BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

MODELADO E IMPLEMENTACIÓN DEL SERVICIO DE CONFIDENCIALIDAD USANDO AES-CCM (ESTÁNDAR AVANZADO DE CIFRADO, MODO-CONTADOR/CBC-MAC) PARA EL ESTÁNDAR IEEE 802.11

TESIS

PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

LIC. SUGEHI MARINA MERINO HIGAREDA

ASESOR

DR. MIGUEL ANGEL LEON CHAVEZ

Puebla, 2005

ÍNDICE GENERAL

ÍNDICE GENERAL

ÍNDICE FIGURAS

ÍNDICE TABLAS

ÍNDICE GRÁFICAS

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Antecedentes

Es trascendental el papel que en nuestra sociedad desempeñan las terminales móviles hoy en día. La idea de estar siempre comunicables en tiempo y en espacio se ha convertido en una necesidad, lo que ha llevado al diseño de nuevas tecnologías y redes de acceso inalámbricas.

Las redes de área local inalámbricas (Wireless LANs) han tenido mucho auge y desarrollo en estos últimos años. La función principal de este tipo de redes es proporcionar conectividad y acceso a las tradicionales redes cableadas (Ethernet, Token Ring, etc.), pero con la flexibilidad y movilidad que ofrecen las comunicaciones inalámbricas.

El estándar IEEE 802.11 [1] define un protocolo con conexión compatible a equipos de comunicaciones que utilizan el aire, radio o infrarrojo como medio de transmisión, dentro de una LAN. El medio de transmisión constituye el soporte físico a través del cual emisor y receptor pueden comunicarse en un sistema de transmisión de datos.

El propósito de este estándar es el de proveer conectividad inalámbrica a maquinaria automatizada, equipo de cómputo o estaciones que requieren una distribución rápida.

Un aspecto importante a considerar cuando se diseñó este estándar fue que tenía que servir a unidades portátiles y móviles. Una unidad portátil es aquella que se mueve de un lugar a otro, pero permanece fija mientras envía y recibe información. Una unidad móvil es la que mientras está recibiendo o enviando información está en constante cambio.

Este estándar ofrece tres servicios de seguridad: Autenticación, Confidencialidad e Integridad mediante el protocolo WEP (Wired Equivalent Privacy) y CRC32. WEP esta basado en el algoritmo simétrico RC4, que es un algoritmo criptográfico muy poderoso, sin embrago, WEP usa una aproximación muy pobre.

Este documento propone el uso de Criptografía de Llave Privada en concreto el algoritmo Rijndael, Estándar Avanzado de Cifrado (Advanced Encryption Standard, AES) [2] utilizando el modo de operación CCM (Counter–Mode/CBC-MAC) [3] para implementar el servicio de confidencialidad y autenticación en el IEEE 802.11.

1.2. Problema

El estándar IEEE 802.11 utiliza el protocolo WEP, este protocolo tiene como función cifrar la información antes de su transmisión por el canal utilizando el algoritmo de cifrado por flujo RC4.

El proceso de cifrado de cada paquete es el siguiente, se cuenta con una llave privada cuya longitud es de 40 bits la cual es concatenada con IV (Vector de Inicialización) la longitud

del campo IV es de 24 bits consecutivamente se aplica RC4 para obtener el "Flujo de llave RC4", realizando una operación de or-exclusivo al resultado anterior con el texto claro obteniendo el texto cifrado a transmitir.

Sin embargo durante el proceso del WEP se presentan una serie de vulnerabilidades [4] entre ellas, que viaja en texto claro el IV, la longitud del campo IV es de 24 bits, lo que nos da 16,8 millones de combinaciones posibles para cifrar un paquete con una misma llave WEP, además que estadísticamente cada 5 horas tiende a repetirse el mismo valor para IV, debido a que la longitud de la llave WEP es de 64 bits puede realizar un ataque de fuerza bruta. También se describe la vulnerabilidad del cifrado RC4 en el documento "*Weaknesses in the Key Scheduling Algorithm of RC4*" [5]. En *"Unsafe at any key size; An analisys of the WEP encapsulation"* [6] se pone en manifiesto las vulnerabilidades del WEP. Se hace mención de ataque para romper el WEP en "*Using the Fluhrer, Mantin, and Shamir Attack to Break WEP"* [7]. Para mayor detalle acerca de los artículos anteriormente mencionados ver sección 3.6.

En concreto se explican los ataques y vulnabilidades del protocolo WEP:

Ataque Activo de Tráfico Inyectado

Si se tiene conocimiento de un texto claro y su correspondiente texto cifrado. Ahora podemos construir paquetes correctamente cifrados con la propiedad RC4(X) xor X xor Y = RC4(Y).

El Valor de Chequeo de Integridad (ICV) utilizado en WEP es implementado como una suma de verificación CRC-32, el cual es lineal, lo cual implica la posibilidad de calcular los bits de diferencia de dos CRC-32 a partir de los bits de diferencia de dos mensajes. Ahora incluso con un conocimiento parcial del contenido de un mensaje, es posible cambiar ciertos bits de un mensaje y ajustar el CRC del mensaje cifrado.

Ataque Activo Desde Ambos Extremos

El ataque previo puede ser extendido. Se intenta adivinar la IP a la cual va dirigida la trama, se alteran bits para indicar una nueva IP fuera de la WLAN en Internet. De esta manera el Punto de Acceso descifra el contenido de la trama, la cual es recibida en texto claro en la nueva máquina receptora.

Robo de Información

Cuando alguien realiza una transmisión de datos en una red, espera que no sea fácilmente leídas por personas no autorizadas. En una red inalámbrica, a diferencia de una red alambrada, si alguien desea realizar una lectura de los paquetes que viajan entre las computadoras, no requiere realizar una intrusión física, basta con que cuente con un receptor adecuado, para poder leer dichos paquetes. En las especificaciones del 802.11, se hace uso del cifrador de flujo de datos RC4, a pesar de que RC4 es un cifrador comúnmente usado, su uso en 802.11 es cuestionable. En 802.11, el cifrador es inicializado con una llave privada y un vector de inicialización de 24 bits.

El problema es el siguiente, debido a que las redes inalámbricas son muy susceptibles a errores, la sincronización del cifrador en el emisor y en el receptor se ve comprometida, por lo que se requiere que, el vector de inicialización sea transmitido en cada mensaje de manera clara. Debido a que los paquetes de 802.11 son relativamente cortos, se espera que el vector se repita de manera rápida. Debido a la baja entropía de los mensajes, el atacante puede almacenar una cantidad suficiente de mensajes y calcular la llave de cifrado empleada.

Autenticación Falsa

El estándar 802.11 establece dos tipos de autenticación, la autenticación abierta y la autenticación de llave compartida.

La autenticación abierta se refiere básicamente a no autenticación, es decir, todo quien solicita una autenticación, es autenticado.

En el método de llave compartida, cuando un cliente desea autenticarse ante el servidor (Punto de Acceso), este genera un número de 128 bits de manera pseudoaleatoria (llamado reto), que es enviado a quien solicita la autenticación. El cliente emplea la llave privada (compartida por cliente y servidor) para cifrar el número y se lo retorna a servidor, que verifica la validez del cifrado. Dependiendo de la validez del cifrado, el acceso es otorgado o negado. El cifrado del mensaje es el resultado de la operación or-exclusivo entre el reto y la llave, compuesta de un secreto (compartido entre cliente y servidor) y un vector de inicialización público. En este método de autenticación, lo único que cambia entre autenticación y autenticación, es el número pseudoaleatorio, generado por el servidor. Un ataque a este método de autenticación es: cuando el intruso almacena el reto y la respuesta, para después calcular la llave mediante la operación de or-exclusivo del reto y la respuesta. De esta manera, el impostor podrá conocer la respuesta a los siguientes retos, sin necesidad de conocer la parte secreta de la llave.

Punto de Acceso Falso

En el esquema de llave compartida, se observa que no existe una autenticación mutua. El cliente se autentica ante el punto de acceso, pero el punto de acceso jamás se autentica ante el cliente. Este esquema abre las puertas a ataques en los cuales, algún intruso se haga pasar por el punto de acceso y pueda redirigir el tráfico de los clientes.

1.3. Solución

Este trabajo de tesis propone como solución el uso del algoritmo Rijndael utilizando como modo de operación CCM para el servicio de confidencialidad y autenticación, debido que éste es un servicio primordial para ofrecer una arquitectura de seguridad [8] como ha sido definida por ISO (International Organization for Standardization) [9]. Se utiliza AES ya que es altamente improbable que existan llaves débiles o semidébiles, debido a la estructura de su diseño, que busca eliminar la simetría en las subllaves.

También se ha comprobado que AES es resistente a *criptoanálisis lineal* [10], descubierto por Mitsuru Matsui, basa su funcionamiento en tomar algunos bits del texto claro y efectuar una operación XOR entre ellos, tomar algunos del texto cifrado y hacerles lo mismo y finalmente hacer un XOR de los dos resultados anteriores, obteniendo un único bit. Efectuando esa operación a una gran cantidad de pares de texto claro y criptograma diferentes podemos ver si se obtienen más ceros o más unos.

Como al *criptoanálisis diferencial* [10] descubierto por Biham y Shamir en 1990, su funcionamiento es tomar dos mensajes cualesquiera (incluso aleatorios) idénticos salvo en un número concreto de bits. Usando las diferencias entre los textos cifrados, se asignan probabilidades a las diferentes llaves de cifrado. Conforme tenemos más y más pares, una de las llaves aparece como la más probable, esa será la llave buscada.

Se utiliza el modo CCM por que proporciona autenticación y privacidad mediante el algoritmo de cifrado AES utilizando una sola llave que se establezca de antemano. Además de las siguientes características que tiene el modo CCM:

- Permite manejar mensajes en los cuales existan partes únicamente para autenticar y no para cifrar, sin provocar una disminución en la eficiencia.
- El cifrador por bloques se utiliza únicamente en su etapa de cifrado, lo cual permite un ahorro de recursos.
- Todos los derechos intelectuales han sido liberados para el dominio público.
- Propuesto CCM por tres compañías importantes en seguridad Hifn Inc., MacFergus BV y RSA Security Inc.

1.4. Objetivos

Objetivo general: Realizar el modelado en UML (Unified Modeling Language) e implementación en C++ para análisis y evaluación de rendimiento de AES-CCM para el servicio de confidencialidad y autenticación en el estándar IEEE 802.11.

O***bjetivos específicos:***

- Analizar la arquitectura y funcionamiento del estándar IEEE 802.11.
- Estudiar la notación y semántica de UML.
- Crear el modelado en UML del estándar IEEE 802.11.
- Incluir en este modelo la nueva propuesta AES-CCM (Estándar Avanzado de Cifrado, Modo–Contador/CBC-MAC) para ofrecer el servicio de confidencialidad y autenticación.
- Construir un prototipo del servicio de confidencialidad y autenticación a nivel de aplicación utilizando AES-CCM.
- Realizar pruebas del desempeño.
- Elaborar la implementación AES-CCM a nivel enlace de datos.
- Efectuar análisis del rendimiento.

1.5. Organización de la tesis

El contenido de esta tesis cubre el marco, tanto teórico como práctico, el cual nos permiten alcanzar los objetivos anteriormente planteados.

El marco teórico está formado por los capítulos 2, 3, 4 y 5 los cuales cubren temas específicos necesarios para la comprensión del desarrollo del modelado en UML del servicio de confidencialidad mediante el protocolo WEP en el estándar IEEE 802.11 y la propuesta de este trabajo de tesis, es decir, crear el modelo en UML mediante la utilización de AES-CCM para ofrecer el servicio de confidencialidad y autenticación en dicho estádar.

El marco práctico está formado por los capítulos 6 y 7 los cuales abarcan el modelado en UML del estándar con WEP y AES-CCM, además de la implementación de AES-CCM con sus respectivas pruebas de desempeño. Se da una visión general de cada uno de los capítulos de este trabajo, para así tener una concepción de la organización del mismo.

El CAPÍTULO 2 contiene temas introductorios para ubicar al lector en el contexto general del documento y para que éste conozca los términos y conceptos en los que se basa el trabajo. Se abarca la definición y las configuraciones de redes de área local inalámbricas como tema introductorio a las redes inalámbricas.

El CAPÍTULO 3 presenta el estándar IEEE 802.11, se abarca su arquitectura en componentes, interfaces de servicio, los servicios ofrecidos a nivel de la subcapa de Control de Acceso al Medio (MAC) y los servicios de seguridad. Incluye los ataques y vulnerabilidades en el protocolo WEP. Se explica el funcionamiento del protocolo WEP. Los temas anteriores nos proporcionan conceptos básicos que se utilizan en el modelado de dicho estándar.

El CAPÍTULO 4 se da una breve introducción a la criptografía. En concreto contiene información de la historia, algoritmo y seguridad de AES.

El CAPÍTULO 5 se describe diferentes modos de operación para algoritmos de cifrado por bloques. Entre estos modos está CCM utilizando AES como algoritmo de cifrado. Siendo temas muy útiles para el modelado en UML y la implementación de AES-CCM en el estándar IEEE 802.11.

El CAPÍTULO 6 indica el modelado en UML del estándar IEEE 802.11, en donde se muestra su funcionamiento a nivel de subcapa MAC usando la Función de Coordinación Distribuida (DCF) que implementa el protocolo de Acceso Múltiple por Sensado de Portadora Evitando Colisiones (CSMA/CA) y Request To Send/Clear To Send (RTS/CTS). Además presenta el modelado en UML del servicio de seguridad de confidencialidad definido por el estándar por medio del protocolo WEP. Se muestra el modelo de AES-CCM para ofrecer el servicio de confidencialidad y autenticación. El modelado incluye los modelos de análisis y diseño.

El CAPÍTULO 7 contiene la implementación de los servicios de seguridad mediante AES-CCM. Utilizando como sistema operativo Linux, en concreto Suse versión 9.2; Además se desarolla la implementación en el lenguaje de programación C++ utilizando el compilador g++. Se presenta la implementación a nivel aplicación de AES-CCM y se realiza un análisis de desempeño. Se anexa el uso de sockets a la aplicación de AES-CCM, obteniendo los tiempos de transferencia. Se hace un análisis de desempeño del algoritmo de cifrado por flujo RC4 y AES-CCM a nivel aplicación, finalmente se realiza una comparación entre AES-CCM (D. Whiting, R. Housley y N. Ferguson) y la implementación AES-CCM (S. Merino).

El CAPÍTULO 8 presenta las conclusiones de este trabajo de tesis obtenidas durante la elaboración de la tesis, indicando cuales podrían ser las mejoras al mismo siendo parte del trabajo futuro.

Por último se agregan los APÉNDICES A y B, el apéndice A contiene un manual de instalación de Suse Linux Professional 9.2, mientras que el apéndice B indica la instalación de la Tarjeta Inalámbrica SMC 2802W V. 2.

CAPÍTULO 2

2. REDES DE ÁREA LOCAL INALÁMBRICAS

El papel que en nuestra sociedad desempeñan las terminales móviles hoy en día es trascendental. La movilidad se ha vuelto un requerimiento cada vez mayor dentro de los ambientes de trabajo y gracias a las redes inalámbricas se ha obtenido una movilidad real en los dispositivos. Las redes inalámbricas de área local (WLANs), donde el estándar IEEE 802.11 ha destacado, ofrecen una alternativa de bajo costo a los sistemas cableados. La norma 802.11 ha sufrido diferentes extensiones sobre la norma para obtener modificaciones y mejoras. De esta manera, tenemos las siguientes especificaciones:

- **802.11** [1] Especificación para 1-2 Mbps en la banda de los 2.4 GHz, usando salto de frecuencias (FHSS) o secuencia directa (DSSS).
- **802.11a** Extensión de 802.11 para proporcionar 54 Mbps usando frecuencia ortogonal (OFDM).
- **802.11b** Extensión de 802.11 para proporcionar 11 Mbps usando DSSS.
- **Wi-Fi (Wireless Fidelity)** Promulgado por el WECA (Wireless Ethernet Compatibillity Alliance) para certificar productos 802.11b capaces de interoperar con los de otros fabricantes.
- **802.11g** [11] Extensión de 802.11 para proporcionar 20-54 Mbps usando DSSS y OFDM. Es compatible con 802.11b. Tiene mayor alcance y menor consumo de potencia que 802.11a.
- **802.11i** [12,13] Extensión que presenta las propuestas para implantar una arquitectura de seguridad en este tipo de redes.

Este trabajo se centra en el estándar IEEE 802.11. Un punto importante es que los sistemas WLAN no pretenden sustituir a las tradicionales redes cableadas, sino más bien complementarlas. En este sentido el objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red total donde coexistan los dos tipos de sistemas.

2.1. Generalidades sobre redes de Área Local Inalámbricas

En los últimos años las redes inalámbricas han ganado muchos partidarios y popularidad en mercados como hospitales, fábricas, bodegas, tiendas de autoservicio, tiendas departamentales, pequeños negocios y áreas académicas. Las redes inalámbricas permiten a los usuarios accesar información y recursos en tiempo real sin necesidad de estar físicamente en un sólo lugar. Con WLANs la red por sí misma es móvil y elimina la necesidad de usar cables y establece nuevas aplicaciones añadiendo flexibilidad a la red y lo más importante incrementa la productividad y eficiencia en las actividades diarias de la empresa. Un usuario dentro de una red inalámbrica puede transmitir y recibir voz, datos y video dentro de edificios,

entre edificios o campus universitarios e inclusive sobre áreas metropolitanas a velocidades de hasta 11 Mbps.

Muchos de los fabricantes de computadoras y equipos de comunicaciones como PDAs (Personal Digital Assistants), módems, microprocesadores inalámbricos, lectores de punto de venta y otros dispositivos están introduciendo aplicaciones en soporte a las comunicaciones inalámbricas. Las nuevas posibilidades que ofrecen las WLANs son permitir una fácil incorporación de nuevos usuarios a la red, ofrecen una alternativa de bajo costo a los sistemas cableados, además de la posibilidad ubicua para accesar cualquier base de datos o cualquier aplicación localizada dentro de la red.

2.2. Definición de Red de Área Local Inalámbrica

Una *red de área local inalámbrica* puede definirse como a una red de alcance local que tiene como medio de transmisión el aire. El medio de transmisión constituye el soporte físico a través del cual emisor y receptor pueden comunicarse en un sistema de transmisión de datos. Por red de área local se interpreta una red que cubre un entorno geográfico limitado, con una velocidad de transferencia de datos relativamente alta (mayor o igual a 1 Mbps tal y como especifica el IEEE), con baja tasa de errores y administrada de forma privada. Por red inalámbrica se define una red que utiliza ondas electromagnéticas como medio de transmisión de la información que viaja a través del canal inalámbrico enlazando los diferentes equipos o terminales móviles asociados a la red. Estos enlaces se implementan básicamente a través de tecnologías de microondas y de infrarrojos.

En las redes tradicionales cableadas esta información viaja a través de cables coaxiales, pares trenzados o fibra óptica. WLAN es un sistema flexible de comunicaciones que puede implementarse como una extensión o directamente como una alternativa a una red cableada. Este tipo de redes utiliza tecnología de radiofrecuencia minimizando así la necesidad de conexiones cableadas. Este hecho proporciona al usuario una gran movilidad sin perder conectividad.

El atractivo fundamental de este tipo de redes es la facilidad de instalación. Las redes inalámbricas son la alternativa ideal para hacer llegar una red tradicional a lugares donde el cableado no lo permite, y en general las WLAN se utilizarán como un complemento de las redes fijas.

2.3. Ventajas de los sistemas WLAN

A continuación se describen cada una de las ventajas de WLANs sobre las redes alámbricas:

- *Movilidad*: Las redes inalámbricas pueden proveer a los usuarios de una LAN acceso a la información en tiempo real en cualquier lugar dentro de la organización. Esta movilidad incluye oportunidades de productividad y servicio que no es posible con una red alámbrica.

- *Simplicidad y rápidez en la instalación*: La instalación de una red inalámbrica puede ser tan rápida, fácil y además que puede eliminar la posibilidad de tirar cable a través de paredes y techos.

- *Flexibilidad en la instalación*: La tecnología inalámbrica permite a la red ir donde la alámbrica no puede ir.

- *Costo de propiedad reducido*: Mientras que la inversión inicial requerida para una red inalámbrica puede ser más alta que el costo en hardware de una alámbrica, la inversión de toda la instalación y el costo del ciclo de vida puede ser significativamente inferior. Los beneficios y costos a largo plazo son superiores en ambientes dinámicos que requieren acciones y movimientos frecuentes.

- *Escalabilidad*: Los sistemas de WLANs pueden ser configurados en una variedad de topologías para satisfacer las necesidades de las instalaciones y aplicaciones específicas. Las configuraciones son muy fáciles de cambiar y además es muy fácil la incorporación de nuevos usuarios a la red.

2.4. Aplicaciones de los sistemas WLAN

Las aplicaciones más típicas de las redes inalámbricas de área local que podemos encontrar actualmente son las siguientes:

- Implementación de WLAN en edificios históricos, de difícil acceso y en general en entornos donde la solución cableada es no viable.

- Posibilidad de reconfiguración de la topología de la red sin añadir costos adicionales. Esta solución es muy típica en entornos cambiantes que necesitan una estructura de red flexible que se adapte a estos cambios.

- Redes locales para situaciones de emergencia o congestión de la red cableada.

- WLAN permiten el acceso a la información mientras el usuario se encuentra en movimiento. Habitualmente esta solución es requerida en hospitales, fábricas, almacenes...

- Generación de grupos de trabajo eventuales y reuniones Ad-Hoc. En estos casos no valdría la pena instalar una red cableada. Con la solución inalámbrica es viable implementar una red de área local aunque sea para un plazo corto de tiempo.

- En ambientes industriales con severas condiciones ambientales este tipo de redes sirve para interconectar diferentes dispositivos y máquinas.

- Interconexión de redes de área local que se encuentran en lugares físicos distintos. Por ejemplo, se puede utilizar una red de área local inalámbrica para interconectar dos o más redes de área local cableada situadas en dos edificios distintos.

2.5. Tecnologías

Existen varias tecnologías utilizadas en redes inalámbricas. El empleo de cada una de ellas depende mucho de la aplicación. Cada tecnología tiene sus ventajas y desventajas. A continuación se listan las más importantes en este género.

- **Infrarrojo:** Los sistemas de comunicación por infrarrojo utilizan muy altas frecuencias, justo abajo del espectro de la luz visible para transportar datos. Como la luz, el infrarrojo no puede penetrar objetos opacos, ya sea directamente (línea de vista) o indirectamente (tecnología difundida/reflectiva). El alto desempeño del infrarrojo directo es impráctico para usuarios móviles pero su uso es prácticamente para conectar

dos redes fijas. La tecnología reflectiva no requiere línea de vista pero está limitada a cuartos individuales en zonas relativamente cercanas.

- **Banda Angosta:** Un sistema de radio de banda angosta transmite y recibe información en una radio frecuencia específica. La banda amplia mantiene la frecuencia de la señal de radio tan angostamente posible para pasar la información. El cruzamiento no deseado entre canales es evitado al coordinar cuidadosamente diferentes usuarios en diferente canal de frecuencia. En un sistema de radio la privacidad y la no-interferencia se incrementa por el uso de frecuencias separadas de radio. El radio receptor filtra todas aquellas frecuencias que no son de su competencia. La desventaja de esta tecnología es el uso amplio de frecuencias, uno para cada usuario, lo cual es impráctico si se tienen muchos.
- **Espectro Extendido:** Está diseñado para intercambiar eficiencia en ancho de banda por confiabilidad, integridad y seguridad. Es decir, más ancho de banda es consumido con respecto al caso de la transmisión en banda angosta, pero el 'trueque' [ancho de banda/potencia] produce una señal que es en efecto más fuerte y así más fácil de detectar por el receptor que conoce los parámetros de la señal de espectro extendido que está siendo difundida. Si el receptor no está sintonizado a la frecuencia correcta, una señal de espectro extendido se miraría como ruido en el fondo. Otra característica del espectro disperso es la reducción de interferencia entre la señal procesada y otras señales no esenciales o ajenas al sistema de comunicación. Existen dos tipos de señales de Espectro Extendido:

    - *Espectro extendido con salto en frecuencia (FHSS)* utiliza una portadora de banda angosta que cambia la frecuencia en un patrón conocido tanto por el transmisor como por el receptor. Tanto el transmisor como el receptor están debidamente sincronizados comunicándose por un canal que está cambiado a cada momento en frecuencia. FHSS es utilizado para distancias cortas, en aplicaciones por lo general punto a multipunto, donde se tienen una cantidad de receptores dispersados en un área relativamente cercana al punto de acceso.

    - *Espectro extendido en secuencia directa (DSSS)* genera un patrón de bits redundante para cada bit que sea transmitido. Este patrón de bit es llamado código chip. Entre más grande sea este chip, es más grande la probabilidad de que los datos originales puedan ser recuperados (pero, por supuesto se requerirá más ancho de banda). Más sin embargo si uno o mas bits son dañados durante la transmisión, técnicas estadísticas abstraídas dentro del radio transmisor podrán recuperar la señal original sin necesidad de retransmisión. DSSS se utilizará comúnmente en aplicaciones punto a punto.

2.6. Organismos

Debido al gran crecimiento de las redes inalámbricas han surgido nuevas organizaciones en esta industria tales como alianzas, consorcios y foros, las cuales se encargan de proponer estándares y definir nuevas tecnologías. Se pueden dividir estas organizaciones en tres categorías: alianzas de tecnología, organizaciones de estándares y asociaciones de la industria.

Alianzas de tecnología: Está formada para introducir al mercado una tecnología o protocolo específico y proveer interoperatibilidad y certificación de productos de diferentes compañías que utilizan esa tecnología o protocolo. Ejemplos de este tipo de organizaciones están las siguientes:

- *Bluetooth SIG* [14]: basado en la especificación BluetoothTM especificación que utiliza la tecnología de radio para proveer conectividad a Internet a bajo costo a computadoras portátiles, teléfonos móviles u otros dispositivos portátiles.

- *HiperLAN1, HiperLAN Alliance e HiperLAN2 Global Forum* [15]: organizaciones europeas que utilizan enlaces de radio de alto desempeño a frecuencias en el rango de 5 GHz.

- *HomeRF* [16]: Basada en una especificación para comunicaciones inalámbricas en hogares conocida por sus siglas en inglés SWAP (Shared Wireless Access Protocol). El HRFWG (HomeRF Working Group) fue fundado para proveer los cimientos para un amplio rango de dispositivos al establecer una especificación abierta a la industria para comunicaciones digitales inalámbricas entre PCs y dispositivos domésticos alrededor de los hogares.

- *OFDM*: Esta organización está basada básicamente en una tecnología patentada conocida como W-OFDM (Wide-band Orthogonal Frecuency División Multiplexing).

- *WLI forum*: WLIF estableció un estándar interoperable en 1996 conocido como OpenAir, el estándar está disponible a cualquier compañía que se une al Forum. OpenAir es una tecnología de espectro extendio con salto en frecuencia a 2.4 GHz.

- *WECA*: La misión de la WECA (Wireless Ethernet Compatibillity Alliance) es certificar la interoperatibilidad del estándar conocido como Wi-FiTM que es una versión de alta velocidad del estándar 802.11b de la IEEE.

Organizaciones de estándares: Este tipo de organizaciones crean, definen y proponen estándares internacionales oficiales abiertos a la industria a través de un proceso abierto a todas las compañías. Ejemplos de estas organizaciones:

- *IEEE* (Institute of Electrical and Electronics Engineers)
- *ETSI* (European Telecommunications Standards Isntitute)

Asociaciones de la industria: Estás organizaciones son creadas para promover el crecimiento de la industria a través de educación y promoción, proveyendo información objetiva sobre la industria en general, tecnologías, tendencias, organizaciones, oportunidades independientemente de la tecnología. La organización más importante en esta categoría es la WLANA (Wireless LAN Association) cuya misión es ayudar y fomentar el crecimiento de la industria a través de la educación que puede ser caracterizada por asociaciones industriales y comerciales.

Organizaciones como estás promueven la competencia y avances tecnológicos lo cual significa mejores soluciones para los usuarios de redes inalámbricas e incrementar el crecimiento de la industria. La fuerza del mercado decidirá el valor de cada organización.

2.7. Futuro de los sistemas WLAN

Los fabricantes de WLANs migraron de la banda de 900 MHz a la banda de 2.4 GHz para mejorar la velocidad de información. Este patrón continua al abrirse el estándar IEEE 802.11a en la banda de 5.7 GHz operando con una velocidad de datos de hasta 54 Mbps. Esta banda de 5.7 GHz promete otras mejoras en velocidad permitiendo quizá algún día romper la barrera de los 100 Mbps.

Otras tecnologías para redes inalámbricas también han emergido paralelamente a las definidas por la IEEE 802.11x [17, 18], tales como Bluetooth [14], HomeRF [16], LMDS (Local Multipoint Distribution Service), WLL (Wireless Local Loop), también la entrada de nuevos protocolos, lenguajes y esquemas de seguridad ha sido de gran importancia en el avance de las redes inalámbricas tales como WAP (Wireless Application Protocol) [19], WML (Wireless Markup Language), WEP (Wired Equivalent Privacy), entre otros.

Con relación al costo los equipos de WLANs han abierto nuevos mercados. Para esta tecnología, la demanda continua incrementándose, la reducción del costo en la ingeniería y eficiencia en la fabricación permitirán la reducción mas de los costos, hasta que llegue un día en que un adaptador de un cliente inalámbrico cueste lo mismo que un adaptador alámbrico. Si tomamos en cuenta el cableado y el costo de mano de obra que involucra instalar una red alámbrica, esta diferencia será muy poca entre ambas tecnologías.

2.8. Configuraciones WLAN

El grado de complejidad de una red de área local inalámbrica es variable, dependiendo de las necesidades a cubrir y en función de los requerimientos del sistema que se desean implementar se puede utilizar diversas configuraciones de red.

2.8.1. Peer to Peer o Redes Ad-Hoc

La configuración más básica es la llamada *de igual a igual* o *ad-hoc* y como muestra la figura 2.1 consiste de una red con dos terminales móviles equipados con la correspondiente tarjeta adaptadora para comunicaciones inalámbricas. Para que la comunicación entre estas dos estaciones sea posible hace falta que se vean mutuamente de manera directa, es decir, que cada una de ellas esté en el rango de cobertura radioeléctrica de la otra. Las redes de tipo ad-hoc son muy sencillas de implementar y no requieren ningún tipo de gestión administrativa.

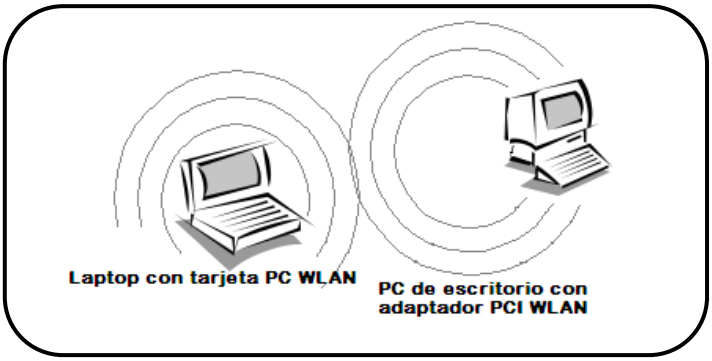


Figura 2.1. Redes Ad-hoc.

2.8.2. Modo Infraestructura

Para aumentar el alcance de una red del tipo anterior hace falta la instalación de un *punto de acces*o, como muestra la figura 2.2. Con este nuevo elemento se dobla el alcance de la red inalámbrica (ahora la distancia máxima permitida no es entre estaciones, sino entre cada

estación y el punto de acceso). Además, los *puntos de acceso* se pueden conectar a otras redes, y en particular a una red fija, con lo cual un usuario puede tener acceso desde su terminal móvil a otros recursos. Para dar cobertura en una zona determinada habrá que instalar varios puntos de acceso de tal manera que podamos cubrir la superficie necesaria con las celdas de cobertura que proporciona cada punto de acceso y ligeramente solapadas para permitir el paso de una celda a otra sin perder la comunicación.

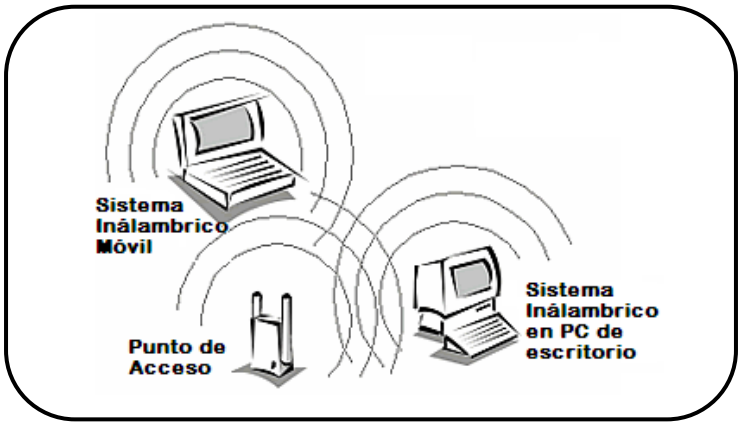


Figura 2.2. Redes Modo Infraestructura.

2.8.3. Enlace entre varias LAN o WMAN

Otra de las configuraciones de red posibles es la que incluye el uso de antenas direccionales. El objetivo de estas antenas direccionales es el de enlazar redes que se encuentran situadas geográficamente en sitios distintos tal y como se muestra en la figura 2.3.

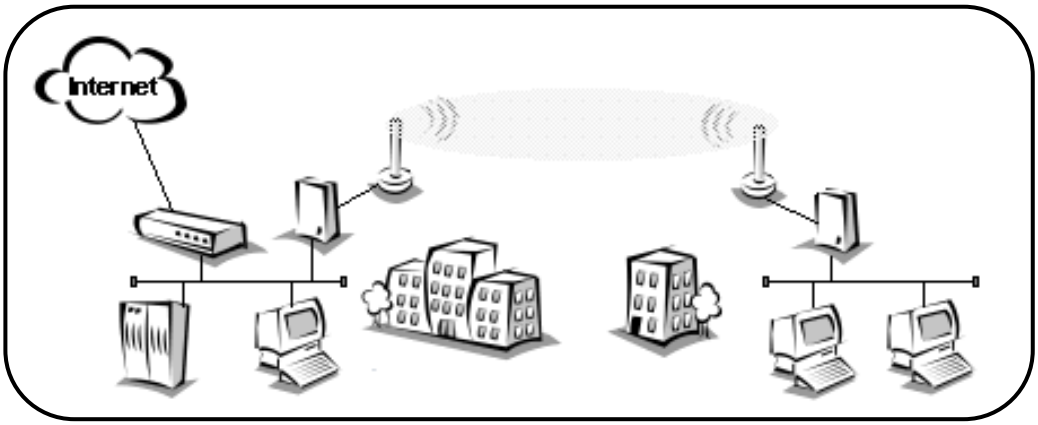


Figura 2.3. Enlace entre LAN y WLAN.

Un ejemplo de esta configuración se tiene en el caso en que una red local en un edificio y se desea extender a otro edificio. Una posible solución a este problema consiste en instalar una antena direccional en cada edificio apuntándose mutuamente. A la vez, cada una de estas antenas está conectada a la red local de su edificio mediante un punto de acceso. De esta manera podemos interconectar las dos redes locales.

CAPÍTULO 3

3. ESTÁNDAR IEEE 802.11

El propósito del estándar IEEE 802.11 [1] es el de proveer conectividad inalámbrica a maquinaria automatizada, equipo de cómputo o estaciones que requieren una distribución rápida que pueden ser portátiles, o que pueden montarse en vehículos móviles dentro de un área local. Un aspecto importante a considerar cuando se diseño este estándar fue que tenía que servir a unidades portátiles y móviles. Una unidad portátil es aquella que se mueve de un lugar a otro, pero permanece fija mientras envía y recibe información. Una unidad móvil es la que mientras está recibiendo o enviando información está en constante cambio.

El estándar IEEE 802.11 es parte de una familia de estándares para redes LAN, el cual sólo define dos capas OSI (Open Systems Interconection): Capa Física y Capa de Enlace de Datos, esta última dividida en subcapa de Control del Enlace Lógico (LLC) y subcapa de Control del Acceso al Medio (MAC).

A nivel de capa física, el estándar define métodos de transmisión compatibles con equipos de comunicaciones que utilizan el aire, radio o infrarrojo como medio de transmisión. El medio de transmisión constituye el soporte físico a través del cual emisor y receptor pueden comunicarse en un sistema de transmisión de datos.

