corresponde a la reducción "aparente" de la capacidad del canal que observa una estación que utilice estos protocolos, con respecto a la capacidad nominal.

5.4.1 Expansión de longitudes en el protocolo *SBP*

5.4.1.1 Variantes V1 y V3

Las versiones V1 y V3 del protocolo *SBP* son idénticas exceptuando el mecanismo de fragmentación de tramas largas en el caso de V1. De la figura 4.2, es inmediato que la longitud m del campo *SBP_Data* (datos cifrados) es

$$m = n + PADL + 4 = 8 \cdot \left\lceil \frac{n+4}{8} \right\rceil,$$

de donde la longitud l' de la nueva trama T' viene dada por

$$l' = 20 + m = 20 + 8 \cdot \left\lceil \frac{n+4}{8} \right\rceil = 20 + 8 \cdot \left\lceil \frac{(l-18)+4}{8} \right\rceil = 20 + 8 \cdot \left\lceil \frac{l-14}{8} \right\rceil \qquad (5.1)$$

La anterior ecuación se satisface en el caso de V3 para cualquier longitud de la trama original T ($64 \leq l \leq 1518$). En el caso de V1, la ecuación (5.1) nos daría para tramas con $l > 1510$ una longitud de la nueva trama $l' = 1524$, claramente superior a la longitud máxima especificada por los estándares Ethernet/802.3 (1518 bytes). Por tanto se hace necesario el mecanismo de fragmentación: T será transportada sobre el núcleo inseguro sobre dos tramas T_1 y T_2 de longitudes l_1 y l_2, respectivamente.

---

[6] Actualmente es fácil conseguir en el mercado circuitos integrados para cifrado DES que ofrecen velocidades bastante por encima de los 10 Mbits/seg. Además, se dispone aproximadamente de 67.2 μseg. para construir los diferentes campos del protocolo *SBP*, inicializar el dispositivo de cifrado y cifrar una trama de longitud mínima, lo cual es holgadamente alcanzable con la tecnología disponible hoy en día.

Al tener que generar una trama adicional al fragmentar, se tiene una expansión equivalente con respecto a (5.1) de 14+2+4=20 bytes (correspondientes a los campos *Header*, *SBP_Header* y *CRC* de la trama extra). De manera que la suma de longitudes de las dos tramas T_1 y T_2:

$$l_1 + l_2 = l' + 20 = 40 + 8 \cdot \left\lceil \frac{l-14}{8} \right\rceil; \text{ con } 1510 \leq l \leq 1518$$

En definitiva, para este rango de valores de l, se tiene fijada la suma de longitudes de ambos fragmentos ya que:

$$l_1 + l_2 = 1544 \tag{5.2}$$

Queda por tanto decidir si la fragmentación se hace de manera balanceada ($l_1=l_2=772$), o no balanceada (p.ej. $l_1=64$ y $l_2=1480$). A primera vista, la alternativa más atractiva es la segunda puesto que:

- Cuando el *bridge SBP* destinatario recibe el primer fragmento T_1, deberá almacenarlo en espera de recibir el segundo (el cual pudiera no llegar, pues la fiabilidad de la red es alta pero no absoluta). Escogiendo l_1 mínimo se minimiza la cantidad de memoria necesaria para almacenar temporalmente los primeros fragmentos, disminuyendo por tanto el coste del dispositivo.
- Otra razón menos obvia que la anterior pero que puede en algún caso puede ser más importante es la siguiente. Una vez cifrado el campo *SBP_Data*, puede ser necesario[7] mover en memoria parte del resultado a fin de añadir a cada fragmento los campos adicionales para formar T_1 y T_2. Si este es el caso, interesa que la porción a mover sea lo más pequeña posible a fin de minimizar el tiempo requerido.

No obstante, hay que ser cautelosos antes de decidir si la fragmentación ha de ser, o no, balanceada. Por ejemplo, es un resultado conocido que en condiciones de carga elevada las redes CSMA en general se comportan mejor con tráfico de tramas largas que no con cortas. Esto podría hacer pensar que la opción no balanceada pueda afectar a las prestaciones de la red significativamente más que la opción balanceada. La solución adoptada fue realizar las medidas de evaluación para ambas alternativas, y de esta forma poder realizar un diseño óptimo.

5.4.1.2 Variantes V2 y V4

Análogamente a V1 y V3, las versiones V2 y V4 sólo difieren entre sí en la utilización de un mecanismo de fragmentación en el caso de V2. De la figura 4.3 se tiene que la longitud del campo *SBP_Data* es ahora

$$m = 8 \cdot \left\lceil \frac{n+18}{8} \right\rceil = 8 \cdot \left\lceil \frac{l}{8} \right\rceil$$

De donde la trama T' encapsulada según estos protocolos tiene ahora una longitud

$$l' = 20 + 8 \cdot \left\lceil \frac{l}{8} \right\rceil \tag{5.3}$$

Esto es válido para el caso de V4 independientemente del valor de l ($64 \leq l \leq 1518$). En el caso de V2 sólo para $64 \leq l \leq 1496$ ($46 \leq n \leq 1478$), teniéndose que fragmentar la trama para longitudes mayores. Las longitudes l_1 y l_2 de los dos fragmentos vienen dados en función de la longitud l de la trama original según se indica en la Tabla 5.1.

---

[7] Obviamente, la alternativa es dividir los datos durante el cifrado de manera que una vez acabada esta operación ya queden en áreas de memoria separadas.

	$l_1 + l_2$	Opción "balanceada"	Opción "no balanceada"
1496<l≤1504	1544	$l_1 = l_2 = 772$	l_1 =64; l_2 = 1480
1504<l≤1512	1552	$l_1 = l_2 = 776$	l_1 =64; l_2 = 1488
1512<l≤1518	1560	$l_1 = l_2 = 780$	l_1 =64; l_2 = 1496

Tabla 5.1: Longitudes de fragmentación en el protocolo *SBP V2*

5.4.2 *Throughput* máximo con subred destino totalmente descargada

En la Figura 5.5 se muestra el hipotético experimento mediante el cual se mediría el *throughput* máximo que puede cursar un *bridge* cuando dispone para el sólo de toda la capacidad de la subred LAN2. En esta figura se ilustra la secuencia de tramas a la salida del *bridge* (LAN2) cuando a la entrada se inyecta una secuencia de tramas a la velocidad máxima permitida por la red (y todas ellas de una misma longitud l). Si cada trama a la entrada debe ser retransmitida y transformada en una nueva trama de longitud $l' \geq l$, es obvio que la cadencia de salida va a ser menor que la de entrada. Se debe observar que en este experimento se supone que las colas internas de este *bridge* tienen capacidad infinita.

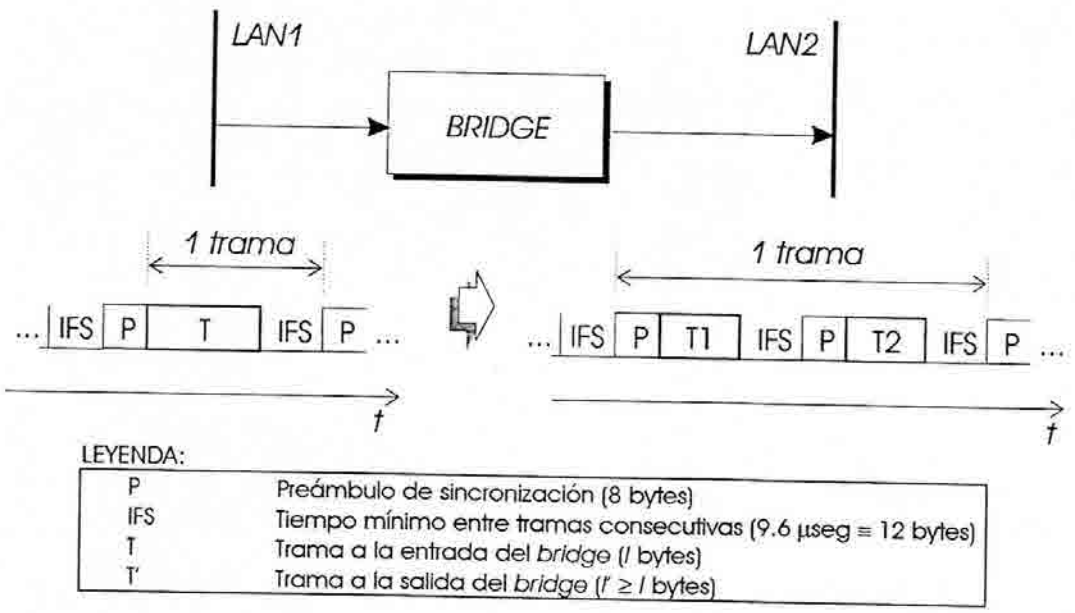


Figura 5.4

Si definimos $Th(l)$ como el número máximo de tramas que se puede transmitir por segundo sobre la subred de salida, entonces podemos escribir $Th(l)$ en función de l':

$$Th(l) = \frac{v_T}{8 \cdot (|P| + l' + |IFS|)} = \frac{10^7}{8(20 + l')} \tag{5.4}$$

donde l' vendrá dada a su vez en función de l dependiendo del tipo de transformación que se aplique a las tramas.

5.4.2.1 *Bridge* transparente convencional

En el caso de *bridges* transparentes "convencionales" (y en general para cualquier estación conectada a un segmento Ethernet totalmente desocupado) , se tiene:

$$Th(l) = \frac{10^7}{8(20+l)}$$

En la Figura 5.5 se muestra la dependencia del *throughput* con respecto a l para este caso.

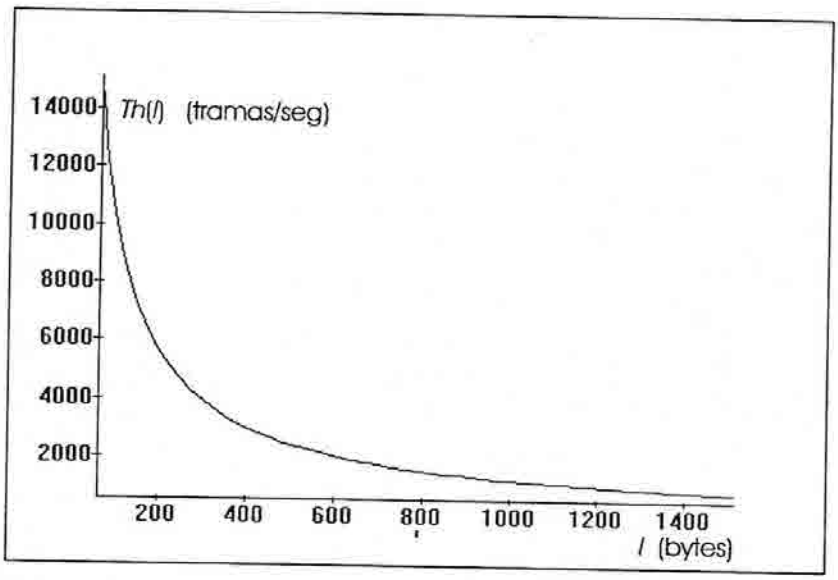


Figura 5.5: *Throughput* máximo para el protocolo Ethernet/802.3 original

5.4.2.2 Bridge SBP

5.4.2.2.1 *Protocolos V1 (sin fragmentación) y V3*

De (5.1), se tiene

$$l' = 20 + 8 \cdot \left\lceil \frac{l-14}{8} \right\rceil$$

de donde

$$Th(l)_{SBPv1\&v3} = \frac{10^7}{8\left(40 + 8 \cdot \left\lceil \frac{l-14}{8} \right\rceil\right)}$$

5.4.2.2.2 *Protocolo V1 con fragmentación (l>1510)*

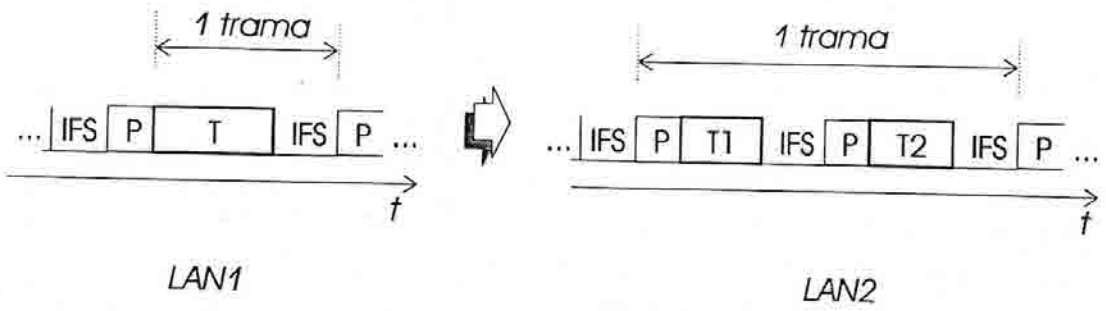


Figura 5.6

De la Figura 5.6 es inmediato modificar (5.4) para tramas fragmentadas:

$$Th(l)=\frac{v_T}{8\cdot\left(2\cdot|P|+2\cdot|IFS|+l_1+l_2\right)}=\frac{10^7}{8(40+l_1+l_2)} \tag{5.5}$$

Se puede observar que el *throughput* es independiente de la elección de l_1 y l_2, ya que la suma de ambas es invariante. En el caso de V1, se tiene $l_1 + l_2 = 1544$, de donde

$Th(l>1510)_{SBPv1f}$= 789 tramas/seg.

5.4.2.2.3 *Protocolos V2 (sin fragmentación) y V4*

De (5.3), se tiene

$$l'=20+8\cdot\left\lceil\frac{l}{8}\right\rceil$$

de donde

$$Th(l)_{SBPv2\&v4}=\frac{10^7}{8\left(40+8\cdot\left\lceil\frac{l}{8}\right\rceil\right)}$$

5.4.2.2.4 *Protocolo V2 con fragmentación (l>1496)*

En el caso de V2, bytes se tiene fragmentación para longitudes de la trama T superiores a 1496. A partir de la Tabla 5.1 y la ecuación (5.4), es inmediato obtener el *throughput* máximo en estas condiciones, obteniendo los resultados indicados en la Tabla 5.2.