A nivel MAC el estándar define dos métodos de acceso al medio, uno centralizado (Función de Coordinación Puntual, PCF) y otro distribuido (Función de Coordinación Distribuida, DCF) el cual es implementado obligatoriamente en todas las estaciones por medio del protocolo de Acceso Múltiple por Sensado de Portadora Evitando Colisiones (CSMA/CA) y Request To Send/Clear To Send (RTS/CTS)

Específicamente, este estándar:

- Describe las funciones y servicios que requiere cumplir un dispositivo para operar en redes inalámbricas del tipo Ad-Hoc, al igual que los aspectos de estaciones móviles (transición) dentro de esas redes.
- Define los procedimientos de la MAC para soportar los servicios de entrega asíncronos de MSDUs (*MAC* Service Data Unit).
- Define varias técnicas de señalización y funciones de interfaz de la capa física que son controladas por la IEEE 802.11 MAC.
- Permite la operación de un dispositivo (que cumple con el estándar) en múltiples redes inalámbricas que se solapen.
- Describe los requerimientos y procedimientos para ofrecer confidencialidad de la información del usuario que se transmite sobre los medios inalámbricos y autenticación de los dispositivos que cumplen con el IEEE 802.11.

3.1. Arquitectura en Capas

El estándar IEEE 802.11 es parte de una familia de estándares para redes LAN, el cual sólo define dos capas OSI como se muestra en la figura 3.1: Capa Física y Capa de Enlace de Datos, esta última dividida en subcapa de Control del Enlace Lógico (LLC) y subcapa de Control del Acceso al Medio (MAC). La Capa de superior (Aplicación) esta abierta a implementaciones y propuestas.

La capa física tiene como función codificar y decodificar las señales, la transmisión y recepción de bits, la sincronización. Además a este nivel, el estándar define métodos de transmisión compatibles con equipos de comunicaciones que utilizan el aire, radio o infrarrojo como medio de transmisión.

El estándar IEEE 802.11 se ha desarrollado en varias etapas. La primera llamada simplemente IEEE 802.11 incluye la capa MAC y la especificación de tres capas físicas, dos en la banda de 2.4 GHz, la primera utiliza Espectro Extendido con Salto en Frecuencia (Frequency Hopping Spread Spectrum, FHSS), la segunda capa física utiliza Espectro Extendido en Secuencia Directa (Direct Sequence Spread Spectrum, DSSS) y la última utiliza el infrarrojo, todas operando a 1 y 2 Mbps. El IEEE 802.11a opera en la banda de 5 GHz utilizando Frecuencia Ortogonal (Orthogonal Frequency Division Multiplexing, OFDM) a una velocidad de hasta 54 Mbps. El IEEE 802.11b opera en la banda de 2.4 GHz utilizando DSSS a 11 Mbps. El IEEE 802.11g opera en la banda de 2.4 GHz utilizando OFDM y DSSS hasta 54 Mbps. En proceso de estandarización se encuentran las versiones 802.11x [17, 18] y 802.11i [12, 13].

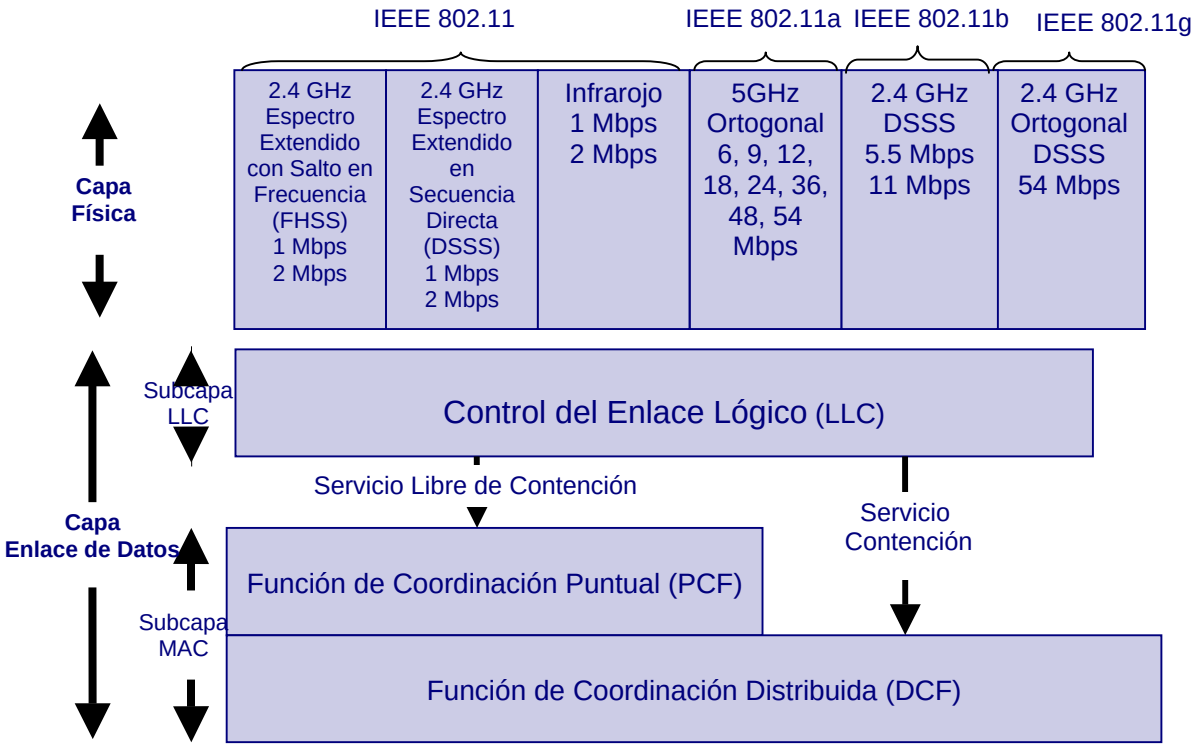


Figura 3.1. Arquitectura en Capas del IEEE 802.11.

La capa física de servicios consiste en dos protocolos:

- Una función de convergencia de capa física, que adapta las capacidades del sistema físico dependiente del medio (PMD). Esta función es implementada por el protocolo PLCP o *Procedimiento de Convergencia de Capa Física*, que define una forma de mapear MPDUs o unidades de datos MAC en un formato de tramas susceptibles de ser transmitidas o recibidas entre diferentes estaciones o STASs a través de la capa PMD.

- Un sistema PMD, cuya función define las características y un medio de transmitir y recibir a través de un medio sin cables entre dos o más STAs.

La comunicación entre MACs de diferentes estaciones se realizará a través de la capa física mediante de una serie de puntos de acceso al servicio, donde la capa MAC invocará las primitivas de servicio.

Además de estas capas, podemos distinguir la capa física de gestión. En esta capa podemos distinguir la estructura MIB (Management Information Base) que contienen por definición las variables de gestión, los atributos, las acciones y las notificaciones requeridas para gestionar una estación. Consiste en un conjunto de variables donde podemos especificar o contener el estado y la configuración de las comunicaciones de una estación.

La Capa de Enlace de Datos se divide en dos subcapas MAC y LLC, las cuales se explican a continuación:

- *Control de Acceso al Medio* (Medium Access Control, MAC) tiene como funciones la división y reensamblado de tramas, la detección de errores, gobernar el acceso al medio. Además a este nivel, el estándar define dos métodos de acceso al medio:

    - Centralizado (Función de Coordinación Puntual, PCF).

    - Distribuido (Función de Coordinación Distribuida, DCF) el cual es implementado obligatoriamente en todas las estaciones por medio del protocolo de Acceso Múltiple por Sensado de Portadora Evitando Colisiones (CSMA/CA) y Request To Send/Clear To Send (RTS/CTS).

- *Control del Enlace Lógico* (Logical Link Control, LLC) ofrece una interfaz a las capas superiores para el control de flujo y control de errores.

Obsevación únicamente para la capa física del IEEE 802.11g es obligatorio el uso de PCF y DCF.

3.2. Arquitectura en Componentes

La arquitectura del estándar IEEE 802.11 consiste de varios componentes que interactúan para proveer una WLAN (Redes Inalámbricas tipo LAN) que soporta la movilidad de estaciones de una forma totalmente transparente para las capas superiores.

Una *estación* (STA) es un dispositivo que soporta el uso de este estándar. Un *Punto de Acceso* (AP) es una estación que provee acceso al *Sistema de Distribución* (DS) por medio de los servicios de Sistema de Distribución, además de actuar como una simple estación. La información se mueve entre un *Conjunto de Servicios Básicos* (BSS) y un DS a través de un AP.

El DS y los BSSs permiten crear una red inalámbrica de tamaño y complejidad arbitraria. El estándar se refiere a esta clase de red como Red de *Conjunto de Servicios Extendidos* (ESS). Un *Portal* integra la arquitectura IEEE 802.11 con una LAN tradicional. Estos conceptos se pueden apreciar de forma gráfica en la figura 3.2.

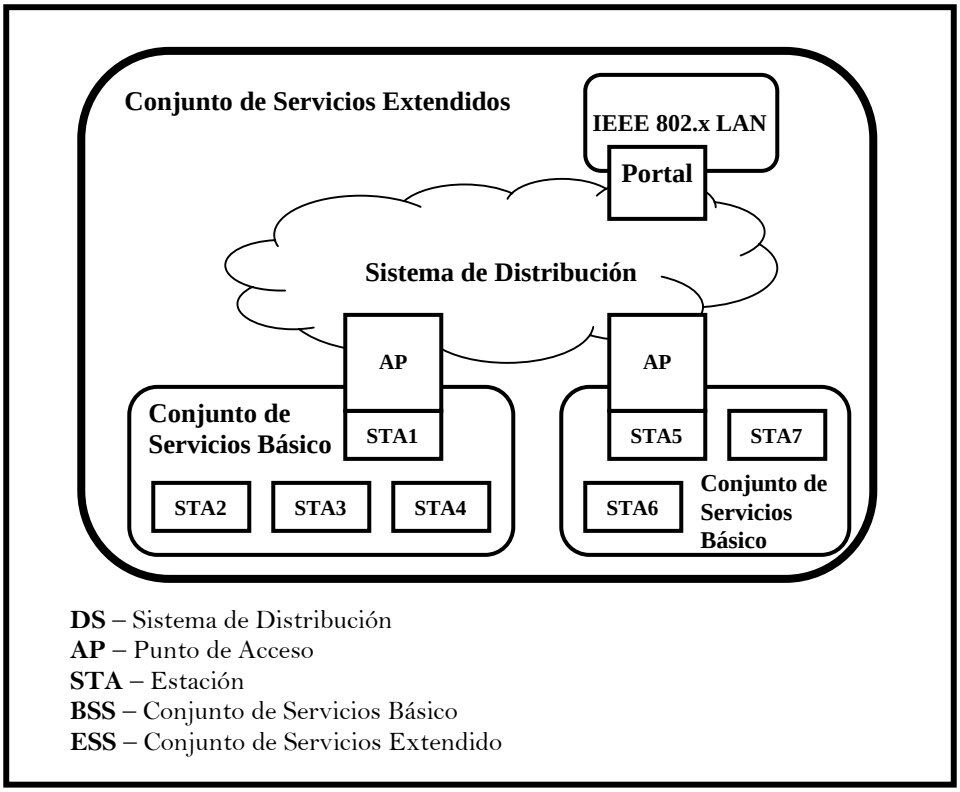


Figura 3.2. Componentes IEEE 802.11.

3.3. Servicios

Existen nueve servicios especificados por el IEEE 802.11. Seis de estos servicios son usados para soportar la entrega de MSDUs *(MAC Service Data Unit)* entre STAs y los tres restantes para controlar el acceso a la LAN IEEE 802.11 y la confidencialidad.

Cada uno de estos servicios hace uso de uno, o más de uno, segmentos MAC. Algunos de estos servicios son Mensajes de Administración MAC y otros Mensajes de Datos MAC.

La subcapa MAC de este estándar usa tres tipos de mensajes: datos, administración y control.

La distribución de mensajes con el ***Sistema de Distribución (DS)*** esta compuesto por dos servicios: *Distribución* e *Integración*.

Los servicios que soporta el ***Servicio de Distribución*** son: *Asociación*, *Reasociación*, *Desasociación* y *Tipos de Movilidad*. El propósito principal de la subcapa MAC es la

transferencia de MSDUs *(MAC Service Data Unit)* entre entidades MAC. La información requerida para que el servicio de distribución funcione es ofrecida por el servicio de asociación. Antes de que un mensaje de datos pueda ser manejado por el Sistema de Distribución, la estación debe estar asociada.

Los ***Servicios de Confidencialidad y Control de Acceso*** son ofrecidos para proveer a las redes inalámbricas de los servicios de seguridad básicos equivalentes a los que las redes alámbricas ofrecen: *Autenticación*, *Deautenticación* y *Confidencialidad*

3.3.1. Asociación

Para entregar un mensaje dentro de un DS, el servicio de distribución debe saber cual AP debe acceder para poder llegar a la STA. Esta información es provista por el DS mediante el concepto de asociación. La asociación es necesaria, pero no suficiente para soportar transición BSS. La asociación es suficiente para soportar movilidad sin transición. Antes de que un STA pueda enviar un mensaje a través de un AP, primero debe asociarse con el AP. En cualquier instante dado, una STA debe estar asociada con no más de un AP. Un AP puede estar asociado con varios STAs al mismo tiempo. Este servicio es un DSS.

3.3.2. Autenticación

En LAN alámbricas, la seguridad física puede ser usada para prevenir accesos no autorizados. Esto es impráctico en las redes inalámbricas ya que el medio no tiene límites precisos. El estándar IEEE 802.11 provee la habilidad de controlar accesos a la LAN por medio de servicios de autenticación. El servicio es usado por todas las estaciones para establecer su identidad ante las estaciones con las cuales se comunica. Si no es establecido un nivel de autenticación mutuo la asociación no se realiza. Este servicio es un SS.

Este estándar soporta autenticación de llave compartida implementado en el algoritmo WEP (Wired Equivalent Privacy).

3.3.3. Confidencialidad

En una red LAN alámbrica, sólo aquellas estaciones físicamente conectadas al cable pueden oír el tráfico. Con un medio inalámbrico compartido, este no es el caso, cualquiera que conozca el rango de frecuencia puede escuchar el tráfico, por lo que una conexión de este tipo degrada en mucho la seguridad de una conexión.

Para evitar este problema, este estándar ofrece la posibilidad de cifrar el contenido de un mensaje a través del servicio de confidencialidad, que es un SS. Este proceso se realiza con ayuda del algoritmo WEP.

3.3.4. Deautenticación

El servicio de deautenticación se invoca siempre que una autenticación se va a terminar. En un ESS, ya que la autenticación es un pre-requisito para la asociación, el acto de deautenticar provocará que la STA sea desasociada. La deautenticación puede ser invocada por cualquier entidad autenticada, sea un AP o una STA. Este servicio no es una petición, es una

notificación; no puede ser rehusada por ninguna parte, cuando se notifica es porque ya se ha realizado. Este servicio es un SS.

3.3.5. Desasociación

El servicio de Desasociación es invocado siempre que una asociación existente necesite terminarse. Puede ser invocado por cualquiera que pertenezca a una asociación (STAs que no son AP y APs). La Desasociación es una notificación, no una petición, por lo que esta no puede ser rechazada por ninguna de las partes.

Los APs pueden necesitar disasociarse de las STAs para permitir que el AP se elimine de la red. Este servicio es un DSS.

3.3.6. Distribución

Este es el servicio primario usado por las STAs. Es conceptualmente invocado por cada mensaje de datos que se envía o se recibe entre STAs dentro de un ESS (cuando el paquete es enviado por el DS). La distribución se lleva a cabo mediante DSSs.

3.3.7. Integración

Si el servicio de distribución determina que el receptor de un mensaje es miembro de una LAN integrada, el punto de salida del DS debe ser un portal en vez de un AP. Los mensajes se distribuyen a un portal porque el DS invoca una función de integración. La función de integración es responsable de llevar a cabo todo lo necesario para entregar un mensaje desde el DSM al medio de la LAN integrada. Los mensajes recibidos desde una LAN integrada a través de un portal por el DS para una STA, invocarán la función de integración antes de que el mensaje se distribuya por el servicio de distribución. Los detalles de la función de integración son dependientes de la implementación de un DS específico y se encuentran fuera del ámbito del estándar IEEE 802.11.

3.3.8. Reasociación

La asociación es suficiente para el envío de mensajes sin transición entre STAs. Se necesita funcionalidad extra para poder soportar movilidad de transición BSS. La funcionalidad adicional requerida es ofrecida por el servicio de reasociación. La reasociación es invocada para "mover" una asociación de un AP a otro. Este servicio también permite cambiar los atributos de una asociación ya establecida. Este servicio siempre es iniciado por las estaciones. Este servicio es un DSS.

3.3.9. Tipos de Movilidad

Existen tres tipos de transición que describen la movilidad de estaciones dentro de una red:

- *Sin transición*: En este tipo, se reconocen dos subclases que son:
    - *Estático*: sin movimiento.

- *Movimiento local*: un movimiento dentro del rango físico de las STAs, es decir, movimiento dentro del Área Básica de Servicio (BSA).

- *Transición BSS*: Este tipo se define como el movimiento de una estación de un BSS dentro de un ESS a otro BSS dentro del mismo ESS.

- *Transición ESS*: Este tipo se define como el movimiento de una estación de un BSS en un ESS a otro BSS en otro ESS.

3.4. Interfaces de Servicio

Este estándar permite que el Sistema de Distribución sea creado por diferentes tecnologías inalámbricas, no solo la del mismo estándar. Es por esto que dentro del IEEE 802.11 no detalla explícitamente la implementación de un DS, sólo especifica los servicios que este debe ofrecer. Estos servicios están asociados con diferentes componentes de la arquitectura.

Existen dos categorías de servicio: los Servicios de Estación (Station Services, SS) y los Servicios del Sistema de Distribución (Distribution System Services, DSS), ambas categorías son usadas por la subcapa MAC.

El conjunto completo de servicios arquitectónicos que ofrece el estándar 802.11 son: Autenticación, Asociación, Deautenticación, Desasociación, Distribución, Integración, Confidencialidad, Reasociación y Entrega de MSDU's.

Este conjunto esta dividido en dos grupos: unos son parte de las STAs y otros del DS.

3.4.1. Servicios de Estación (SS)

Los servicios ofrecidos por las estaciones son conocidos como *Servicios de Estación (SS).* Los SS están presentes en cada estación IEEE 802.11 (incluyendo los APs, ya que también son considerados estaciones) y son usados por las entidades la subcapa MAC. Los SS son los siguientes: Autenticación, Deautenticación, Confidencialidad y Entrega de MSDU's.

3.4.2. Servicios del Sistema de Distribución (DSS)

Los servicios ofrecidos por los Sistemas de Distribución son conocidos como *Servicios del Sistema de Distribución.*

Los DSS son ofrecidos por los Sistemas de Distribución. Estos son accedidos a través de una estación que también provee los DSS's. Una estación que ofrece acceso a los DSS en un Punto de Acceso. También son usados por las entidades de la subcapa MAC. Los DSS son los siguientes: Asociación, Desasociación, Distribución, Integración y Reasociación.

3.5. Servicios de Seguridad

Para hacer frente a las amenazas a la seguridad del sistema se definen una serie de servicios para proteger los sistemas de proceso de datos y de transferencia de información de una organización. Estos servicios hacen uso de uno o varios mecanismos de seguridad. A continuación se describen los servicios de seguridad definidos por el ISO (International Organization for Standardization) y los servicios de seguridad ofrecidos por el estándar IEEE 802.11.

3.5.1. Servicios de Seguridad definidos por el ISO

Los servicios de seguridad definidos por el ISO son:

- Autenticación. Su función es verificar la supuesta identidad de un usuario o sistema.
- Control de Acceso. Protege los recursos del sistema de usuarios no autorizados.
- Confidencialidad. Consiste en proteger los datos antes de enviarlos a través del medio o canal para evitar la interpretación de su significado por terceros no autorizados.
- No-Rechazo. Protege al emisor/receptor no niege a ver enviado/recibido información.
- Integridad. Su función es proteger los datos contra modificaciones, inserciones y borrados no autorizados.

3.5.2. Servicios de Seguridad definidos por el Estándar IEEE 802.11

Mientras que los servicios de seguridad definidos por el estándar IEEE 802.11 son: Autenticación, Confidencialidad e Integridad.

Para la ***autentificación***, el estándar especifica dos modalidades:

- *OSA (Autentificación de Sistema Abierto).* Es un algoritmo nulo y más simple de autenticación. Cualquier estación que requiera autenticación con este algoritmo puede ser autenticada si la estación esta activada para este tipo de autenticación. Este tipo de autenticación no garantiza el éxito del proceso autenticación de una estación a otra, ya que en cualquier momento la estación puede rechazar autenticar a otra estación.
- *Autenticación de Llave Compartida.* Soporta la autenticación de estaciones que conozcan la llave privada o bien que no la conozcan. Si la estación no conoce la llave la otra estación se la enviará cifrándola mediante el algoritmo WEP, para que se pueda realizar el proceso. Para autentificar una estación, el AP envía un desafío en texto claro a la estación, que esta devuelve cifrada usando la llave compartida. A su vez el AP también realiza la misma operación y compara ambos resultados. En caso de que ambos coincidan se permite el acceso a la estación.

Para la ***confidencialidad*** el estándar IEEE 802.11 a nivel de la subcapa MAC define el protocolo *WEP*, la utilización es opcional. El WEP es definido para proteger la información de los usuarios autorizados de una WLAN de escuchas externos. Este servicio provee seguridad en WLAN equivalentes a los de las redes alámbricas. Su principal función es proveer mecanismos de seguridad en el flujo de datos en redes inalámbricas.

Las principales metas del WEP son:

- Denegar acceso a la red a usuarios no autorizados que no posean la llave WEP apropiada.
- Prevenir la decodificación de trafico WLAN capturado sino se posee la llave WEP.
- Con WEP activado, el transmisor toma el contenido de una trama de datos y usa el algoritmo de codificación. Entonces reemplaza la trama original con la información cifrada. Las tramas de datos que son cifradas se envían con un bit en alto en el campo de control de las cabeceras MAC para indicar el cifrado WEP. El receptor de una trama de datos cifrada descifra el cuerpo de la trama usando el mismo algoritmo de cifrado usado por el transmisor. El resultado es el cuerpo de datos original, el cual es enviado a los protocolo de las capas superiores.

La confidencialidad de los datos depende del Servicio de Administración de Llave Externa para distribuir las llaves (que se comparte entre una STA y un AP) de los datos cifrados/descifrados.

El protocolo WEP es el que cifra los paquetes de datos en texto claro antes de su transmisión por el canal. Para ello se usa el cifrador RC4.

Al texto en claro se le aplican dos procesos: uno que cifra el texto y otro que lo protege de modificaciones no autorizadas mientras es transmitido.

Cada paquete se cifra con una llave distinta y como parte del proceso de cifrado, WEP prepara una semilla al concatenar la llave privada (cuya longitud es de 40 bits) con un Vector de Inicialización (IV, que viaja en texto claro en el paquete) de 24 bits generado de manera aleatoria, hay por tanto 16,8 millones de combinaciones posibles para cifrar un paquete con una misma llave WEP. El IV alarga la vida de la llave privada porque la estación puede cambiar el IV en cada trama transmitida. WEP da como entrada el IV al Generador de Número Pseudoaleatorio (PRNG) que produce un flujo de llave igual a la longitud de la trama más un Valor de Verificación de la Integridad (ICV) de 32 bits. Antes de que la transmisión se lleve a cabo, WEP combina todos los flujos con la operación XOR aplicada a cada bit, lo que produce el texto cifrado. El ICV es un código de verificación que la STA receptora recalcula eventualmente y lo compara con el enviado por la STA remitente para determinar si la transmisión de datos fue alterada.

WEP solo se cifra los campos de datos y el ICV. Este último utiliza el CRC de 32 bits para comprobar la ***integridad*** de los datos.

Cabe mencionar que CRC-32 sólo detecta alteraciones en la información cuando es accidental (clima, etc.) y no intencional (atacante).

Es importante destacar que la implementación de WEP es opcional según el estándar, y por ello inicialmente no se implementó en muchos equipos.

La figura 3.3 muestra el funcionamiento del protocolo WEP. Se tiene un emisor que en este caso es *Alicia* y un receptor que es *Beto*. *Alicia* y *Beto* de antemano se pusieron de acuerdo con la llave privada K_{AB}, la cual es utilizada para cifrar y descifrar la información.

Alicia tiene el *mensaje* a enviar el cual es procesado por *CRC-32* para generar el valor de chequeo de integridad *ICV*, este valor es anexado al *texto claro*.

El proceso de cifrado de cada paquete es el siguiente, se cuenta con una llave privada K_{AB} cuya longitud es de 40 bits la cual es concatenada con *IV* (Vector de Inicialización) la longitud del campo *IV* es de 24 bits consecutivamente se aplica cifrador por flujo *RC4* para obtener el *"Flujo de llave RC4"*, realizando una operación de or-exclusivo (*XOR*) al resultado anterior con el *texto claro* obteniendo el *texto cifrado* a transmitir.

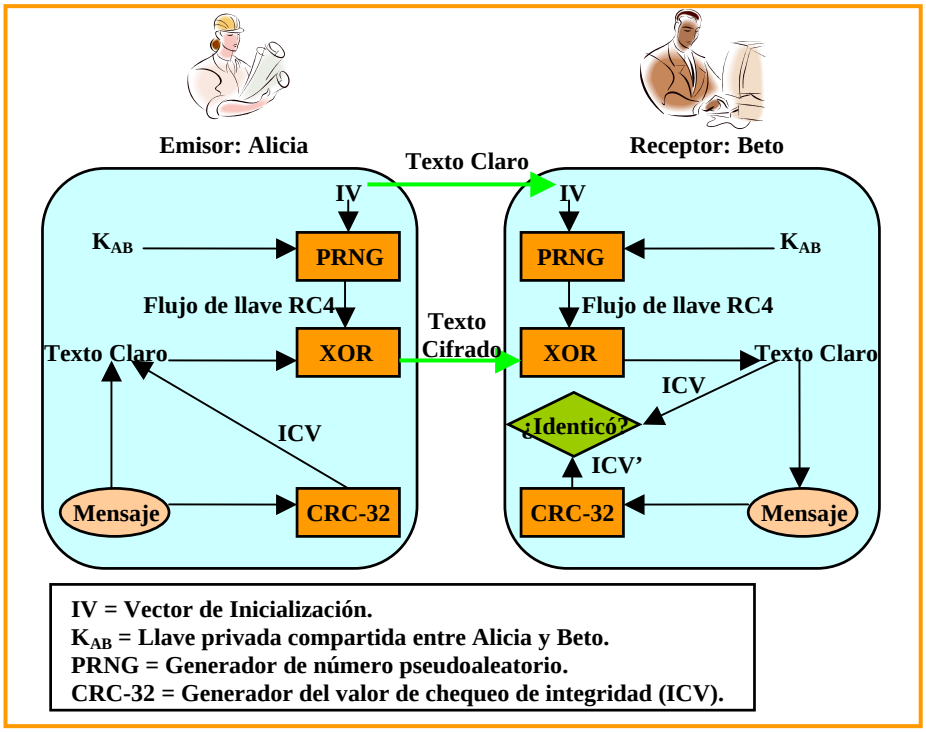


Figura 3.3. Funcionamiento del protocolo WEP.

Beto recibe el *IV* en *texto claro* y a su vez el *texto cifrado* para descifrar el mensaje recibido se realiza el proceso de descifrado. El proceso de descifrado de cada paquete es el siguiente, se cuenta con una llave privada K_{AB} la cual es concatenada con *IV* consecutivamente se aplica cifrador por flujo *RC4* para obtener el *"Flujo de llave RC4"*, realizando un *XOR* al resultado anterior con el *texto cifrado* obteniendo el *texto claro*.

Beto tiene el *mensaje* recibido el cual es procesado por *CRC-32* para generar el valor de chequeo de integridad *ICV'*, este valor es comparado con *ICV* que esta concatenado al *texto claro*, si son identicos entonces la integridad se cumplio en caso contrario la información sufrio alteraciones.

3.6. Ataques y vulnerabilidades en el protocolo WEP

El artículo *"WLAN Security: Currentand Future"* siendo los autores J. Park y D. Dicoi publicado en Septiembre-Octubre del 2003 [4]. Los autores describen las fallas encontradas en WEP. La seguridad del WEP sufré una pobre solución para la administración de llaves entre el Punto de Acceso (AP) y los dispositivos inalámbricos, debido a que pueden dejar la misma llave por largos períodos de tiempo. Si el dispositivo fue perdido o robado, un atacante podría utilizar la llave para comprometer ese dispositivo y cualquier otro que compartía la misma llave.

El artículo discute que el Vector de Inicialización (IV) viaja en la red en texto claro, lo cual permite que un atacante puede obtener los 24 bits de cada llave enviada. La longitud del IV nos da 16,8 millones de combinaciones posibles para cifrar un paquete con una misma llave WEP, siendo que estadísticamente cada 5 horas tiende a repetirse el mismo valor para IV.

Se menciona la utilización de CRC-32 para ofrecer el servicio de integridad, los autores destacan que CRC-32 no da un valor criptografico, lo que compromete fácilmente la integridad de los datos.

El artículo *"Weaknesses in the Key Scheduling Algorithm of RC4"* cuyos autores son Fluhrer, Mantin y Shamir publicado en Agosto del 2001 [5]. Los autores introducen un nuevo ataque sobre WEP que es pasivo, un ataque basado únicamente en el texto cifrado que es capaz de recuperar de forma completa la llave privada (no solamente el *"Flujo de llave RC4"* generado por un particular IV, sino que la verdadera llave) en un período relativamente corto de tiempo, alrededor de 4,000,000 paquetes. Además, el ataque crece de forma lineal sin importar la llave o el tamaño del IV.

El ataque hace uso de una falla en el protocolo WEP que permite a un atacante recolectar información acerca de los bytes de la llave dado cierto conocimiento del IV y el primer byte de salida. El primer byte en la mayoría, sino es que todas las transmisiones cifradas con WEP contiene 0XAA. Entonces, como el IV es transmitido de forma abierta como se especifica en 802.11, los atacantes tienen los dos requerimientos para ejecutar este ataque.

Aunque los autores de este artículo mencionan que ellos "no han intentado atacar una conexión WEP real, y que por lo tanto no afirman que WEP sea vulnerable a este ataque", los investigadores en AT&T en Agosto del 2001 [7] si implementaron el ataque contra WEP utilizando un NIC 802.11 de USD $100 en un cliente Linux. Ellos fueron capaces de recuperar la llave privada completa en alrededor de 5,000,000 paquetes, que representan más o menos tres horas en una red con carga mediana.

El artículo "*Unsafe at any key size; An analisys of the WEP encapsulation*" cuyo autor es J. R. Walker publicado en Octubre del 2000 [6]. El algoritmo RC4 utiliza una llave secreta previamente compartida de 40 bits y un vector de inicialización de 24 bits. Se ha probado que este tamaño de llave es inseguro, desafortunadamente parece ser que el tamaño de llave no es en sí la principal causa de la falla de WEP pues aunque se han hecho versiones con tamaños de llave mucho mayores la falla persiste. En este artículo se discute que la falla del WEP radica en el uso del IV. Esta vulnerabilidad evita que la encapsulación de WEP proporcione una eficiente seguridad. Se discuten cada uno de los problemas que son ocasionados por IV y el cifrador por flujo RC4.

El artículo "*Intercepting Mobile Communications: The Insecurity of 802.11"* cuyos autores son Nikita Borisov, Ian Goldberg, y David Wagner publicado en el año 2001 [20], también se le conoce como el *"Berkeley paper"*, este fue el primero en una serie de artículos que expusieron a detalle las vulnerabilidades del algoritmo criptográfico RC4 y de la forma en la que es usado en el estándar 802.11.

El artículo indica que la forma en que el RC4 es usado en WEP expone el protocolo a ataques pasivos y activos que permite espiar o modificar las transmisiones inalámbricas. Lo que hace posible estos ataques es el hecho de que el IV se transmite en texto claro. Vea la sección 3.5.2 para mayor información en el uso del IV en el WEP.

El foco principal del artículo de *Berkeley* es proveer que es posible descifrar información cifrada con WEP sin tener la llave privada. Al capturar dos transmisiones que usan el mismo IV un atacante puede cancelar de forma efectiva el *"Flujo de llave RC4"* haciendo el XOR de los dos textos cifrados. Esto, entonces, produce el XOR de los dos textos claros originales. Si se

conoce uno de los textos claros, entonces el otro puede ser deducido, así como el *"Flujo de llave RC4"* que fue utilizado para generar ambos. Se puede crear un diccionario que especifique el *"Flujo de llave RC4"* usado por cada IV. De esta forma, un atacante puede eventualmente descifrar todas las transmisiones en el medio inalámbrico sin conocer la llave privada.

Los autores demuestran que la reutilización del IV es casi imposible de evitar, porque el 802.11 especifica un tamaño de IV de 24 bits. Aún con un WEP de 128 bits, la situación no mejora, porque aunque la llave es ahora de 104 bits de longitud, el IV sigue siendo solamente de 24 bits tal como se especifica en el 802.11. Además de esto, muchos fabricantes reinician el IV a 0 cada vez que la tarjeta es reiniciada y lo incrementan uno para cada transmisión subsecuente. Esto trae la indeseable consecuencia de reutilizar muchos de los primeros valores del IV repetidamente.

El artículo *"Your Wireless Network Has No Clothes"* cuyos autores son Arbaugh, Shankar y Wan publicado en el año 2001 [21]. Los autores se enfocan en los protocolos utilizados para la autentificación y el control de acceso y marcan varios puntos que deben ser bastante obvios para cualquiera que esté familiarizado con el estándar 802.11 y haya trabajado con los productos.

Los autores de forma correcta apuntan que el SSID (Service Set Identifier) no sirve como mecanismo de seguridad. Porque es transmitido de forma abierta dentro de muchos de los marcos administrativos del 802.11, es muy sencillo utilizar un sniffer de red para capturar el SSID y obtener acceso a la WLAN. También de forma correcta señalan que ambos mecanismos de autentificación en la especificación, la autentificación abierta y la autentificación de llave compartida, son muy débiles. La autentificación abierta es esencialmente una autentificación "nula". Cualquier solicitud de autentificación de una estación inalámbrica hacia la WLAN será permitida. La autentificación de llave compartida, como se describe en la sección 3.5.2, es básicamente una autentificación de respuesta a un desafío o reto que permite al atacante determinar el *"Flujo de llave RC4"* utilizado para cifrar la respuesta y utilizar este mismo *"Flujo de llave RC4"* para obtener autentificación a la WLAN, aunque el texto de desafío se ha generado por el PRNG (Generador de número pseudoaleatorio) para cada intercambio de autentificación.

Los autores también comentan acerca de la inseguridad de las Listas de Control de Acceso (ACLs) encontradas en muchos productos hoy en día. La mayor parte de las ACLs se usan para restringir el acceso a una lista de direcciones MAC conocidas. Sin embargo, debido a que la mayoría de los adaptadores 802.11 permiten que su dirección MAC sea modificada por software, ésta es una forma muy débil de seguridad. Es un procedimiento relativamente simple el de sniffear una WLAN para encontrar direcciones MAC que tengan permitido el acceso, y después cambiar la dirección MAC del adaptador 802.11 para obtener acceso.

3.7. Control de Acceso al Medio (MAC)

Los diferentes métodos de acceso del IEEE 802.11 están diseñados según el modelo OSI y se encuentran ubicados en el nivel físico y en la parte inferior del nivel de enlace o subnivel MAC.

Además, la capa de gestión MAC controlará aspectos como sincronización y los algoritmos del sistema de distribución, que se define como el conjunto de servicios que precisa o propone el modo infraestructura.

3.7.1. Descripción Funcional MAC

La arquitectura MAC del estándar 802.11 [22] se compone de dos funcionalidades básicas: la *Función de Coordinación Distribuida* (DCF) y la *Función de Coordinación Puntual* (PCF).

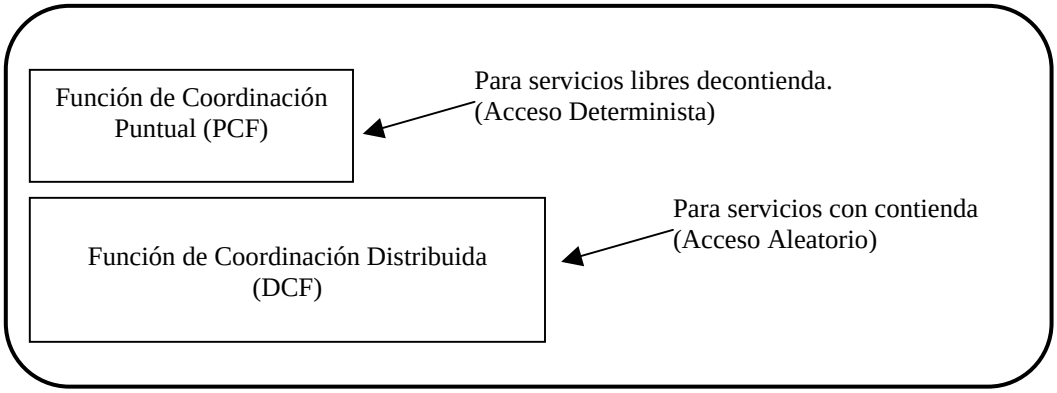


Figura 3.4. Arquitectura MAC.