	$l_1 + l_2$	$Th(l)$ (tramas/seg.)
1496<l≤1504	1544	807
1504<l≤1512	1552	785
1512<l≤1518	1560	781

Tabla 5.2

5.4.2.3 Otros protocolos de seguridad para redes Ethernet/802.3

Se estudia a continuación el *throughput* máximo que puede cursar una estación que utilice protocolos de seguridad como el estándar 802.10 SDE [IEE91] o el protocolo "propietario" EESS [HER88]. Aunque estos protocolos no están en principio concebidos para ser integrados en *bridges*, en cualquier caso su utilización supone una expansión de las tramas transmitidas y en consecuencia una disminución aparente del *throughput* que en el mejor de los casos podría cursar esta estación. Hay que señalar que en ambos casos no se ha considerado fragmentación. No obstante, los efectos de la fragmentación sobre el *throughput* son relativamente inapreciables. Esto se debe, como se ve más adelante, a que la fragmentación supone tan sólo una pequeña sobrecarga para las tramas de gran longitud a las que afecta.

5.4.2.3.1 Protocolo EESS

En la figura 2.10 se muestra el formato de las tramas EESS, de donde es inmediato que

$$l' = 40 + 8 \cdot \left\lceil \frac{l+4}{8} \right\rceil$$

siendo l' la longitud de la trama protegida y l la longitud de la trama equivalente que se transmitiría sin protección EESS. De manera que

$$Th(l)_{EESS} = \frac{10^7}{8\left(60 + 8 \cdot \left\lceil \frac{l+4}{8} \right\rceil\right)}$$

5.4.2.3.2 Protocolo 802.10 SDE

En la figura 2.5 se muestra el formato de las tramas 802.10 SDE. A efectos de poder comparar todos los protocolos en condiciones similares, suponemos que se utiliza un algoritmo en bloque de 64 bits, que el vector de comprobación de integridad es de 4 octetos, y que todos los campos del protocolo están presentes excepto el campo *User_Defined*. En estas condiciones se tiene

$$l' = |HEADER| + |SDE\_PDU| + |CRC| = 14 + |SDE\_PDU| + 4 = 18 + |SDE\_PDU|$$

con

$$|SDE\_PDU| = |CLEAR\_HEADER| + 8 \cdot \left\lceil \frac{|PROT\_HEADER| + n + |PAD\_Length| + |ICV|}{8} \right\rceil = \ldots$$

$$= 7 + 8 \cdot \left\lceil \frac{l-5}{8} \right\rceil$$

Finalmente

$$l' = 27 + 8 \cdot \left\lceil \frac{l-5}{8} \right\rceil$$

siendo l' la longitud de la trama protegida y l la longitud de la trama equivalente que se transmitiría sin protección 802.10 SDE. De manera que

$$Th(l)_{SDE} = \frac{10^7}{8\left(47+8\cdot\left\lceil\frac{l-5}{8}\right\rceil\right)}$$

5.4.3 Factor de sobrecarga. Comparación de los protocolos *SBP* V1/2/3/4, 802.10 *SDE* y EESS

En la Tabla 5.3 se muestra el *trhoughput* máximo soportable por los protocolos *SBP* V1/2/3/4, 802.10 *SDE* y EESS para algunas longitudes de trama (y suponiendo disponibilidad total de la capacidad del canal de salida).

	l = 128 bytes	l = 128 bytes	l = 128 bytes	l = 128 bytes	l = 1518 bytes
ORIGINAL	14881	8446	4529	2350	813
SBP V1 y V3	13021	7812	4340	2298	809
SBP V1	"	"	"	"	789
SBP V2 y V4	12019	7440	4223	2264	801
SBP V2	"	"	"	"	781
802.10 *SDE*	11261	7143	4125	2236	798
EESS	9470	6377	3858	2155	787

Tabla 5.3: Número máximo de tramas por segundo en cada protocolo

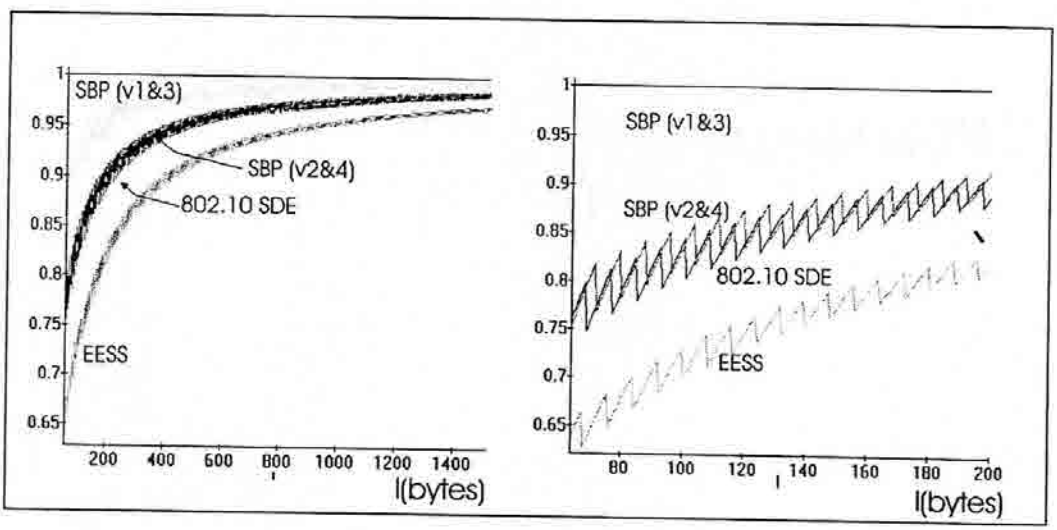


Figura 5.7: Coeficientes de sobrecarga para los diferentes protocolos estudiados

Definimos ahora "factor de sobrecarga" del protocolo X, a la relación entre el *throughput* máximo para este protocolo y el *throughput* máximo con el protocolo Ethernet/802.3 original, y lo denotamos mediante $c_X(l)$. Es decir:

$$c_X(l) = \frac{Th_X(l)}{Th(l)} = \frac{20+l}{20+l'} \tag{5.6}$$

La interpretación "física" de esta función es clara: representa la reducción aparente en la capacidad de la red que observa un *bridge* (o una estación) al utilizar el protocolo de seguridad

X. En la Figura 5.7 se muestran los coeficientes de sobrecarga para todos los protocolos estudiados (sin considerarse fragmentación. En esta figura se muestra la misma gráfica en dos escalas diferentes. La primera gráfica recorre todo el intervalo [64,1518], mientras que la segunda presenta, a mayor escala, el subintervalo [64,200].

Es de remarcar el carácter "aserrado" de las gráficas del factor de sobrecarga. Obviamente, esto es efecto del relleno exigido por el algoritmo de cifrado en bloque empleado. Como consecuencia se tiene que para un mismo protocolo, la sobrecarga producida en tramas de longitudes muy similares puede diferir hasta un 5% aproximadamente. Las conclusiones más interesantes que se pueden extraer de la observación de estas gráficas son:

- Para longitudes de trama muy grandes, la sobrecarga introducida por todos los protocolos es muy similar (la reducción de *throughput* está por debajo del 5%). Esto es cierto incluso en presencia de fragmentación (ver Tabla 5.3).
- Las diferencias entre los diferentes protocolos se ponen de manifiesto con tramas cortas para las que los campos añadidos pueden significar una sobrecarga importante.
- El protocolo EESS presenta una sobrecarga comparativamente alta con respecto a los otros casos.
- El caso particular 802.10 *SDE* analizado ofrece los mismos servicios de seguridad que las versiones V1 y V3 del protocolo *SBP* (Confidencialidad e integridad sólo del campo de datos de la trama original). Por tanto, queda patente una mayor eficiencia en el diseño de nuestro protocolo frente al estándar.
- El caso particular 802.10 *SDE* analizado presenta una sobrecarga similar a la de las versiones V2 y V4 del protocolo *SBP*. No obstante, estos dos últimos garantizan "al mismo precio" confidencialidad e integridad también de la cabecera de la trama original.

5.4.4 *Throughput* máximo con subred destino parcialmente ocupada

En condiciones normales, cuando un *bridge* intenta retransmitir sobre el *backbone*, va a encontrárselo parcialmente ocupado por tramas transmitidas por otras estaciones o *bridges*. A fin de realizar un estudio más completo, parece interesante evaluar cómo afecta la introducción de protocolos de seguridad en el nivel de enlace al *throughput* máximo que puede cursar un *bridge* sobre el *backbone*, cuando este se encuentra parcialmente ocupado. Para realizar este estudio se llevaron a cabo una serie de experimentos simulados sobre la red de referencia descrita en el Anexo A. Los experimentos consistieron básicamente en inyectar a uno de los *bridges* una secuencia indefinida de tramas a velocidad máxima y longitud constante[8], midiéndose la velocidad de retransmisión sobre el *backbone* cuando en el se tenían previamente niveles de tráfico (multimodal) bajo, medio o altos. Estos niveles de tráfico de "fondo" se generaron a partir de las otras cuatro subredes del sistema de referencia. El experimento se repitió para cinco longitudes de trama diferentes y para todos los protocolos de seguridad considerados (i.e. protocolo original, *SBP* V1/2/3/4, 802.10 *SDE* y *EESS*). Los resultados obtenidos se exponen en las figuras 5.8, ..., 5.12. Cada una de estas gráficas corresponde a una longitud de trama para la que se ha evaluado el *throughput* máximo, según los protocolos considerados:

"t": Protocolo Ethernet original. Corresponde a las medidas para un *bridge* transparente convencional;

"SBPv1": En las gráficas correspondientes a 64, 128, 256 y 512 bytes representa las versiones V1 y V3 del protocolo *SBP*. En la gráfica correspondiente a 1518 bytes representa la versión V1 con fragmentación balanceada ($l_1 = l_2 = 772$ bytes).

"SBPv2": En las gráficas correspondientes a 64, 128, 256 y 512 bytes representa las versiones V2 y V4 del protocolo *SBP*. En la gráfica correspondiente a 1518

[8] Se utilizó el *bridge* correspondiente a la subred LAN1 de la red de referencia del anexo A.

bytes representa la versión V2 con fragmentación balanceada ($l_1 = l_2 = 780$ bytes).

"802.10 SDE": Protocolo de seguridad *SDE* según el estándar IEEE 802.10 según se indica en el apartado 5.4.2.3.2.

"EESS" Protocolo de seguridad EESS.

"SBPv1u": Protocolo *SBP* V1 con fragmentación no balanceada (l_1=64, l_2=1480).

"SBPv2u": Protocolo *SBP* V2 con fragmentación no balanceada (l_1=64, l_2=1496).

"SBPv3": Protocolo *SBP* V3.

"SBPv4": Protocolo *SBP* V4.

Para generar el tráfico de fondo se simuló previamente la red "desactivando" la subred del *bridge* utilizado para las medidas, ofreciendo diferentes niveles de carga al *backbone* desde las restantes cuatro subredes. De esta manera se obtuvo la curva de carga representada en la Figura 5.13. Después, se replicó la experiencia para los puntos de la anterior gráfica correspondientes a caudales cursados del 20, 50 y 90 % (nivel de tráfico de fondo, S_f), pero ahora inyectando la secuencia de tramas correspondiente a la entrada del *bridge* bajo medida.