3.7.2. Función de Coordinación Distribuida (DCF)

Definimos función de coordinación como la funcionalidad que determina, dentro de un conjunto básico de servicios (BSS), cuándo una estación puede transmitir y/o recibir unidades de datos de protocolo a nivel de la subcapa de Control de Acceso al Medio (MAC) a través del medio inalámbrico. En el nivel inferior de la subcapa MAC se encuentra la Función de Coordinación Distribuida y su funcionamiento se basa en técnicas de acceso aleatorias de contienda por el medio.

El tráfico que se transmite bajo esta funcionalidad es de carácter asíncrono ya que estas técnicas de contienda introducen retardos aleatorios, no predecibles y no tolerados por los servicios síncronos.

Las características de DCF son las siguientes:

- Utiliza el protocolo de Acceso Múltiple por Sensado de Portadora Evitando Colisiones (CSMA/CA).
- Utiliza Request To Send/Clear To Send (RTS/CTS).
- Utiliza el Algoritmo de Backoff.
- Necesario reconocimientos (ACKs), provocando retransmisiones si no se recibe.
- Usa campo Duration/ID que contiene el tiempo de reserva para transmisión y ACK. Esto quiere decir que todos los nodos conocerán al escuchar cuando el canal volverá a quedar libre.
- Implementa fragmentación de datos.
- Concede prioridad a tramas mediante el espaciado entre tramas (IFS).

- ❖ Soporta Broadcast y Multicast sin ACKs.

3.7.3. Protocolo de Acceso Múltiple por Sensado de Portadora Evitando Colisiones (CSMA/CA)

El algoritmo básico de acceso CSMA/CA [22, 23] a este nivel funciona tal y como se describe a continuación:

1.- Antes de transmitir información una estación debe sensar el medio, o canal inalámbrico, para determinar su estado (libre/ocupado).

2.- Si el medio no esta ocupado por ninguna otra trama la estación ejecuta una espera adicional llamada espaciado entre tramas (IFS).

3.- Si durante este intervalo temporal, o bien ya desde el principio, el medio se determina ocupado, entonces la estación debe esperar hasta el final de la transacción actual antes de realizar cualquier acción.

4.- Una vez finaliza esta espera debida a la ocupación del medio la estación ejecuta el llamado algoritmo de Backoff, el cual determina una espera adicional y aleatoria escogida uniformemente en un intervalo llamado Ventana de Contención (CW). El algoritmo de Backoff nos da un número aleatorio y entero de ranuras temporales y su función es la de reducir la probabilidad de colisión que es máxima cuando varias estaciones están esperando a que el medio quede libre para transmitir.

5.- Mientras se ejecuta la espera marcada por el algoritmo de Backoff se continúa escuchando el medio de tal manera que si el medio se determina libre durante un tiempo de al menos IFS esta espera va avanzando temporalmente hasta que la estación consume todas las ranura temporales asignadas. En cambio, si el medio no permanece libre durante un tiempo igual o superior a IFS el algoritmo de Backoff queda suspendido hasta que se cumpla esta condición.

Cada retransmisión provocará que el valor de CW, que se encontrará entre CWmin y CWmax se duplique hasta llegar al valor máximo. Por otra parte, el valor de la ranura de tiempo es de 20μseg.

La figura 3.5 muestra el funcionamiento del protocolo de acceso CSMA/CA. Se tiene la estación fuente, destino y otra. Las estaciones mencionadas realizan el proceso de acceso mediante el algoritmo descrito con anterioridad. En este caso la estación fuente encuentra el canal libre y transmite la información; Mientras que la última estación esta esperando a que el medio quede libre para poder acceder a esté.

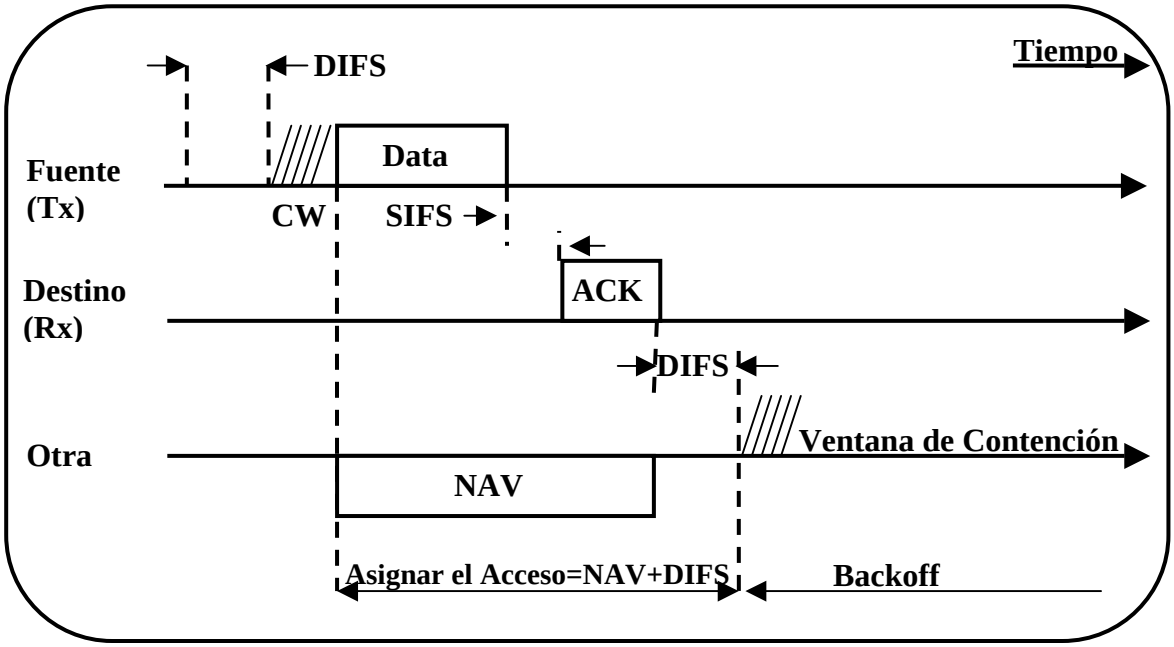


Figura 3.5. Protocolo de Acceso CSMA/CA.

3.7.4. Protocolo de Acceso Request To Send/Clear To Send (RTS/CTS)

Sin embargo, CSMA/CA en un entorno inalámbrico y celular presenta una serie de problemas. Los dos principales problemas que podemos detectar son:

- Nodos ocultos. Una estación cree que el canal está libre, pero en realidad está ocupado por otro nodo que no se oye.
- Nodos expuestos. Una estación cree que el canal está ocupado, pero en realidad está libre pues el nodo al que oye no le interferiría para transmitir a otro destino.

La solución que propone el estándar IEEE 802.11 es Request To Send/Clear To Send (RTS/CTS) [22, 23]. La figura 3.6 muetra el funcionamiento de este protocolo, antes de transmitir el emisor envía una trama RTS (Request to Send), indicando la longitud de datos que quiere enviar. El receptor le contesta con una trama CTS (Clear to Send), reservando el canal para los datos. Al recibir el CTS, el emisor envía sus datos (Data). Una vez que finalizó la transferencia de información se envía un reconocimiento (ACK) a la estación destino. Los nodos seguirán una serie de normas para evitar los nodos ocultos y expuestos:

- Al escuchar un RTS, hay que esperar un tiempo por el CTS.
- Al escuchar un CTS, hay que esperar según la longitud de datos.

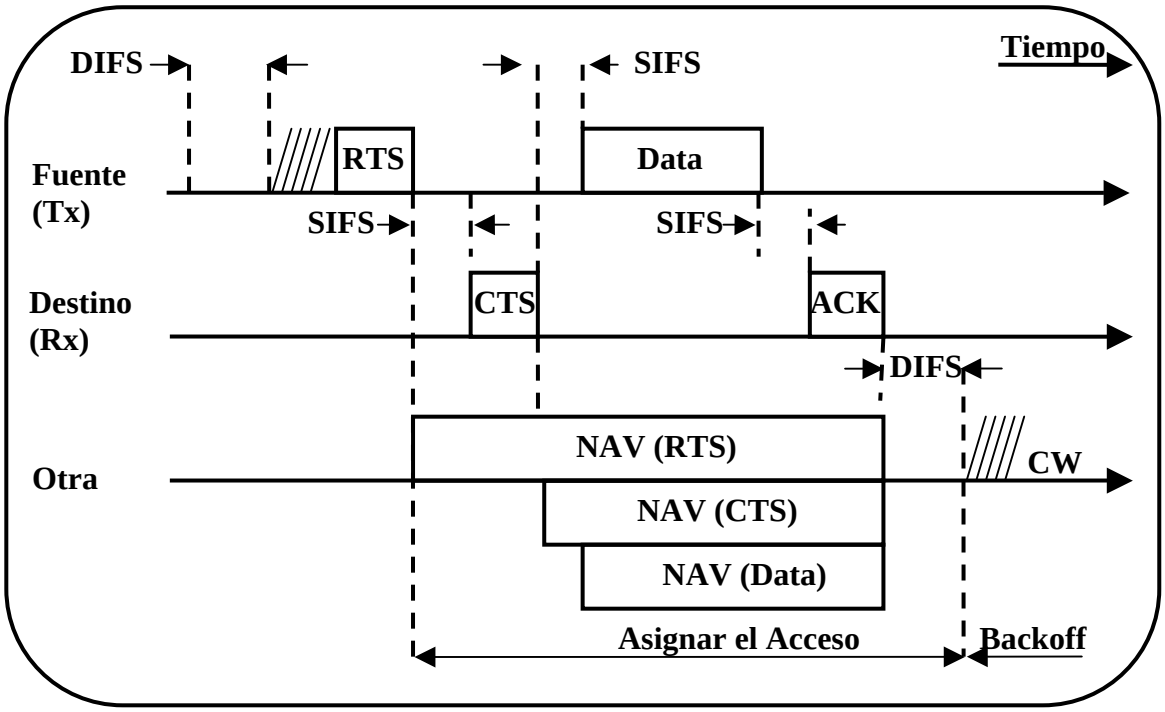


Figura 3.6. Protocolo de Acceso RTS/CTS.

3.7.5. Espaciado entre tramas (InterFrame Space, IFS)

El tiempo de intervalo entre tramas se llama IFS. Durante este período mínimo, una estación (STA) esta escuchando el medio antes de transmitir. Se definen cuatro espaciados para dar prioridad de acceso al medio inalámbrico. A continuación se describen del tiempo de espera más corto al más largo:

- ***SIFS (Short IFS).*** Este es el período más corto. Se utiliza fundamentalmente para transmitir los reconocimientos y las tramas CTS. También es utilizado para transmitir cada uno de los fragmentos de una trama. Por último, es usado por el PC o Point Control para enviar testigo a estaciones que quieran transmitir datos síncronos

- ***PIFS (PCF).*** Es utilizado por STAs para ganar prioridad de acceso en los períodos libres de contienda. Lo utiliza el PC para ganar la contienda normal, que se produce al esperar DIFS.

- ***DIFS (DCF).*** Es el tiempo de espera habitual en las contiendas con el protocolo de acceso CSMA/CA y RTS/CTS. Se utiliza pues para el envío de tramas MAC MPDUs y tramas de gestión MMPDUs.

- ***EIFS (Extended IFS).*** Controla la espera en los casos en los que se detecta la llegada de una trama errónea. Espera un tiempo suficiente para que le vuelvan a enviar la trama u otra solución.

La figura 3.7 muestra algunas relaciones del tiempo de espera IFS. Cuando una estación desea acceder al medio o canal, primero sensa si está libre u ocupado, en caso de que este libre accede al medio en un tiempo de espera DIFS. En caso contrario, es decir que se

encuentre ocupado el tiempo de espera varia de un DIFS hasta un SIFS anexando el tiempo de ejecución del algoritmo Backoff.

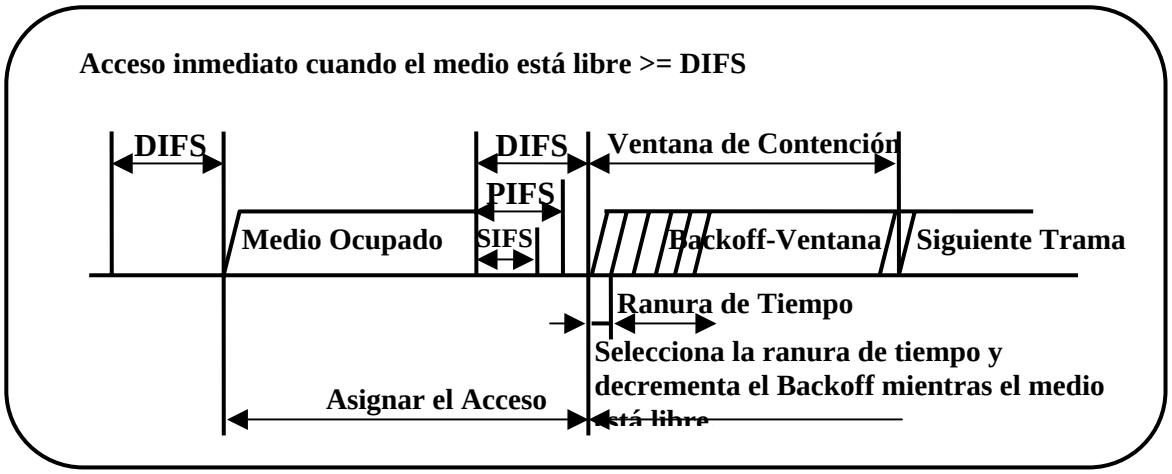


Figura 3.7. Ejemplo de algunas relaciones del tiempo de espera IFS.

3.7.6. Conocimiento del Medio, NAV (Network Allocation Vector)

Las estaciones tienen un conocimiento específico de cuando la estación, que en estos momentos tiene el control del medio porque está transmitiendo o recibiendo, va a finalizar su período de reserva del canal.

Esto se hace a través de una variable llamada NAV (Network Allocation Vector) que mantendrá una predicción de cuando el medio quedará liberado.

Tanto al enviar un RTS como al recibir un CTS, se envía el campo Duration/ID con el valor reservado para la transmisión y el subsiguiente reconocimiento. Las estaciones que estén a la escucha modificarán su NAV según el valor de este campo Duration/ID. En realidad, hay una serie de normas para modificar el NAV, una de ellas es que el NAV siempre se situará al valor más alto de entre los que se disponga.

La figura 3.8 muestra un ejemplo en el momento que es utilizado el NAV por las estaciones que desean acceder al medio. Mediante el NAV tienen una aproximación de cuando el canal estará libre, para poder acceder al canal y transmitir la información.Teniendo una estación fuente que desea acceder al medio para realizar su transmisión, una estación destino en espera de información y otra estación tratando de acceder al canal.

Una vez que la estación fuente encuentra el canal libre transmite RTS siendo una solicitud para enviar información y después de un intervalo de tiempo SIFS si se recibe CTS entonces se reserva el canal por el período de tiempo en el que se va enviar la información, consecutivamente se envía la información (Data) y espera por el reconocimiento como confirmación de haber recibido la información, todas las STAs esperan un intervalo de tiempo DIFS.

Mientras que la otra estación ejecuta el NAV para predecir en que momento queda libre el canal y poder acceder.

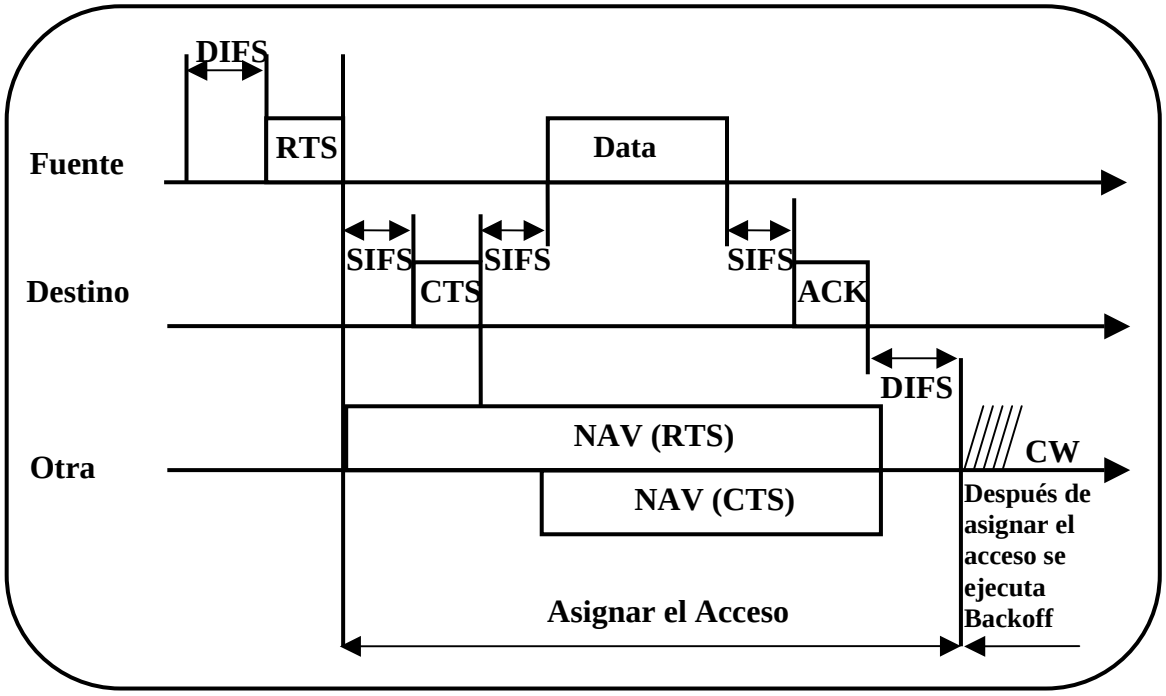


Figura 3.8. Ejemplo del NAV.

3.7.7. Función de Coordinación Puntual (PCF)

En este modo, un Punto de Acceso controla el acceso al medio de varias estaciones. Si un BSS tiene activada la opción de PCF, en el caso de las versiones a y b, o la versión g donde los dos métodos son obligatorios, los dos métodos de acceso al medio (PCF y DCF) se alternaran. Esto se hará con el uso de un Periodo Libre de Contención (CFP, Contetion Free Period) seguido de un Periodo de Contención (CP, Contention Period), como se muestra en la figura 3.9.

Estos dos métodos de acceso pueden operar conjuntamente dentro de una misma celda o conjunto básico de servicios dentro de una estructura llamada supertrama. Una parte de esta supertrama se asigna al período de contienda permitiendo al subconjunto de estaciones que lo requieran transmitir bajo mecanismos aleatorios. Una vez finalizado este período el punto de acceso toma el medio y se inicia un período libre de contienda *n* el que pueden transmitir el resto de estaciones de la celda que utilizan técnicas deterministas.

Existe un nodo organizador o director, llamado punto de coordinación o PC. Este nodo tomará el control mediante el método PIFS, y enviará un CF-Poll a cada estación que pueda transmitir en CFP, concediéndole poder transmitir una trama MPDU. El PC mantendrá una lista Pollable donde tendrá todos los datos de las estaciones que se han asociado al modo CF-Pollable. La concesión de transmisiones será por riguroso listado y no permitirá que se envíen dos tramas hasta que la lista se haya completado.

El nodo utilizará una trama para la configuración de la supertrama, llamada Beacon, donde establecerá una CFRate o tasa de períodos de contienda. Pese a que el período de contienda se puede retrasar por estar el medio ocupado, la tasa se mantendrá en el siguiente período con medio libre.

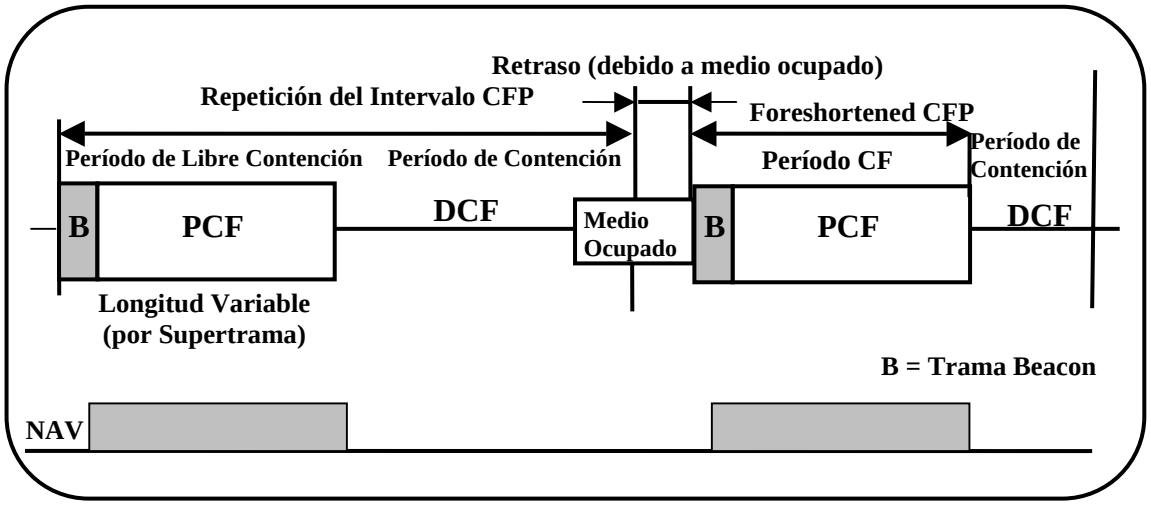


Figura 3.9. Acceso al medio PCF y DCF.

La figura 3.10 muestra la transmisión de CF-Polls espera un tiempo SIFS. También se vee que si una estación no aprovecha su CF-Poll se transmite a la siguiente en el listado Pollable.

Las estaciones que no usen el CF, situarán su NAV al valor del final del CF y luego lo resetearán para poder modificarlo en el período de contienda en igualdad de condiciones.

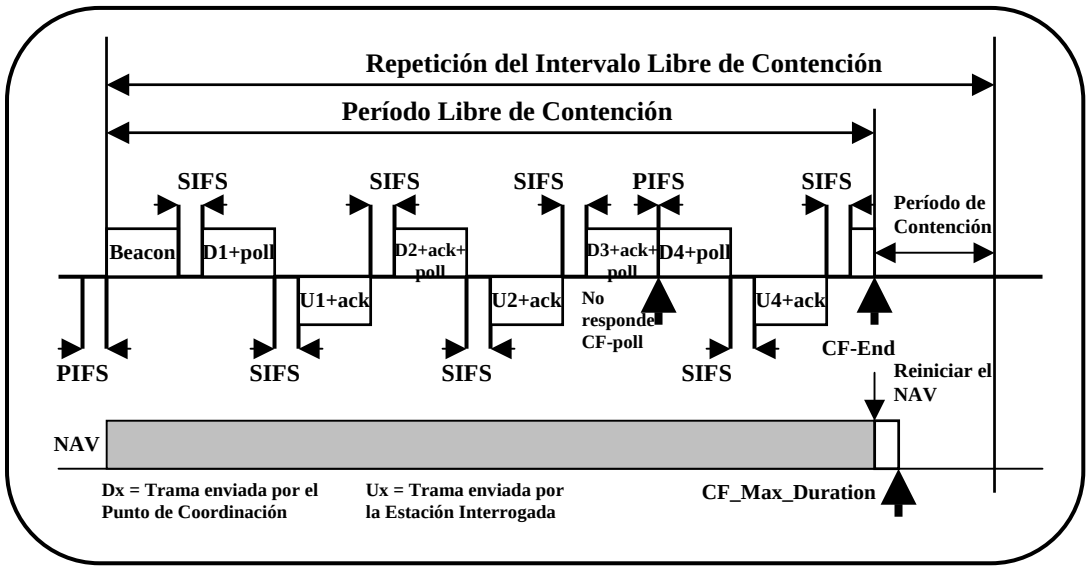


Figura 3.10. Ejemplo del PCF.

CAPÍTULO 4

4. CRIPTOGRAFÍA

La primera aplicación conocida de la criptografía se remonta a 4000 años atrás, cuando los egipcios utilizaban jeroglíficos crípticos para narrar la vida y hazañas de sus faraones. El cifrado no se empleaba para esconder el significado del texto sino para darle un carácter más solemne.

En la antigua China, el carácter ideográfico del idioma servía para esconder el significado de las palabras, aunque no parece que esta particularidad se hubiera empleado para cifrar/descifrar mensajes.

Varios pueblos de la antigüedad emplearon diversos métodos de cifrado/descifrado de escritos, como los Griegos, los Espartanos y los Hebreos, pero los Árabes y los Indios fueron los que mayor desarrollo lograron en este campo.

El método de Julio César es el más antiguo. El sistema reemplaza cada letra por la situada tres posiciones delante en el alfabeto. Por ejemplo: OOHJXH = LLEGUE. A través de prueba y ensayo (con 26 intentos) y métodos estadísticos fácilmente se rompe este método.

Con el tiempo, además de los métodos manuales aparecieron máquinas simples, como la rueda de Thomas Jefferson. La llegada del telégrafo significó un importante avance en la criptografía, al generalizarse el uso de máquinas electromecánicas para el cifrado de mensajes. Las dos guerras mundiales también impulsaron significativamente el avance de la criptografía y del criptoanálisis.

Se define criptografía [10] (Kriptos=ocultar, Graphos=escritura) como la técnica de transformar un mensaje legible, denominado *texto en claro*, en otro que sólo puedan entender las personas autorizadas a ello, que se llama *texto cifrado*. El método o sistema empleado para cifrar el texto en claro se denomina *algoritmo de cifrado*. La criptografía se clasifica en clásica y moderna.

Los sistemas criptográficos clásicos presentaban una dificultad en cuanto a la relación complejidad-longitud de la llave/tiempo necesario para cifrar y descifrar el mensaje. En los sistemas criptográficos modernos esta barrera clásica se rompió, debido principalmente a los siguientes factores: Velocidad de cálculo, avance de las matemáticas, necesidades de seguridad. A partir de estas bases surgieron nuevos y complejos sistemas criptográficos, que se clasificaron en dos subtipos: *Criptografía de Llave Pública* (se basan en complejas operaciones matemáticas) y *Criptografía de Llave Simétrica* (mezclan la trasposición y la permutación).

La *Criptografía de Llave Pública o Asimétrica* [24], utiliza dos llaves diferentes, llaves que poséen una propiedad fundamental: una llave puede descifrar lo que la otra ha cifrado. Generalmente una de las llaves de la pareja, denominada llave privada, es usada por el propietario para cifrar los mensajes, mientras que la otra, llamada llave pública, es usada para descifrar el mensaje cifrado.

Las llaves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves públicas son diferentes, entonces sus llaves privadas asociadas también lo son y viceversa.

Mientras que la llave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la llave pública es difundida ámpliamente por Internet, para que está al alcance el mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas llaves.

En este sistema, para enviar un documento con seguridad (ver figura 4.1), el emisor A cifra el mismo el mensaje claro con la llave pública del receptor B, obtiene el texto cifrado y lo envía por el medio inseguro. Este documento está totalmente protegido en su viaje, ya que sólo se puede descifrar con la llave privada correspondiente, conocida sólamente por B. Al llegar el mensaje cifrado a su destino, el receptor usa su llave privada para obtener el mensaje en claro.

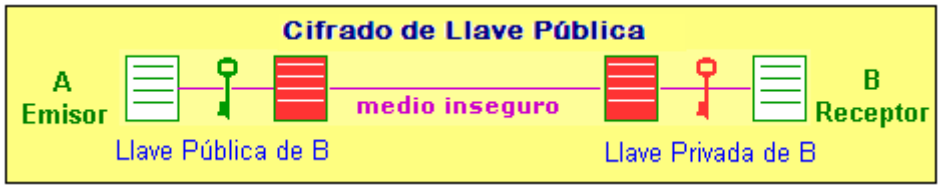


Figura 4.1. Cifrado de Llave Pública.

Los sistemas de criptografía asimétrica son: el DH (Diffie & Hellman) [25], el Gamal, el DSA (Digital Signature Algorithm), el Merkle-Hellman, el Chor-Rivest, el LUC, el McEliece, y finalmente el RSA (Rivest, Shamir & Adleman) [10] que es el más ampliamente usado.

La *Criptografía de Llave Privada o Simétrica* [26] se caracteriza por que utiliza la misma llave para cifrar y para descifrar, como se muestra en la figura 4.2. El emisor A cifra el mismo el mensaje claro con la llave privada AB, obtiene el texto cifrado y lo envía por el medio inseguro. Este documento está totalmente protegido en su viaje, ya que sólo se puede descifrar con la llave privada AB correspondiente. Al llegar el mensaje cifrado a su destino, el receptor usa su llave privada AB para obtener el mensaje en claro.

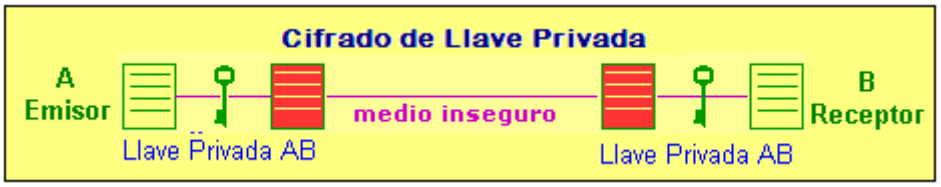


Figura 4.2. Cifrado de Llave Privada.

Toda la seguridad de este sistema está basada en la llave simétrica, por lo que es misión fundamental tanto del emisor como del receptor conocer esta llave y mantenerla en secreto. Si la llave cae en manos de terceros, el sistema deja de ser seguro, por lo que habría que desechar dicha llave y generar una nueva.

Generalmente el algoritmo de cifrado es conocido, se divulga públicamente, por lo que la fortaleza del mismo dependerá de su complejidad interna y sobre todo de la longitud de la llave empleada. Los algoritmos simétricos cifran bloques de texto del documento original, y son más sencillos que los sistemas de llave pública, por lo que sus procesos de cifrado y descifrado son más rápidos. Los principales algoritmos simétricos actuales son: AES [28], DES [27], IDEA [27] y RC5.

Las principales desventajas de los métodos simétricos son la distribución de las llaves, el peligro de que muchas personas deban conocer una misma llave y la dificultad de almacenar y proteger muchas llaves diferentes.

Tanto la criptografía simétrica como la asimétrica basan su fortaleza en problemas matemáticos difíciles de resolver, como por ejemplo la factorización de números enteros grandes. Sin embargo y debido al avance en la potencia de computación, se estima hoy en día que solamente ofrecen muy buena seguridad las llaves de 2048 bits en el caso de RSA y 256 bits para AES.

La criptografía, en el contexto de redes informáticas, es la ciencia que estudia los métodos y procedimientos, mediante algoritmos matemáticos, para modificar los datos de tal manera que solamente las personas que tengan la llave adecuada puedan a) tener acceso a la versión original de los mismos (confidencialidad) y b) asegurar que estos datos no fueron modificados entre el remitente y el destinatario (integridad). Considerando que los servicios de seguridad definidos por el ISO son: Autenticación, Control de Acceso, Confidencialidad, No-Rechazo e Integridad. Se puede decir que la criptografía nos proporciona dos de los servicios definidos por el ISO.

La criptografía hoy día involucra varias formas de cifrar/descifrar, así como diferentes métodos de autenticación. Aunque sus métodos y aplicaciones siguen siendo cada vez más complejos, la criptografía como tal sigue girando fundamentalmente alrededor de problemas matemáticos difíciles de solucionar. Un problema puede ser difícil de resolver porque su solución requiere de cierto conocimiento secreto, como la llave para descifrar un mensaje cifrado o para firmar un documento digital. También puede ser que sea intrínsecamente difícil de solucionar, en términos de los requerimientos matemáticos o de cómputo necesarios para solucionar o descifrar el mensaje cifrado.

4.1. Historia Estándar Avanzado de Cifrado (AES)

El primer concurso para un sistema estándar de cifrado fue lanzado en 1973 por la NBS (antecesora del NIST), y lo ganó el DES (Data Encryption Standard).

El 23 de Noviembre de 1976 se establece el primer estándar de cifrado para comunicaciones, el denominado algoritmo DES [27]. Este algoritmo se ha mantenido hasta hace poco como estándar de los organismos oficiales de los EEUU y como algoritmo simétrico más utilizado por las empresas privadas.

Desde entonces se han descrito multitud de ataques que permiten criptoanalizarlo más rápidamente que con un ataque por fuerza bruta. Pero sin duda la publicación de la asociación EFF (Electronic Frontier Foundation) del diseño de una máquina que permite atacarlo por fuerza bruta en un tiempo ínfimo (DES-CRACKER). El estándar había consumido su tiempo de vida.

El Instituto Nacional de Estándares y Tecnología de EEUU (NIST) aprobó el DES desde 1994 hasta diciembre de 1998. Pero en 1997 decidió no volver a utilizarlo y, por lo tanto, convocar un concurso para buscar un nuevo sistema.

La última reafirmación de DES en Octubre de 1999 realmente fue suplantado por TDES (Triple Data Encryption Standard) [27], que es una versión múltiple de DES, designado como TDEA (Triple Data Encryption Algorithm) [27].

En el año 1997, el NIST, emprende un proceso abierto para la selección de un nuevo algoritmo de cifrado, que sustituya al actual estándar de cifrado, criticado por especialistas e instituciones en seguridad.

Este nuevo algoritmo sería útil no sólo para proteger la información del Gobierno EEUU, sino que también sería utilizado masivamente por el sector privado, y adoptado como estándar por el resto de países, entre ellos los europeos.

Para evitar las quejas que se procedieron con la implantación del algoritmo DES debido a partes del algoritmo no documentadas que daban la sensación que el gobierno EEUU mantenía puertas traseras, se decide iniciar un proceso abierto para seleccionar el algoritmo que formaría el nuevo estándar de cifrado AES [26].

Se inicia entonces los primeros pasos para la consolidación de un Estándar de Cifrado Avanzado (AES) que permita proteger los datos confidenciales del gobierno, así como la información sensible de los ciudadanos.

En Septiembre de 1997 se presentan los criterios de evaluación y requisitos mínimos que debían cumplir todos los algoritmos que optarán a ganar el concurso, entre ellos destacaban:

- El algoritmo debe ser público.
- Debe ser un algoritmo de cifrado en bloque simétrico.
- La longitud de la llave debe ser como mínimo 128 bits.
- Su diseño debe permitir aumentar la longitud de la llave según las necesidades.
- Debe ser implementable tanto en HW como en SW

Los algoritmos que cumplieran los requisitos anteriores serían juzgados por los siguientes factores:

- Seguridad.
- Eficiencia computacional.
- Requisitos de memoria.
- Implementable en HW y SW.
- Simplicidad de diseño.
- Flexibilidad.

Los algoritmos que se presentaron a este concurso además tenían que soportar obligatoriamente una longitud de bloque de 128 bits como mínimo, y una longitud de llave de 128, 192 y 256 bits, al margen de cualesquiera otras longitudes posibles.

NIST, propuso que cualquier organización, institución o persona pudiera participar de forma activa en este concurso, ya fuera presentando algoritmos, o enviando informes o pruebas de cualquier tipo para poner en evidencia las características de cualquier de los algoritmos candidatos.

La intención de este estándar es que sea robusto, por lo menos, hasta la mitad del presente siglo, o por lo menos hasta que se publiquen estudios criptoanalíticos o incluso posibles máquinas futuras de supercomputación, como la soñada computación cuántica, que debilite seriamente su seguridad.

Para llevar a cabo la elección del algoritmo se propuso crear dos rondas de selección. En la primera ronda se seleccionaría los 5 algoritmos mejores, que cumplieron las especificaciones iniciales, y en la segunda ronda se decidiría el algoritmo ó algoritmos ganadores.

Para realizar estas rondas de selección, se convocaron tres Conferencias, en distintos lugares del mundo, en las que los algoritmos candidatos, pudieron ser probados, comentados y revisados con lupa por todo el mundo que lo deseó.

Durante todo el desarrollo del proceso AES, todos los algoritmos y criterios de diseño estuvieron disponibles de forma pública y abierta, por lo que la indagación al que han sido sometidos todos los finalistas ha sido enorme, acorde con la importancia del nuevo AES. Todos los participantes contribuyeron al proceso, analizando las posibles vulnerabilidades de sus competidores.

Las propuestas fueron presentadas antes de junio de 1998 y después se realizó una primera ronda para eliminar candidatos. En agosto de 1998 se publicó la lista de los 15 algoritmos candidatos como se muestra en la tabla 2.1.

Nombre del Algoritmo	***Autores del Algoritmo***
CAST-256	Entrust Technologies, Inc.
CRYPTON	Future Systems, Inc.
DEAL	Richard Outerbridge, Lars Knudsen
DFC	CNRS - Centre National pour la Recherche Scientifique – Ecole Normale Superieure
E2	NTT - Nippon Telegraph and Telephone Corporation
FROG	TecApro Internacional S.A.
HPC	Rich Schroeppel
LOKI97	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
MAGENTA	Deutsche Telekom AG
MARS	IBM
RC6	RSA Laboratories
RIJNDAEL	Joan Daemen, Vincent Rijmen
SAFER+	Cylink Corporation
SERPENT	Ross Anderson, Eli Biham, Lars Knudsen

Tabla 2.1. Algoritmos Candidatos.

En Marzo de 1999, se celebraría la "Segunda Conferencia de Candidatos AES", en la que se discutió los resultados de las numerosas pruebas y criptoanálisis realizados por la

comunidad criptográfica mundial sobre los quince candidatos iniciales. Basándose en estos comentarios y análisis, NIST seleccionó cinco candidatos finalistas. Cabe decir como anécdota que uno de los quince algoritmos seleccionados, concretamente "Magenta", fue criptoanalizado en el mismo encuentro en el que se presentó. Los cinco algoritmos afortunados fueron:

- MARS. [29]
- RC6.
- RIJNDAEL. [2]
- SERPENT.
- TWOFISH.

En abril del 2000 se celebró la "Tercera Conferencia de Candidatos AES" en Nueva York, durante la cual los asistentes presentaron nuevos documentos de evaluación y criptoanálisis de los últimos cinco candidatos. Varios de los algoritmos recibieron una paliza criptográfica. RC6 resultó el más afectado: dos grupos se las ingeniaron para romper 15 de 20 ciclos del algoritmo más rápidamente que con fuerza bruta. RIJNDAEL resistió algo mejor: 7 ciclos rotos de 10/12/14 ciclos. Se presentaron varios ataques contra MARS; el más interesante rompió 11 de 16 ciclos del núcleo criptográfico. SERPENT y TWOFISH se comportaron mejor: el ataque más fuerte contra SERPENT rompió 9 de 32 ciclos, y no se presentaron nuevos ataques contra TWOFISH.

Por fin, el 2 de octubre de 2000, el NIST anunció el algoritmo ganador. Las votaciones del concurso establecieron el siguiente orden:

- RIJNDAEL 86 votos.
- SERPENT 59 votos.
- TWOFISH 31 votos.
- RC6 23 votos.
- MARS 13 votos.

El algoritmo Rijndael ganó el concurso, por permitir la mejor combinación de seguridad-velocidad-eficiencia, sencillez y flexibilidad. Destacando su sencillez, que había permitido un análisis muy intenso de su estructura.

Los creadores de este ingenio son dos ingenieros electrónicos belgas, un equipo bastante modesto, teniendo en cuenta que en el proceso de selección se enfrentaban a algoritmos creados por equipos de multinacionales tan fuertes y poderosas como IBM, Deuche Telekom, así como equipos de criptólogos de reputada fama mundial como, por ejemplo, Bruce Schneier (Twofish), autor de varios libros de criptografía, o Ronald Rivest (RC6), coautor del algoritmo de llave asimétrica RSA.

En octubre de 2000 el NIST anunciaba oficialmente la adopción del algoritmo Rijndael como nuevo Estándar Avanzado de Cifrado (AES) para su empleo en aplicaciones criptográficas no militares, culminando así un proceso de más de tres años, encaminado a proporcionar a la comunidad internacional un nuevo algoritmo de cifrado potente, eficiente, y fácil de implementar. DES tenía por fin un sucesor.

La palabra Rijndael —en adelante, para referirnos a este algoritmo, emplearemos la denominación AES— es un acrónimo formado por los nombres de sus dos autores, los belgas Joan Daemen y Vincent Rijmen.

Joan Daemen, nacido en 1965, ingeniero electrónico y especialista en sistemas de seguridad electrónica bancaria. Vincent Rijmen, nacido en 1970, matemático de la facultad de Ciencias de la Universidad Católica de Lovaina.

La pregunta básica es ¿Cómo pronunciar Rijndael?, la respuesta encontrada en la páguina Web oficial es la siguente: sí es usted Alemán, de Indonesia, de Sur Africa, Flemish y Surinamer, éste se pronuncia como es, sin embargo usted puede pronunciarlo como "Reign Dahl", "Rain Doll", "Rhine Dahl". Pero no pronunciar como "Region Deal".

Su interés radica en que todo el proceso de selección, revisión y estudio tanto de este algoritmo como de los restantes candidatos, se ha efectuado de forma pública y abierta, por lo que, prácticamente por primera vez, toda la comunidad criptográfica mundial ha participado en su análisis, lo cual convierte a Rijndael en un algoritmo perfectamente digno de la confianza de todos.

Este algoritmo soporta diferentes tamaños de bloque y llave, en el estándar adoptado por el Gobierno Estadounidense en noviembre de 2001 [30], se especifica una longitud fija de bloque de 128 bits, y la longitud de llave a escoger entre 128, 192 y 256 bits.

4.2. Algoritmo AES

AES es un sistema de cifrado por bloques, diseñado para manejar una longitud fija de bloque de 128 bits, y la longitud de llave a escoger entre 128, 192 y 256 bits.

Realiza varias de sus operaciones internas a nivel de byte, interpretando éstos como elementos de un campo de Galois GF(2^8). El resto de operaciones se efectúan en términos de registros de 32 bits.

Si bien, como ya se ha dicho, este algoritmo soporta tamaño de bloque de 128 bits y diferentes tamaños de llave, por simplicidad y dado que en la práctica sólo se utiliza, se restringe a 128 bits para el tamaño del bloque y de la llave.

El algoritmo consiste de 10 rondas. Cada ronda tiene una llave, derivada de la llave original. La llave original se utiliza en la ronda cero. Una ronda inicia con una entrada de 128 bits y produce una salida de 128 bits.

Hay cuatro pasos básicos, llamados capas, que son usados para formar las rondas:

1. **Transformación de ByteSub (Sustitución de Byte):** Esta capa no lineal es para prevenir y resistir ataques de criptoanálisis lineal y diferencial. Además consiste en la aplicación paralela de s-cajas con propiedades óptimas de no linealidad

2. **Transformación ShiftRow (DesplazarFila):** Esta capa lineal mezcla pasos causando difusión de los bits sobre múltiples rondas.

3. **Transformación MixColumn (MezclarColumnas):** Esta capa lineal permite obtener un alto nivel de difusión a lo largo de varias rondas.

4. **AddRoundKey (Adición de llave):** es un simple or-exclusivo entre el estado intermedio y la subllave correspondiente a cada ronda.

Una ronda es entonces:

Algoritmo Rijndael Cifrado

1. ARK, usando la llave original para la ronda cero.
2. Nueve rondas de BS, SR, MC, ARK, usando las llaves del uno al nueve.
3. Ronda final o diez, BS, SR, ARK, usando la llave diez.

La salida del bloque de texto cifrado es de 128 bits.

Algoritmo Rijndael Descifrado

1. ARK, usando la llave diez.
2. Nueve rondas de IBS, ISR, IMC, IARK, usando las llaves del nueve al uno.
3. Ronda final o diez, IBS, ISR, ARK, usando la llave original.

Siendo I igual a Inverse, es decir el proceso inverso.

A continuación se describen los pasos más a detalle. La entrada (texto claro o bien conocido como mensaje a cifrar) de 128 bits son agrupados en 16 bytes. Los 16 bytes son almacenados en una matriz de 4 x 4 llamada State, como se muestra consecutivamente:

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{1,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{1,2} & a_{3,3} \end{pmatrix}$$

Matriz State

4.2.1. Transformación ByteSub

La transformación ByteSub es una substitución no lineal que se aplica a cada byte de la matriz State, mediante la tabla de substitución (o S-Box) 256 invertible, que se obtiene mediante la composición dos transformaciones:

1. Cada byte es considerado como un elemento del GF(2^8) que genera el polinomio irreducible $m(x) = x^8 + x^4 + x^3 + x + 1$, y substituido por su inverso multiplicativo. El valor cero queda inalterado.
2. Después se aplica la siguiente transformación afín en GF(2^8), siendo $x^0, x^1, \ldots, x^7$ los bits del byte correspondiente, e $y^0, y^1, \ldots, y^7$ los del resultado:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

En esta transformación, cada byte de la matriz State es cambiado por otro byte en S-Box. Un byte esta compuesto por ocho bits: *abcdefgh*. Los primeros cuatro bits *abcd* representan la fila y los últimos cuatro bits *efgh* la columna. Los pasos a seguir para realizar la transformación de ByteSub son:

- Tomar el 1er byte $a_{0,0}$ de la matriz State.
- Ir a S-Box, *abcd* indica la fila y *efgh* la columna, la intercesión de la fila con la columna indica el nuevo byte.
- Reemplazar el 1er byte de la matriz State por el nuevo byte.
- Los pasos del 1 al 3 se repiten 16 veces, es decir hasta finalizar con la substitución todos los elementos de la matriz.

S-Box															
99	124	119	123	242	107	111	197	48	1	103	43	254	215	171	118
202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
4	119	35	195	24	150	5	154	7	18	128	226	235	39	178	117
9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
231	200	55	109	141	213	78	169	108	86	244	234	101	122	174	8
186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

Tabla 2.2. S-Box para Rijndael.

El proceso inverso de ByteSub es la aplicación de la inversa de S-Box correspondiente a cada byte de la matriz State.

4.2.2. Transformación ShiftRow

Esta transformación consiste en desplazar a la izquierda cíclicamente las filas de la matriz State, es decir:

- Fila 0 – 0 Rotación
- Fila 1 – 1 Rotación
- Fila 2 – 2 Rotaciones
- Fila 3 – 3 Rotaciones

El proceso inverso ShiftRow es desplazar a la derecha cíclicamente las filas de la matriz de State, es decir:

- Fila 0 – 0 Rotación
- Fila 1 – 1 Rotación
- Fila 2 – 2 Rotaciones
- Fila 3 – 3 Rotaciones

4.2.3. Transformación MixColumn

Para esta transformación, cada columna de State se considera un polinomio cuyos coeficientes pertenecen a $GF(2^8)$ y se multiplica módulo $x^4 + 1$ por:

$$c(x) = 03x^4 + 01x^2 + 01x + 02$$

donde 03 es el valor hexadecimal que se obtiene concatenando los coeficientes binarios del polinomio correspondiente en $GF(2^8)$, en este caso 00000011, o sea, x+1, y así sucesivamente.

La inversa de MixColumn se obtiene multiplicando cada columna de la matriz State por el polinomio:

$$d(x) = 0Bx^4 + 0Dx^2 + 09x + 0E$$

4.2.4. AddRoundKey

Se realiza una operación de XOR del resultado de la transformación MixColumn con la llave de ronda correspondiente. La llave de ronda se obtiene apartir de la llave inicial mediante el proceso de Key Schedule.

Se ilustra la transformación de AddRound Key:

$$\begin{pmatrix} d_{0,0} & d_{0,1} & d_{0,2} & d_{0,3} \\ d_{1,0} & d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,0} & d_{2,1} & d_{1,2} & d_{2,3} \\ d_{3,0} & d_{3,1} & d_{1,2} & d_{3,3} \end{pmatrix} \text{XOR} \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{1,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{1,2} & k_{3,3} \end{pmatrix} = \begin{pmatrix} e_{0,0} & e_{0,1} & e_{0,2} & e_{0,3} \\ e_{1,0} & e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,0} & e_{2,1} & e_{1,2} & e_{2,3} \\ e_{3,0} & e_{3,1} & e_{1,2} & e_{3,3} \end{pmatrix}$$

El proceso inverso de AddRoundKey es utilizar la llave de ronda correspondiente al algoritmo descifrado, es decir es similar.

4.3. Key Schedule

La llave original tiene un tamaño de 128 bits, la cual es almacenada en una matriz de 4x4 bytes. Esta matriz es expandida uniendo 40 columnas más, como sigue. Se etiquetan las cuatro primeras columnas como W(0), W(1), W(2), W(3). Las nuevas columnas se generan recursivamente. Se definen las nuevas columnas ha obtener como W(i-1).

Si i no es múltiplo de 4, entonces:

$$W(i) = W(i - 4) \text{ XOR } W(i - 1)$$

Si i es múltiplo de 4, entonces:

$$W(i) = W(i - 4) \text{ XOR } T(W(i - 1)),$$

donde T(W(i - 1)) es la transformación de W(i - 1) obtenida como sigue. Sean a, b, c, d los elementos de la columna W(i - 1). Se cambian los elementos cíclicamente obteniendo b, c, d, a. Ahora substituya cada uno de estos bytes por el elemento correspondiente en S-Box de la transformación ByteSub, obteniendo 4 bytes e, f, g, h. Finalmente calcule la ronda constante,

$$r(i) = 00000010^{(i-4)/4}$$

en GF(2^8). Entonces T(W(i - 1)) es el vector columna (e XOR $r(i)$, f, g, h)…

Mediante lo anterior, las columnas W(4), . . ., W(43) son generadas a partir de las cuatro columnas iniciales.

La llave de ronda para la ronda i consiste de las columnas W($4i$), W($4i$+1), W($4i$+2), W($4i$+3).

4.4. Seguridad AES

Según sus autores, es altamente improbable que existan llaves débiles o semidébiles en AES, debido a la estructura de su diseño, que busca eliminar la simetría en las subllaves.
También se ha comprobado que es resistente a criptoanálisis tanto lineal como diferencial. En efecto, el método más eficiente conocido hasta la fecha para recuperar la llave a partir de un par texto cifrado-texto claro es la búsqueda exhaustiva, por lo que podemos considerar a este algoritmo uno de los más seguros en la actualidad.

CAPÍTULO 5

5. MODOS DE OPERACIÓN PARA ALGORITMOS DE CIFRADO POR BLOQUES

En esta sección se explican algunos métodos para aplicar cifrado por bloques a mensajes de gran longitud, eligiendo un método para utilizar con el algoritmo de cifrado por bloques AES. En primer lugar, independientemente del método empleado para cifrar, se tiene en cuenta lo que ocurre cuando la longitud de la cadena que se desea cifrar no es un múltiplo exacto del tamaño de bloque. Entonces hay que añadir información al final para que sí lo sea. El mecanismo más sencillo (ver figura 5.1) consiste en rellenar con ceros (o algún otro patrón) el último bloque que se cifra. El problema ahora consiste en saber cuando se descifra por dónde hay que cortar. Lo que se suele hacer es añadir como último byte del último bloque el número de bytes que se han añadido. Esto tiene el inconveniente de que si el tamaño original es múltiplo del bloque, hay que alargarlo con otro bloque entero. Por ejemplo, si el tamaño de bloque fuera 64 bits, y sobran cinco bytes al final, se añade dos ceros y un tres, para completar los ocho bytes necesarios en el último bloque. Si por el contrario no sobrara nada, se tiene que añadir siete ceros y un ocho.

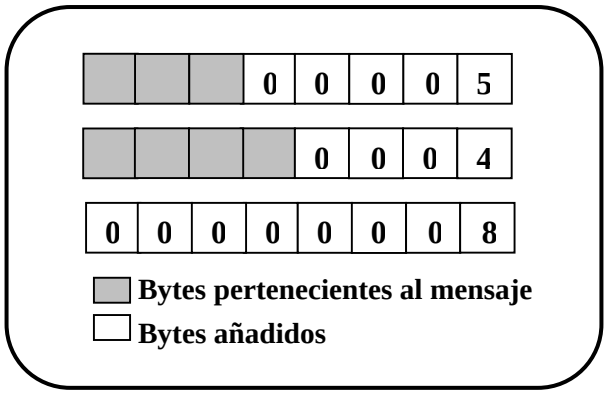


Figura 5.1. Relleno de los bytes del último bloque al emplear un algoritmo de cifrado por bloque.

5.1. Modo ECB (Electronic CodeBook)

El modo ECB es el método más sencillo y obvio de aplicar un algoritmo de cifrado por bloques. Simplemente se subdivide la cadena que se quiere codificar en bloques del tamaño adecuado y se cifran todos empleando la misma llave. Se muestra en la figura 5.2 el procedimiento de Cifrado el cual consiste en insertar el bloque del mensaje P_1 y la llave K en el tiempo 1 al cifrador obteniendo el texto cifrado C_1 hasta terminar con el bloque del mensaje P_N en el tiempo N obteniendo el texto cifrado C_N.

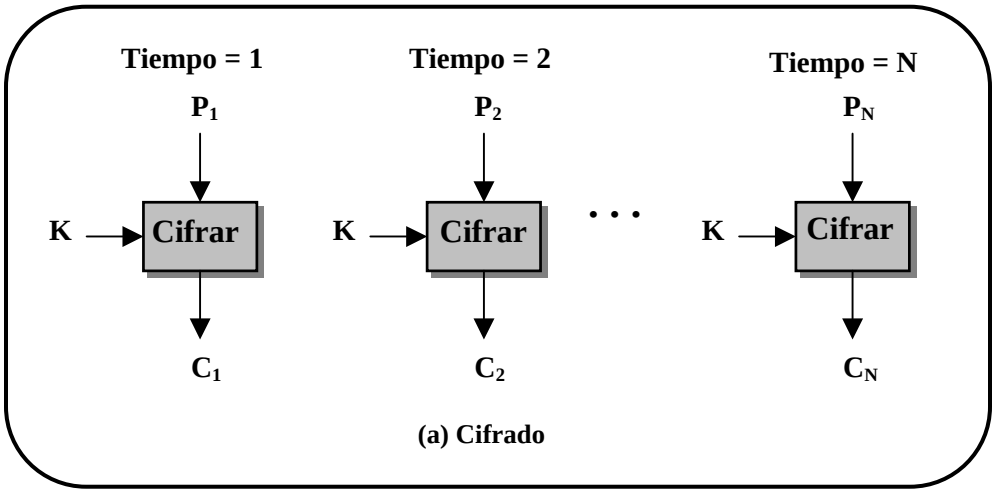


Figura 5.2. Modo de operación ECB. (a): Cifrado.

Se ilustra en la figura 5.3 el procedimiento de Descifrado, tiene como entrada el texto cifrado C_1 y la llave K en el tiempo 1 obteniendo el bloque del mensaje P1 hasta finalizar con el texto cifrado C_N en el tiempo N teniendo como resultado el bloque del mensaje P_N.

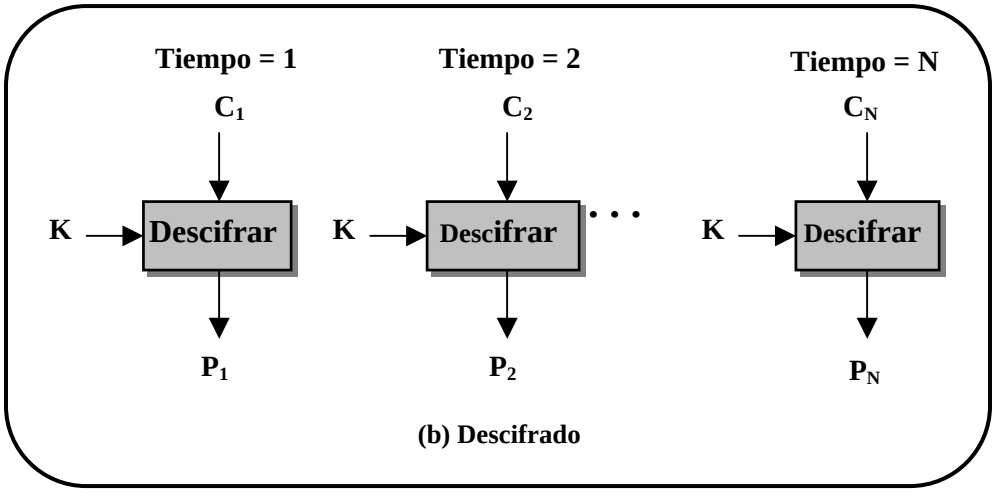


Figura 5.3. Modo de operación ECB. (b): Descifrado.

Este método permite cifrar los bloques independientemente de su orden, lo cual es adecuado para codificar bases de datos o ficheros en los que se requiera un acceso aleatorio. También es resistente a errores, pues si uno de los bloques sufriera una alteración, el resto quedaría intacto.

Por contra, si el mensaje presenta patrones repetitivos, el texto cifrado también los presentará, y eso es peligroso, sobre todo cuando se codifica información muy redundante (como ficheros de texto), o con patrones comunes al inicio y final (como el correo electrónico). Un contrincante puede en estos casos efectuar un ataque estadítico y extraer bastante información.

5.2. Modo CBC (Cipher Book Chaining)

El modo CBC incorpora un mecanismo de retroalimentación en el cifrado por bloques. Esto significa que el cifrado de bloques anteriores condiciona el cifrado del bloque actual, por lo que será imposible sustituir un bloque individual en el mensaje cifrado. Esto se consigue efectuando una operación XOR entre el bloque del mensaje que se va ha cifrar y el último mensaje cifrado obtenido.

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

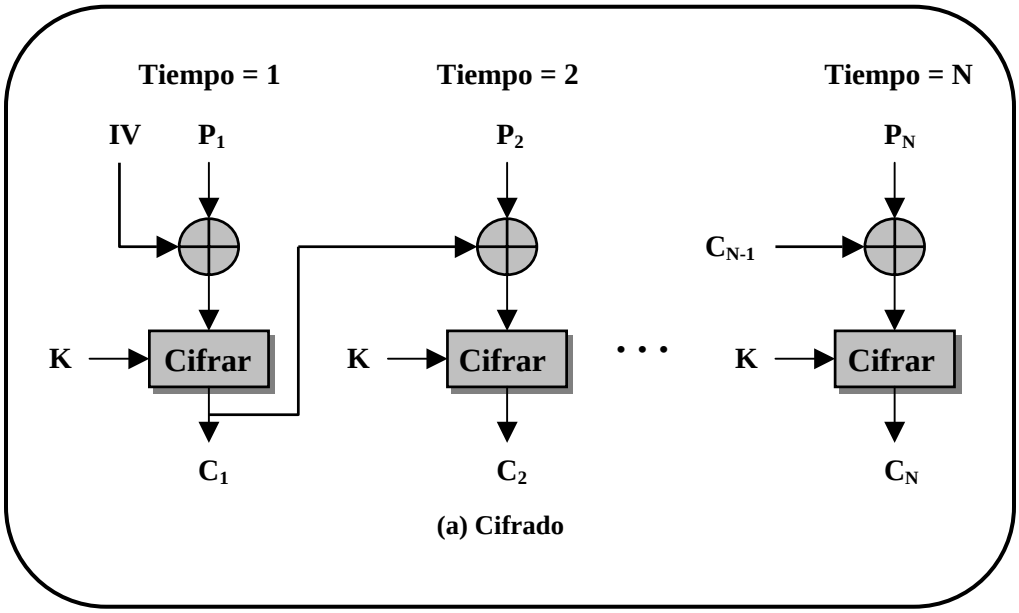


Figura 5.4. Modo de operación CBC. (a): Cifrado.

El funcionamiento de este modo se muestra en la figura 5.4 y es el siguiente:

- **Cifrado**
  1. Inicia tiempo 1, bloque del mensaje P_1 operación XOR con IV.
  2. Resultado anterior junto con la llave K entra en el cifrador.
  3. Obteniendo texto cifrado C_1, finaliza tiempo 1.
  4. Inicia tiempo 2, bloque del mensaje P_2 operación XOR con C_1.
  5. Resultado anterior junto con la llave K entra en el cifrador.
  6. Se obtiene texto cifrado C_2, finaliza tiempo 2.
  7. Se repiten los pasos del número 4 al 6 hasta obtener el texto cifrado C_N en el tiempo N.

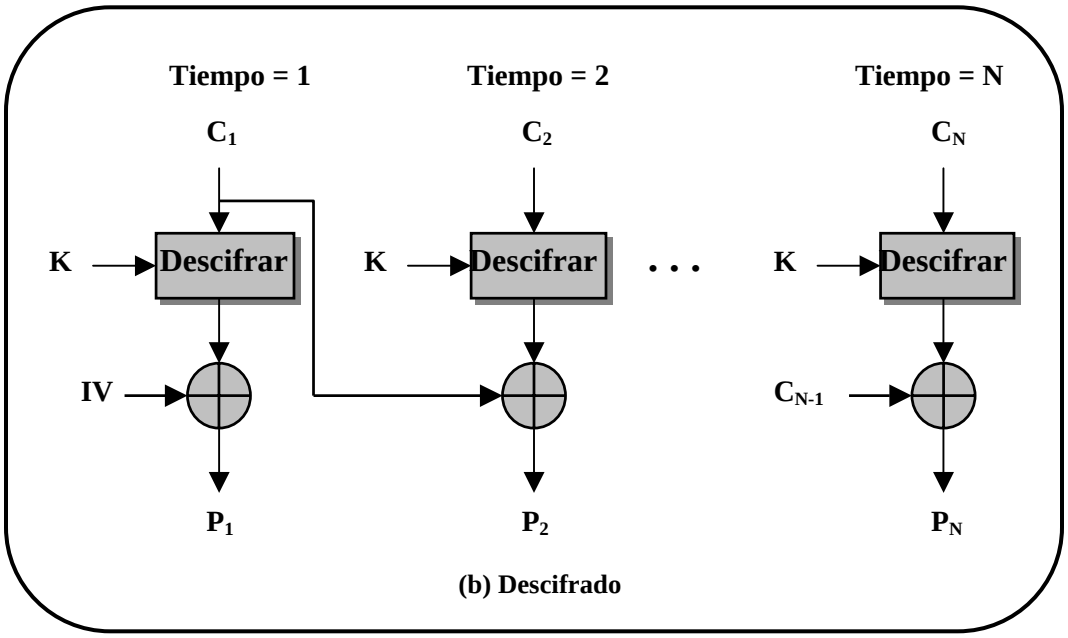


Figura 5.5. Modo de operación CBC. (b): Descifrado.

El funcionamiento del modo CBC se muestra en la figura 5.5 y es el siguiente:

- **Descifrado**
  1. Inicia tiempo 1, entra texto cifrado C_1 y la llave K a descifrar.
  2. Resultado anterior se le aplica una operación de XOR con IV.
  3. Obteniendo bloque del mensaje P_1, finaliza tiempo 1.
  4. Inicia tiempo 2, entra texto cifrado C_2 y la llave K a descifrar.
  5. Resultado anterior se le aplica una operación de XOR con el texto cifrado C_1.
  6. Se obtiene el bloque del mensaje P_2, finaliza tiempo 2.
  7. Se repiten los pasos del número 4 al 6 hasta obtener el bloque del mensaje P_N en el tiempo N.

5.3. Modo CFB (Cipher-FeedBack)

El modo de operación CFB permite cifrar la información en unidades inferiores al tamaño del bloque, lo cual aprovecha totalmente la capacidad de transmisión del canal de comunicaciones, manteniendo además un nivel de seguridad adecuado.

Este modo utiliza un registro de desplazamiento con una longitud de un bloque y dividido en secciones. Por ejemplo, si el tamaño del bloque es de ocho bytes, y se procesa un byte cada vez, el registro de desplazamiento se divide en ocho secciones.

El funcionamiento de este modo se muestra en la figura 5.6 y es el siguiente:

- **Cifrado**
  1. Inicia tiempo 1, el registro de desplazamiento es de 64 bits y es inicializado con IV, siendo s = 8 bits.
  2. Resultado anterior junto con la llave K entra en el cifrador.
  3. Se obtiene un nuevo resultado, al cual se le aplica operación XOR con el bloque del mensaje P_1.
  4. Se obtiene texto cifrado C_1, finaliza tiempo 1.

5. Del tiempo = 2 hasta M la operación para (a) Cifrado es:
    - *Ci = Pi XOR Ek(Ci-1)*

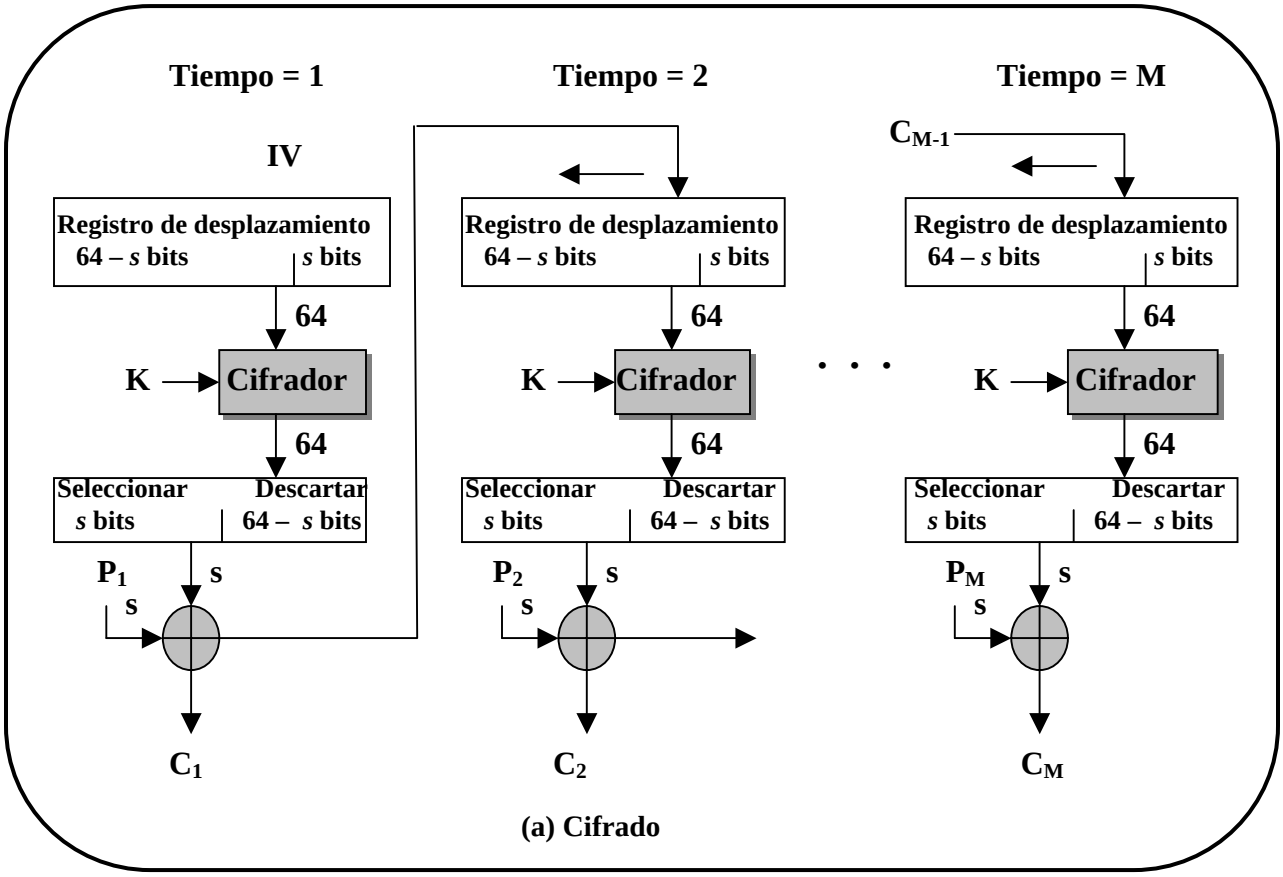


Figura 5.6. Modo de operación CFB. (a): Cifrado.

El funcionamiento del modo CFB se muestra en la figura 5.7 y es el siguiente:

❖ **Descifrado**

1. Inicia tiempo 1, el registro de desplazamiento es de 64 bits y es inicializado con IV, siendo s = 8 bits.
2. Resultado anterior junto con la llave K entra en el cifrador.
3. Se obtiene un nuevo resultado, al cual se le aplica operación XOR con el bloque del texto cifrado C1.
4. Se obtiene el bloque del mensaje original P_1, finaliza tiempo 1.
5. Del tiempo = 2 hasta M la operación para (b) Descifrado es:
    - *Pi = Ci XOR Ek(Ci-1)*

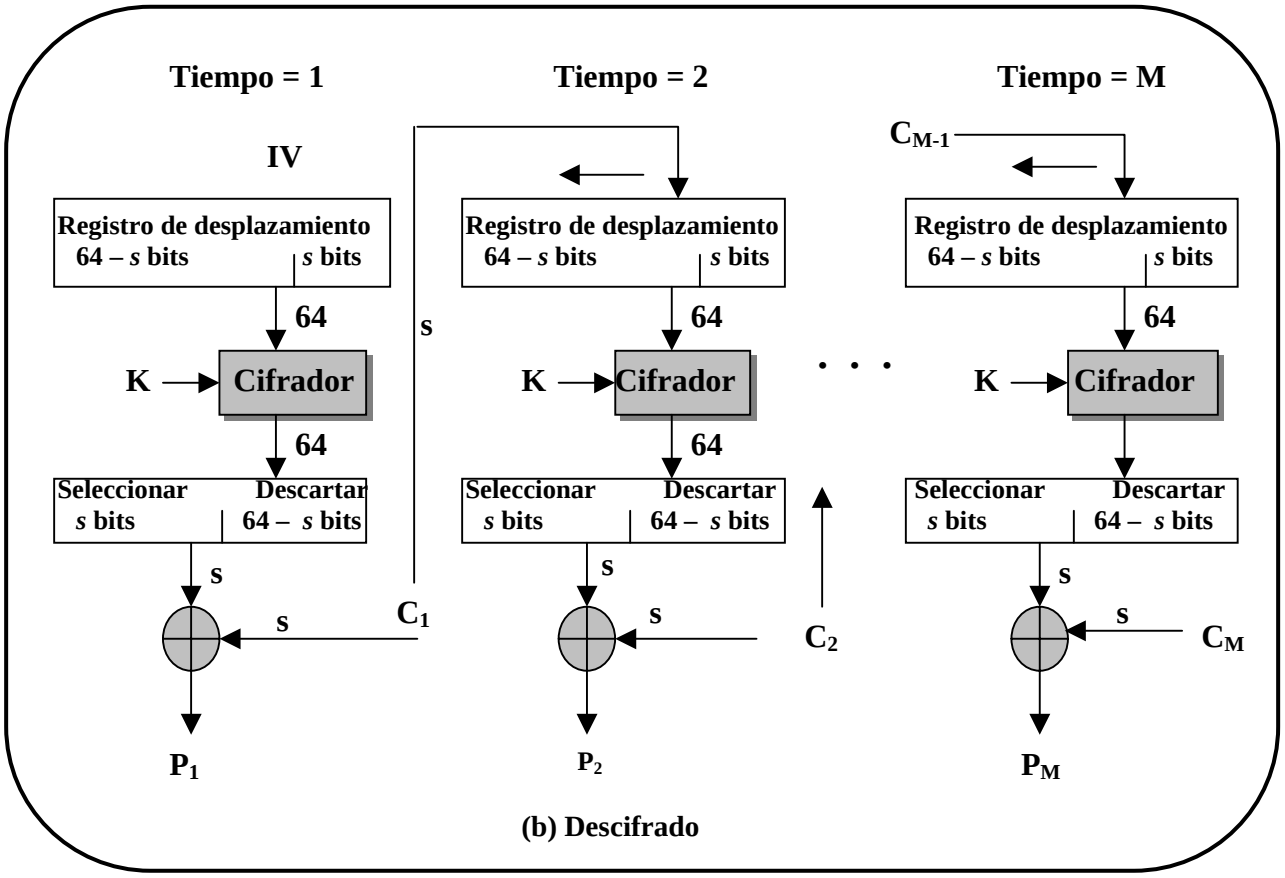


Figura 5.7. Modo de operación CFB. (b): Descifrado.

5.4. Modo CCM (Counter Mode/CBC-MAC)

Dado que los modos anteriormente explicados (ver sección 5.1 hasta 5.3) sólo proveen privacidad y no autenticación se vuelve necesario combinarlos con algún mecanismo de autenticación, típicamente un algoritmo MAC (Message Authentication Code) [31, 32]. Es necesario mencionar que existen formas seguras e inseguras de combinar un mecanismo de cifrado seguro con uno de autenticación también seguro, sin embargo ciertas de estas combinaciones son fáciles de romper debido a la mala interacción de los componentes. Mientras otras tienen una seguridad demostrable en términos de la combinación de sus componentes.

En los últimos años se han desarrollado diversos modos de operación para los cifradores de bloque que ofrezcan de manera simultanea privacidad y autenticidad. Estos modos son referidos como "modos combinados" [33] (un término menos ambiguo frecuentemente utilizado es el de "modos de autenticación y cifrado"). Idealmente, estos bloques deberían tener una seguridad comprobable, lo cual significa que existe una prueba matemática de que el esquema no puede ser roto a menos que sea encontrada una debilidad en el cifrador por bloque en el que se basa.