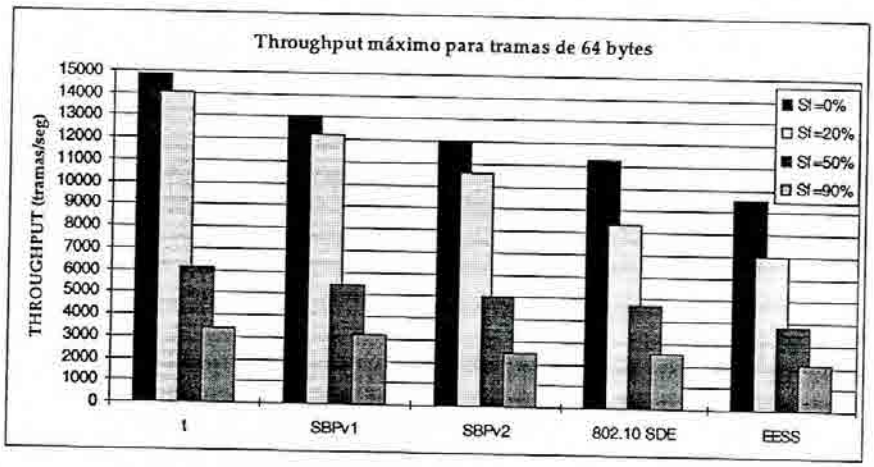


Figura 5.8

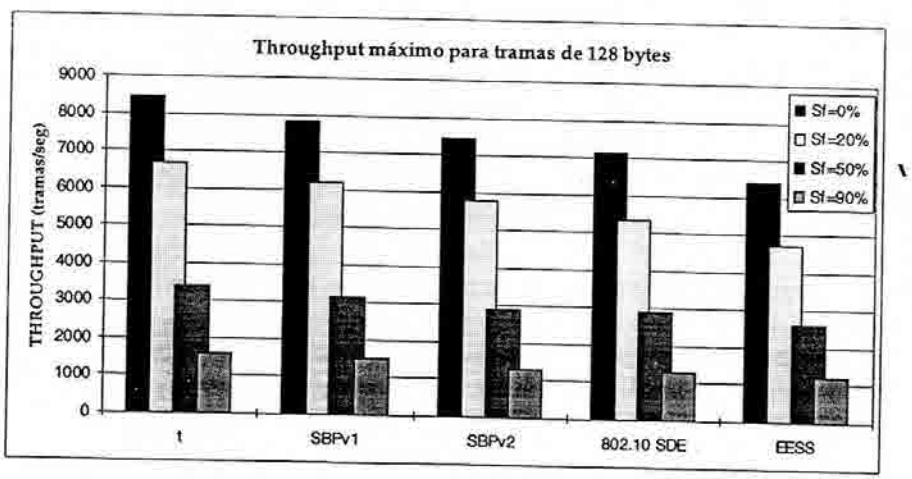


Figura 5.9

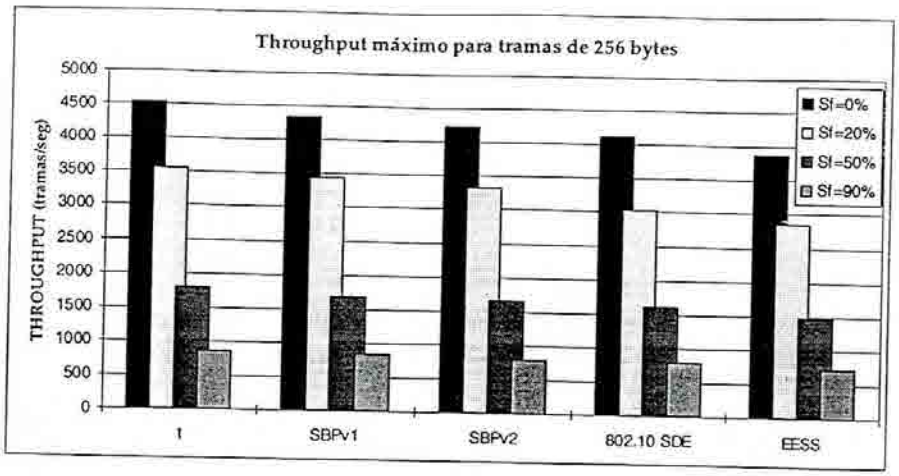


Figura 5.10

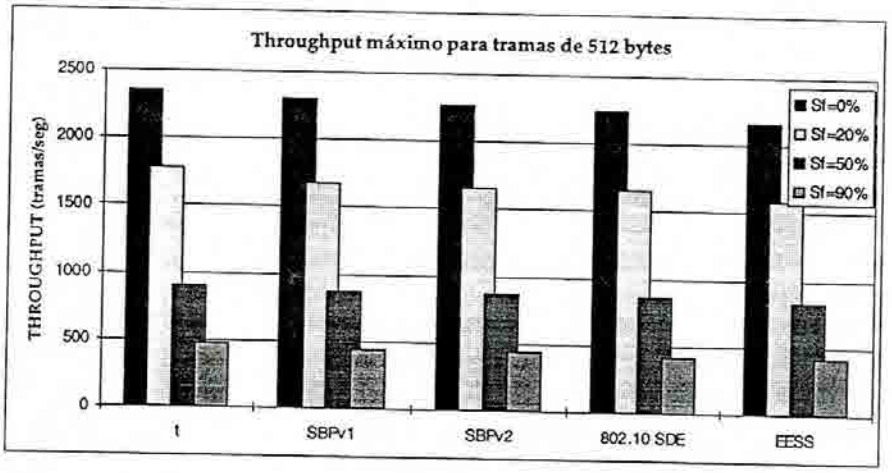


Figura 5.11

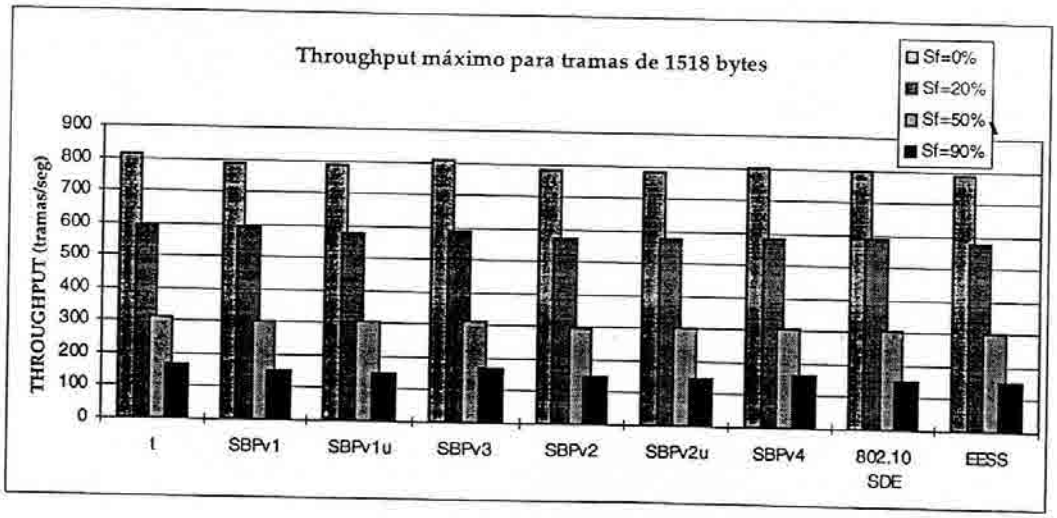


Figura 5.12

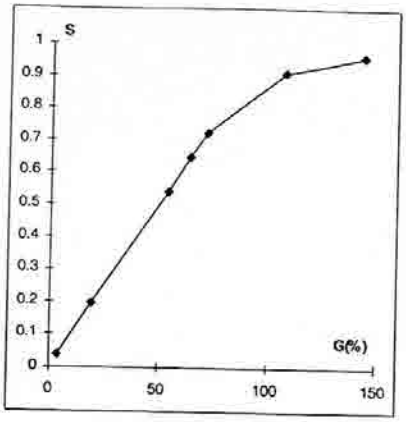


Figura 5.13: Curva de carga del *backbone* con tráfico procedente sólo de LAN2, ..., LAN5

Las conclusiones más importantes que se pueden extraer de estas medidas de *throughput* son las siguientes:

- La sobrecarga que suponen los protocolos de seguridad evaluados sólo es apreciable para tramas de longitud corta. En concreto, para las gráficas correspondientes a longitudes de trama de 512 y 1518 bytes no existe efecto "apreciable" al introducir cualquiera de los protocolos. Esto se podría calificar como noticia "muy buena". Como se comenta en el anexo A, los periodos de tiempo durante los que se observan periodos de alta actividad en redes de este tipo "reales", se deben precisamente a ráfagas de tramas de longitud máxima, siendo extraño que ráfagas de tramas de longitud corta provoquen tasas de ocupación elevadas. En definitiva, el *bridge SBP* no tendrá que retransmitir normalmente secuencias rápidas de tramas pequeñas. De forma que, en cuanto a lo que *throughput* se refiere, la ya relativamente pequeña sobrecarga que se introduce en estas tramas va a afectar inapreciablemente al sistema.
- La "insensibilidad" del *throughput* para ráfagas de longitud grande se traduce adicionalmente en que el mecanismo de fragmentación no afecta apreciablemente a las prestaciones de la red en cuanto a *throughput*. Por tanto, únicamente la expansión de las tramas puede afectar esta característica de la red.

5.5 Impacto sobre el tiempo de respuesta de los *bridges*

La introducción de los mecanismos de seguridad en los *bridges* puede potencialmente aumentar el tiempo que una trama protegida necesita para recorrer la red desde la estación emisora hasta la destinataria. Este aumento en el tiempo de respuesta de la red puede darse en las partes (1) y (2) del camino de la trama, según la Figura 5.1. Suponiendo que el efecto del mecanismo de seguridad sobre el tráfico local a las subredes es insignificante[9], entonces el retardo introducido se deberá básicamente a los dos siguientes factores:

- Tiempo de desencapsulación en los *bridges* seguros destinatarios. Suponiendo que el procesador de desencapsulación es suficientemente rápido como para no representar un cuello de botella en ningún caso (i.e. el tiempo de desencapsulación es menor o igual al tiempo necesario para su transmisión), entonces este tiempo se puede acotar determinísticamente. Efectivamente, en la cola QSBP se pueden acumular un total de tramas cuyo tiempo de transmisión (en este caso recepción) es idéntico al tiempo de desencapsulación de la trama más larga. A su vez, las tramas acumuladas en esta cola se

[9] Las tramas que circulan por estas subredes en presencia del sistema de seguridad son exactamente las mismas que se tendrían sin este sistema, únicamente se verán ligeramente afectados los instantes en que estas tramas se intentan transmitir.

servirán a un ritmo no inferior al de su transmisión sobre la red (condición necesaria para que no haya cuello de botella). Obviamente, el tiempo de transmisión de esta secuencia de tramas será, como máximo, igual al de desencapsulación de una trama de longitud máxima. En conclusión, bajo estas condiciones una vez se ha recibido una trama protegida, esta invertirá un tiempo en el bloque *SBP* que como mucho será el tiempo de transmisión de una trama de máxima longitud (i.e. 1.23 mseg).

- Tiempo de respuesta del *bridge* seguro emisor. Este tiempo a su vez se puede ver modificado por dos componentes: el tiempo de espera y encapsulamiento en el bloque *SBP* que por razones idénticas a las del anterior punto estará acotado en 1.23 mseg.; y el incremento es el del tiempo de acceso al núcleo inseguro, debido a las modificaciones que tiene la introducción del protocolo de seguridad sobre el tráfico ofrecido. Dado el carácter no determinista de la segunda componente, se intentó realizar una evaluación, mediante simulación, del tiempo de respuesta del *bridge* seguro "origen".

A fin de evaluar los efectos sobre el tiempo de respuesta del *bridge* origen (t_B en la Figura 5.3), se midió mediante simulación este tiempo para *bridges* transparentes convencionales y *bridges* seguros que utilizasen los protocolos *SBP* V1/2/3/4 (para V1 y V2 se experimentó con fragmentación balanceada y no balanceada), y los protocolos 802.10 *SDE* y *EESS*. Las medidas se realizaron para diversas condiciones de carga del *backbone*, y con diferentes porcentajes de tráfico seguro con respecto al total. El coeficiente de tráfico externo utilizado es del 50% para las cinco subredes LAN1, ..., LAN5 y se protegen todas las tramas retransmitidas por el *bridge* (excepto, claro está, en el caso del *bridge* convencional). Bajo estas condiciones se ofertaron cargas a las cinco subredes de 5, 20, 25, 28 y 30% (de tráfico multimodal según se describe en el anexo A), y se midió el tiempo de respuesta medio para los protocolos mencionados. De las curvas de trabajo de estas subredes (ver anexo A), se deduce que en estos cinco niveles de carga, el caudal cursado en cada una de ellas es aproximadamente igual a la carga ofrecida. De esta manera, se fija la carga ofrecida del *backbone* aproximadamente[10] en 12.5, 50, 62.5, 70 y 75 %.

	2 BRIDGES					3 BRIDGES					4 BRIDGES					5 BRIDGES				
	G=12	G=50	G=62	G=70	G=75	G=12	G=50	G=62	G=70	G=75	G=12	G=50	G=62	G=70	G=75	G=12	G=50	G=62	G=70	G=75
T	0.6	1.3	2.6	8.3	19.3	0.6	1.3	2.6	8.3	19.3	0.6	1.3	2.6	8.3	19.3	0.6	1.3	2.6	8.3	19.3
SBPv1	0.8	1.8	4.7	14.7	30.2	0.8	1.9	5.6	18.0	35.5	0.8	2.0	7.1	21.3	40.0	0.8	2.3	8.2	26.0	44.9
SBPv1u	0.9	1.7	3.6	8.9	20.6	0.9	1.7	3.4	10.0	24.1	0.9	1.7	3.7	10.7	22.8	0.9	1.8	3.7	10.8	24.0
SBPv3	0.9	1.7	3.5	8.8	22.5	0.9	1.7	3.4	9.4	22.7	0.9	1.7	3.6	10.4	24.6	0.9	1.7	3.8	10.9	25.2
SBPv2	0.8	1.8	4.6	15.4	31.5	0.8	2.0	5.9	19.4	39.3	0.8	2.0	7.3	23.8	44.4	0.8	2.3	9.6	28.6	47.8
SBPv2u	0.9	1.7	3.6	9.4	23.2	0.9	1.7	3.5	10.7	24.9	0.9	1.8	3.9	11.0	25.8	0.9	1.8	4.2	12.7	27.4
SBPv4	0.9	1.7	3.4	10.0	23.6	0.9	1.8	3.5	10.6	24.9	0.9	1.7	3.8	11.5	26.7	0.9	1.8	4.2	12.5	29.6
SDE	0.9	1.7	3.8	11.1	24.6	0.9	1.8	3.7	11.3	27.1	0.9	1.7	4.4	12.0	27.5	0.9	1.8	4.5	13.9	31.3
EESS	0.9	1.8	3.7	11.1	26.9	1.0	1.8	4.0	13.6	32.3	1.0	1.9	4.5	15.6	35.5	1.0	1.9	5.3	17.7	39.9

Tabla 5.4: Tiempo medio de respuesta de un *bridge* (mseg)

Los tiempos medios de respuesta medidos se muestran numéricamente en las Tabla 5.4 y gráficamente en las figuras 5.14, ..., 5.17. Como se puede observar, las simulaciones se llevaron a cabo "protegiendo" el 0, 40, 60, 80 y 100% del tráfico sobre el *backbone* (configurando 0, 2, 3, 4 y 5 *bridges*, respectivamente, como *bridges* seguros. Las conclusiones más importantes que se extraen de la observación de estas medidas son las siguientes:

- El tiempo de respuesta se ve afectado tanto por la expansión de las tramas como por su fragmentación. No obstante, como se comenta a continuación, el factor que más afecta de manera desfavorable al tiempo de respuesta es la fragmentación balanceada. Comportándose las versiones con fragmentación balanceada de los protocolos v1 y v2 notoriamente peor que el resto.
- En condiciones de carga ofrecida baja-media (por debajo del 60%, como es usual en este tipo de redes) no se observa en ningún caso un aumento significativo del tiempo medio de respuesta del *bridge* (p.ej. en el peor caso se observa un incremento de 1 mseg). En

[10] No se considera aquí el efecto de la expansión de las tramas. Es decir, estas cargas ofrecidas son las equivalentes a las que se tendrían sin protocolo de seguridad.

condiciones de carga elevada, exceptuando los protocolos con fragmentación balanceada, el tiempo de respuesta se ve aumentado en una cantidad "aceptable" (de 19.3 a 27.4 mseg en el peor caso). De todas maneras, se debe tener muy presente lo desfavorable de las condiciones en las que se han realizado las medidas: todas las tramas cursadas por el *bridge* bajo observación se protegen y se han considerado casos protegiendo hasta el 100% de las tramas que viajan por el *backbone*.