Dentro de estos modos de operación surge el CCM para el AES, propuesto por D. Whiting de Hfin Inc, R. Housley de RSA Labs y N. Ferguson de MacFergus BV [3, 34, 35]. En [3] los tres autores proponen el uso de este modo de operación como un modo genérico para el

AES, brindan las especificaciones de los parámetros que se utilizan, y la explicación de como funciona el modo de operación.

El modo CCM es el nombre corto de CTR + CBC-MAC. Como su nombre lo indica el CCM combina el modo de cifrado CTR con el modo de autenticación CBC-MAC. El CCM realiza ambos servicios usando un cifrador por bloque, esta propuesto para los modos de operación genéricos para el NIST [36], y su uso es ya aplicado en el estándar IEEE 802.11i [12, 13]. Así mismo existe la propuesta de D. Johnston y J Walker de Intel [37] para que sea incluido en el nuevo estándar 802.16.

Algunas de las propiedades del modo CCM que lo hacen atractivo son:

- Permite manejar mensajes en los cuales existan partes únicamente para autenticar y no para cifrar, sin provocar una disminución en la eficiencia.
- El cifrador por bloques se utiliza únicamente en su etapa de cifrado, lo cual permite un ahorro de recursos.
- Todos los derechos intelectuales han sido liberados para el dominio público.
- Propuesto el modo CCM por tres compañías importantes en seguridad Hifn Inc., MacFergus BV y RSA Security Inc.

En [33] Jakob Jonsson estudia la seguridad brindada por este nuevo modo de operación, y como lo menciona en su artículo, el hecho de que CCM este basado en dos modos que son confiables no es suficiente para hacer especulaciones sobre su seguridad, pues como hemos mencionado antes, existen métodos de combinación que hacen que un esquema a partir de dos bloques confiables sea fácil de romper.

En su artículo, Jonsson realiza el estudio de la seguridad del AES-CCM en dos aspectos fundamentales:

- Privacidad: Debe ser imposible para un adversario obtener cualquier tipo de información de un texto cifrado sin tener acceso a la llave privada.
- Autenticidad: Debe ser imposible para un adversario obtener un texto cifrado válido sin tener acceso a la llave privada.

Concluye finalmente que el modo de operación tiene una seguridad considerable y comparable a diferentes modos de operación como el OCB [38] (O_set CodeBook, que es otro modo de operación genérico que brinda tanto privacidad como autenticidad).

El modo de operación CCM proporciona confidencialidad y la autenticidad de datos. CCM se basa en un algoritmo de cifrado simétrico por bloques, con un tamaño de bloque de 128 bits, tal como el AES. CCM se puede considerar un modo de operación para algoritmo de cifrado por bloque. CCM es diseñado para utilizarse en un ambiente de paquetes de datos y no es diseñado para procesamiento por flujo de bits.

La entrada a CCM incluye tres elementos: 1) datos que serán autenticados y cifrados, llamados *payload*; 2) *datos asociados*, ejemplo, una cabezera, que será autenticada pero no cifrada; y 3) un valor único, llamado *nonce*, que se asigna al *payload* y a los *datos asociados*.

CCM consiste en dos procesos relacionados: autenticación-cifrado y descifrado-verificación; los cuales usan la combinación de dos primitivas criptográficas: CRT (Counter Mode) para ofrecer privacidad y CBC (Cipher Book Chaining) ofreciendo autenticación.

En el proceso autenticación-cifrado, CBC se aplica al payload, a los datos asociados y al nonce para generar un código de la autenticación del mensaje (MAC); entonces, el cifrador CRT se aplica al MAC y al payload para transformarlos en una forma ilegible, llamada texto cifrado.

En el proceso descifrado-verificación, el descifrado CRT se aplica al texto cifrado para recuperar el MAC y el payload; entonces, CBC se aplica al payload, los datos asociados y al nonce recibidos para verificar el MAC. La verificación exitosa proporciona que la información recibida proviene de una fuente con acceso a la llave.

5.4.1. Etapa Autenticación

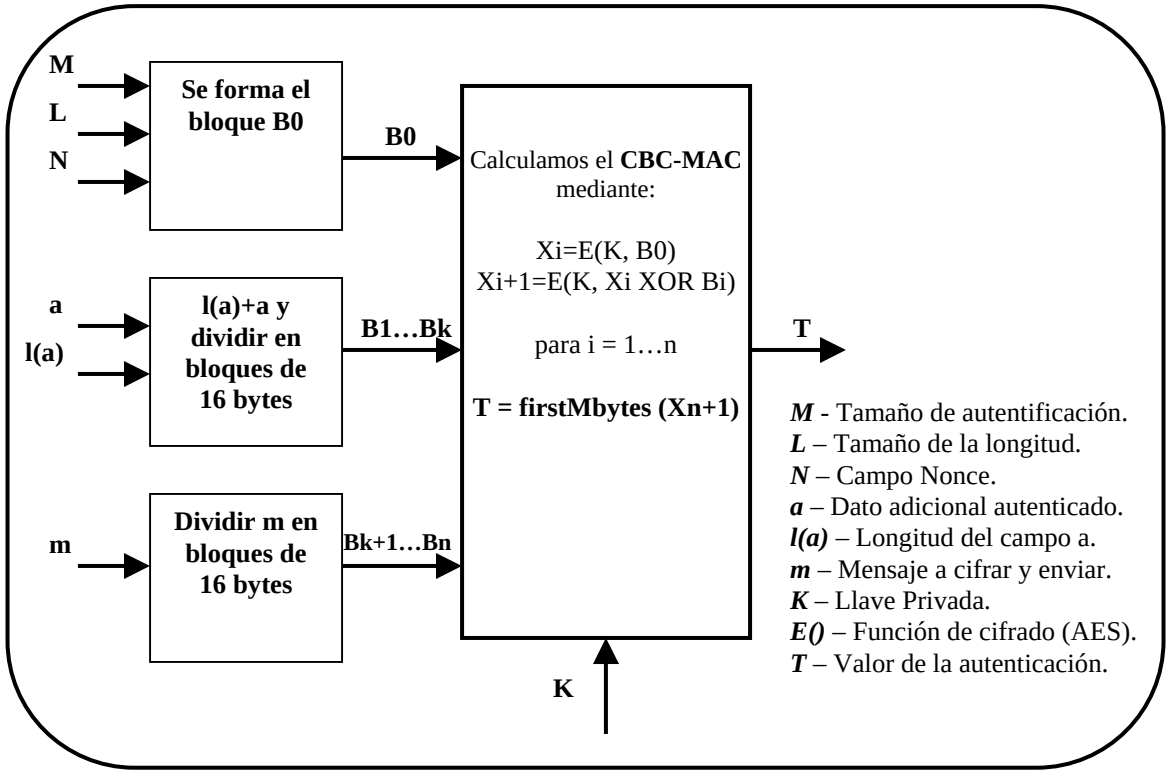


Figura 5.8. Autenticación AES-CCM.

La figura 5.8 muestra la *etapa de Autenticación*, el cual necesita como parámetros *M*, *L*, *N* y las entradas *a*, *l(a)* y *m*.

Siendo *M* el tamaño del campo autenticación, valores validos: 4, 6, 8, 10, 12, 14 y 16 bytes. Mientras *L* es el tamaño del campo longitud del mensaje, valores validos: rango entre 2 y 8 bytes. El parámetro *N* es el tamaño del campo Nonce que se obtiene mediante la siguiente operación 15-*L* bytes.

El dato *a* es el valor del dato adicional para autenticar siendo este campo opcional, es decir, puede tener un valor asignado o no ya que como su nombre lo dice es adicional. Además este dato adicional es autenticado pero no cifrado y no es incluido en la salida de este modo CCM. Este dato puede ser usado para autenticar las cabeceras de los paquetes del texto en claro. Siendo *l(a)* la longitud del dato *a*. La entrada *m* es el mensaje que va hacer cifrado y autenticado.

Con los parámetros de entrada *M*, *L*, *N* y los datos a y m se forma el bloque *B0*. El formato del bloque B0 es el siguiente:

Número de octetos:	0	1 … 15 - L	16 - L …15
Contenido:	Flags	Nonce N	l(m)

Para el bloque B0, el formato del campo Flags es el siguiente:

Número de Bit:	7	6	5	4	3	2	1	0
Contenido:	Reservado	Adata	M			L		

El bit *Reservado* es reservado para expansiones futuras y se fija siempre en cero. El bit *Adata* se fija a cero si l(a)=0 y se fija a uno si l(a)>0, *Adata* es el dato adicional para autenticar. *M* codifica el valor de *M* como *(M-2)/2*. *L* codifica el tamaño de la longitud del mensaje. El campo *L* puede tomar valores desde 2 hasta 8 bytes, siendo *L=1* reservado.

Los bloques *B1* hasta *Bk* se forman con *a* y *l(a)*, se codifica *l(a)* en un determinado número de bytes y se concatena a *a*. Los bloques *Bk+1* hasta *Bn* se forman con el mensaje de entrada *m*, el cual se divide en bloques de 16 bytes rellenando con ceros si es necesario.

Se autentican los datos usando el CBC-MAC, utilizando el procedimiento que se muestra en la figura 4.10, donde *E(k,Bi)*, representa el uso del cifrador AES con su llave *K* y como entrada la información del bloque *Bi*. Su salida es almacenada en *Xi*, y *Xi+1* incluye la operación *XOR* entre *Bi* y *Bi+1*.

Del *Xn+1* resultante se toman los primeros *M* bytes y se almacenan en *T*, que se denomina como el valor de la autenticación. El valor T obtenido es enviado al proceso de cifrado para calcular U que es el valor de la autenticación cifrado.

5.4.2. Etapa Cifrado

En la *etapa del Cifrado*, se calculan los bloques *Ai*, los cuales sirven como entrada para el modo de cifrado *CRT*. Para cifrar se usa el proceso mostrado en la figura 5.9, donde *E(k,Ai)* significa el cifrado con AES del bloque *Ai* usando la llave *K*.

La salida de AES se denomina bloques Si. Del bloque *S0* resultante se toman los primeros *M* bytes operación *XOR* con *T*, obteniendo *U* que se denomina como el valor de la autenticación cifrado. Los bloques *Si* menos el bloque *S0* son usados para hacer la operación de *XOR* con el mensaje *m*, dando como resultado *c* que corresponde a la información cifrada. *U* y *c* son enviados al receptor para que sean descifrados y verificados.

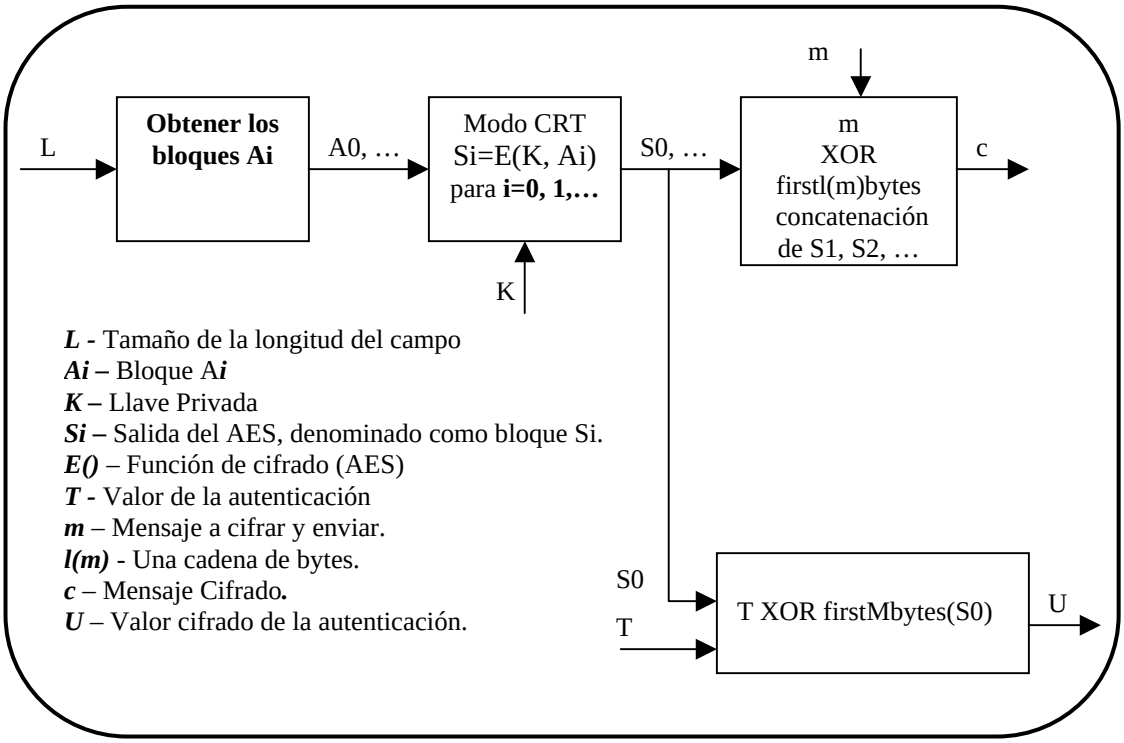


Figura 5.9. Cifrado AES-CCM.

5.4.3. Etapa Descifrado

En la *etapa del descifrado* es el proceso es muy similar al del cifrado, la variación se explica a continuación. Para descifrar se usa el proceso mostrado en la figura 5.10, donde *E(k,Ai)* significa el cifrado con AES del bloque *Ai* usando la llave *K*.

La salida de AES se denomina bloques Si. Del S0 resultante se toman los primeros M bytes operación XOR con U, obteniendo T que se denomina como el valor de la autenticación. Los bloques Si menos el bloque S0 son usados para hacer la operación de XOR con el mensaje cifrado c, dando como resultado m que corresponde a la información original.

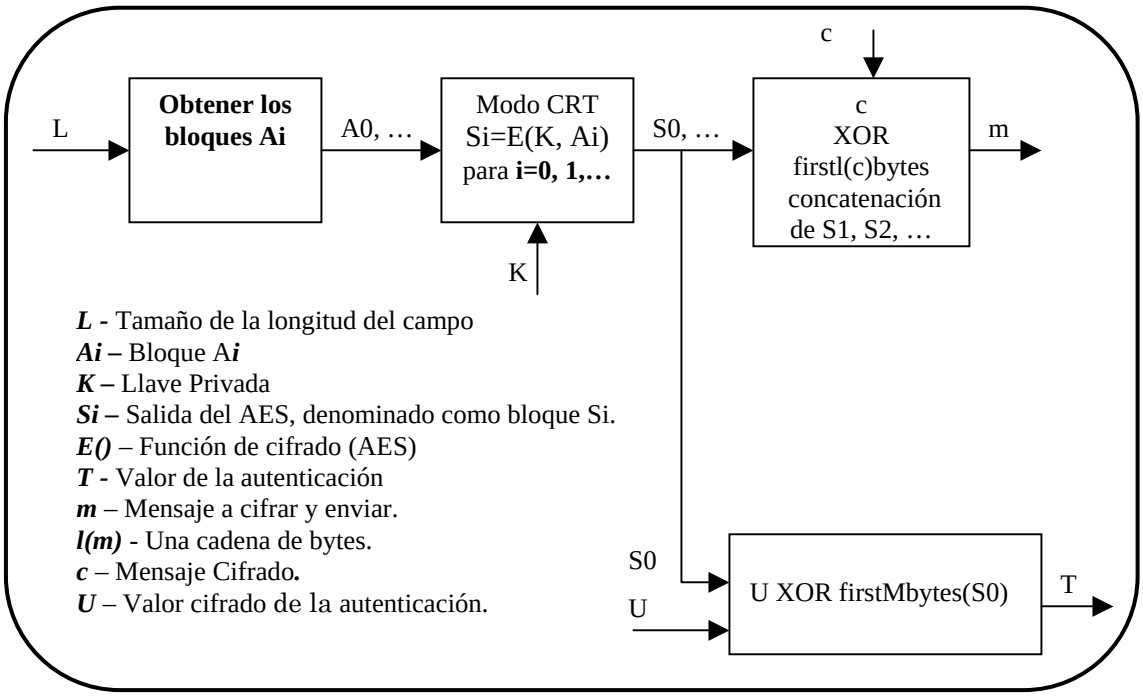


Figura 5.10. Descifrado AES-CCM.

5.4.4. Etapa Verificación

Una vez obtenidos *T* y *m*, se realiza el *proceso de verificación* mostrado en la figura 5.11, el cual es exactamente igual al de autenticación. Obteniendo *T'*, la cual se compara con el valor *T* obtenido en la etapa de descifrado, si *T'* es igual a *T* entonces la información verifico y puede ser utilizada por el receptor, en caso contrario la información se desecha y notifica al emisor que *T* no verifico.

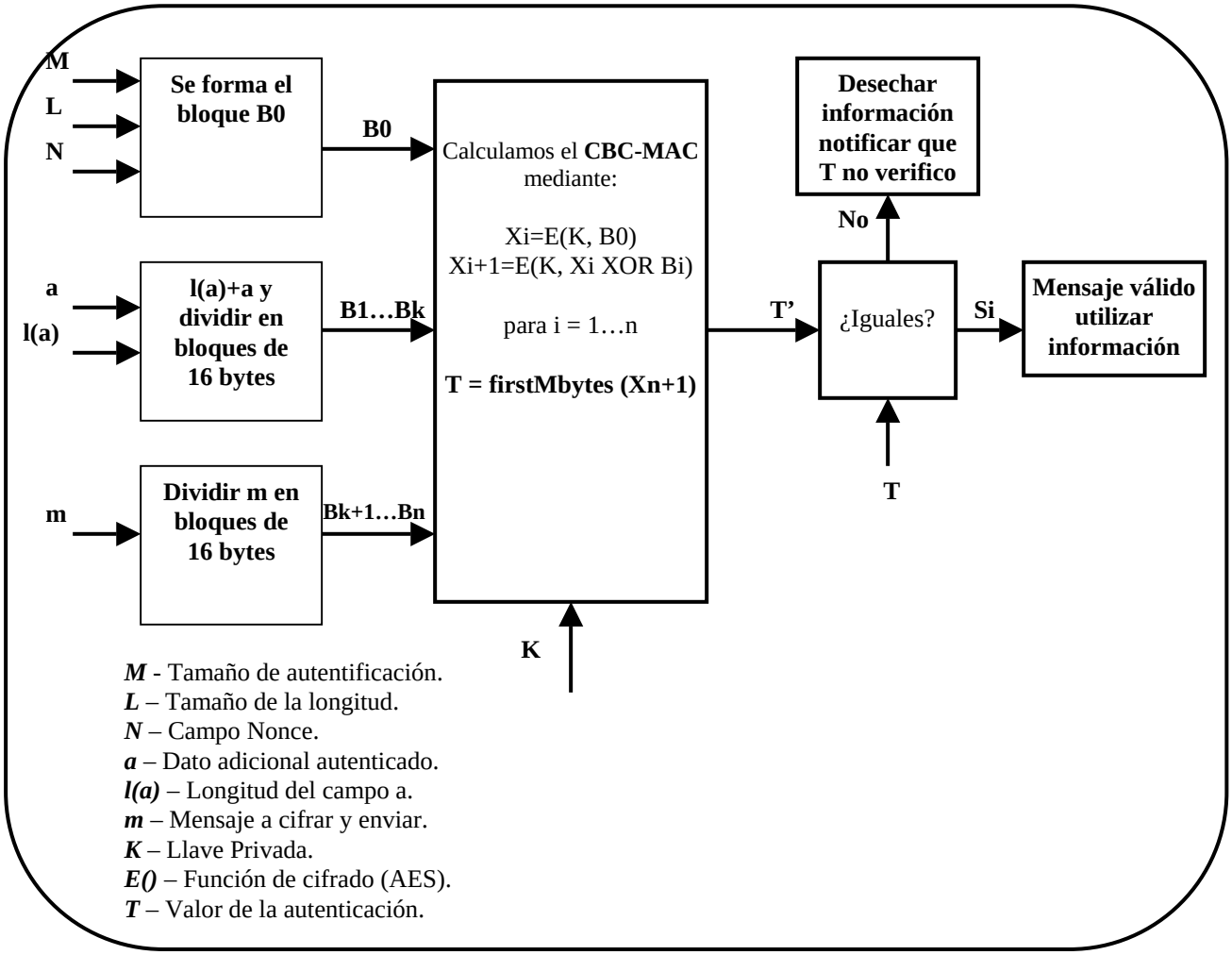


Figura 5.11. Verificación AES-CCM.

CAPÍTULO 6

6. MODELADO EN UML

Este proyecto de tesis utiliza el Proceso Unificado de Desarrollo de Software [39], el cual nos permite construir los siguientes modelos: análisis, diseño, implementación y prueba.

El Proceso Unificado utiliza el Lenguaje de Modelo Unificado (Unified Modeling Language, UML) [40], para modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

La base de UML son los diagramas. Cada diagrama usa la anotación pertinente y la suma de estos diagramas crean las diferentes vistas. Las vistas existentes en UML son:

- **Diagrama de casos de uso:** Muestra los casos de uso, actores y sus relaciones. Es decir quien puede hacer que y que relaciones existen entre acciones. Su función es modelar y organizar el comportamiento del sistema.
- **Diagrama de clases:** Muestra las clases, interfaces, colaboraciones y sus relaciones. Dan una vista estática del proyecto.
- **Diagrama de secuencia:** Muestra a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos y el manejo de la información.

6.1. Modelado IEEE 802.11

Se presenta el modelado [41] en UML del estándar IEEE 802.11 en modo DCF (Función de Coordinación Distribuida). Este modelo está formado por los modelos de análisis y diseño. El *modelo de análisis* lo componen los diagramas de casos de uso y de clases. El *modelo de diseño* los diagramas de clases refinadas y secuencia. Este modelo fue realizado en la herramienta Rational Rose 2000 Enterprise Edition.

La figura 6.1 muestra el diagrama de casos de uso donde se identifica un actor que es una Capa Superior. Este actor accede a los servicios de la subcapa MAC, en específico a los servicios de estación (SS), en este caso sólo a la entrega de paquetes ya que el resto del proceso es transparente para el actor. Note que el servicio de Entrega MSDU (MAC Service Data Unit) hace uso del servicio de Confidencialidad para utilizar el algoritmo de cifrado WEP, siendo este proceso opcional debido a que se puede o no utilizar tal y como lo define el éstandar IEEE 802.11.

La figura 6.2 muestra el diagrama de casos de uso donde se identifica un actor que es una estación (STA). Ésta accede a los servicios de la subcapa MAC utilizando tanto los servicios SS, si sólo es una STA, o los servicios del Sistema de Distribución (DSS), siendo un Punto de Acceso (AP). El servicio de Asociación hace uso del servicio de Autenticación, debido a que una STA debe ser autenticada antes de ser asociada; por lo tanto Desasociación utiliza Deautenticación para realizar el proceso inverso. El servicio de Entrega MSDU hace uso del servicio de Confidencialidad para utilizar el algoritmo de cifrado WEP.

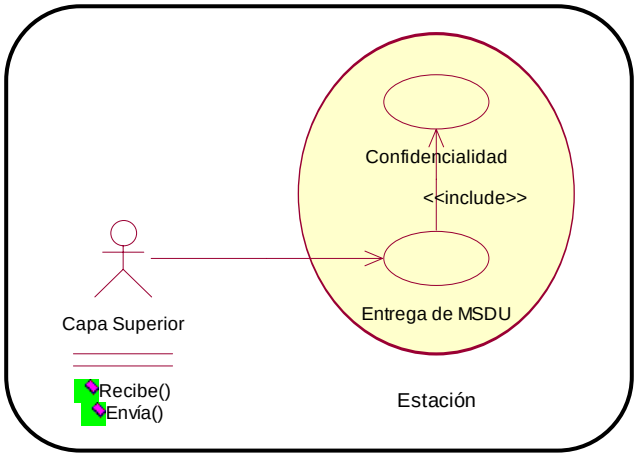


Figura 6.1. Diagrama de Casos de Uso de la Capa Superior.

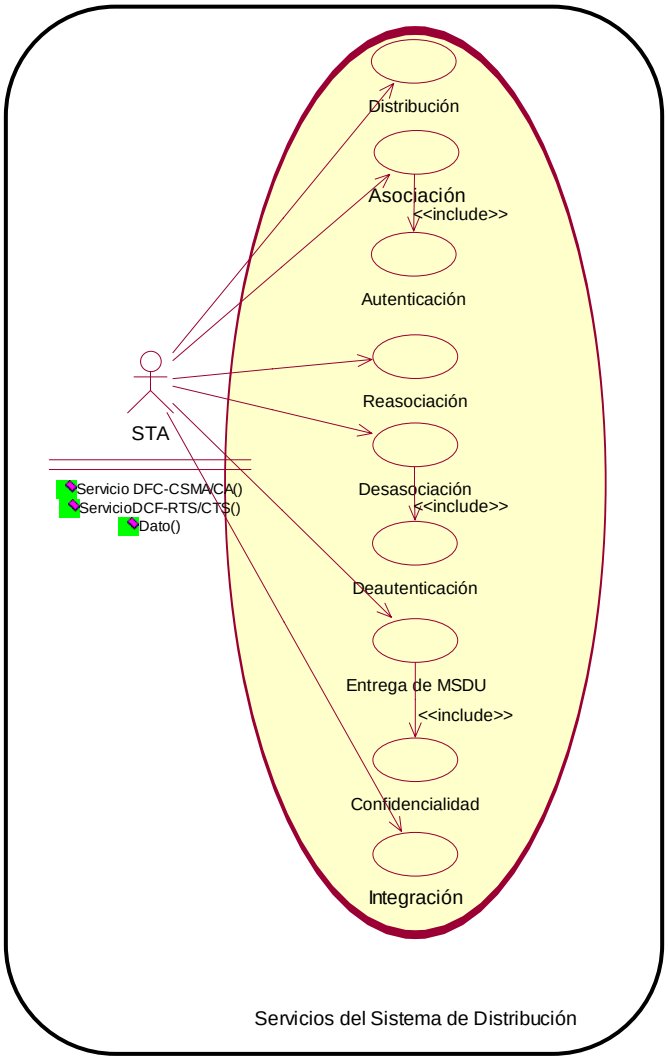


Figura 6.2. Diagrama de Casos de Uso de un Punto de Acceso.

El diagrama de clases mostrado en la figura 6.3 es el modelo del Estándar IEEE 802.11. Este modelo muestra su funcionamiento a nivel de subcapa MAC utilizando el modo DCF. Las clases que conforman el diagrama son las siguientes:

- **MAC:** Implementa todos los servicios definidos por la subcapa MAC, es decir SS y DSS. A nivel MAC el estándar define dos métodos de acceso al medio, uno centralizado (Función de Coordinación Puntual, PCF) y otro distribuido (DCF).

- **PCF:** Representa la Función de Coordinación Puntual [1, 42]. Esta función no es implementada.

- **DCF:** Implementa la Función de Coordinación Distribuida con CSMA/CA y RTS/CTS. En este modo de acceso al medio se centra el trabajo de tesis.

- **SeguridadWEP:** Implementa el protocolo WEP, el algoritmo de cifrado RC4 y el algoritmo de detección de errores CRC-32, para ofrecer el servicio de confidencialidad e integridad definidos por el estándar IEEE 802.11.

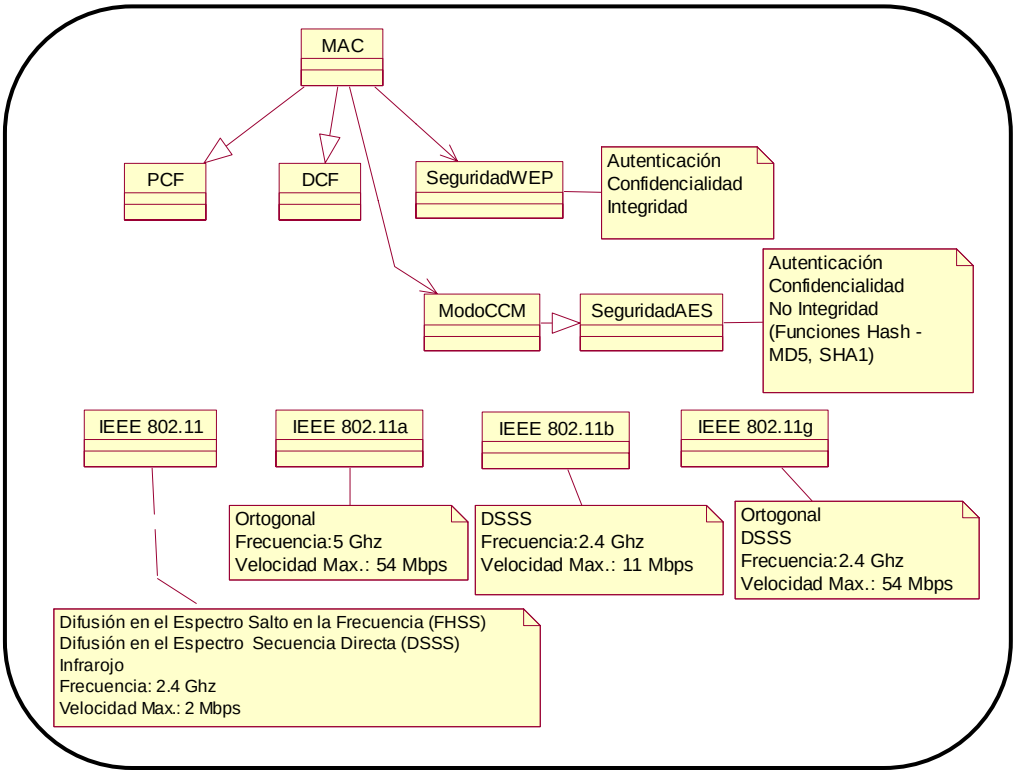


Figura 6.3. Diagrama de Clases del IEEE 802.11 y AES-CCM.

La figura 6.4 muestra la clase MAC refinada del diagrama de clases del IEEE 802.11 en modo DCF, con los atributos y métodos de la clase MAC, en la tabla 6.1 se explica el funcionamiento de cada atributo y en la tabla 6.2 se explica el funcionamiento de cada método definido.

La clase MAC implementa todos los servicios definidos por la subcapa MAC, es decir SS y DSS.

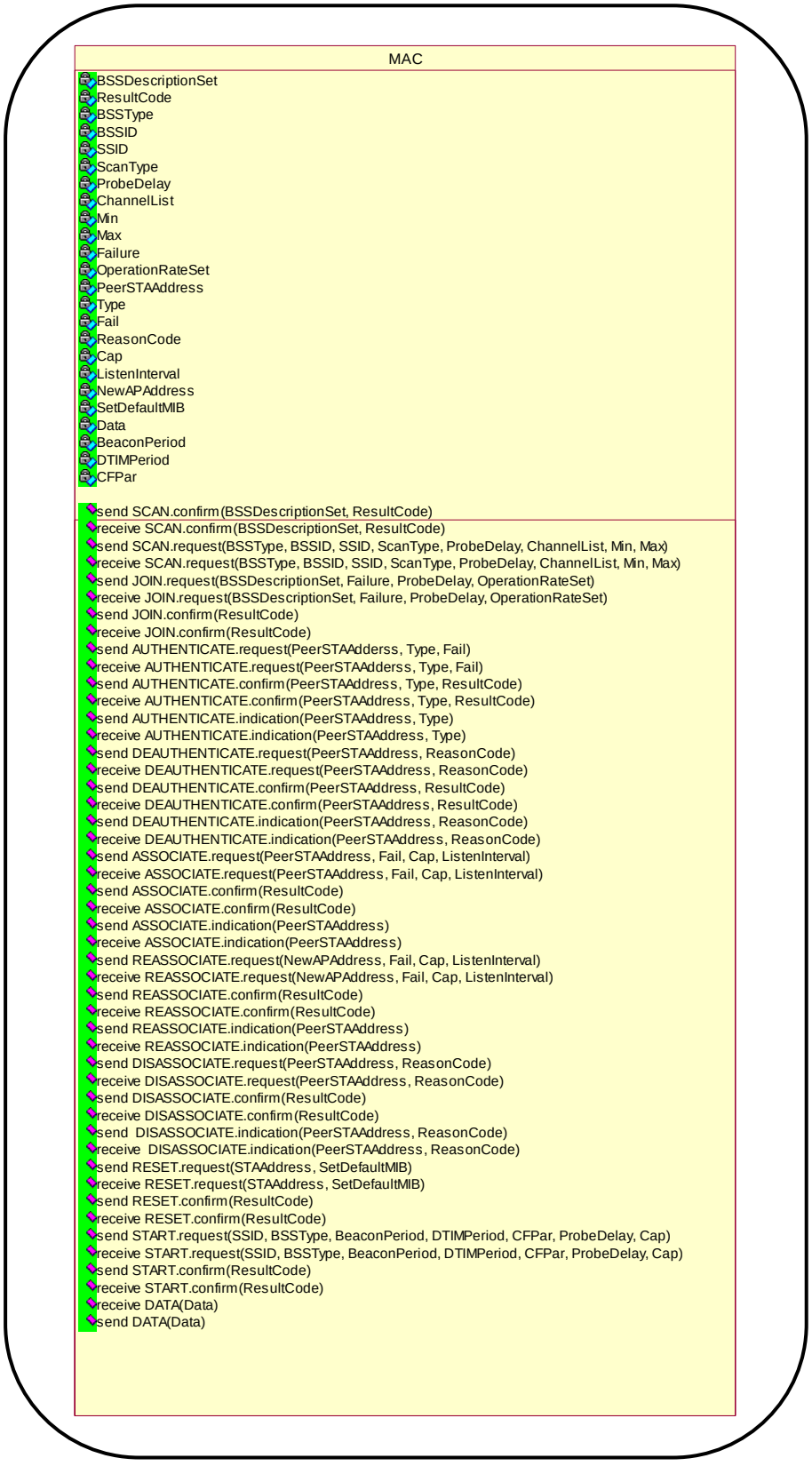


Figura 6.4. Clase Refinada MAC.

La tabla 6.1 muestra a detalle cada uno de los atributos definidos en la Clase MAC.

Clase MAC	
Nombre del Atributo	*Función*
BSSDescriptionSet	Indica el resultado de la primitiva de servicio SCAN.request
ResultCode	Indica el resultado de la primitiva de servicio SCAN.confirm Indica el resultado de la primitiva de servicio JOIN.request Indica el resultado de la primitiva AUTHENTICATE.request Indica el resultado de DEAUTHENTICATE.request Indica el resultado de ASSOCIATE.request Indica el resultado de REASSOCIATE.request Indica el resultado de DISASSOCIATE.request Indica el resultado de RESET.request Indica el resultado de START.request
BSSType	Determina la infraestructura BSS
BSSID	Identifica un BSSID específico o Broadcast
SSID	Especifica el SSID deseado o SSID Broadcast
ScanType	Indica la exploración activa o pasiva
ProbeDelay	Retrasa (en μs) para ser utilizado antes de transmitir una trama de prueba durante la exploración activa
ChannelList	Especifica una lista de los canales que se examinan al explorar un BSS
Min	El tiempo mínimo (en TU) utilizado en cada canal al explorar
Max	El tiempo máximo (en TU) utilizado en cada canal al explorar
Failure	Indica el tiempo límite en el cual la primitiva debio haber terminado
OperationRateSet	Indica el grupo de rango de datos en unidades de 500 kbits/s que una estación puede utilizar para comunicarse en un Grupo de Servicios Básicos (BSS)
PeerSTAAddress	Especifica la dirección MAC de la entidad
Type	Indica el tipo de trama
Fail	Indica si la operación falla
ReasonCode	Especifica la rázon por la cual se va a realizar el método DEAUTHENTICATE Indica la rázon por el cual fue iniciado el método DEAUTHENTICATE Especifica la rázon por la cual se va a realizar el método DISASSOCIATE Indica la rázon por el cual fue iniciado el método DISASSOCIATE
Cap	Especifica la capacidad operacional para ser usada por la entidad MAC
ListenInterval	Especifica el número de intervalos beacon antes de que una estación despierte y escuche el siguiente beacon
NewAPAddress	Especifica la dirección de la entidad MAC para realizar la REASSOCIATE
SetDefaultMIB	Si el valor es TRUE, todos los atributos MIB se les asignan los valores por default. Si el valor es FALSE, se reinicia el MAC
Data	Almacena la información
BeaconPeriod	Período de tiempo beacon
DTIMPeriod	Período de tiempo DTIMP
CFPar	Indica control

Tabla 6.1. Atributos definidos en la Clase MAC.

La tabla 6.2 muestra a detalle cada uno de los métodos definidos en la Clase MAC.