- Las versiones del protocolo *SBP* con fragmentación balanceada afectan comparativamente de manera mucho más notoria que el resto al tiempo de respuesta. La razón seguramente se debe a que en estos casos la transmisión del primer fragmento requiere un tiempo relativamente alto. Ello significa que la probabilidad de que otras estaciones difieran algún intento de transmisión hacia el final del primer fragmento es mucho mayor que en el caso de fragmentación no balanceada. De manera que con fragmentación balanceada la probabilidad de que el segundo fragmento deba competir por el medio con otras tramas será mucho mayor que en el caso de fragmentación no balanceada.

En definitiva, estas observaciones permiten calificar al protocolo *SBP*V2 seguramente como el protocolo de seguridad con mejor comportamiento de entre los evaluados. En cualquier caso, la conclusión más importante a extraer es que la utilización de fragmentación balanceada es desaconsejable en este tipo de redes.

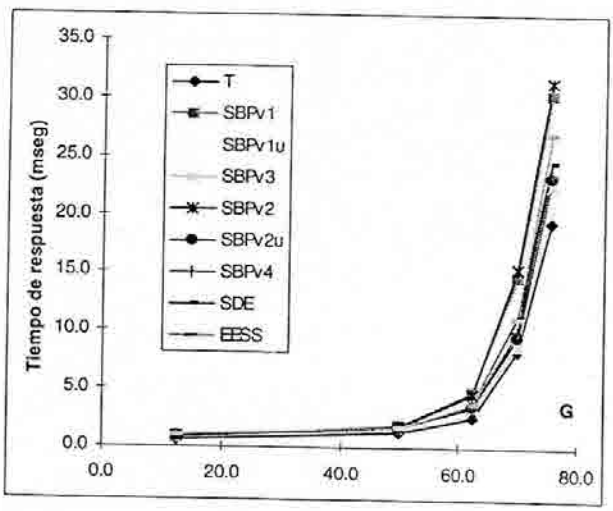


Figura 5.14: Tiempo medio de respuesta protegiendo el 40% del tráfico en el *backbone* (2 *bridges*)

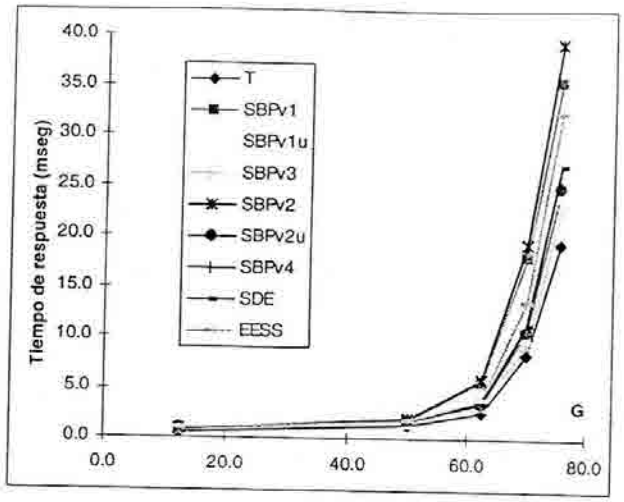


Figura 5.15: Tiempo medio de respuesta protegiendo el 60% del tráfico en el *backbone* (3 *bridges*)

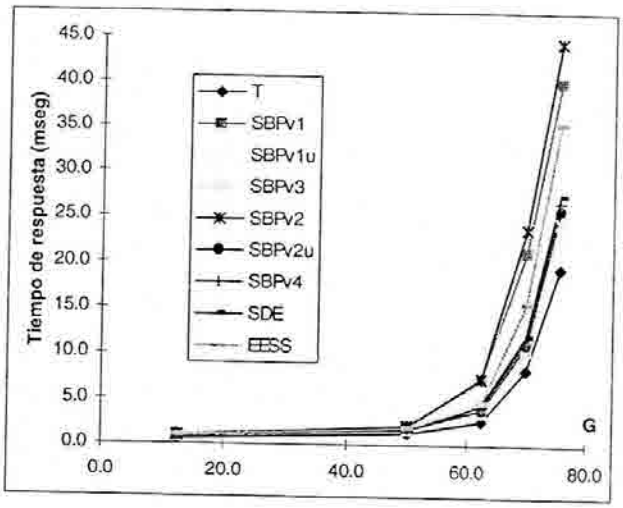


Figura 5.16: Tiempo medio de respuesta protegiendo el 80% del tráfico en el *backbone* (4 *bridges*)

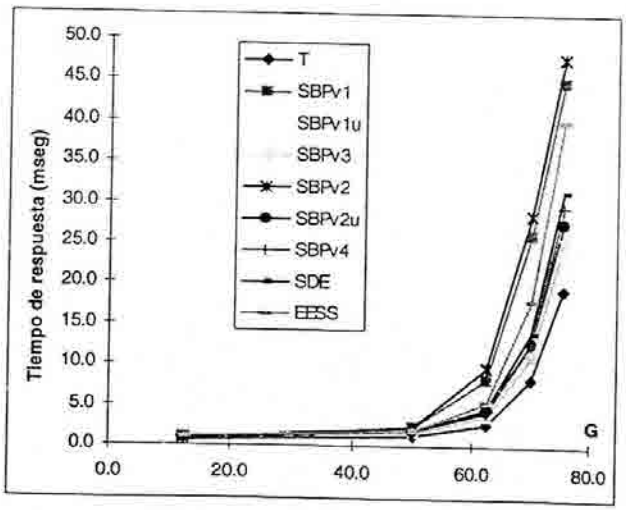


Figura 5.17: Tiempo medio de respuesta protegiendo el 100% del tráfico en el *backbone* (5 *bridges*)

6. Conclusiones y líneas futuras.

6.1 Conclusiones

Se proponen en este trabajo mecanismos de seguridad cuyo objetivo es dotar de los servicios de confidencialidad e integridad a los servicios de comunicación prestados por el subnivel de control de acceso al medio de redes locales extendidas 802.3/Ethernet. Las contribuciones principales aportadas en esta tesis son las siguientes:

- Propuesta de una arquitectura de seguridad de las comunicaciones en redes locales. Esta arquitectura se fundamenta en la integración de mecanismos de seguridad en los dispositivos de interconexión entre las subredes que componen el sistema (capítulo 2). Las funciones desarrolladas por estos dispositivos son básicamente las de un *bridge* transparente convencional con mecanismos añadidos de confidencialidad e integridad para proteger la información especialmente sensible.

- Diseño de un modo de encadenamiento para cifradores en bloque IOBC (*Input and Output Block Chainning*) que permite garantizar la confidencialidad y la integridad de los datos en una sola "pasada" (capítulo 3). El principal interés de este modo reside en la minimización de los recursos, o del tiempo de procesamiento, requeridos para realizar mecanismos de integridad y confidencialidad de datos. El interés de esta contribución es independiente del resto del trabajo, pues la aplicación del modo de encadenamiento propuesto se puede extender a otras áreas de la seguridad.

- Diseño de cuatro versiones de un protocolo para el intercambio seguro de la información entre *bridges* de subredes distintas. Este protocolo se diseña utilizando las "virtudes" del modo de encadenamiento de cifrado propuesto, consiguiéndose minimizar la sobrecarga introducida en la red (capítulo 4).

- Evaluación del impacto que provoca la introducción de cualquiera de los protocolos de seguridad sobre las prestaciones de la red, así como una comparación con otros protocolos afines (capítulo 5). Es de destacar el especial interés de este estudio de evaluación, ya que debido a la relativa "juventud" de la disciplina de seguridad en comunicaciones, se da actualmente una especial escasez de estudios de este tipo. En la evaluación realizada se estudia el impacto de los mecanismos de seguridad sobre la capacidad y sobre el tiempo de respuesta del sistema (las dos características más interesantes en este tipo de redes). Por un lado se concluye que el tiempo de respuesta crece debido a la expansión y la fragmentación de las tramas, presentando las versiones con fragmentación balanceada un retardo notoriamente superior a todos los demás

protocolos estudiados. No obstante, las versiones con fragmentación no balanceada presentan un tiempo de respuesta similar a la de los protocolos sin fragmentación y que es razonablemente aceptable (en el peor caso observado se introducían 5 mseg. de retardo medio adicional en el *bridge* emisor). Por otro lado, la capacidad del sistema se ve afectada principalmente por la expansión de las tramas. Afortunadamente, este efecto sólo es apreciable cuando la carga ofrecida a la red es elevada y compuesta principalmente de tramas cortas. En una red "real" aparecen tráficos elevados básicamente sólo con tramas de longitud elevada cuando se realizan transferencias de ficheros o aplicaciones similares que necesitan transportar grandes cantidades de información. Concluyéndose en definitiva que el impacto de los mecanismos de seguridad sobre las prestaciones de la red es relativamente pequeño incluso en situaciones extremas "de laboratorio", e inapreciable en situaciones reales. También es de señalar que el comportamiento de los protocolos propuestos es mejor que el de los otros dos protocolos con los que se ha comparado (excepto en el caso menciado de fragmentación balanceada).

6.2 Líneas futuras de trabajo

Como líneas de continuación del trabajo expuesto cabe citar las siguientes dos:

- Estudio de integración de mecanismos similares en otras redes.
- Estudio de la problemática de administración de seguridad.

Dentro de la primera línea cabría continuar con un estudio de integración de mecanismos de seguridad en redes locales extendidas donde se haga uso de protocolos de acceso diferentes a Ethernet/802.3 (i.e. redes locales "token-ring", "token-bus", etc). El siguiente paso sería la integración de mecanismos de este tipo en redes metropolitanas, las cuales presentan características similares a las estudiadas aquí (múltiples subredes locales interconectadas mediante un bus o anillo metropolitano que hace las veces de *backbone*). Tal como se apunta en [REC95], en redes metropolitanas se agudiza la problemática de seguridad debido al (usual) carácter "multidominio" que tienen esta redes.

En la segunda línea de continuación del trabajo cabe estudiar los diferentes aspectos de la administración de seguridad tal y como se apunta en el cuarto capítulo de esta memoria. Además, es de interés el estudiar cómo integrar estos mecanismos de gestión de seguridad en sistemas estándares de gestión de red. Finalmente, señalar que la problemática de administración de seguridad presenta un peculiar interés en entornos multidominio como es el caso de las redes metropolitanas. En este tipo de entornos se debe resolver satisfactoriamente la implantación coordinada de las diferentes políticas de seguridad.

Anexo A: Simulador de redes 802.3/Ethernet extendidas

A.1. Descripción del simulador de segmentos Ethernet

Existen tres vías para evaluar las prestaciones de un sistema: el estudio analítico, la simulación y la medida sobre el sistema real [JAI91]. Existen multitud de trabajos publicados sobre el estudio de prestaciones de redes Ethernet/802.3 (véase, por ejemplo, [MET76] [SCH80] [BOG88] [LEL91] [SMI91]). Es de señalar no obstante que aunque se pueden encontrar multitud de análisis de las prestaciones de redes locales CSMA/CD (por ejemplo, [HAM86] [TOB87] [BUX87] [MED87] [KLE76]), las redes Ethernet tienen ciertas características que las hacen especialmente difíciles de evaluar analíticamente [GON87] [BOG88]. Entre estas características cabe citar la utilización del algoritmo *"back-off"* (usado para diferir transmisiones que hayan sufrido colisiones), la distribución física de las estaciones, y el modelo de tráfico [LEL91] [SMI91a] [SMI91b]. En consecuencia, esta dificultad en el análisis formal provoca que las herramientas de evaluación más "asequibles" sean la simulación o la medida en "tiempo real" sobre redes ya en funcionamiento.

Si a los anteriores argumentos les añadimos la dificultad de disponer de toda una red local 802.3/Ethernet extendida con la que "jugar a nuestro antojo" sometiéndola a las condiciones de trabajo que requieran nuestros experimentos de evaluación, entonces se hace patente la conveniencia de disponer de un simulador de este tipo de redes.

El simulador construido permite la interconexión de varios segmentos Ethernet en número y topología arbitrarios. Cada uno de estos segmentos es a su vez simulado mediante un simulador Ethernet "convencional" (i.e. simulador por eventos discretos en tiempo continuo). Por otro lado, la interconexión de segmentos para modelar redes extendidas se realiza mediante técnicas de programación concurrente mediante paso de mensajes entre los simuladores de cada segmento. Esta filosofía nos ha permitido simplificar la complejidad de programación del modelo con respecto a técnicas de programación secuencial mediante lenguajes convencionales, y al mismo tiempo ha permitido obtener un simulador más eficiente que el que se hubiera obtenido usando un entorno de simulación específico.