Clase MAC	
Nombre del Método	*Función*
send SCAN.confirm	Retorna las descripciones del sistema BSSs detectado por SCAN
receive SCAN.confirm	Retorna las descripciones del sistema BSSs detectado por SCAN
send SCAN.request	Solicita un SCAN del BSSs que se pueda elegir más adelante
receive SCAN.request	Solicita un SCAN del BSSs que se pueda elegir más adelante
send JOIN.request	Solicita la sincronización con un BSS
receive JOIN.request	Esta primitiva solicita la sincronización con un BSS
send JOIN.confirm	Esta primitiva confirma la sincronización con un BSS
receive JOIN.confirm	Confirma la sincronización con un BSS

send AUTHENTICATE.request	Solicita autenticación con una entidad MAC específica
receive AUTHENTICATE.request	Solicita autenticación con una entidad MAC específica
send AUTHENTICATE.confirm	Reporta los resultados de la solicitud de autenticación
receive AUTHENTICATE.confirm	Reporta los resultados de la solicitud de autenticación
send AUTHENTICATE.indication	Reporta el establecimiento de la autenticación
receive AUTHENTICATE.indication	Reporta el establecimiento de la autenticación
send DEAUTHENTICATE.request	Solicita la deautenticación
receive DEAUTHENTICATE.request	Solicita la deautenticación
send DEAUTHENTICATE.confirm	Reporta los resultados de la solicitud de deautenticación
receive DEAUTHENTICATE.confirm	Reporta los resultados de la solicitud de deautenticación
send DEAUTHENTICATE.indication	Reporta la deautenticación con una entidad MAC específica
receive DEAUTHENTICATE.indication	Reporta la deautenticación con una entidad MAC específica
send ASSOCIATE.request	Solicita la asociación con una entidad que esté actuando como AP
receive ASSOCIATE.request	Solicita la asociación con una entidad que esté actuando como AP
send ASSOCIATE.confirm	Reporta los resultados de la solicitud de asociación
receive ASSOCIATE.confirm	Reporta los resultados de la solicitud de asociación
send ASSOCIATE.indication	Reporta el establecimiento de la asociación con una entidad MAC
receive ASSOCIATE.indication	Reporta el establecimiento de la asociación con una entidad MAC
send REASSOCIATE.request	Solicita una reasociación
receive REASSOCIATE.request	Solicita una reasociación
send REASSOCIATE.confirm	Reporta los resultados de la solicitud de reasociación
receive REASSOCIATE.confirm	Reporta los resultados de la solicitud de reasociación
send REASSOCIATE.indication	Reporta el establecimiento de la reasociación
receive REASSOCIATE.indication	Reporta el establecimiento de la reasociación
send DISASSOCIATE.request	Solicita la desasociación a un AP
receive DISASSOCIATE.request	Solicita la desasociación a un AP
send DISASSOCIATE.confirm	Reporta los resultados de la solicitud de desasociación
receive DISASSOCIATE.confirm	Reporta los resultados de la solicitud de desasociación
send DISASSOCIATE.indication	Reporta el establecimiento de la desasociación
receive DISASSOCIATE.indication	Reporta el establecimiento de la desasociación
send RESET.request	Esta primitiva solicita que la entidad MAC se reajuste
receive RESET.request	Esta primitiva solicita que la entidad MAC se reajuste
send RESET.confirm	Esta primitiva reporta los resultados de la solicitud de reajuste
receive RESET.confirm	Esta primitiva reporta los resultados de la solicitud de reajuste
send START.request	Solicita que la entidad MAC inicie en un nuevo BBS
receive START.request	Solicita que la entidad MAC inicie en un nuevo BBS
send START.confirm	Reporta los resultados de la solicitud de inicio de un nuevo BBS
receive START.confirm	Reporta los resultados de la solicitud de inicio de un nuevo BBS
receive DATA(Data)	Define la transferencia de datos de la MAC a la entidad local PHY
send DATA(Data)	Define la transferencia de datos de la MAC a la entidad local PHY

Tabla 6.2. Métodos definidos en la Clase MAC.

La figura 6.5 muestra la clase refinada DCF del diagrama de clases del IEEE 802.11, con los atributos y métodos de la clase DCF, en la tabla 6.3 se explica el funcionamiento de cada atributo y en la tabla 6.4 se explica el funcionamiento de cada método definido. La clase DCF implementa la Función de Coordinación Distribuida con CSMA/CA y RTS/CTS.

La tabla 6.3 muestra a detalle cada uno de los atributos definidos en la Clase DCF.

Clase DCF	
Nombre del Atributo	*Función*
DIFS	Define el tiempo de espera DIFS (espacio de intertrama)
SIFS	(Short IFS). Define el período más corto. Se transmiten los reconocimientos
FrameControl	Define la trama de control
Duration	Define el campo Duration/ID contiene el tiempo de reserva para la transmisión
ReceiverAddress	Contiene la dirección receptora

TransmitterAddress	Almacena la dirección donde se va a realizar la transmisión
FCS	Tiene el valor de Secuencia de chequeo de trama
Data	Guarda el Data
RTS	Recopila el tiempo en que se realizó la señal Request to Send
CTS	Obtiene el tiempo en que se realizó la señal Clear to Send
ACK	Tiempo en que se realizó la señal de Reconocimiento
CW	Obtiene el valor actual de la Ventana de Contención
State	Valor booleano, que sirve para informar si el medio se encuentra libre u ocupado

Tabla 6.3. Atributos definidos en la Clase DCF.

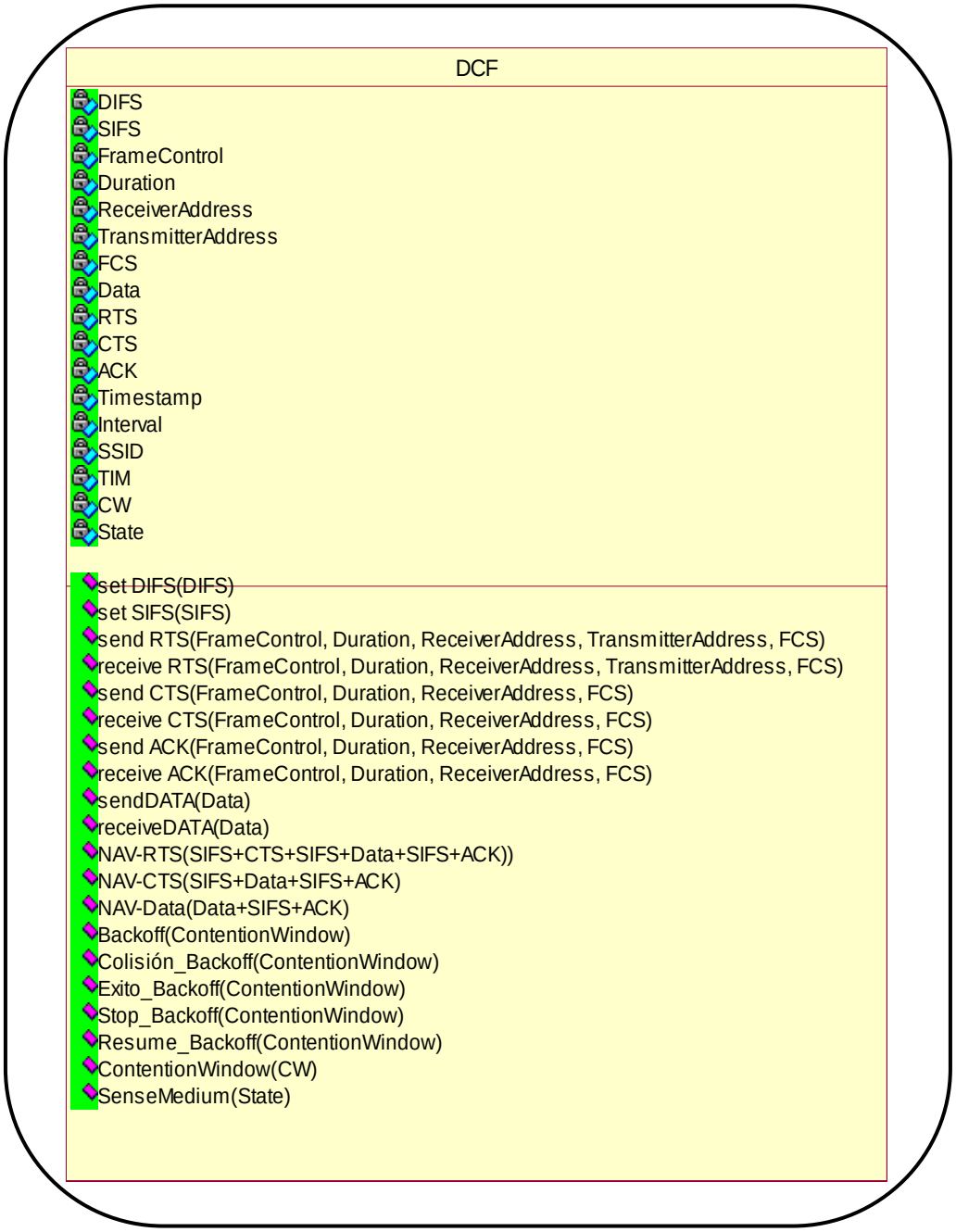


Figura 6.5. Clase Refinada DCF.

La tabla 6.4 muestra a detalle cada uno de los métodos definidos en la Clase DCF.

Clase DCF	
Nombre del Método	*Función*
set DIFS	Método que ejecuta un tiempo de espera DIFS
set SIFS	Activa el tiempo de espera SIFS
send RTS	Envío de la señal Request to Send
receive RTS	Recepción de la señal Request to Send
send CTS	Envío de la señal Clear to Send
receive CTS	Recepción de la señal Clear to Send
send ACK	Envío del Reconocimiento
receive ACK	Recepción del Reconocimiento
sendDATA	Envío de la información
receiveDATA	Recepción de la información
NAV-RTS	Calcula el Network Allocation Vector para la señal Request to Send
NAV-CTS	Obtiene el Network Allocation Vector para la señal Clear to Send
NAV-Data	Determina el Network Allocation Vector para Data
Backoff	Ejecuta el algoritmo de Backoff
Colisión_Backoff	Se ejecuta este método cuando existe una colisión
Exito_Backoff	Se ejecuta este método cuando todo funciona correctamente
Stop_Backoff	Se detiene el algoritmo de Backoff
Resume_Backoff	Se reinicia el algoritmo de Backoff
ContentionWindow	Calcula el valor de la Ventana de Contención
SenseMedium	Método que obtiene el estado actual del medio o canal mediante un sensado

Tabla 6.4. Métodos definidos en la Clase DCF.

La figura 6.6 muestra la clase refinada SeguridadWEP que implementa el protocolo WEP, el algoritmo de cifrado RC4 y el algoritmo de detección de errores CRC-32. En la tabla 6.5 se explica el funcionamiento de cada atributo definido en la clase SeguridadWEP y en la tabla 6.6 se explica el funcionamiento de cada método definido en la clase SeguridadWEP.

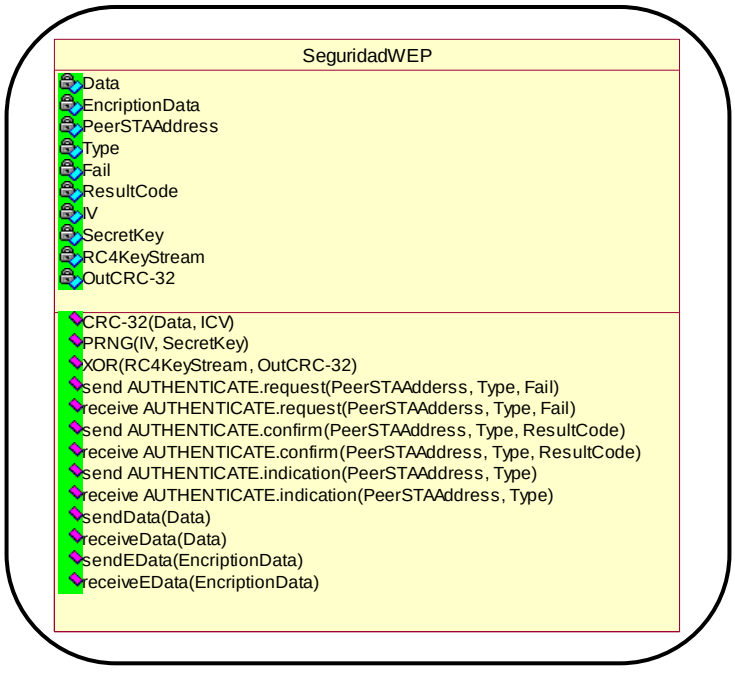


Figura 6.6. Clase Refinada SeguridadWEP.

La tabla 6.5 muestra a detalle cada uno de los atributos definidos en la Clase SeguridadWEP.

Clase SeguridadWEP	
Nombre del Atributo	*Función*
Data	Almacena la información sin cifrar
EncriptionData	Guarda la información cifradad
PeerSTAAddress	Contiene la dirección de la estación
Type	Se define el tipo
Fail	Almacena erl valor del campo Fail
ResultCode	Código del resultado
IV	Recopila el vector de inicialización
SecretKey	Tiene el valor de la llave privada
RC4KeyStream	Acumula el resultado de RC4
OutCRC-32	Obtiene el resultado del método CRC-32

Tabla 6.5. Atributos definidos en la Clase SeguridadWEP.

La tabla 6.6 muestra a detalle cada uno de los métodos definidos en la Clase SeguridadWEP.

Clase SeguridadWEP	
Nombre del Método	*Función*
CRC-32	Método que implementa el CRC-32
PRNG	Implementa el Generador de Número Pseudoaleatorio
XOR	Operación de XOR entre el Vector de Inicialización (IV) y Llave Privada(Secretkey)
send AUTHENTICATE.request	Señal de envío, solicita autenticación
receive AUTHENTICATE.request	Señal de recepción de solicitud para la autenticación
send AUTHENTICATE.confirm	Señal de envío, reporta los resultados para realizarse la autenticación
receive AUTHENTICATE.confirm	Señal de recepción para reportar resultados de la autenticación
send AUTHENTICATE.indication	Señal de envío, establece la autenticación
receive AUTHENTICATE.indication	Señal de recepción para establecer la autenticación
sendData	Señal de envío de información sin cifrar
receiveData	Señal de recepción de información sin cifrar
sendEData	Señal de envió de información cifrada
receiveEData	Señal de recepción de información cifrada

Tabla 6.6. Métodos definidos en la Clase SeguridadWEP.

La figura 6.7 muestra el diagrama de secuencia del DCF utilizando CSMA/CA (Acceso Múltiple por Sensado de Portadora Evitando Colisiones) a nivel de la subcapa MAC con una configuración Ad-Hoc.

Se tiene una STA emisora que desea acceder al medio para realizar la transmisión de información, una STA receptora en espera de información y otra STA tratando de acceder al canal.

Una vez que la STA emisora encuentra el canal libre transmite la información y después de un intervalo de tiempo SIFS se envía el reconocimiento de haber recibido la información, todas las STAs esperan un intervalo de tiempo DIFS. En caso que el canal esté ocupado, se detiene el proceso de Backoff y se reanuda cuando la otra STA termina la transmisión de la información. Entonces nuevamente todas las estaciones pelean por el medio.

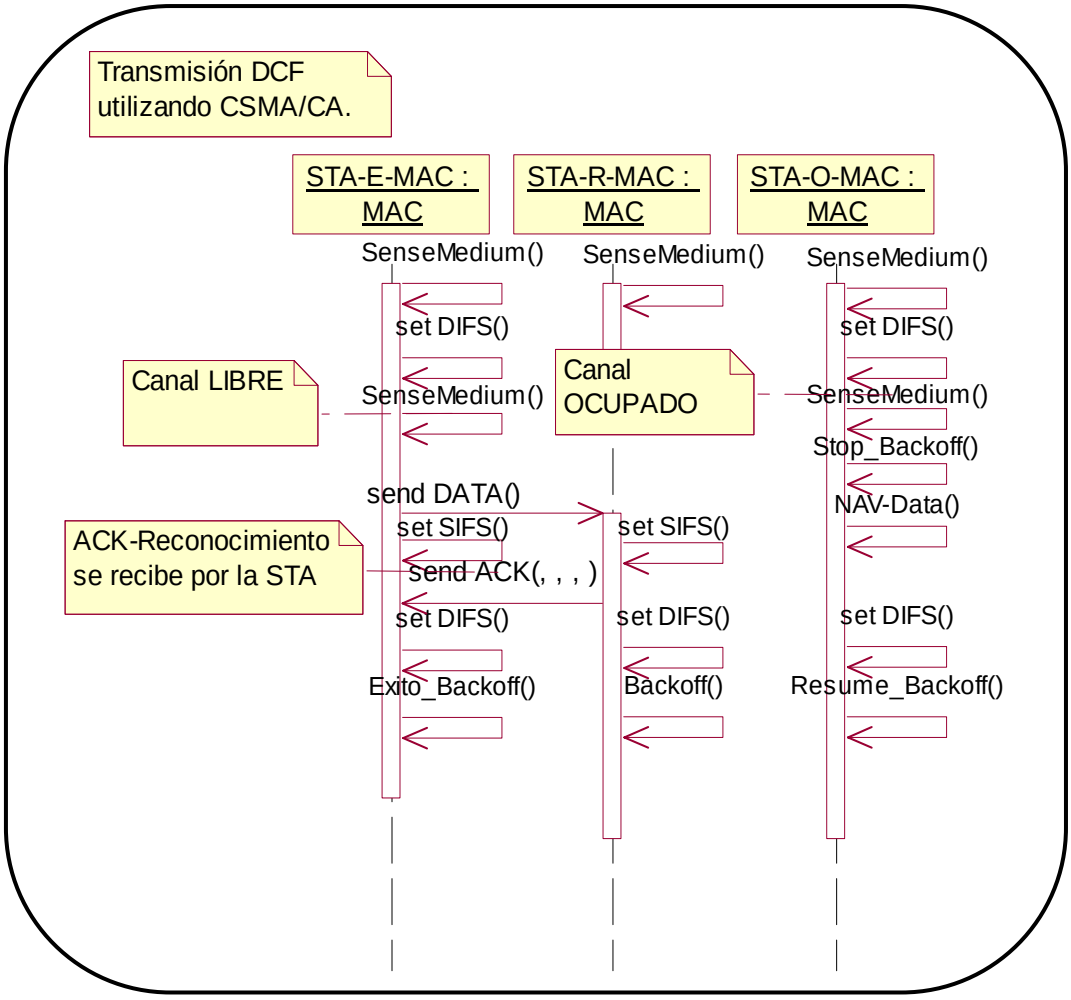


Figura 6.7. Diagrama de Secuencia del DCF usando CSMA/CA.

La figura 6.8 muestra el diagrama de secuencia del DCF utilizando CSMA/CA a nivel de la subcapa MAC con una configuración Ad-Hoc en el caso que el protocolo CSMA/CA no reciba la información (DATA) por algún problema.

Se pelean por el medio inalámbrico tres estaciones: la estación emisora, la estación receptora y otra estación.

Antes de transmitir información una STA debe sensar el medio, para determinar su estado (libre/ocupado). Si el medio esta libre se ejecuta un tiempo de espera DIFS, finalizando el tiempo se sensa el medio nuevamente, si se encuentra libre se transmite la información (DATA) por la estación que gane el medio en este caso la estación emisora. Debido a que DATA no es recibido durante el tiempo de espera SIFS, la estación receptora no transmite el reconocimiento a la estación emisora, durante el tiempo DIFS asignado. Una vez finalizada esta espera las STAs ejecutan el algoritmo de espera llamado Backoff.

En caso en que el medio este ocupado, se detiene el algoritmo Backoff, y se transmite un Network Allocation Vector (NAV), finalmente se reanuda Backoff.

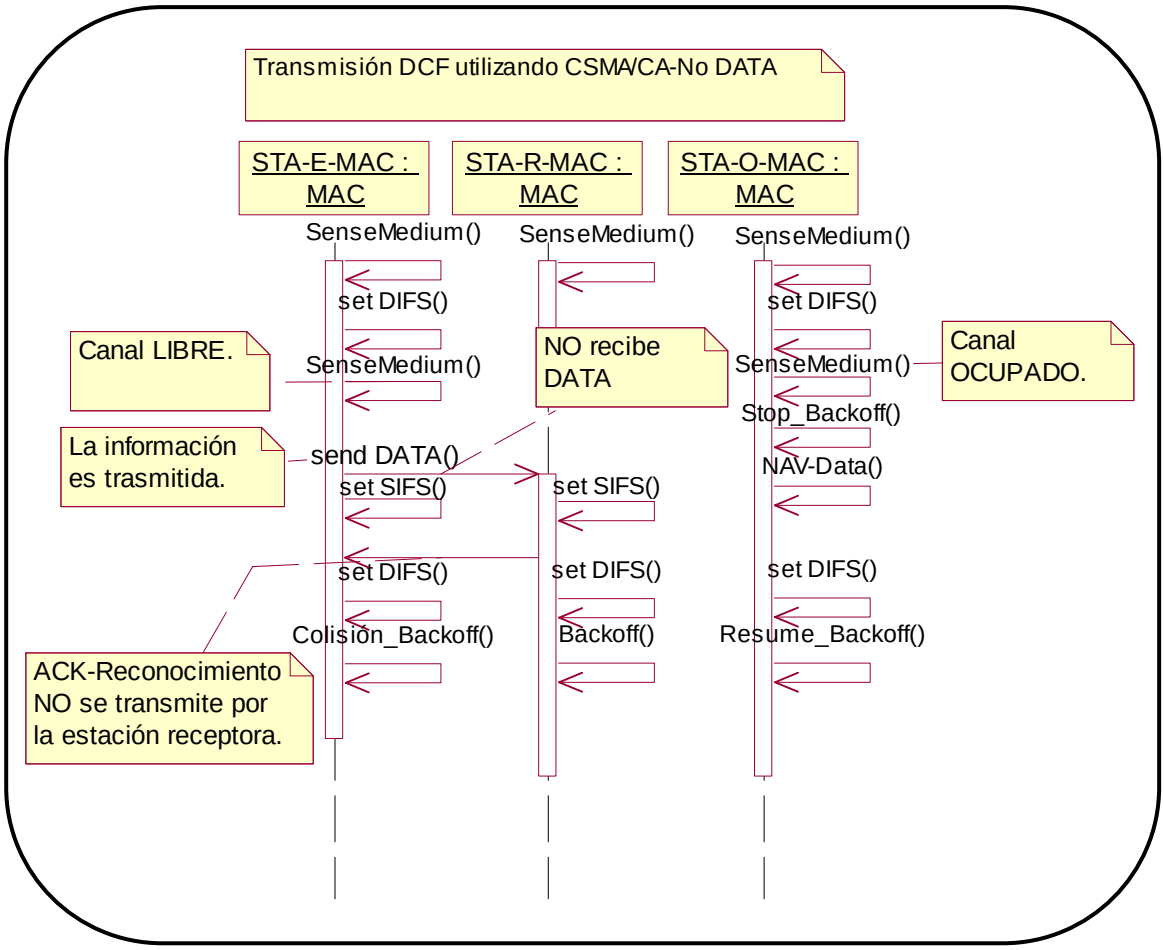


Figura 6.8. Diagrama de Secuencia del DCF usando CSMA/CA-No DATA.

La figura 6.9 muestra el diagrama de secuencia en el caso que el protocolo CSMA/CA no reciba el reconocimiento (ACK) por alguna cirscunstancia presentada.

Antes de transmitir información una estación debe sensar el medio, para determinar su estado (libre/ocupado). Si el medio esta libre se ejecuta un tiempo de espera DIFS, finalizando el tiempo se sensa el medio nuevamente, si se encuentra libre se transmite la información por la estación que gana el canal, en este caso la estación emisora. Las STAs emisora y receptora ejecutan un tiempo SIFS, al finalizar el tiempo la STA receptora envía el reconocimiento (ACK) si ésta recibió exitosamente la información. Las STAs ejecutan otro tiempo de espera DIFS, si durante este lazo de tiempo no se recibe el ACK entonces las STAs ejecutan el algoritmo de espera llamado Backoff.

En caso en que el medio este ocupado, se detiene el algoritmo Backoff, y se transmite un Network Allocation Vector (NAV), finalmente se reanuda Backoff.

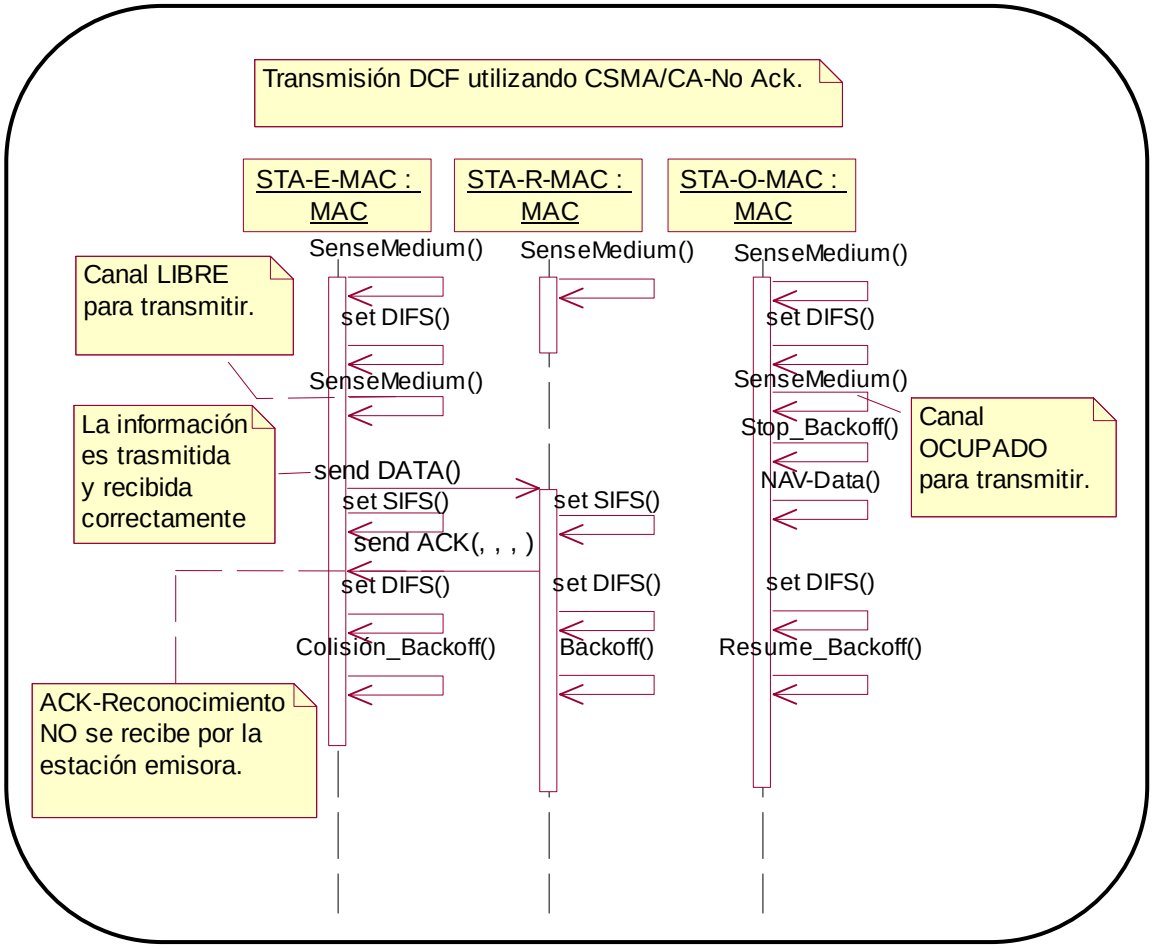


Figura 6.9. Diagrama de Secuencia del DCF usando CSMA/CA-No ACK.

La figura 6.10 muestra el diagrama de secuencia del DCF utilizando RTS/CTS a nivel de la subcapa MAC con una configuración Ad-Hoc.

Teniendo una STA emisora que desea acceder al medio para realizar su transmisión, una STA receptora en espera de información y otra STA tratando de acceder al canal.

Una vez que la STA emisora encuentra el canal libre transmite RTS siendo una solicitud para enviar información y después de un intervalo de tiempo SIFS, la estación receptora responde con CTS. Entonces CTS reserva el canal durante el período de tiempo en el que la estación emisora envía la información, consecutivamente se transmite la información y se espera un tiempo SIFS por el reconocimiento como confirmación de haber recibido la información, todas las STAs esperan un intervalo de tiempo DIFS.

En caso en que el medio este ocupado, se detiene el algoritmo Backoff, y se transmite un Network Allocation Vector (NAV), finalmente se reanuda Backoff.

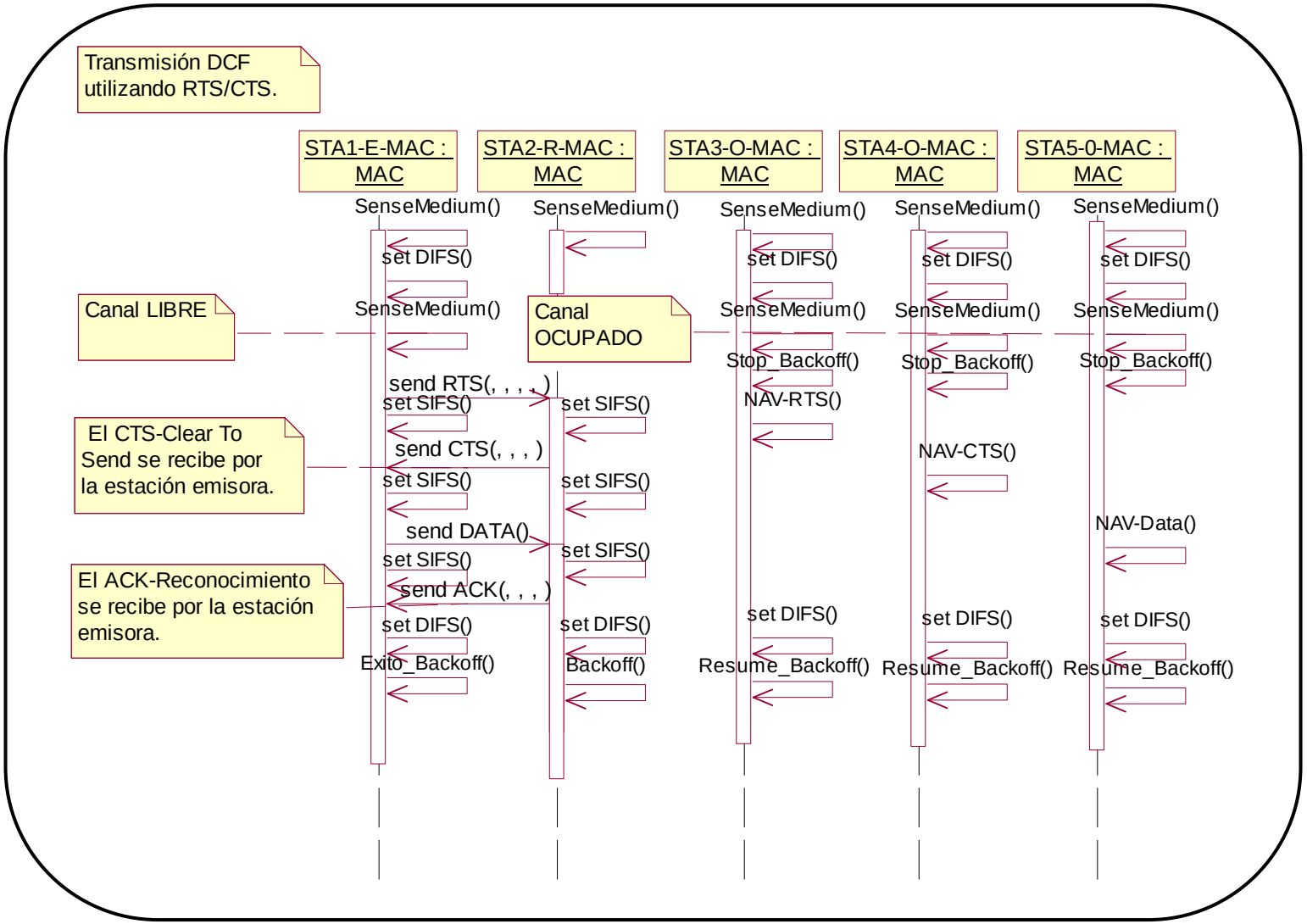


Figura 6.10. Diagrama de Secuencia del DCF usando RTS/CTS.

La figura 6.11 muestra el diagrama de secuencia utilizando el protocolo RTS/CTS cuando la solicitud de envio (RTS) no es recibida por la estación receptora.

Consecutivamente se usa el NAV para que una estación pueda conocer cuando la transmisión actual termina y el medio queda libre. Una vez que termina el tiempo de espera DIFS se ejecuta el algoritmo de Backoff.

El Backoff tiene como función reducir la probabilidad de colisión que se máximiza cuando varias estaciones están esperando que el medio o canal quede libre para poder transmitir.

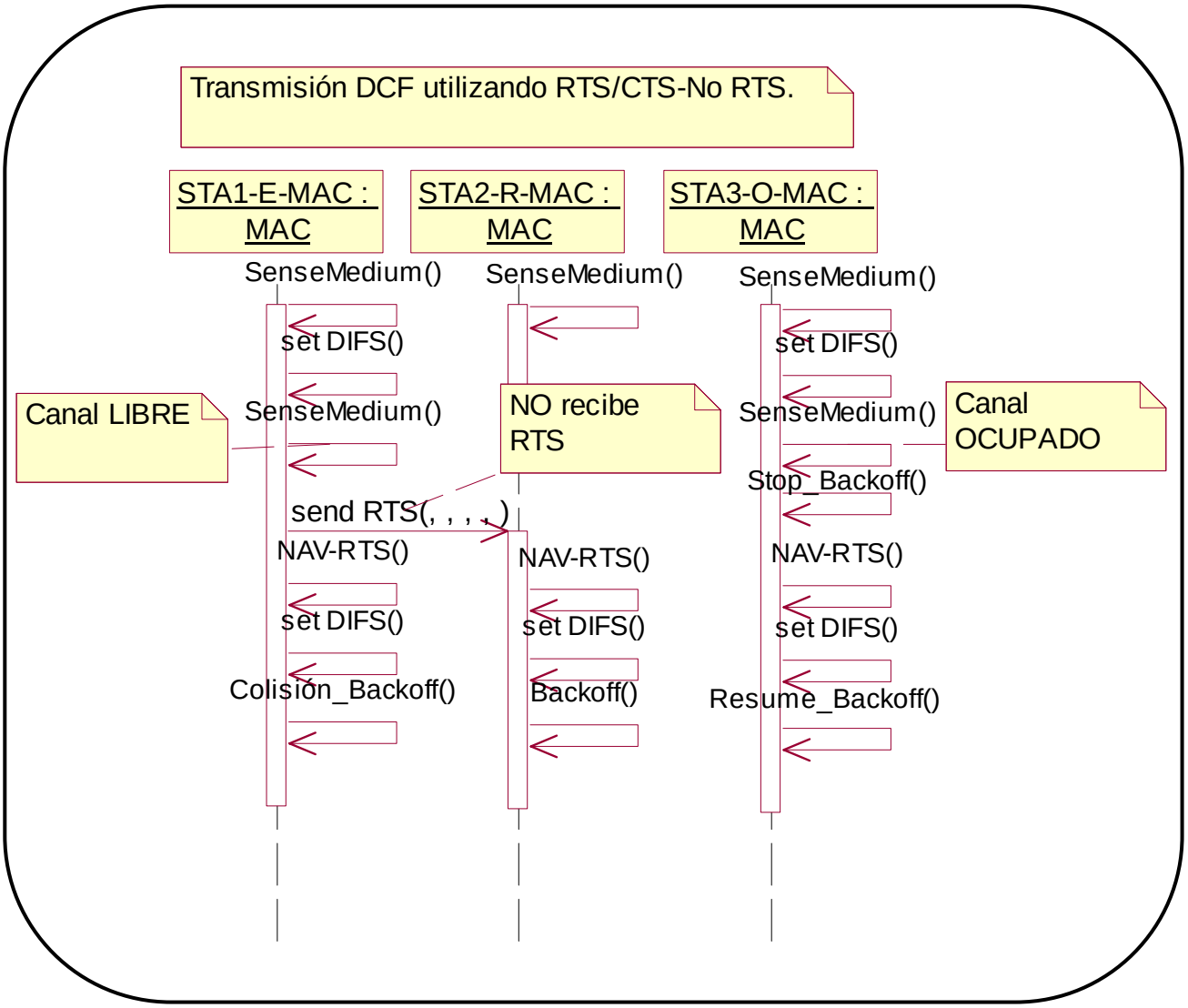


Figura 6.11. Diagrama de Secuencia del DCF usando RTS/CTS-No RTS.

La figura 6.12 muestra el diagrama de secuencia utilizando el protocolo RTS/CTS cuando receptora transmite un Clear To Send (CTS) y esta señal no es recibida por la estación emisora.