A.1.1.Simulador de un segmento

El simulador de segmentos se construyó intentando que permitiera modelar las especificaciones 802.3/Ethernet de la forma más fiel posible. La Tabla A-1 muestra los parámetros más relevantes del simulador construido. Las característica más destacable de este simulador es la modelación de los efectos de propagación sobre el cable, permitiendo de esta forma incorporar en los resultados los posibles efectos de la distribución física de las estaciones sobre el segmento. No menos importante es la posibilidad de simular perfiles de

tráfico arbitrarios. Esto se consigue configurando la longitud de las tramas que emite cada estación y la carga ofrecida de esta a los valores que se deseen.

• Longitud del segmento	configurable
• Velocidad de propagación eléctrica	0.65c (195000 m/seg.)
• Velocidad de transmisión	10 Mbit/seg.
• Preámbulo (duración t_{PRE})	64 bits (6.4 µseg.)
• Guarda entre tramas (t_{IFS})	96 bits time (9.6 µseg.)
• Duración señal *Jammig*	48 bits time (4.8 µseg.)
• N° colisiones antes de descarte	16 intentos
• N° estaciones	configurable
• Ubicación de cada estación	configurable
• Tamaño de las tramas	configurable para cada estación
• Carga ofrecida por estación	ver A.1.2

Tabla A-1: Parámetros de configuración del simulador de 1 segmento.

A.1.2.Modelo de tráfico para las estaciones terminales

Con respecto al modelo de generación de tráfico se opta por una distribución exponencial del tiempo entre tramas, con bloqueo del generador entre el instante de generación de la trama y el instante en el que el controlador de red finaliza su transmisión (o la descarta por número excesivo de colisiones). Se fuerza a que la estación i ofrezca una carga G_i (entre 0 y 1) configurando el tiempo medio λ_i entre las tramas que genera[1]. Definimos[2] como "carga ofrecida" por la estación i, G_i, a la porción del ancho de banda que ocuparía esta estación en caso de tener el medio totalmente a su disposición (es decir, que cuando se generase una trama ésta se pudiera transmitir inmediatamente). En esta definición tenemos en cuenta los tiempos de preámbulo y de espaciado entre tramas que preceden y siguen, respectivamente, a cada trama. Esta definición se ilustra en la Figura A-1. Observar que en este modelo el buffer equivalente que utiliza el usuario sólo tiene capacidad para almacenar una trama, permaneciendo el usuario bloqueado mientras este buffer no se vacía[3].

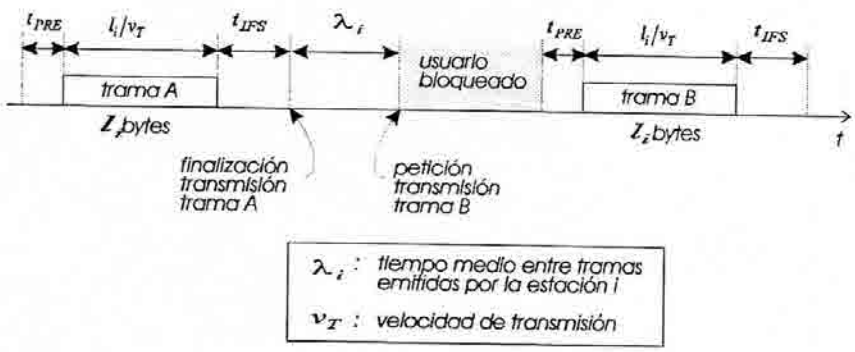


Figura A-1: Modelo de generación de tráfico.

[1] Este λ_i no se corresponde, como es usual, con el inverso de la tasa de generación de tramas debido a que en este segundo se recoge el efecto del periodo de bloqueo entre generaciones consecutivas.

[2] En este contexto la definición de carga ofrecida no es única y difiere de autor en autor [BOG88]. La definición que se utiliza aquí es la que nos ha parecido que tiene una interpretación "intuitiva" más clara.

[3] El modelo de *bridge* utilizado difiere de este modelo de estación "terminal" , entre otras cosas, en que la capacidad de este buffer será infinita. De esta forma, en el caso del *bridge* el usuario generador correspondiente (todo un segmento de red) continuará generando aún cuando no se haya completado la transmisión de una trama.

A partir de la definición de carga ofrecida por estación y de la Figura A-1, se tiene

$$G_i = \frac{t_{PRE} + \frac{8 \cdot l_i}{v_T} + t_{IFS}}{t_{PRE} + \frac{8 \cdot l_i}{v_T} + t_{IFS} + \lambda_i} = \frac{t_i}{t_i + \lambda_i}$$

donde t_i corresponde al tiempo de ocupación del medio por una trama transmitida por la estación i. De aquí, si se quiere fijar la carga ofrecida por la estación i a un valor dado, el parámetro λ_i viene dado por

$$\lambda_i = t_i \cdot \left(\frac{1}{G_i} - 1\right) = \left(t_{PRE} + \frac{l_i}{v_T} + t_{IFS}\right)\left(\frac{1}{G_i} - 1\right); \qquad \text{con } 0 < G_i \leq 1.$$

Ahora definimos como "carga ofrecida al segmento", G, como la suma de cargas ofrecidas por cada una de sus estaciones. Observar que G representa la proporción del ancho de banda del canal que se utilizaría en el caso ideal de que todas las tramas se pudieran transmitir tan pronto se generasen (i.e. que no se produjeran colisiones y que las estaciones encontrasen el medio libre siempre que deseen transmitir).

A.1.3. Validación simulador de un segmento

Con el objetivo de validar el simulador de segmentos Ethernet se llevaron a cabo varias simulaciones a fin de comparar los resultados con los existentes en la bibliografía. Para ello, se simuló un segmento Ethernet de longitud 10000 mts sin repetidores (equivalente a un tiempo de propagación de 512 μseg., tiempo de transmisión de 512 bits, el máximo permitido en las especificaciones) y con 38 estaciones equiespaciadas a lo largo del segmento. Se efectuaron simulaciones para cargas ofrecidas de 5, 10, 25, 50, 75, 100, 200 y 1000 % con longitudes fijas de 64, 128, 256, 512 y 1518 bytes (tráfico monomodal). Para realizar las medidas estadísticas, se ignoraron las primeras 1000 tramas de cada simulación, recogiéndose medidas para las siguientes 10000 tramas (transmitidas o descartadas). Los resultados obtenidos se presentan en la Figura A-2. Las variables medidas son las más extensamente estudiadas en la bibliografía: caudal cursado (o eficiencia), S, tiempo de acceso al medio, T, y probabilidad de pérdida de una trama por sufrir un número excesivo de colisiones en intentos consecutivos de transmisión (gráficas (a), (b) y (c) respectivamente en la Figura A-2). El caudal S corresponde a la relación entre el número total de bits transmitidos (incluyendo cabecera y CRC de cada trama, pero no el preámbulo ni la guarda entre tramas) y la capacidad del medio (10 Mbits/seg.). El tiempo de servicio T corresponde al tiempo medio entre el instante de generación de una trama y el instante en que la red consigue transmitirla (finalización de transmisión de su último bit). A su vez, la probabilidad de pérdida corresponde a la porción de tramas generadas y que son descartadas debido a un número excesivo de colisiones.

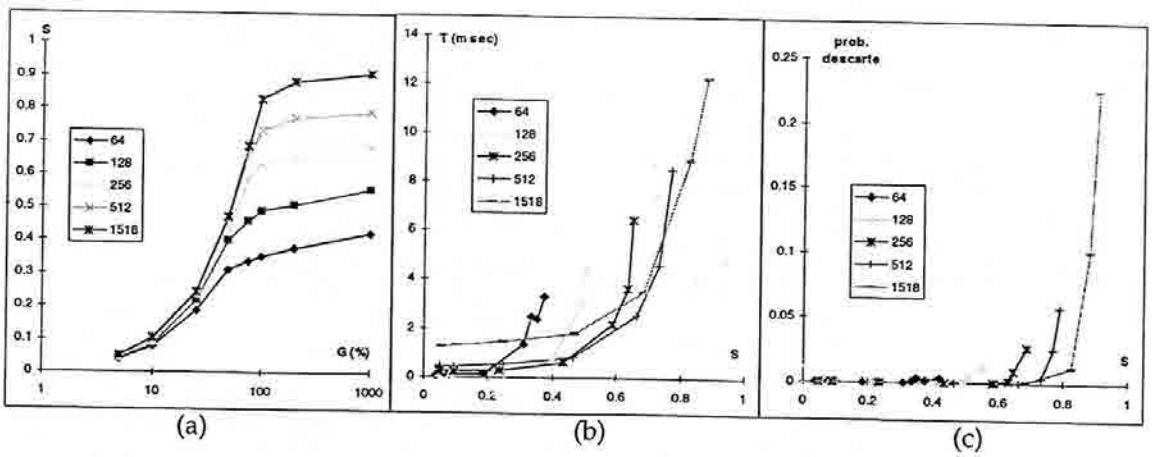


Figura A-2: Comportamiento simulado con tráfico monomodal (NOTA: en (b) no se incluye G=1000%).

A.1.4.Simulador de redes extendidas

Para construir el simulador de redes extendidas se utilizó el lenguaje de programación concurrente PADD basado en paso de mensajes entre procesos, y diseñado en el Dpto. de Matemática Aplicada y Telemática de la UPC y para el que se disponen entornos de desarrollo en diferentes plataformas (DOS, UNIX, ...) [FOR95]. La incorporación de técnicas de programación concurrente y de orientación a objeto en el terreno de diseño de simuladores es un área de investigación abierta [JAI91]. En nuestro caso, la utilización de un lenguaje con construcciones de concurrencia permitió acortar considerablemente la fase de desarrollo y depuración del simulador gracias a que el modelo programado presenta una correspondencia "natural" con el sistema físico (cada segmento de red y cada bridge en el sistema son procesos independientes en el simulador, mientras que el tránsito de tramas entre diferentes segmentos es modelado mediante intercambio de mensajes entre los procesos afectados). Por otro lado, la utilización del concepto de procesos ligeros [SIL94] permite obtener una eficiencia en la ejecución de los simuladores que hasta la fecha sólo se podía conseguir con lenguajes de programación secuencial convencionales. En resumen, la programación del simulador mediante un lenguaje concurrente ha permitido conseguir un compromiso entre la sencillez de programación y la eficiencia del código (características más destacables de lenguajes/entornos de simulación específicos y de lenguajes de propósito general, respectivamente).

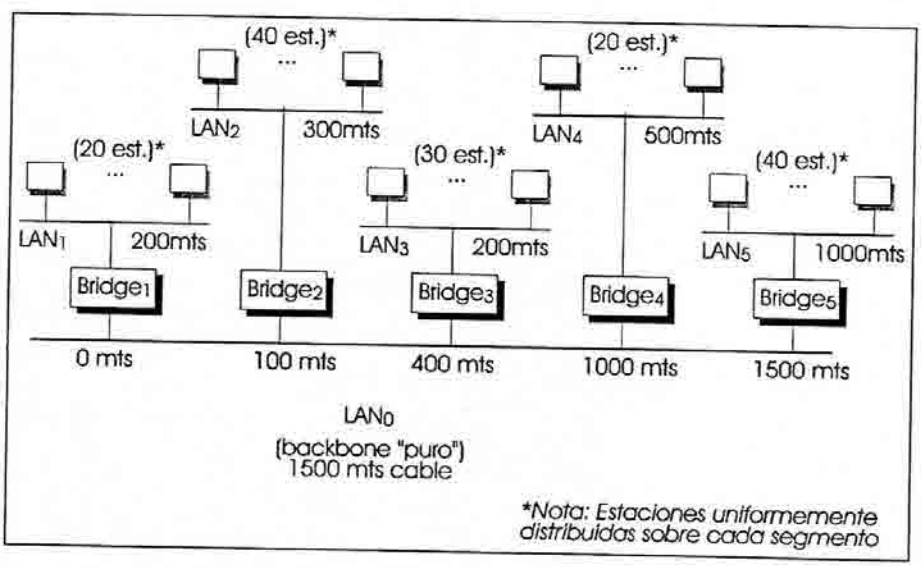


Figura A-3: Escenario de experimentación

La configuración topológica de una red local extendida real puede variar mucho de una instalación a otra. Como se comenta en el segundo capítulo, la topología que considera el sistema de seguridad estudiado en esta tesis modela la red como un conjunto de subredes locales interconectadas a través de otra subred local a la que se denomina *backbone*. A la hora de diseñar los experimentos de simulación para evaluar los protocolos de seguridad propuestos se decidió, por sencillez, realizar todos ellos sobre una misma configuración de referencia. Esta configuración de referencia se muestra en la Figura A-3, y está compuesta por cinco segmentos Ethernet/802.3 conectados cada uno de ellos mediante un *bridge* a un sexto segmento. Este sexto segmento hace funciones de *backbone* "puro", es decir no tiene ninguna estación conectada al mismo aparte de los cinco *bridges* mencionados. En cada uno de los segmentos (de longitudes diferentes) se distribuyen uniformemente un número determinado de estaciones (ver figura). Se debe observar que el segmento que hace las funciones de *backbone* tiene una longitud elevada (1500 mts, o el máximo permitido por las especificaciones para un segmento de cable sin repetidores). La razón de haber escogido una longitud elevada es doble: por un lado, este segmento debe recorrer todo el dominio de la organización por lo que cabe esperar que en la práctica la longitud de este segmento sea normalmente grande; por otro lado,

el utilizar una longitud elevada provoca mayores retardos de propagación y el comportamiento de la subred va a ser peor [GON87], de manera que cabe esperar que los resultados que se obtengan sean una cota "pesimista".

En el simulador construido, cada *bridge* recibe la secuencia de tramas desde su subred LAN_i en los mismos instantes en que estas son transmitidas sobre el medio (determinados por el mecanismo de acceso y los procesos de generación). Una vez se ha recibido una trama, esta será retransmitida con probabilidad p ("coeficiente de tráfico externo" de la LAN_i), no obstante antes deberá ganar el acceso al medio sobre el *backbone*. Obviamente este tiempo no es determinista y puede ocurrir que antes de que se consiga enviar esta trama, se reciban otras más desde LAN_i debiéndose almacenar éstas hasta que la primera se consiga transmitir. Dejando otros detalles aparte (véase el capítulo 5 para una descripción más completa de los modelos de *bridge* empleados), el modelo utilizado para simular los *bridges* tiene dos importantes características:

- la capacidad de las colas internas es ilimitada; de forma que nunca se producen descartes por desbordamiento de capacidad.
- la operación de los bridges es unidireccional. Esto significa que el simulador permite por ejemplo evaluar el tiempo de tránsito de una trama desde que es generada por la estación emisora hasta que llega al backbone, pero no el tiempo que tardaría en llegar a la estación destinataria en la subred correspondiente, pues en el simulador esta trama "muere" en el *backbone*.

La segunda característica puede parecer una limitación a primera vista. No obstante, el objetivo de este simulador es sólo estudiar cómo afecta el protocolo de seguridad propuesto al transito de las tramas sobre el *backbone*. De manera que incorporar la segunda dirección (*backbone*->subred) en los *bridges* hubiera aumentado la complejidad del simulador y no habría aportado información adicional significativa.

A.1.5.Modelo de tráfico para las simulaciones de red extendida

El modelo de tráfico utilizado en las simulaciones se diseñó intentando emular el tráfico presente en este tipo de redes en periodos de carga elevada (con la intención de situar al sistema en una situación desfavorable). Para ello, partiendo del modelo de generación de tráfico descrito en A.1.2, la longitud de las tramas que emite cada estación, l_i, el tráfico que ofrece ésta, G_i, y el número de estaciones que emite a cada longitud, se ajustan para simular cargas de tráfico diferentes con distribuciones similares a las observadas en este tipo de redes durante periodos de carga elevada. Es un resultado conocido [SMI91a] que los periodos de mayor ocupación de este tipo de redes se identifican con ráfagas "rápidas" de tramas de longitud elevada que corresponden, por ejemplo, a transferencias de ficheros de tamaño elevado. A fin de ilustrar el criterio de diseño del modelo del tráfico en periodos de alta carga, se muestran a continuación una serie de medidas estadísticas llevadas a cabo durante una semana sobre la red Ethernet/802.3 del Dpto. de Matemática Aplicada y Telemática de la UPC[4]. En la Figura A-1 se muestra el caudal por minuto observado a lo largo de uno de los días en que tuvo lugar la medida. En la Figura A-5 se muestra el perfil de tráfico observado promediado durante todo un día de observación[5]. En la Figura A-6 se muestra el perfil promediado en la hora más cargada observada durante el anterior experimento. Finalmente, en Figura A-7 se muestra el perfil de tráfico promediado sobre el minuto más cargado observado el día 7 de Junio de 1996.

[4] Esta es una red Ethernet a la que están conectadas aproximadamente tres decenas de estaciones de trabajo UNIX y un centenar de ordenadores PC y Macintosh.

[5] Los días a que corresponden esta y la anterior gráfica son diferentes.

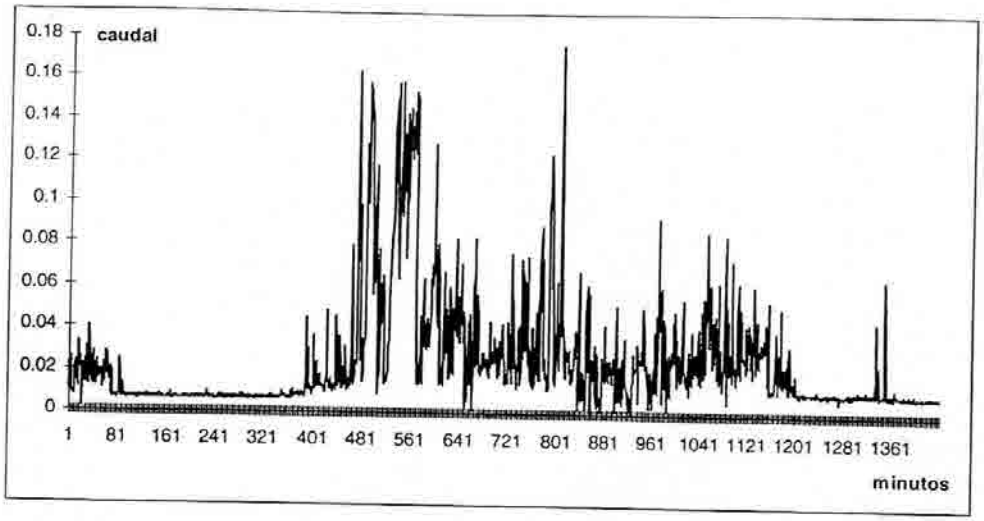


Figura A-4: Histograma de caudal observado durante un día (10 de Junio de 1996)

La conclusión más importante que se extrae de estas medidas es precisamente lo que ya se ha comentado: los periodos de mayor ocupación se producen durante la transferencia de ráfagas de tramas de máxima longitud.

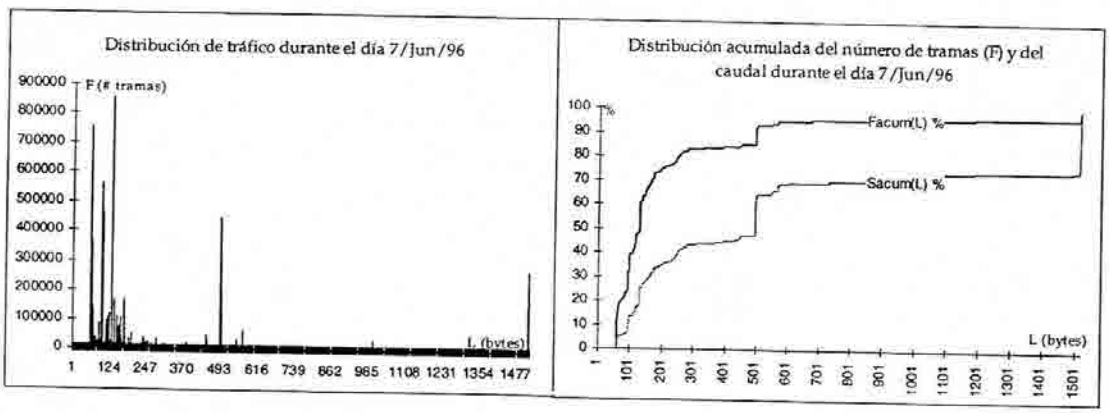


Figura A-5: Perfil de tráfico observado durante el día 7 de Junio de 1996

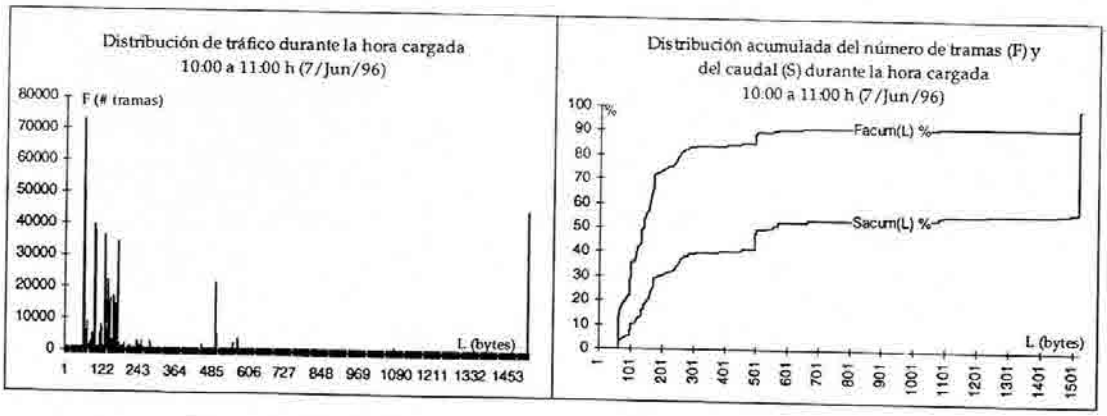


Figura A-6: Perfil de tráfico observado durante la hora más cargada

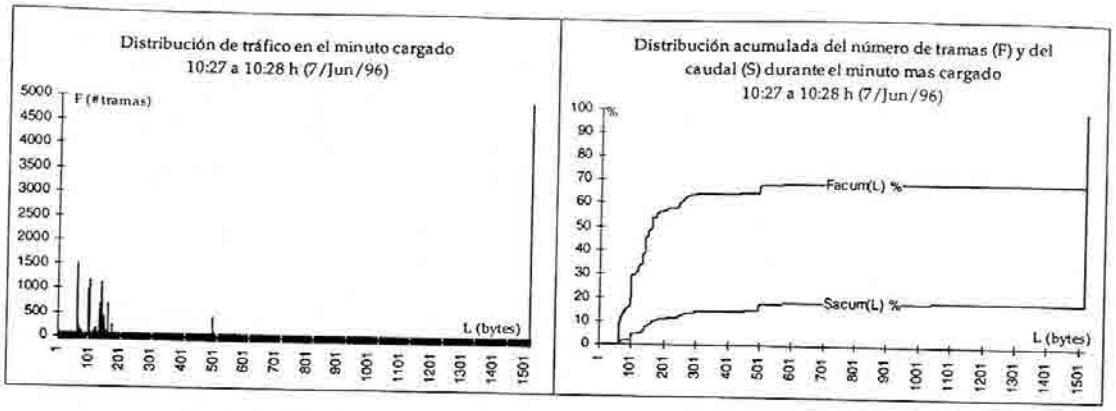


Figura A-7: Perfil de tráfico observado durante el minuto más cargado

El modelo de tráfico utilizado en las simulaciones intenta reproducir de la manera más fiel posible los periodos de carga elevada[6]. Para ello, independientemente del nivel de carga ofrecida a cada segmento de la red de referencia, G_j, ésta se distribuye en tramas de longitudes 64, 128, 256, 512 y 1518 bytes según se indica en la Tabla A-2.

l_i (bytes)	64	128	256	512	1518
$G_j(l_i)$	1% G_j	3% G_j	11% G_j	5% G_j	80% G_j
N° tramas(l_i)	10%	20%	35%	5%	30%

Tabla A-2: Perfil de tráfico "artificial"

En la Figura A-8 se compara el perfil de tráfico "artificial" de la Tabla A-2 con el del minuto cargado de la Figura A-7. Las diferencias que se pueden observar en la franja de longitudes pequeñas se debe a que el diseño del tráfico de simulación se efectuó con medidas anteriores a las que aquí se muestran y de las que se perdió registro.

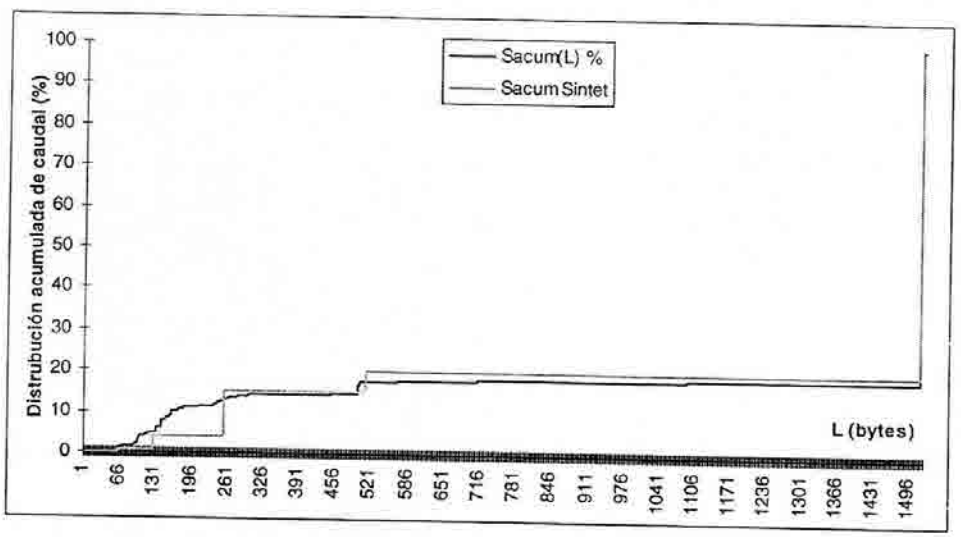


Figura A-8: Perfil del minuto cargado y del tráfico usado en las simulaciones

Para forzar este perfil de tráfico, puesto que cada estación emite tráfico monomodal, se sitúan en cada segmento estaciones emitiendo a las cinco diferentes longitudes consideradas.

[6] Es en estos periodos donde interesa más evaluar el comportamiento del sistema pues será aquí donde primero se pondrá de manifiesto el impacto sobre las prestaciones de la red que pudieran producir los mecanismos de seguridad.

En la Tabla A-3 se muestra el número de estaciones en cada una de las cinco subredes del sistema de referencia según las longitudes de las tramas que emiten[7].

l_i (bytes)	64	128	256	512	1518	TOTAL
LAN_1	10	2	2	2	4	20
LAN_2	2	7	10	15	6	40
LAN_3	14	6	4	3	3	30
LAN_4	1	1	1	2	15	20
LAN_5	12	12	5	4	7	40

Tabla A-3: Número de estaciones por longitud de trama en las cinco subredes

Además, una vez fijada la carga que se desea ofrecer para una longitud de trama determinada, esta carga se distribuye de manera uniforme entre todas las estaciones que trabajan a esta longitud. Para clarificar todo esto parece conveniente exponer un ejemplo. Supongamos que se desea ofrecer una carga del 75% a la subred LAN_4. Esta carga se distribuirá de la siguiente manera (ver Tabla A-2):

- 0.75% en tramas de longitud 64 bytes;
- 2.25% en tramas de longitud 128 bytes;
- 8.25% en tramas de longitud 256 bytes;
- 3.75% en tramas de longitud 512 bytes;
- y 60% en tramas de longitud 1518.