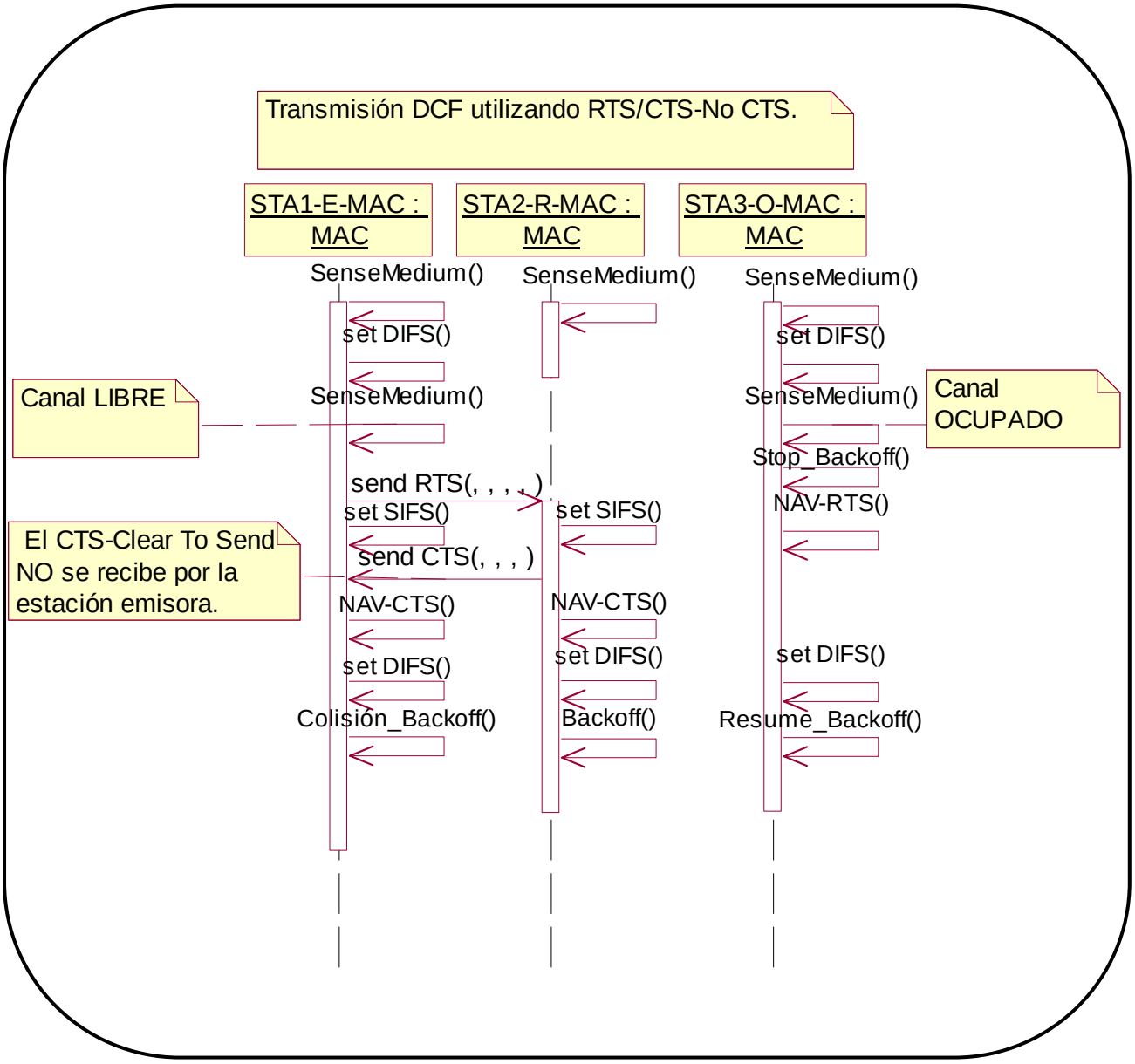


Figura 6.12. Diagrama de Secuencia del DCF usando RTS/CTS-No CTS.

Este diagrama (ver figura 6.13) muestra el funcionamiento del protocolo RTS/CTS teniendo como problema la recepción de la información.

Las estaciones pelean por acceder al medio, una vez que el canal está libre, la estación emisora envía un RTS siendo una solicitud de envío de información a la estación receptora (STA2-R-MAC: MAC). La estación responde con un CTS, reservando el canal para transmitir, por lo tanto la estación emisora trasmité la informción. Sin embargo en este caso DATA no es recibido por la estación receptora. Por lo tanto se ejecuta el NAV-Data consecutivamente el Backoff.

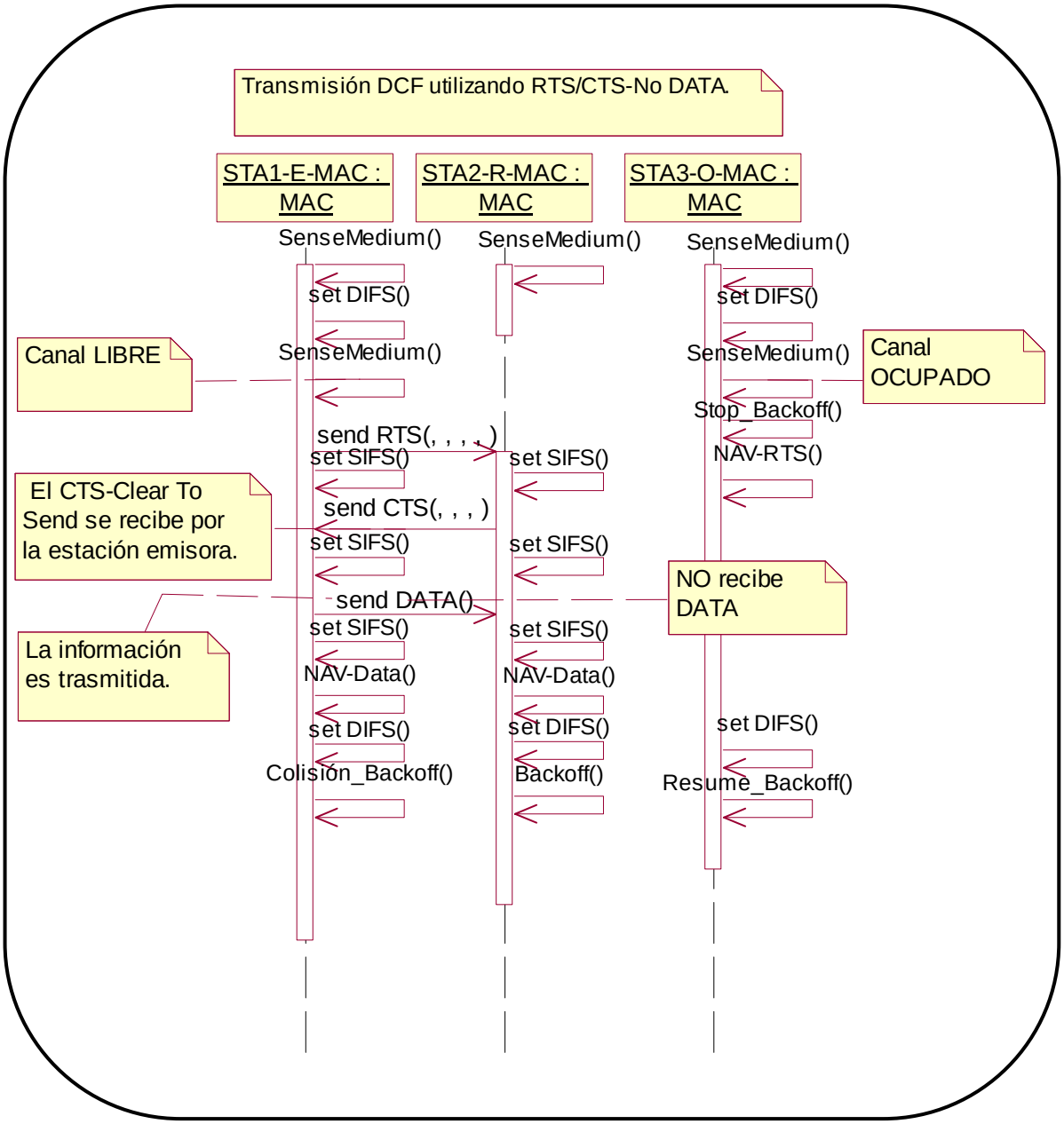


Figura 6.13. Diagrama de Secuencia del DCF usando RTS/CTS-No DATA.

Ahora se plantea el caso cuando la estación emisora no recibe la señal ACK, la figura 6.14 ilustra lo anteriormente planteado.

Nuevamente, las tres estaciones pelean por acceder al medio, cuando el canal está libre entonces la estación emisora transmite un RTS, siendo recibida exitosamente por la estación receptora entonces esta estación responde con un CTS. Una vez recibida esta señal la estación emisora tiene el canal reservado para la transmisión de información. Sí se recibe la información correctamente por la estación receptora entonces se envía ACK, sin embargo ACK no es recibido. Al finalizar el tiempo de espera DIFS se ejecuta el Backoff.

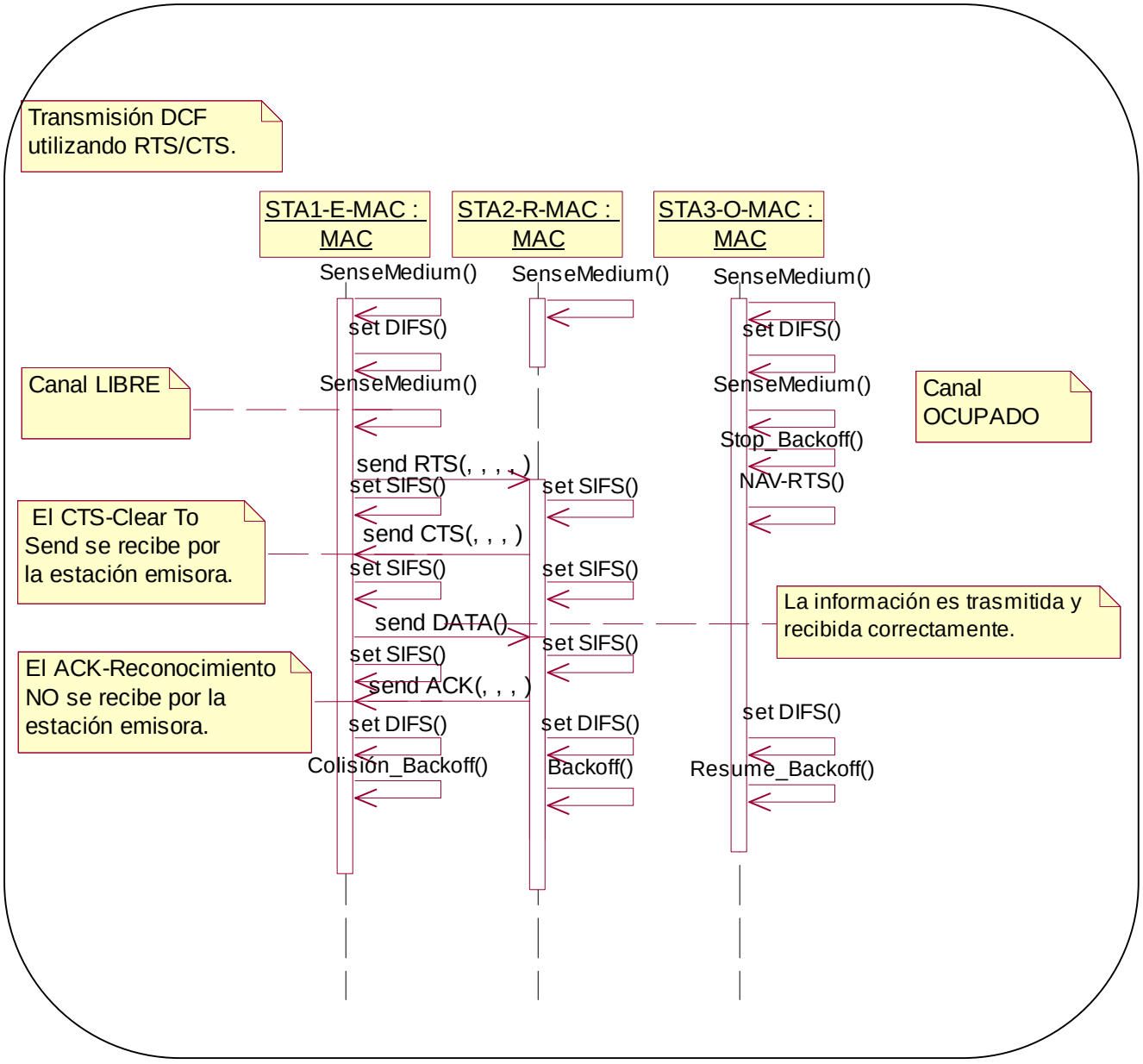


Figura 6.14. Diagrama de Secuencia del DCF usando RTS/CTS-No ACK.

6.2. Modelado AES-CCM

Esta sección propone una solución a los servicios de confidencialidad y autenticación usando Criptografía Simetríca debido a su eficiencia y alta seguridad, como se explicó en el capítulo 4. Debido a las vulnerabilidades y ataques del protocolo WEP discutidas en la sección 3.6.

La figura 6.3 muestra el diagrama de clases con la propuesta de este trabajo de tesis, la cual consiste en reemplazar el protocolo WEP por AES-CCM para ofrecer los servicios de confidencialidad y autenticación [41, 43]. Las clases que conforman el diagrama son las siguientes:

- **ModoCCM:** Implementa el modo de operación CCM, el cual está conformado por dos primitivas criptográficas: CRT (Counter Mode) para ofrecer privacidad y CBC (Cipher Book Chiang) ofreciendo autenticación. ModoCCM hace varios llamados de métodos que pertenecen a la clase SeguridadAES, para poder realizar el proceso de cifrado y descifrado. La clase ModoCCM sólo utiliza el proceso de cifrado de SeguridadAES para realizar el cifrado y descifrado mediante el modo de operación CCM.

- **SeguridadAES:** Implementa el algoritmo de Rijndael incluyendo sus cuatro básicas capas (ByteSub, ShiftRow, MixColumn y AddRoundKey) y el proceso Key Schedule.

Se puede apreciar en la figura 6.3 que la Capa Física del IEEE 802.11 esta compuesta por las siguientes clases:

- **IEEE 802.11:** La especificación de tres capas físicas, dos en la banda de 2.4 GHz, la primera utiliza Espectro Extendido con Salto en Frecuencia (FHSS), la segunda capa física utiliza Espectro Extendido en Secuencia Directa (DSSS) y la última utiliza el infrarrojo, todas operando a 1 y 2 Mbps.

- **IEEE 802.11a:** Opera en la banda de 5 GHz utilizando Frecuencia Ortogonal (OFDM) a una velocidad de hasta 54 Mbps.

- **IEEE 802.11b:** Opera en la banda de 2.4 GHz utilizando DSSS a 11 Mbps.

- **IEEE 802.11g:** Opera en la banda de 2.4 GHz utilizando OFDM y DSSS hasta 54 Mbps.

Se indica mediante la flecha ⟶ que es opcional utilizar la clase de SeguridadWEP y/o ModoCCM. Mientras que la flecha ⟶ indica herencia de una clase a otra. La flecha punteada ---- indica comentario o descripción de la clase.

La figura 6.15 muestra la primera clase refinada ModoCCM. La tabla 6.7 explica el funcionamiento de cada atributo de la clase refinada ModoCCM y la tabla 6.8 contiene la explicación del funcionamiento de cada método definido en la clase refinada ModoCCM.

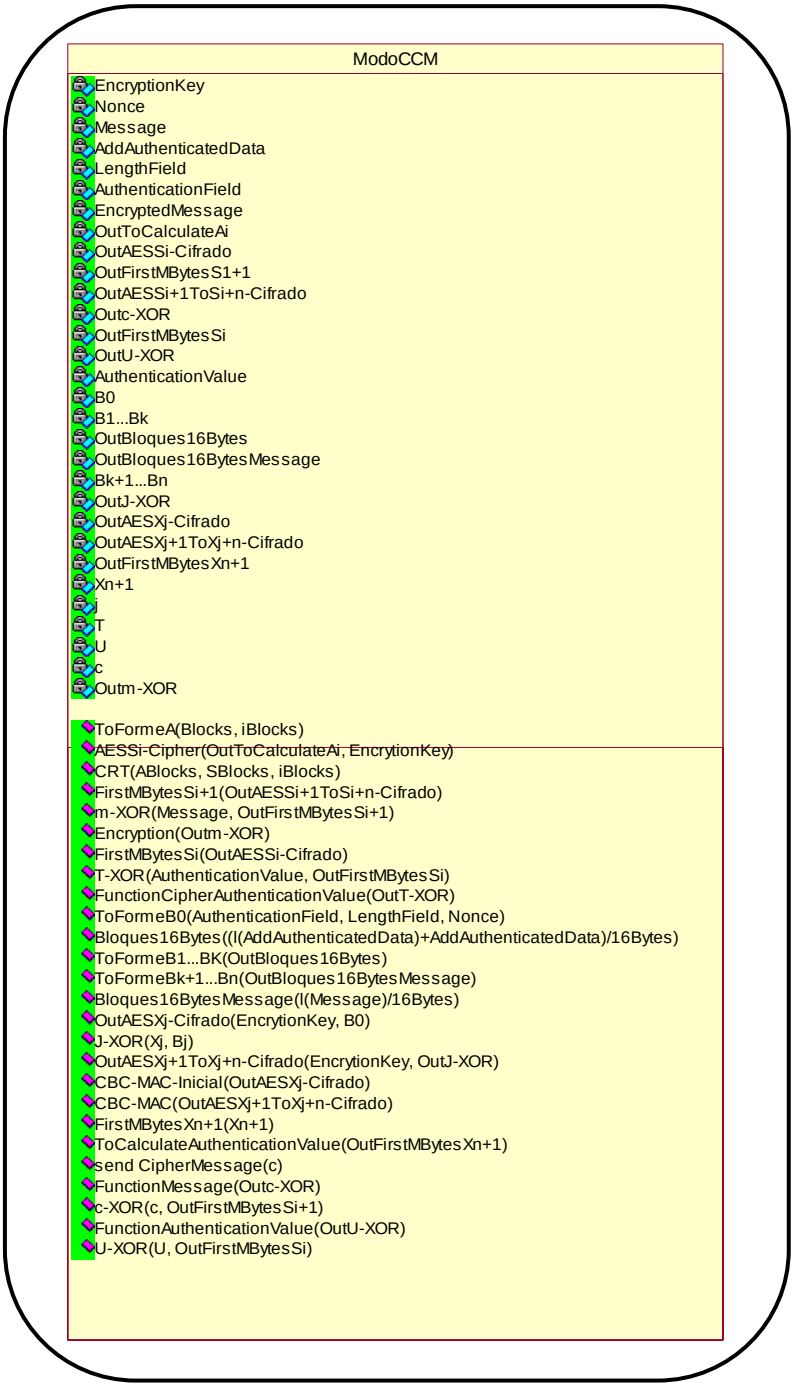


Figura 6.15. Clase Refinada ModoCCM.

La tabla 6.7 muestra a detalle cada uno de los atributos definidos en la Clase ModoCCM. Se divide en dos partes la tabla 6.7, en la primera se pone el nombre del atributo y en la segunda se describe la funcionalidad que tiene.

Clase ModoCCM	
Nombre del Atributo	*Función*
EncryptionKey	Guarda la Llave Privada

Nonce	Almacena el tamaño del campo Nonce 15-L Bytes siendo L=LengthField
Message	Contiene el mensaje Original para Autenticar y Cifrar
AddAuthenticatedData	Tiene el Dato Adicional para Autenticar
LengthField	Acumula el tamaño del campo Longitud del Mensaje
AuthenticationField	Contiene el tamaño del campo Autenticación
EncryptedMessage	Se deposita el Mensaje Autenticado y Cifrado
OutToCalculateAi	Se recopila el resultado obtenido al formar los bloques Ai
OutAESSi-Cifrado	Obtiene el resultado, al aplicar la operación de cifrado utilizando AES, es decir el bloque Si
OutFirstMBytesS1+1	Guarda la salida del método FirstMBytesS1+1
OutAESSi+1ToSi+n-Cifrado	Tiene el resultado obtenido al aplicar la operación de cifrado utilizando AES, es decir los bloque Si+1 hasta Si+n
Outc-XOR	Recopila la salida de la operación c-XOR
OutFirstMBytesSi	Es el resultado del método FirstMBytesSi
OutU-XOR	Obtiene la salida de la operación U-XOR
AuthenticationValue	Acumula el Tamaño del Dato Adicional para Autenticar
B0	Guarda el Bloque B0
B1...Bk	Deposita los Bloques B1 a Bk
OutBloques16Bytes	Contiene el resultado del método Bloques16Bytes
OutBloques16BytesMessage	Tiene el resultado de Bloques16BytesMessage
Bk+1...Bn	Almacena el Bloque Bk+1 hasta Bn
OutJ-XOR	Acumula el resultado de J-XOR
OutAESXj-Cifrado	Guarda el valor obtenido en AESXj-Cifrado
OutAESXj+1ToXj+n-Cifrado	Resultado del método AESXj+1ToXj+n-Cifrado
OutFirstMBytesXn+1	Recopila los datos obtenidos al operar FirstMBytesXn+1
j	Variable j inicializada en cero
T	Valor T = Valor de la Autenticación
U	Valor U = Valor Cifrado de la Autenticación
c	Contiene el Mensaje Autenticado y Cifrado
Outm-XOR	Resultado del método m-XOR

Tabla 6.7. Atributos definidos en la Clase ModoCCM.

La tabla 6.8 muestra a detalle cada uno de los métodos definidos en la Clase ModoCCM.

Clase ModoCCM	
Nombre del Método	*Función*
ToFormeA	Forma Bloques Ai
AESSi-Cipher	Calcula los boques Si mediante el cifrador AES
CRT	Implementa Modo Contador
FirstMBytesSi+1	Obtiene los primeros M (Tamaño del campo Autenticación) bytes de los bloques Si+1
m-XOR	Realiza la operación de or-exclusivo al mensaje original con FirstMBytesSi+1
Encryption	Operación de Cifrado
FirstMBytesSi	Obtiene los primeros M (Tamaño del campo Autenticación) bytes del bloque S0
T-XOR	Realiza la operación de or-exclusivo de T (Valor de la Autenticación) con FirstMBytesSi
FunctionCipherAuthenticationValue	Operación que ontiene el Valor Cifrado de la Autenticación
ToFormeB0	Forma el Bloque B0
Bloques16Bytes	Divide los parámetros de entrada en bloques de 16 bytes
ToFormeB1...BK	Forma los Bloques B1 hasta Bk
ToFormeBk+1...Bn	Forma los Bloques Bk+1 hasta Bn
Bloques16BytesMessage	Divide el mensaje en bloques de 16 bytes
OutAESXj-Cifrado	Calcula el bloque Bj mediante el cifrador AES
J-XOR	Realiza la operación de XOR entre el bloque Xj y Bj, desde j = 0,

	1, 2, 3, 4, …, n
OutAESXj+1ToXj+n-Cifrado	Calcula el bloque Bj+1 hasta Bj+n mediante el cifrador AES
CBC-MAC-Inicial	Implementa la primera operación de CBC-MAC
CBC-MAC	Implementa CBC-MAC
FirstMBytesXn+1	Toma los primeros M (Tamaño del campo Autenticacion) bytes del bloque Xn+1
ToCalculateAuthenticationValue	Operación que calcula el Valor de la Autenticación para realizar la Verificación
send CipherMessage	Señal que envía el Mensaje Cifrado
FunctionMessage	Obtiene el mensaje original en la etapa de descifrado
c-XOR	Se utiliza en la etapa de Descifrado, realiza la operación de or-exclusivo de c (mensaje cifrado) con Firstl(c)BytesSi
FunctionAuthenticationValue	Operación que calcula el Valor de la Autenticación
U-XOR	Realiza la operación de or- exclusivo de U (Valor Cifrado de la Autenticación) con FirstMBytesS0

Tabla 6.8. Métodos definidos en la Clase ModoCCM.

La figura 6.16 muestra la segunda clase refinada SeguridadAES que implementa el algoritmo de Rijndael incluyendo sus cuatro básicas capas (ByteSub, ShiftRow, MixColumn y AddRoundKey) y el proceso Key Schedule. La tabla 6.9 explica el funcionamiento de cada atributo de la clase refinada SeguridadAES y la tabla 6.10 explica el funcionamiento de cada método definido en la clase refinada SeguridadAES.

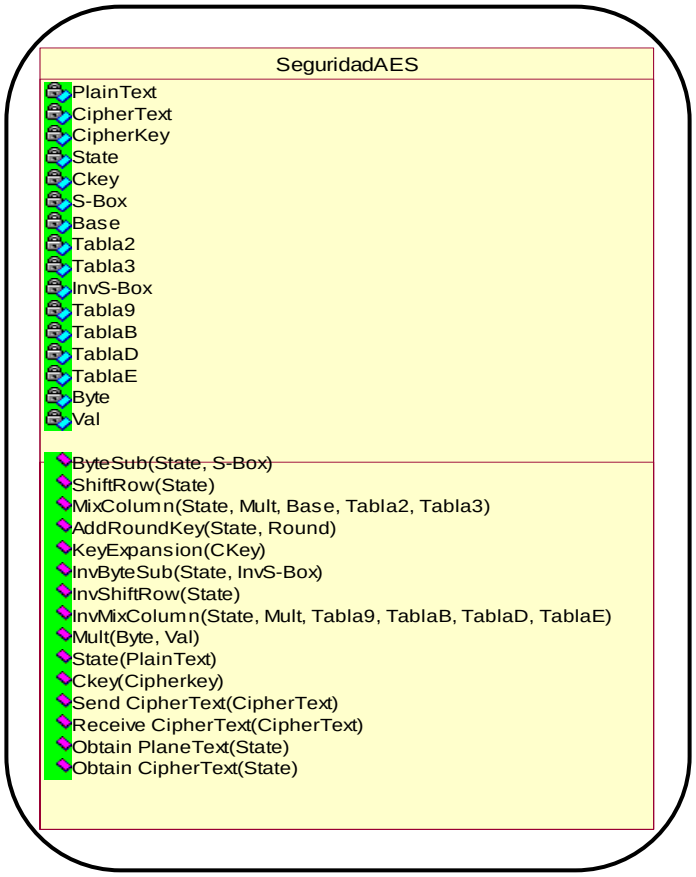


Figura 6.16. Clase Refinada SeguridadAES.

La tabla 6.9 muestra a detalle cada uno de los atributos definidos en la Clase SeguridadAES.

Clase SeguridadAES	
Nombre del Atributo	*Función*
PlainText	Arreglo de tipo unsigned char, donde se almacena el mensaje original para ser procesado
CipherText	Arreglo de tipo unsigned char, que se utiliza para guardar el mensaje ya cifrado
CipherKey	Arreglo de tipo char, se deposita el valor de la llave privada utilizada para realizar el cifrado y descifrado del mensaje
State	Es de tipo unsigned char, su función es recopilar los valores obtenidos después de cada transformación del algoritmo Rijndael
Ckey	Tipo unsigned char, almacena los valores obtenidos al expandir la llave
S-Box	Caja S definida de forma estática como arreglo de [16] [16] de tipo unsigned char, los números definidos están en formato hexadeximal
Base	Es de tipo unsigned char como arreglo de [4] [4]
Tabla2	Se utiliza esta tabla en la transformación de MixColumn, la Tabla2 es un arreglo de [256] que contiene de forma estática valores hexadecimales únicos
Tabla3	Se utiliza esta tabla en la transformación de MixColumn, la Tabla3 es un arreglo de [256] que contiene de forma estática valores hexadecimales, los valores son únicos
InvS-Box	Caja Inversa S definida de forma estática como arreglo de [16] [16] de tipo unsigned char, los números definidos están en formato hexadeximal
Tabla9	Utilizada en la transformación InvMixColumn, Tabla9 es definida como un arreglo de [256] de tipo unsigned char
TablaB	Se ocupa esta tabla en la transformación de InvMixColumn, la TablaB es un arreglo de [256] que contiene de forma estática valores hexadecimales, los valores son únicos
TablaD	Utilizada en la transformación InvMixColumn, TablaD es definida como un arreglo de [256]
TablaE	La transformación de InvMixColumn hace uso de TablaE (es un arreglo de [256])
Byte	Tipo unsigned char
Val	Tipo unsigned char

Tabla 6.9. Atributos definidos en la Clase SeguridadAES.

La tabla 6.10 muestra a detalle cada uno de los métodos definidos en la Clase SeguridadAES.

Clase SeguridadAES	
Nombre del Método	*Función*
ByteSub	Implementa la transformación ByteSub del algoritmo Rijndael
ShiftRow	Implementa la transformación ShiftRow del algoritmo Rijndael
MixColumn	Implementa la transformación MixColumn del algoritmo Rijndael
AddRoundKey	Implementa la transformación AddRoundKey del algoritmo Rijndael
KeyExpansion	Realiza la expansión de llaves a partir de la llave inicial dada
InvByteSub	Implementa la transformación Inversa de ByteSub del algoritmo Rijndael
InvShiftRow	Implementa la transformación Inversa de ShiftRow del algoritmo Rijndael
InvMixColumn	Implementa la transformación Inversa de MixColumn del algoritmo Rijndael
Mult	Función auxiliar en la transformación MixColumn
State	Proceso encargado de almacenar en una matriz de 8 por 8, es decir de 16 Bytes, el mensaje original (texto claro) o bien mensaje cifrado, según sea el caso
Ckey	Proceso encargado de almacenar en una matriz de 8 por 8, es decir de 16 Bytes, la llave privada
Send CipherText	Señal que envía el mensaje cifrado
Receive CipherText	Señal que se utiliza para recibir el mensaje cifrado
Obtain PlaneText	Función que obtiene el texto claro o bien conocido como el mensaje a cifrar
Obtain CipherText	Fúnción que obtiene el texto cifrado

Tabla 6.10. Métodos definidos en la Clase SeguridadAES.

La figura 6.17 muestra el diagrama de secuencia con el proceso de Cifrado de AES (ver sección 4.2), con un bloque y una llave de 128 bits, se considera que cada estación ya tiene su llave privada de antemano. Este proceso es únicamente utilizado por la clase ModoCCM en el proceso de cifrado y descifrado del modo de operación CCM.

Figura 6.17. Diagrama de Secuencia del Cifrado de AES.

La figura 6.18 muestra el diagrama de secuencia con el proceso de Descifrado de AES (ver sección 4.2), con un bloque y una llave de 128 bits, se considera que cada estación ya tiene su llave privada de antemano. El proceso de descifrado no es utilizado por el modo de operación CCM, se muestra el diagrama de secuencia como concepto general del funcionamiento del algoritmo Rijndael.

Figura 6.18. Diagrama de Secuencia del Descifrado de AES.

La figura 6.19 muestra el diagrama de secuencia con el proceso de Cifrado y Descifrado de AES (ver sección 4.2). Una vez que la estación emisora obtiene el texto cifrado mediante el proceso de cifrado, lo envía en el medio inseguro a la estación receptora, la cual obtiene el texto claro mediante el proceso de descifrado.

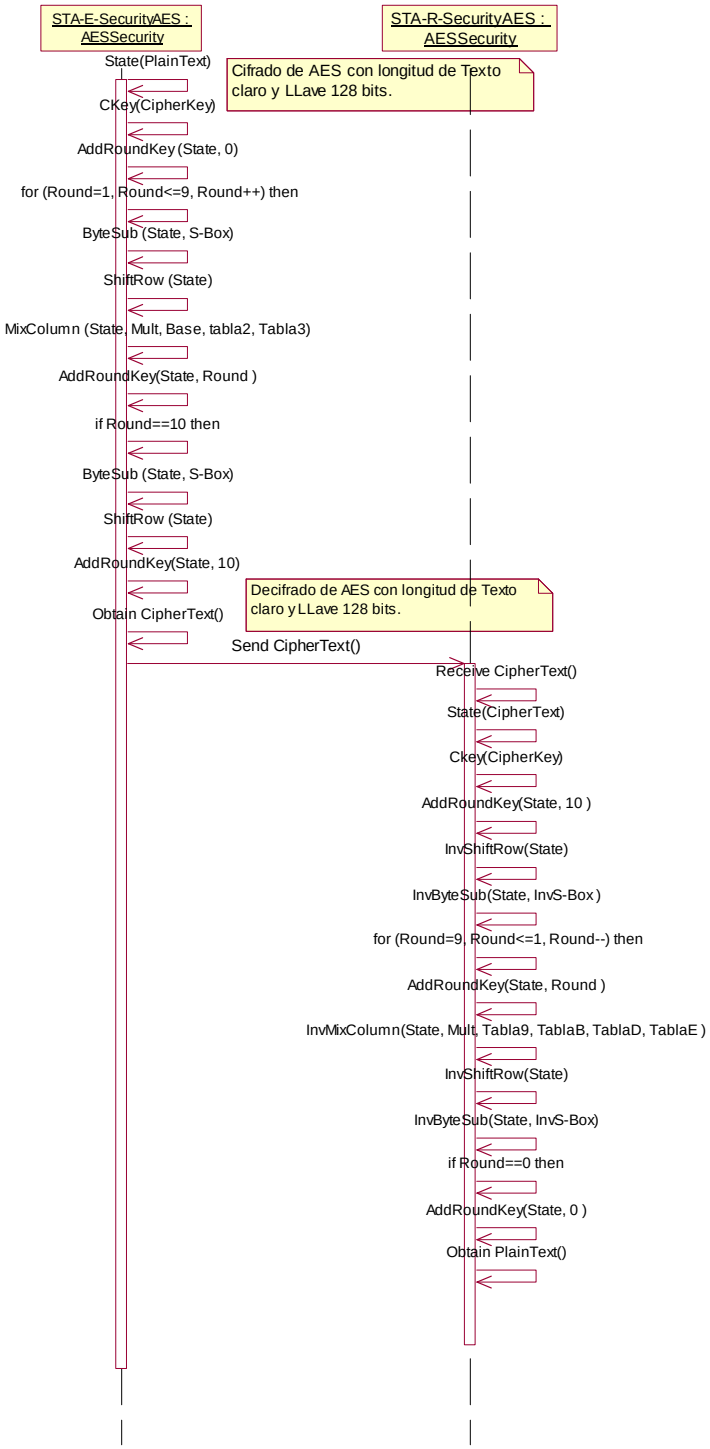


Figura 6.19. Diagrama de Secuencia del Cifrado y Descifrado de AES.

La figura 6.20 muestra el diagrama de secuencia con el proceso de Autenticación/Cifrado AES-CCM (ver sección 4.2 y 5.4). Una vez que la estación emisora obtiene el texto cifrado mediante el proceso de Autenticación/Cifrado, lo envía en el medio a la estación receptora, la cual obtiene el texto claro mediante el proceso inverso, es decir, Verificación/Descifrado.

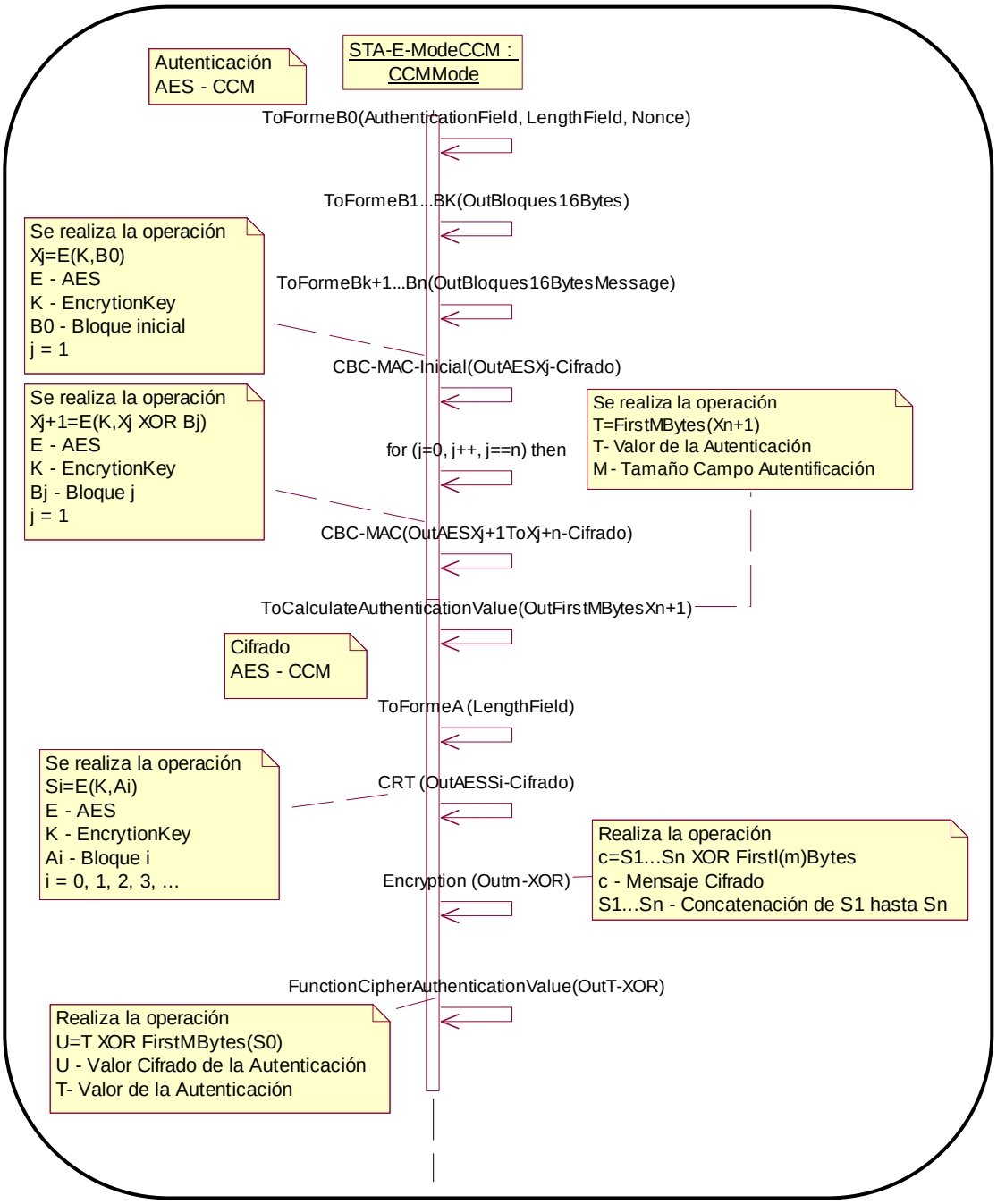


Figura 6.20. Diagrama de Secuencia de Autenticación/Cifrado de AES-CCM.