Dados los números de estaciones para cada longitud en la subred LAN_4 (ver Tabla A-3), cada una de estas deberá ofertar una carga como sigue:

- 0.75% la estación "a" 64;
- 2.25% la estación "a" 128;
- 8.25% la estación "a" 256;
- 1.875% cada estación "a" 512;
- y 4% cada estación "a" 1518.

Finalmente, el tiempo medio entre generaciones se determina en cada estación según se indica en A.1.2.

Para finalizar, hay que decir que en todas las simulaciones realizadas, la relación entre tráfico externo y tráfico local se fijó idéntico para todas las subredes, y en un valor que va desde un 20% hasta un 50% (dependiendo de cada simulación). Este coeficiente de tráfico externo se utilizó junto a la carga ofrecida a cada subred para fijar el punto de trabajo del *backbone* en cada caso.

A.1.6.Curvas de trabajo de las seis subredes del sistema de referencia

Una vez programado el modelo de la red extendida de referencia, se llevaron a cabo simulaciones a fin de obtener curvas de prestaciones que caracterizasen a cada una de las seis subredes. Para ello se inyectó en cada una de estas subredes tráfico multimodal (ver Tabla A-2) según diferentes cargas ofrecidas, midiéndose en cada caso el caudal cursado, S, el tiempo de acceso al medio, T, y la probabilidad de pérdidas. Los resultados obtenidos se muestran en la Figura 9. Los puntos de las curvas correspondientes a las subredes LAN1,...,LAN5 se han obtenido para cargas ofrecidas locales del 5, 25, 75, 100 y 200%. A su vez, los puntos correspondientes a la curva de trabajo del *backbone* (LAN0) se han medido inyectando a todas las subredes periféricas los niveles de tráfico mencionados con un coeficiente de tráfico externo

[7] La ubicación física de cada estación se tomó de manera aleatoria dentro de la secuencia uniforme de distribución de las estaciones en cada segmento.

de 0.2, más un sexto (el situado más a la derecha en el gráfico de caudal) obtenido con una carga ofrecida para las subredes del 200% y el coeficiente de tráfico externo de 0.4.

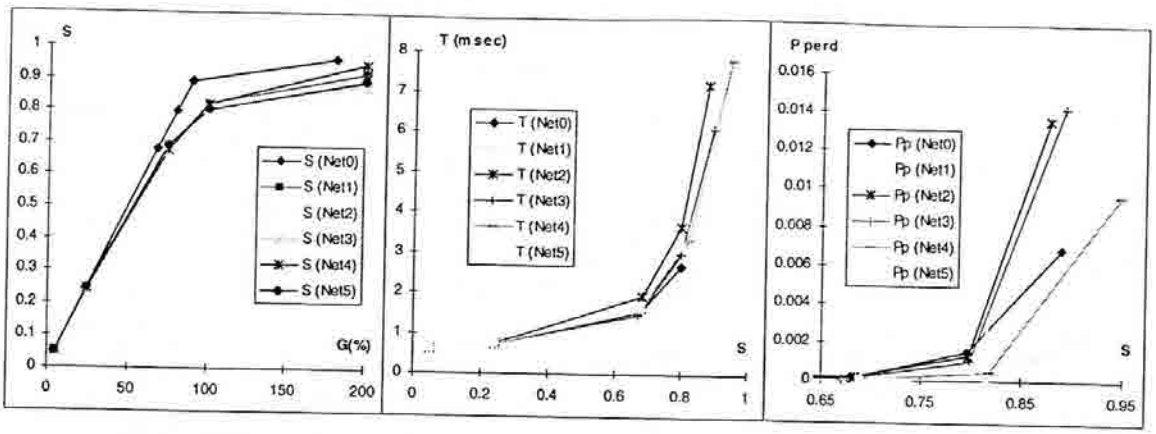


Figura 9: Curvas de trabajo de las subredes LAN0,...,LAN5 con tráfico multimodal[8]

A.1.7.Recogida de estadísticas: eliminación de transitorios, acotación de intervalos de confianza en las medidas

A fin de eliminar de los resultados los efectos que pudieran tener los periodos transitorios iniciales, en todas las medidas realizadas se ignoran los primeros 500 eventos. Este número se determinó mediante prueba y error en las simulaciones de A.1.3. Por otro lado, con el propósito de obtener buenos intervalos de confianza, cada simulación se ejecutó 10 veces usando semillas independientes, y la duración de la simulación se ajustó en cada caso para minimizar el tiempo de ejecución manteniendo buenos intervalos de confianza en las medidas. En todos los casos (exceptuando algún caso aislado de medidas de tiempo en condiciones de carga muy elevada, en los que obtener intervalos de confianza buenos hubiera supuesto tiempos de simulación prohibitivos), los valores medios medidos presentan errores inferiores al 10% para un nivel de confianza del 95% (en la mayoría de casos la cota de este error es mucho menor).

[8] En condiciones de carga elevada la varianza del tiempo de acceso y de probabilidad de pérdidas observadas es muy elevada. En estas gráficas se han excluido aquellos puntos cuya medida no ofrecia un intervalo de confianza satisfactorio.

Artículos publicados

A continuación se detallan los artículos en revistas y comunicaciones en congresos que se han realizado a consecuencia de los estudios y trabajos de esta tesis.

Artículos internacionales:

- F. Recacha, F. Rico, J.L. Melús, "Implementation of a Secure Bridge in an Ethernet Environment". IEEE INFOCOM'92 Proceedings, Firenze (Italy), (May'92) pp 2343-2350.
- M. Soriano, J.L. Melus, F. Recacha, "A Software Design and Implementation for Filtering, Forwading and Ciphering in a Secure Bridge". IEEE CompEuro'92 Proceedings (May'92), pp 487-492.
- X. Simón, J.L. Melús, F. Recacha. "A Cryptographic Protocol to obtain Secure Communications in Extended Ethernet Environment". 17th Annual Conference on Local Computer Networks Proceedings (Sept'92), Minneapolis (USA), pp 254-261.
- F. Recacha , J.L. Melús, X. Simón, M. Soriano, J. Forné. "Secure Data Transmission in Extended Ethernet Environments". *IEEE Journal of Selected Areas on Communications*. vol. 11 n. 5 (June 1993), pp. 794-803.
- J. Forné, M. Soriano, J.L. Melús, F. Recacha. "Hardware Implementation of a Secure Bridge in Ethernet Environment". Globecom'93, Houston (USA), Noviembre 1993.
- M. Soriano, J. Forné, J.L. Melús, F. Recacha. "Implementation of a Security System in a Local Area Network Environment". 18th Annual Conference on Local Computer Networks, Minneapolis (USA), Septiembre 1993, pp. 230-237.
- M. Soriano, J. Forné, F. Recacha, J.L. Melús. "A Particular Solution to Provide Secure Communications in an Ethernet Environment". 1st ACM Conference on Computer and Communications Society, Virginia (USA), Noviembre 1993, pp. 17-25.
- F. Recacha, J. Forné, J.L. Melús, "A solution to secure inter-networking with Metropolitan Area Networks" Minutes of the EEC DG XIII/B (InfoSec) WorkShop on Security Dependability and Safety of Communications Systems and Services, Brussels (June 1995).
- J. Forné, J.L. Melús, F. Recacha, M. Soriano. "The CRIPTO Project: Security in Broadband Communications". Poster presentado en 3rd European Symposium on Research in Computer Security, ESORICS'94, Brighton (UK), Noviembre 1994.
- A. Barba, F. Recacha, J.L. Melús "Security Architecture in the UMTS network. A Comparison with the FPLMTS Network". International II Conference on Universal Personal Communications, pp. 854-860, Ottawa (USA), 12-15 October 1993.
- A. Barba, F. Recacha, J.L. Melús "Security Architecture in the third Generation Networks". SICON/ICIE'93, pp. 421-425, Singapur, 6-11 September 1993.

Artículos nacionales:

- F. Recacha, J. Forné, X. Simón., M. Soriano. "Desarrollo de un Sistema de Seguridad para la red UPCNET basado en bridges cifradores". Actas de la I Reunión Española sobre Criptología, Mallorca (Oct-1991).
- F. Recacha, X. Simón, J.L. Melús. "Desarrollo y evaluación de un protocolo criptográfico para proteger las transmisiones en red local Ethernet" Actas de la II Reunión Española sobre Criptología (Oct'92) pp 25-30.
- F. Recacha, J. Margarit, "Placa hardware universal para cifrado a través de puerto serie". Actas de la II Reunión Española sobre Criptología (Oct-92) pp 31-34.
- F. Recacha, J.L. Melús. "Seguridad en Sistemas Abiertos: Aplicación a la Redes Locales". Mundo Electrónico (Marzo 1994), n. 246, pp. 26-33
- J. Forné, F. Recacha, "Gestión de Claves en un Terminal Multimedia para RDSI-BA". Actas de la III Reunión Española sobre Criptología (Nov-94).
- F. Recacha, A. Barba, E. Cruselles, J.L. Melús. "Comunicaciones Móviles: Seguridad en Redes GSM". Mundo Electrónico (Octubre 1994), n. 251, pp. 42-50. Premio Mundo Electrónico 94/95 al mejor artículo del año.
- E. Cruselles, J.L. Melús, A. Barba, F. Recacha. "Seguridad en Comunicaciones Móviles: Mecanismos y Servicios". Mundo Electrónico (Abril 1994), n. 247, pp. 22-31
- A. Barba, J.L. Melús, F. Recacha, E. Cruselles. "Comunicaciones Móviles: Seguridad en Sistemas de Tercera Generación UMTS". Mundo Electrónico (Sept. 95), n. 259, pp. 62-69.
- J.L. Melús, F. Recacha, A. Barba, E. Cruselles. "Seguridad en Comunicaciones Móviles: Sistema DECT". Mundo Electrónico (por aparecer en breve).
- J. Forné, M. Soriano, F. Recacha, J.L. Melús. "Seguridad en Redes de Banda Ancha". Mundo Electrónico (Octubre 95), n. 260, pp. 64-69.
- F. Recacha. "IOBC: Un nuevo modo para cifrado en bloque". Sometido a la IV Reunión Española sobre Criptología.

Bibliografía

[ABR85] M. Abrams, "Observations on Local Area Network Security", Tutorial: Computer and Network Security, IEEE Comp. Soc. Press, 1987, pp. 317-322. Reprinted from Proceedings of the Aerospace Computer Security Conference: Protecting Individual Property in Space, 1985, pp. 77-82.

[ABR87] M. Abrams, A. Jeng. "Network Security: Protocol Reference Model and the Trusted Computer System Evaluation Criteria", IEEE Network Magazine (April 1987), vol. 1, no. 2, pp. 24-33

[ADA92] J.A. Adam, "Data security", IEEE Spectrum, August 1992, vol. 29, n. 8, pp. 18-20.

[AGN] G. Agnew. "Secrecy and Privacy in a Local Area Network Environment", pp. 349-363.

[ANS83] ANSI X3.106, American National Standard for information systems - Data encryption algorithm - Modes of operation. American National Standards Institute, 1983

[ANS86] ANSI X9.9, Financial Institution Message Authentication (Wholesale). American National Standards Institute, 1986

[BAR89] L.K. Barker. G. A. Evans. "The Impact of Security Services Selection for LANS" Proceedings of the workshop LANSEC´89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 13-18

[BELL94] M. Bellare, J. Kilian, P. Rogaway. "The Security of Cipher Block Chaining". Proceedings of the Crypto'94, pp. 341-358.

[BLA91] U. Black. *OSI: A Model for Computer Communications*. Prentice Hall, NJ, 1991.

[BOG88] D. Boggs, J. Mogul, C. Kent. "Measured Capacity of an Ethernet: Myths and Reality", Proceedings of the ACM SIGCOMM'88 Symposium, pp. 222-228

[BRA87] D. Branstad. "Considerations for Security in the OSI Architecture", IEEE Network Magazine (April 1987), vol. 1, no. 2, pp. 34-39

[BRA80] G. Brassard. Modern Cryptology: A Tutorial (Lecture Notes in Computer Science 325). Springer Verlag, Berlin, 1980.

[BUX87] W. Bux. "Local Area Subnetworks: A Performance Comparision". *Advances in local Area Networks*. IEEE Press (1987), pp. 363-382.

[CCI] CCITT X-509. Open Systems Interconnection - The Directory: Authentication Framework.

[COM88] D.E. Comer, *Internetworking with TCP/IP: Principles, Protocols and Architecture*. Prentice Hall, N.J., 1988.

[COT75] Y. Cotton, P. Meissner. "Approaches to Controlling Personal Access to Computer Terminals". Proceedings of the Computer Networking Symposium (june 1975). pp. 32-39. Also in *Computer Networks: A Tutorial*. IEEE Computer Soc. Press 1984.

[CRY88] Cryptech. *The Cry 12C102 DES Chip. Technical Reference Manual*. CRYPTECH NV/SA Brusels (February 1988)

[DAV89] Davies,D.W. & Price,W.L. *Security for Computer Networks*. John Wiley & Sons 1984.

[DEN82] D.E. Denning. *Cryptography and Data Security*. Addison-Wesley, Reading, MA, 1982.

[DIF76] W. Diffie, M. Hellman, "New Directions in Cryptography", IEEE Transanctions on Information Theory, vol. IT-22, no. 6 (1976), pp. 644-654

[DIF77] W. Diffie, M. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard", Computer (June 1977).

[DOD87] U.S. National Computer Security Center. Trusted Network Interpretation of the Trusted Computer Sustem Evaluation Criteria. NCSC-TG-005 Version 1, July 1987.

[EUR91] Commision of the European Communities. Information Technology Security Evaluation Criteria (ITSEC): Provisional Harmonised Criteria, Bruselas 1991.

[EUR95] InfoSec Unit of DGXIII.B of the European Commission. "Announcement of a Security, Dependability and Safety of Communication System and Services". Electronic mail message (Feb. 1995).

[FOR94] W. Ford. *Computer Communications Security*. PTR Prentice Hall (1994).

[FOR95] J. Forga, F. Recacha. *Notación de Diseño y Simulación de Software Distribuido: PADD. Descripción del Lenguaje y de las Bibliotecas*. Centro de Publicaciones de la ETSIT de Barcelona (1995).

[FOR93] J. Forné, M. Soriano, J.L. Melús, F. Recacha. "Hardware Implentation of a Secure Bridge in Ethernet Environment". Globecom'93, Houston (USA), Noviembre 1993.

[FRE92] H. Freeman. "New Directions in Local Area Networks", (Tutorial Documentation). INFOCOM'92 Tutorials (May 1992).

[FUL93] P. Fulchignoni, G. Marrota, A. Wiley. "Applying Security to SMDS: A Practical Example". Proceedings of INFOCOM'93, pp. 1414-1421

[GAN94] R. Ganesan, "Securing the Information Superhighway", IEEE Communications Magazine, September 1994, vol. 32, n. 9, pp. 28-30.

[GAS89] M. Gasser. "Access Control and Authentication in LANs" Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 19-30

[GIR91] G. Girling. Letter ballot in the Minutes of the LAN Security Working Group for IEEE 802.10, 24th Meeting (802.10-91/3). Oakland, CA (May 1991).

[GON93] L. Gong. "Variations on the Themes of Message Freshness and Replay". Proceedings of the IEEE Computer Foundations Workshop VI (1993), PP. 131-136.

[GON87] T. Gonsalves. "Measured Performance of the Ethernet". Advances in Local Area Networks. IEEE Press 1987, pp. 383-410.

[HAJ91] A. Hajare. "Performance Modelling of LAN Bridges and Routers". Proceedings of the 16th Conference on Local Computer Networks, pp. 554-561

[HAM86] J. Hammond, P. O'Reilly. *Performance Analysis of Local Computer Networks*. Addison-Wesley, 1986.

[HAW84] B. Hawe, A. Kirby, B. Stewart. "Transparent Interconnection of Local Area Networks with Bridges". Journal of Telecommunications Networks (Summer 1984), vol. 4, n. 2. Also in: W. Stallings (Editor), Tutorial: *Local Network Technology*, IEEE Comp. Press 1985, pp. 366-380.

[HER88] B.J. Herbison. "Developing Ethernet Enhanded-Security System". Advances in Cryptology: Proceedings of the Crypto'88, pp. 505-519.

[HOL91] G.J. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall, 1991.

[HOU89] R. Housley "Encapsulation Security Protocol Design for Local Area Networks". Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 103-112

[IEE85] IEEE 802.3. "Standards for local area networks: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications". 1985.

[IEE89] IEEE 802.10a. Standard for Interoperable LAN Security (SILS): Part A - The Model. Unapproved Draft.

[IEE91] IEEE 802.10b. Standard for Interoperable LAN Security (SILS): Part B - Secure Data Exchange. Unapproved Draft.

[IEE92] IEEE 802.10c. Standard for Interoperable LAN Security (SILS): Part C - Key Nanagemenent Proposal. Unapproved Draft.

[ISO84] ISO 7498, Information Processing Systems - Open Systems Interconnection - Basic Reference Model, 15 October 1984.

[ISO88] ISO 7498-2-1988(E), Information Processing Systems - Open Systems Interconnection Reference Model - Part 2 Security Architecture. 1988.

[ISO89] ISO IEC/DIS 10116, Information processing - Modes of operation for an *n*-bit block cipher algorithm. 1989.

[JAI91] Raj Jain. *The Art of Computer Systems Performance Analisis. Techniques for Experimental Design, Measurement, Simulation and Modeling*. John Wiley & Sons, Inc (1991).

[JAN87] C. Jansen, D. Boekee, "Modes of Blockcipher Algorithms and their Protection Against Active Eavesdropping". EUROCRYPT'87 Proceedings, pp. 281 - 286

[JOH95] J. Johnson, "Enterprise Security: Better Safe than Sorry", Data Communications (March 1995), vol. 24, n. 3, pp. 110-127.

[KIR89] K. E. Kirkpatrick. "Why is a LAN a LAN?". Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 3-4.

[KIR90] K. E. Kirkpatrick."P802.10-90/16: SILS Overview", Documentation of a 802.10 plenary. MITRE Corporation (1990).

[KLE76] L. Kleinrock. *Queuing Systems*, vol. 2: *Computer Applications*. John Wiley (1976)..

[KÜM87] K. Kümmerle, M. Reiser. "Local Area Networks - Major Technologies and trends". Advances in Local Area Networks. IEEE Press (1987), pp. 2-26.

[LAN89] Interview with Thomas A. Berson. Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. vii-ix

[LAM89] P.A. Lambert. "Architectural Considerations for LAN Security Protocols" Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 5-12

[LEL91] W. Leland, D. Wilson. "High-Time Resolution Measurement and Analysis of LAN Traffic: Implications for LAN Interconnection". Proceedings of the INFOCOM'91 (1991), pp. 1360-1366.

[MAD92] T. W. Madron. *Network Security in the '90s.* John Wiley & Sons, Inc., 1992.

[MAL89] C. Malaud, *DEC Networks and Architectures*, Intertext Publications McGraw-Hill Book Company (1989).

[MAT93] M. Matsui, "Linear Cryptanalisis Method for DES Cipher". Proceedings, EUROCRYPT'93, Springer-Verlag.

[MED87] J. Meditch, C-T. Lea. "Stability and Optimization of the CSMA and CSMA/CD Channels". In *Advances in Local Area Networks.* IEEE Press (1987), pp. 341-362.

[MET76] R. Metcalfe and D. Boggs. "Ethernet: Distributed Packet Switching for Local Computer Networks". ACM (1987). Also in *The Ethernet Sourcebook*, pp. 3-12.

[MEY82] C. Meyer, S. Matyas. *Cryptography: A New Dimension in Computer Data Security.* New York: Wiley 1982.

[MIN91] D. Minoli. *Telecommunications Technology Handbook.* Artech House, Norwood 1991.

[NBS77] National Bureau of Standards. "Data Encryption Standard" Federal Information Processing Standards Publication 46 (January 1977).

[NEE78] R. Needham, M. Schroeder. "Using Encryption for Authentication in Large Networks of Computers". Communications of the ACM, vol. 21, n. 12 (Dec. 1978).

[NET91] "Source Routing Transparent: Facts and Miths", Netronix (May 91), pp. 24-37. Also in: W. Stallings (Editor), *Advances in Local and Metropolitan Area Networks*, IEEE Comp. Press 1994, pp. 375-381.

[PAR90] R. Parker, "P802.10-90/16: Layer 2 Security Services for Local Area Networks", Documentation of a 802.10 plenary. MITRE Corporation.

[PAS88] J. Pastor, *Notas para un curso monográfico de Criptografía Moderna*, Documentación del curso impartido en el Dpto. de Matemática Aplicada y Telemática (UPC) en 1988.

[PFL89] C.P. Pfleeger. *Security in Computing.* Prentice Hall International, Inc. 1989

[PUR93] M. Purser. *Secure Data Networking.* Artech House, Inc. 1993.

[REC92] F. Recacha, F. Rico, J.L. Melús, "Implementation of a Secure Bridge in an Ethernet Environment". IEEE INFOCOM'92 Proceedings, Firenze (Italy), (May'92) pp 2343-2350.

[REC93] F. Recacha , J.L. Melús, X. Simón, M. Soriano, J. Forné. "Secure Data Transmission in Extended Ethernet Environments". *IEEE Journal of Selected Areas on Communications.* vol. 11 n. 5 (June 1993), pp. 794-803

[REC94] F. Recacha, J.L. Melús. "Seguridad en Sistemas Abiertos: Aplicación a la Redes Locales". Mundo Electrónico (Marzo 1994), n. 246, pp. 26-33

[REC95] F. Recacha, J. Forné, J.L. Melús, "A solution to secure inter-networking with Metropolitan Area Networks" Minutes of the EEC DG XIII/B (InfoSec) WorkShop on Security Dependability and Safety of Communications Systems and Services, Brussels (June 1995).

[REC96] F. Recacha. " IOBC: Un nuevo modo para cifrado en bloque". IV Reunión Española sobre Criptología (Sept. 1996).

[RIC90] J. Rickert Jr., "Evaluating MAC-Layer Bridges - Beyond Filtering and Forwarding". Data Communications Magazine (May 1990), pp. 117-122. Also in: W. Stallings (Editor), *Advances in Local and Metropolitan Area Networks*, IEEE Comp. Press 1994, pp. 392-395.

[RIV77] Rivest, R., Shamir, A. Adlernan, L. "A method for obtaining digital signatures and public-key cryptosystems". MIT Laboratoty for Computer Science. Technical Memo LCS/TM82 (April 1977).

[RUS91] D. Russell, G.T. Gangemi. *Computer Security Basics*. O'Reilly & Associates Inc., July 1991.

[SCH80] J. Schoch, J. Hupp. "Measured Performance of an Ethernet Local Network". *Communications of the ACM*, vol. 23, n. 12 (Dec. 1980), pp. 711-721.

[SCH87] J. Schoch, Y. Dalal, D. Redell, R. Crane. "Ethernet". Advances in Local Area Networks. IEEE Press (1987), pp. 28-48

[SCH89] B. Schanning, "Secure Relays: An Alternative Approach to LANSEC". Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 31-52

[SHI82] R. Shirey. "Security in Local Area Networks". Proceedings of the IEEE Computer Networking Symposium (Dec. 1982), pp. 28-44. Also in Computer Networks: A Tutorial, IEEE Computer Society Press (1984).

[SID82] D.P. Sidhu, M. Gasser. "A Multilevel Secure Local Area Network". Proceedings of the 1982 Symposium on Security and Privacy, pp. 137-143

[SIL94] A. Silberschatz, J. Peterson, P. Galvin. *Sistemas Operativos. Conceptos Fundamentales*. Tercera edición. Addison Wesley Iberoamericana, 1994.

[SIM92] G. J. Simmons (Ed). *Contemporary Cryptology. The Science of Information Integrity*. IEEE Press, 1992.

[SIMO92] X. Simón, J.L. Melús, F. Recacha. "A Cryptographic Protocol to Obtain Secure Communications in Extended Ethernet Environment". 17th Annual Conference on Local Computer Networks Proceedings (Sept'92), Minneapolis (USA), pp 254-261.

[SIR94] P. Siron, B. d'Ausbourg. "A Secure Medium Access Control Protocol: Security versus Performances". Proceedings of the 3rd European Symposium on Research in Computer Security, ESORICS'94, Brighton (UK), Noviembre 1994, pp. 267-279

[SMI91a] W. Smith, R. Kain. "Ethernet Performance under Actual and Simulated Loads". Proceedings of the 16th Conference on Local Computer Networks, 1991, pp. 569-581

[SMI91b] W. Smith, R. Kain. "On the Validity of Assumptions used to Model Local Area Networks". Proc. of 10th Ann. Phoenix Conf. Computers and Communications, 1991, pp. 651-657. Also in *Advances in Local and Metropolitan Area Networks* (Ed. William Stallings). IEEE Press, 1994.

[SOR92] M. Soriano, J.L. Melus, F. Recacha, "A Software Design and Implementation for Filtering, Forwading and Ciphering in a Secure Bridge". IEEE CompEuro'92 Proceedings (May'92), pp 487-492.

[SOR93] M. Soriano, J. Forné, F. Recacha, J.L. Melús. "A Particular Solution to Provide Secure Communications in an Ethernet Environment". 1st ACM Conference on Computer and Communications Security, Virginia (USA), Noviembre 1993, pp. 17-25.

[STA83] W. Stallings."Beyond Local Networks". Datamation, August 1983

[STA94a] W. Stallings. *Data and Computer Communications, Fourth Edition*. New York: Macmillan, 1994.

[STA94b] W. Stallings. "LAN and MAN Performance". In *Advances in Local and Metropolitan Are Networks*, IEEE Comp. Soc. Press, 1994.

[STA95] W. Stallings. *Network and Internetwork Security*. IEEE Press (1995).

[TAN89] Tanenbaum, A.S., *Computer Networks*, Prentice-Hall International, Inc. (1989).

[TAR85] J. Tardo. "Standardizing Cryptographic Services at OSI Higher Layers". IEEE Communications Magazine, vol. 23, n. 7 (July 1985).

[TAY87] D. Taylor, D. Oster, L. Green. "VLSI Node Processor Architecture for Ethernet". *Advances in Local Area Networks*. IEEE Press (1987), pp. 49-61.

[TOB87] F. Tobagi, V. Hunt. "Performance Analysis of Carrier Sense Multiple Access with Collision Detection". In *Advances in Local Area Networks*, IEEE Press (1987), pp. 318-339.

[VOY85] V.L. Voydock, S. T. Kent. "Security in High-Level Network Protocols"IEEE Communications Magazine, July 1985, pp 12-24.

[VAR90] Varios, Notas del curso de doctorado "Estructuras Matemáticas Discretas y Aplicaciones". Dpto. Matemática Aplicada y Telemática. Universidad Politécnica de Cataluña (1990).

[VOT87] G. Votsis. "Ethernet VLSIs Controllers Comparison". in *Local Communication Systems: LAN and PBX*. Elsevier Science Publishers (IFIP 1987).

[WIL89] S. Wilbur, J. Crowcroft, Y. Murayama. "MAC Layer Security Measures in Local Area Networks". Proceedings of the workshop LANSEC'89 in Lecture Notes in Computer Science, Springer Verlag (1989), n. 396, pp. 53-66.

[ZOR94] V. Zorkadis. "Security versus Performance Requirements in Data Communications Systems". Proceedings of the 3rd European Symposium on Research in Computer Security, ESORICS'94, Brighton (UK), Noviembre 1994, pp. 19-30.