La figura 6.21 muestra el diagrama de secuencia con el proceso Verificación/Descifrado AES-CCM (ver sección 4.2 y 5.4).

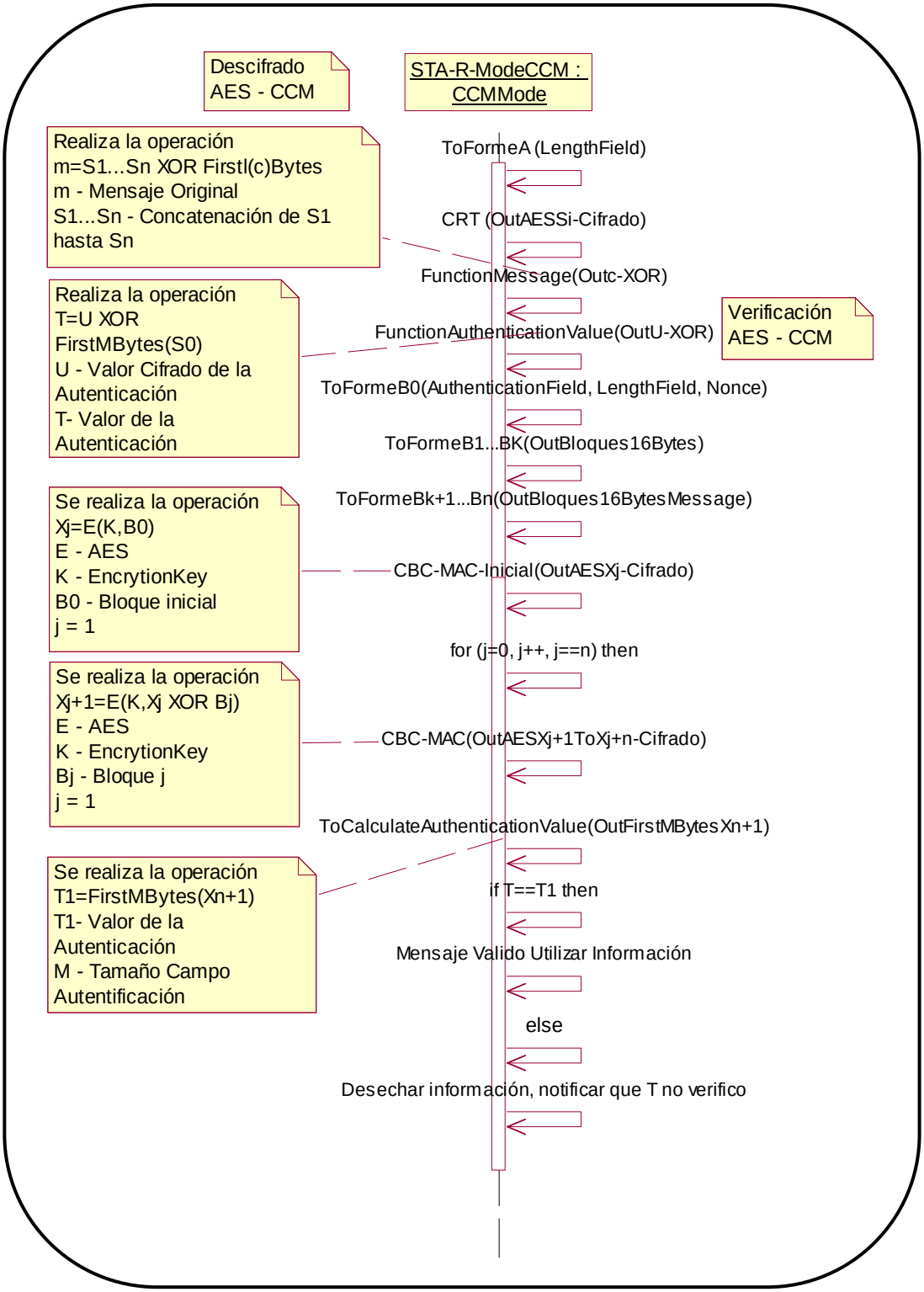


Figura 6.21. Diagrama de Secuencia de Verificación/Descifrado de AES-CCM.

La figura 6.22 muestra el diagrama de secuencia con el proceso de Autenticación/Cifrado y Verificación/Descifrado AES-CCM (ver sección 4.2 y 5.4), teniendo una estación emisora que envía el mensaje cifrado anexando en la cabecera el valor de la autenticación del mismo. Mientras que la estación receptora descifra y verifica este mensaje. Si la verificación es exitosa entonces el mensaje puede ser entregado al receptor, en caso contrario se elimina y notifica al emisor.

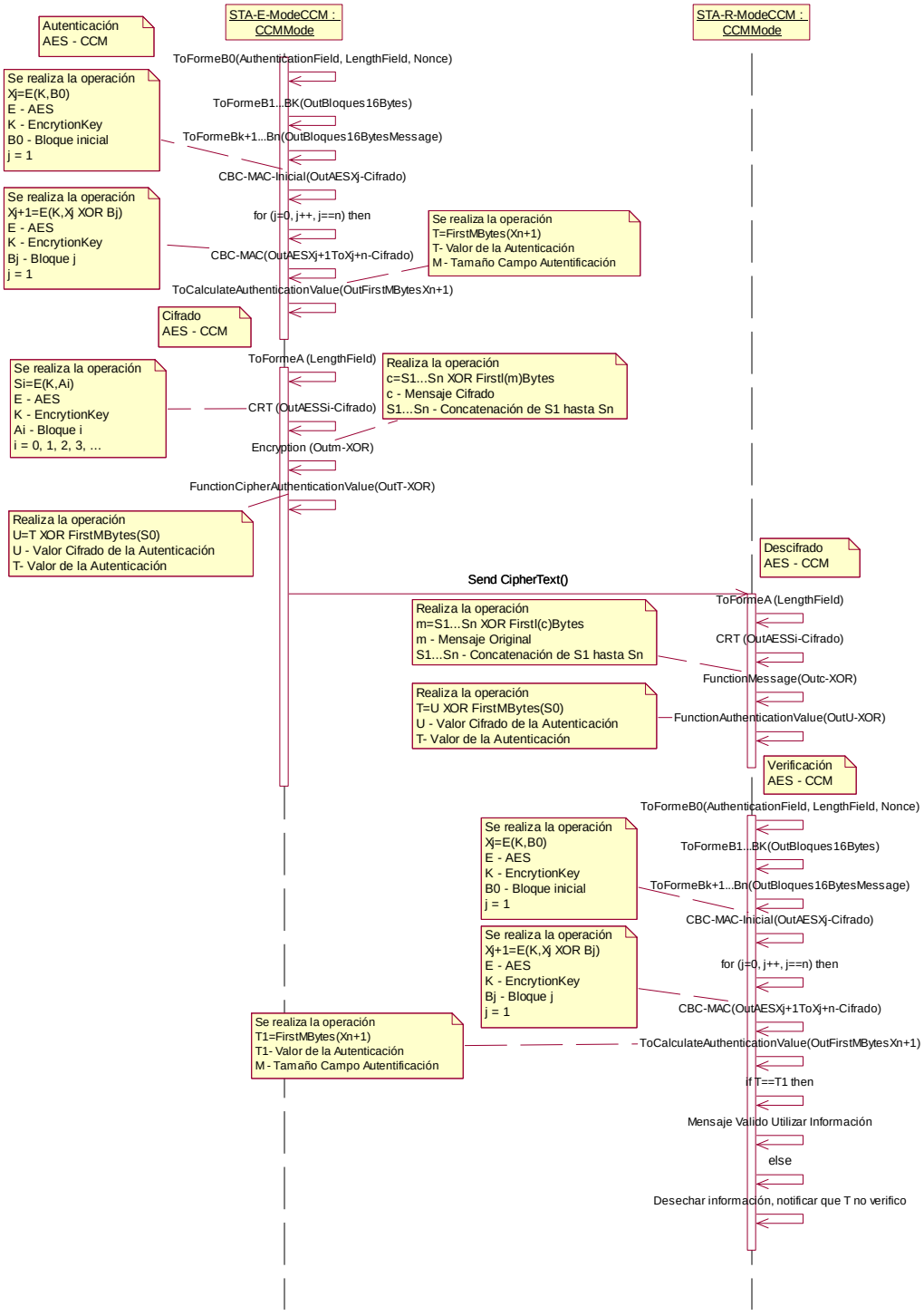


Figura 6.22. Diagrama de Secuencia del Funcionamiento AES-CCM.

CAPÍTULO 7

7. MODELO DE IMPLEMENTACIÓN

7.1. Nivel Aplicación

La implementación [44] utiliza el lenguaje de programación C++. Este lenguaje [45] es flexible, soporta abstracción de datos, programación genérica y es un lenguaje orientado a objetos. Además el lenguaje C++ puede trabajar en las siguientes plataformas: Windows, Linux y Unix. Por lo tanto la implementación se realiza bajo la plataforma de Linux en concreto Suse 9.2, ver apéndice A para la instalación.

El capítulo anterior muestra en la sección 6.2 el modelado de los servicios de seguridad propuestos para el estándar IEEE 802.11. Este modelo ofrece los servicios de confidencialidad y autenticación mediante el uso del cifrador por bloques AES y el modo de operación CCM, tal y como se explicaron en la sección 4.2 para el algoritmo Rijndael en etapa de cifrado y 5.4 para la etapa de autenticación-cifrado y descifrado-verificación.

El siguiente paso fue crear la implementación del algoritmo Rijndael y del modo de operación CCM con la ayuda del modelado, mostrado en el capítulo anterior.

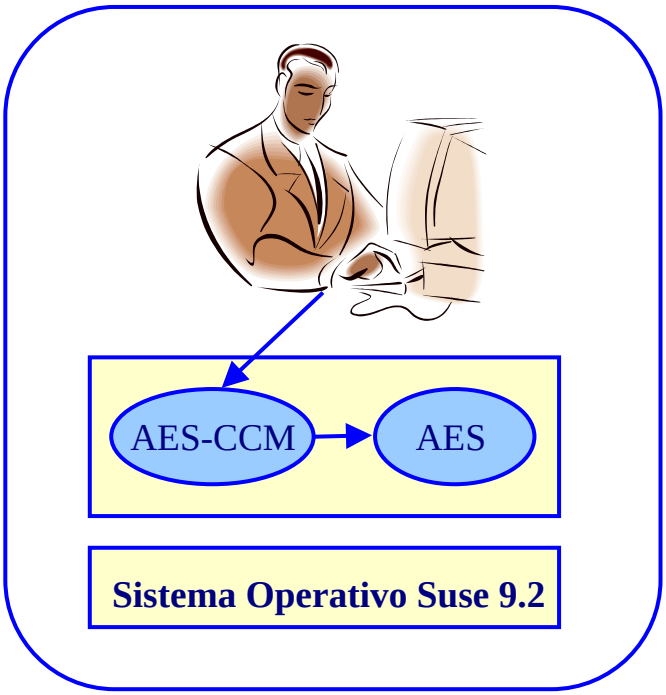


Figura 7.1. Estructura del programa.

La figura 7.1 muestra la estructura del programa, primero se elaboró la librería de AES, la cual implementa el algoritmo Rijndael en la etapa de cifrado con una longitud de bloque y llave de 128 bits y después se realizó el programa de aplicación AES-CCM, este programa implementa el modo de operación CCM para la etapa de autenticación-cifrado y descifrado-verificación. A su vez la aplicación AES-CCM hace uso de la librería AES para usar el algoritmo de Rijndael en la etapa de cifrado.

La figura 7.1 ilustra al usuario empleando el programa AES-CCM a nivel aplicación como una primera aproximación, con el propósito de:

- Probar el funcionamiento correcto de la aplicación mediante vectores de prueba [3, 35].
- Obtener los tiempos de ejecución de cada etapa: cifrado, descifrado, autenticación y verificación.
- Analizar el rendimiento de la aplicación.

7.1.1. Pruebas Realizadas

La tabla 7.1 muestra el tamaño del bloque de datos utilizado para las pruebas. El tamaño máximo del bloque de datos empleado para las pruebas fue de 2312 bytes, por que es el tamaño máximo de los datos que puede contener una trama IEEE 802.11. Se maneja una llave privada de 128 bits que es establecida de antemano. El programa AES-CCM fue probado a nivel aplicación.

Tamaño Bytes
1
8
16
32
64
128
256
512
1024
2048
2312

Tabla 7.1. Tamaño del bloque de datos.

El equipo utilizado para realizar las pruebas cuenta con las siguientes características:

- Pentium IV.
- A 3 GHz
- 512 RAM
- SUSE LINUX Professional 9.2 OS.

Los vectores de prueba [3, 35] fueron utilizados para probar el funcionamiento correcto del programa. A continuación se muestra un ejemplo realizado con el programa AES-CCM aplicando el primer vector de prueba [35], donde se colocan los valores dados para la Llave Privada, el campo Nonce, el Dato Adicional para Autenticar y el Mensaje. Además se ponen los resultados obtenidos mediante la utilización del programa AES-CCM a nivel aplicación.

Valores Dados:

- La Llave Privada: C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
- Valor del campo Nonce: 00 00 00 03 02 01 00 A0 A1 A2 A3 A4 A5
- Valor del Dato Adicional para Autenticar: 00 01 02 03 04 05 06 07
- El mensaje: 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E

Resultados Obtenidos:

- Valor de la Autenticación: 2D C6 97 E4 11 CA 83 A8
- Mensaje Cifrado: 58 8C 97 9A 61 C6 63 D2 F0 66 D0 C2 C0 F9 89 80 6D 5F 6B 61 DA C3 84
- Valor Cifrado de la Autenticación: 17 E8 D1 2C FD F9 26 E0

7.1.2. Resultados Obtenidos

A continuación se muestran los resultados obtenidos de forma independiente de cada etapa del programa AES-CCM. La tabla 7.2 contiene los tiempos de ejecución obtenidos en milisegundos en realizar la etapa de cifrado y descifrado del programa AES-CCM a nivel aplicación.

Tamaño Bytes	Cifrar	Descifrar
1	0.006	0.006
8	0.007	0.007
16	0.007	0.007
32	0.009	0.009
64	0.011	0.011
128	0.015	0.014
256	0.023	0.022
512	0.037	0.037
1024	0.066	0.067
2048	0.113	0.114
2312	0.143	0.145

Tabla 7.2. Tiempos de ejecución obtenidos en la etapa de Cifrado y Descifrado.

La tabla 7.3 tiene los tiempos de ejecución obtenidos en milisegundos en realizar la etapa de autenticación y verificación del programa AES-CCM a nivel aplicación.

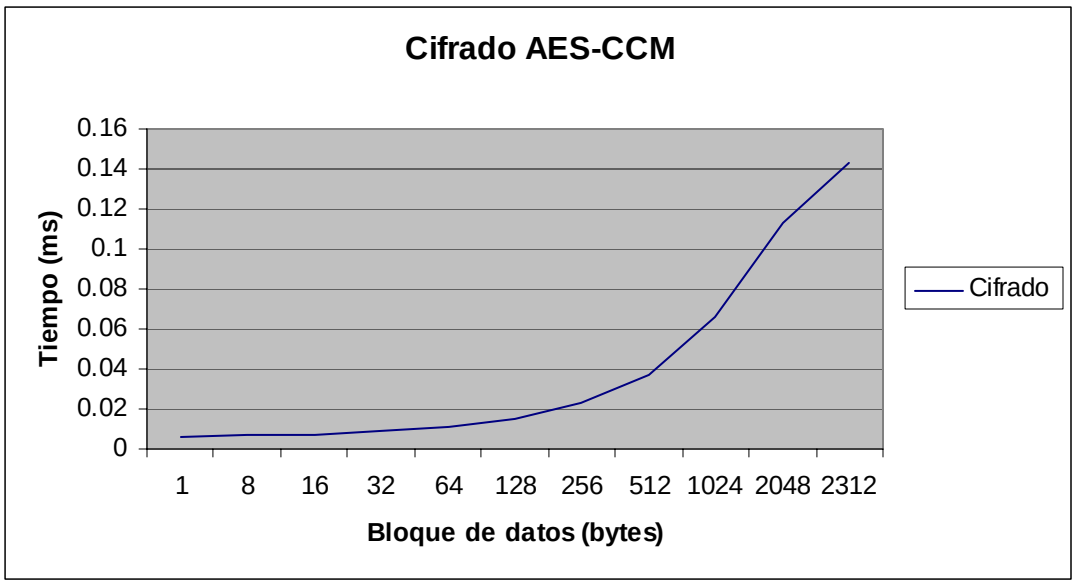
Tamaño Bytes	Autenticar	Verificar
1	0.015	0.014
8	0.016	0.016
16	0.016	0.017
32	0.016	0.016
64	0.018	0.017
128	0.021	0.021
256	0.026	0.026
512	0.039	0.039
1024	0.063	0.063
2048	0.113	0.115
2312	0.126	0.127

Tabla 7.3. Tiempos obtenidos para la etapa de Autenticación y Verificación.

7.1.3. Análisis de Rendimiento

La gráfica 7.1 ilustra los tiempos obtenidos en milisegundos respecto al tamaño de bloque de datos en bytes. Los tiempos mostrados son de la etapa de cifrado del programa AES-CCM a nivel aplicación.

Se observa en la gráfica 7.1 que el tiempo para los tres primeros bloques de datos permanece casi igual, esto se debe al cifrador por bloques AES, por que maneja una longitud fija de bloque de datos de 16 bytes rellenando con ceros cuando es menor el bloque. A partir de un tamaño de bloque de datos de 32 hasta 128 bytes el tiempo incrementa 0.002 milisegundos, sin embargo en los bloques de datos restantes el tiempo va en aumento. Se puede deducir que entre mayor sea el tamaño de bloques de datos mayor es el tiempo en realizar la etapa de cifrado mediante AES-CCM.

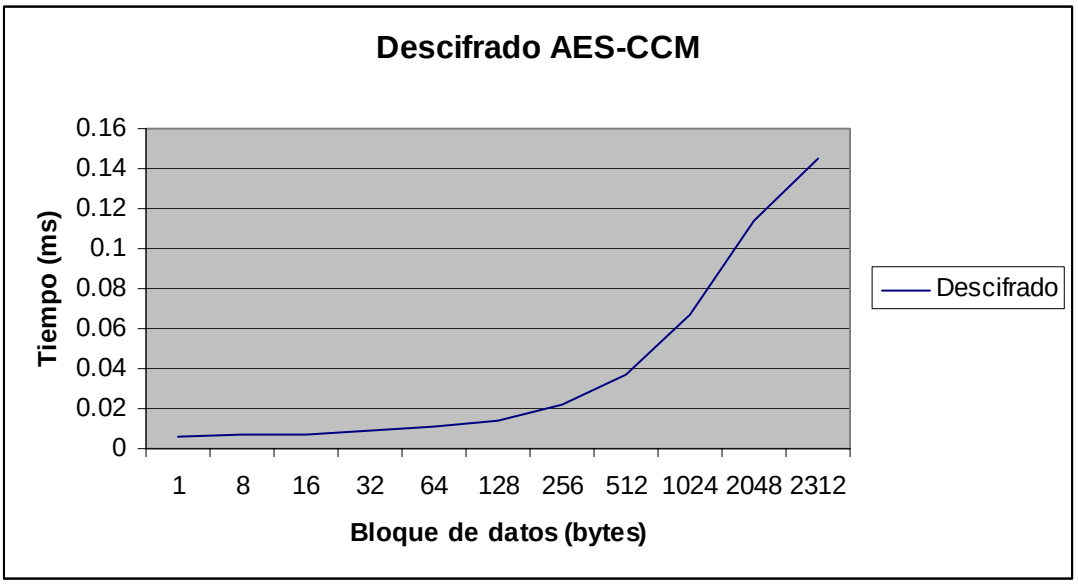


Gráfica 7.1. Rendimiento en la etapa de Cifrado de AES-CCM.

La gráfica 7.2 muestra los tiempos obtenidos en la etapa de descifrado del programa AES-CCM a nivel aplicación. Los tiempos dados están en milisegundos respecto al tamaño de bloque de datos en bytes.

En la gráfica 7.2, el tiempo incrementa respecto a la longitud del bloque de datos, se puede notar que los tiempos dados en el proceso de cifrado son casi iguales (con una variación de 0.01 a 0.02 milisegundos) a la etapa de descifrado.

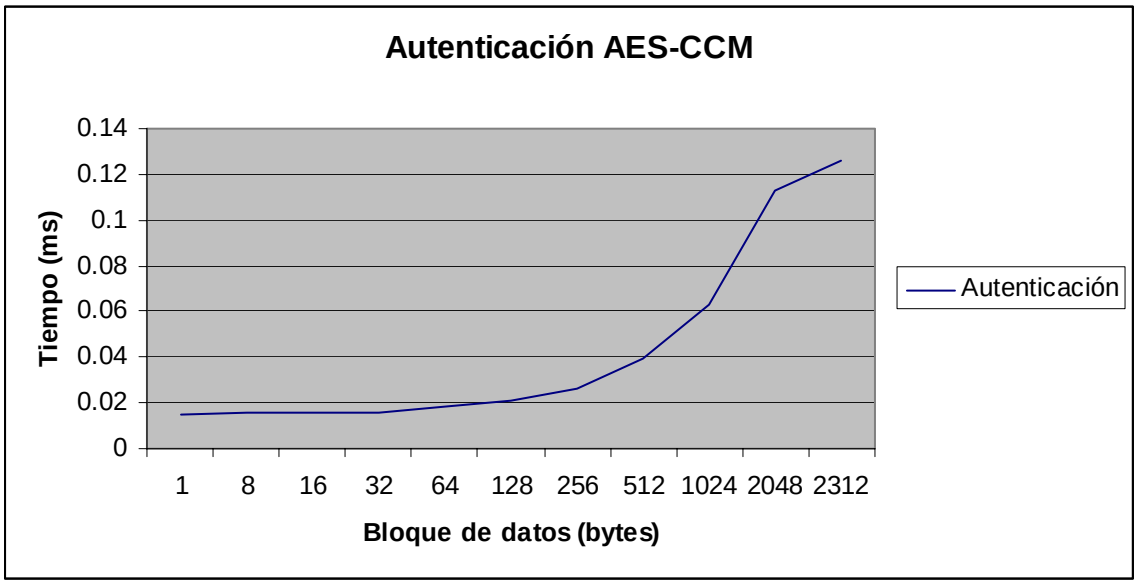
Como se explicó en la sección 5.4.2 la etapa de cifrado y en la sección 5.4.3 la etapa de descifrado, donde se puede notar una mínima diferencia de la figura 5.9 con la figura 5.10, por lo tanto es congruente la similitud de los tiempos obtenidos en la etapa de cifrado y descifrado.



Gráfica 7.2. Rendimiento en la etapa de Descifrado de AES-CCM.

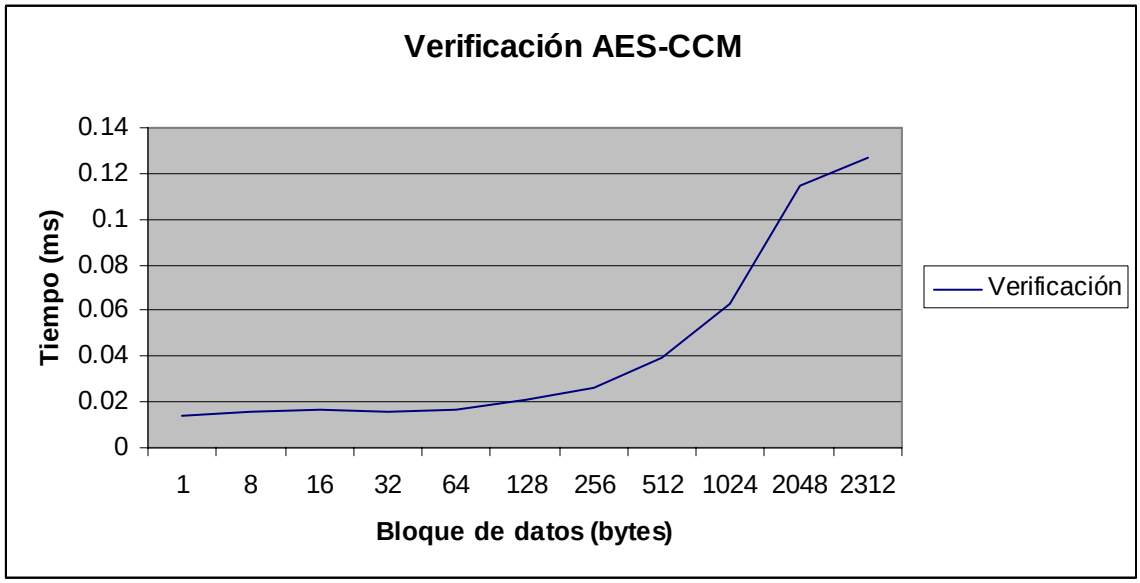
Se observa en la gráfica 7.3 los tiempos obtenidos en milisegundos respecto al tamaño de bloque de datos en bytes. La etapa de autenticación es mostrada en la gráfica 7.3 del programa AES-CCM a nivel aplicación.

La gráfica 7.3 ilustra un acesso mínimo en el tiempo referente al tamaño de bloque de datos. El tiempo medido es la duración total de la etapa de autenticación. Los mejores tiempos obtenidos son en los cuatro primeros bloques de datos.



Gráfica 7.3. Rendimiento en la etapa de Autenticación de AES-CCM.

La gráfica 7.4 indica los tiempos obtenidos en la etapa de verificación del programa AES-CCM a nivel aplicación. Los tiempos dados están en milisegundos respecto al tamaño de bloque de datos en bytes. El tiempo crece respecto a la longitud del bloque de datos, se percata que los tiempos obtenidos en el proceso de autenticación son semejantes (con una variación de 0.01 a 0.02 milisegundos) a la etapa de verificación, debido a la semejanza de la etapa de autenticación como se explica en la sección 5.4.1 y la etapa de verificación expuesta en la sección 5.4.4.



Gráfica 7.4. Rendimiento en la etapa de Verificación de AES-CCM.

7.2. Utilizando Sockets

Una vez realizado el programa a nivel de aplicación se anexa al programa AES-CCM la utilización de Sockets para establecer la comunicación en la red inalámbrica en modo Ad-Hoc, cada PC empleada en la red inalámbrica ocupa la tarjeta inalámbrica SMC 2802W versión 2, la instalación de la tarjeta inalámbrica en Suse 9.2 se explica en el apéndice B.

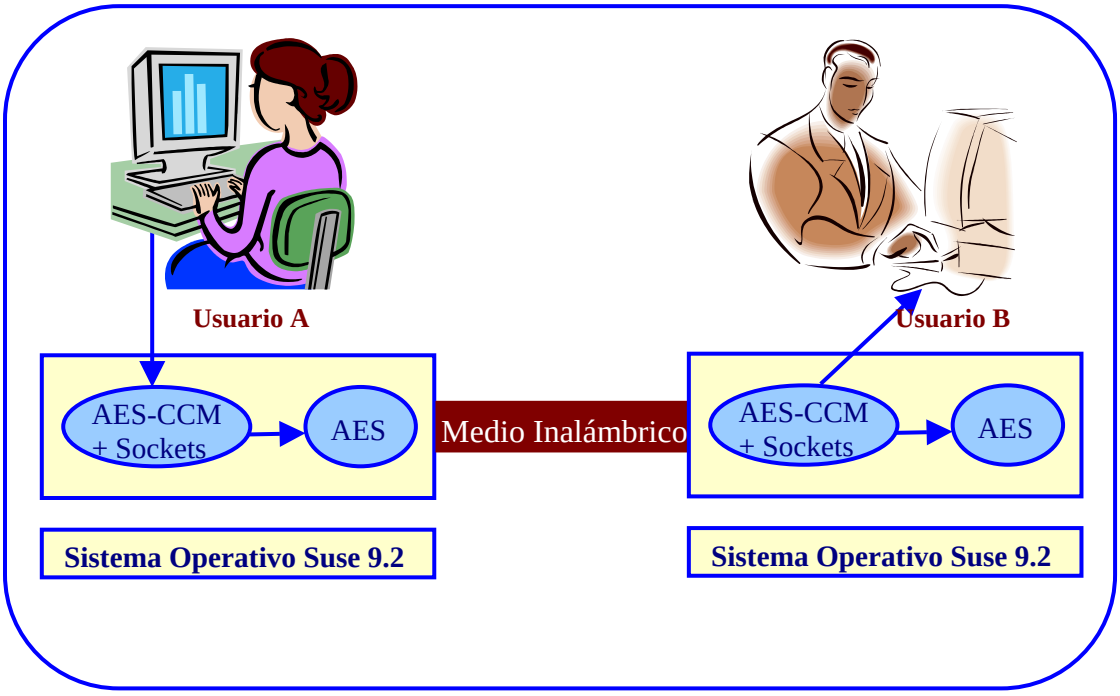


Figura 7.2. Estructura del programa usando sockets.

Se muestra la estructura del programa mediante el uso de sockets en la figura 7.2, donde el *Usuario A* transmite información a través del medio inalámbrico al *Usuario B*. El funcionamiento es el siguiente, se tiene un emisor que en este caso es *Usuario A* y un receptor que es *Usuario B*.

Usuario A tiene el *mensaje* a enviar el cual es procesado por *el programa AES-CCM + Sockets* para generar el *valor cifrado de la autenticación* y el *mensaje cifrado*, este valor junto con el mensaje cifrado se transmite por el medio inalámbrico al *Usuario B* para obtener el *mensaje original* mediante *el programa AES-CCM + Sockets*.

Teniendo como próposito:

- Obtener los tiempos de ejecución de cada etapa autenticación-cifrado y descifrado-verificación.
- Obtener los tiempos de transferencia de la información de una PC a otra.
- Analizar el rendimiento de la aplicación utilizando sockets.

7.2.1. Pruebas Realizadas

La tabla 7.1 indica la capacidad del bloque de datos empleado para las pruebas. La longitud máxima del bloque de datos manejado para las pruebas fue de 2312 bytes, por que es el tamaño límite de los datos que puede abarcar una trama IEEE 802.11. Se ocupa una llave privada de 128 bits que es determinada de antemano. El programa AES-CCM + Sockets fue probado en la red inalámbrica en modo Ad-Hoc.

Cada PC empleada en la red inalámbrica para realizar las pruebas cuenta con las siguientes características:

- Pentium IV.
- A 3 GHz
- 512 RAM
- SUSE LINUX Professional 9.2 OS.

7.2.2. Resultados Obtenidos

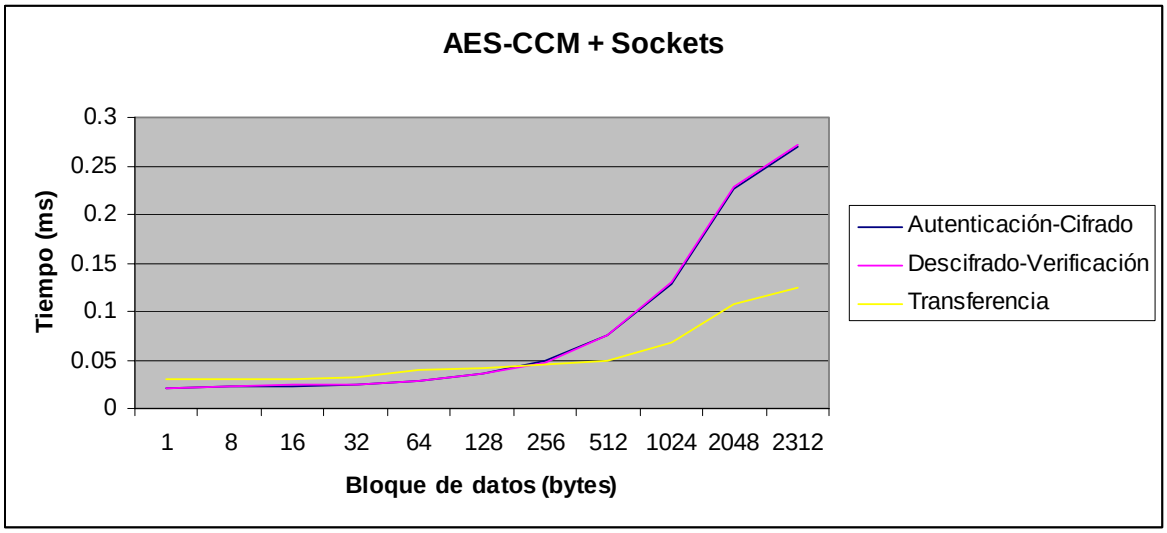
Los resultados obtenidos se exhiben en la tabla 7.4. Esta tabla contiene los tiempos de ejecución obtenidos en milisegundos en realizar la etapa de autenticación-cifrado y descifrado-verificación del programa AES-CCM, además tiene el tiempo de transferencia de envío y recepción de la información.

Tamaño Bytes	Autenticación-Cifrado	Descifrado-Verificación	Transferencia
1	**0.021**	**0.020**	**0.030**
8	**0.023**	**0.023**	**0.031**
16	**0.023**	**0.024**	**0.030**
32	**0.025**	**0.025**	**0.032**
64	**0.029**	**0.028**	**0.039**
128	**0.036**	**0.035**	**0.041**
256	**0.049**	**0.048**	**0.045**
512	**0.076**	**0.076**	**0.050**
1024	**0.129**	**0.130**	**0.067**
2048	**0.226**	**0.229**	**0.108**
2312	**0.269**	**0.272**	**0.125**

Tabla 7.4. Tiempos de ejecución obtenidos en el programa AES-CCM + Sockets.

7.2.3. Análisis de Rendimiento

La gráfica 7.5 ilustra los tiempos obtenidos en milisegundos respecto al tamaño de bloque de datos en bytes. Los tiempos expuestos son de la etapa de autenticación-cifrado y descifrado-verificación del programa AES-CCM + Sockets ocupando la red inalámbrica. Los tiempos de transferencia de envío y recepción de una PC a otra PC son incluidos en la gráfica 7.5.



Gráfica 7.5. Rendimiento en el programa AES-CCM + Sockets.

La línea que representa la etapa de autenticación-cifrado permanece casi idéntica a la línea que simboliza la etapa de descifrado-verificación. Mientras que el tiempo de ejecución obtenido en la transferencia de información se incrementa ligeramente respecto al tamaño del bloque de datos.

7.3. RC4 y AES-CCM

Esta sección presenta un análisis de rendimiento [46] entre el cifrador por flujo RC4 y el cifrador por bloques AES aplicando el modo de operación CCM. La razón de esta comparación es que el protocolo WEP hacer uso de RC4 en el estándar IEEE 802.11 para brindar el servicio de confidencialidad, debido a los ataques y vulnerabilidades presentadas en la sección 3.6. Se emplea AES-CCM para ofrecer el servicio de confidencialidad y autenticación. La comparación entre RC4 y AES-CCM sólo se realiza en la etapa de cifrado a nivel aplicación.

7.3.1. Pruebas Realizadas

La tabla 7.1 contiene la medida del bloque de datos asignado para las pruebas. La capacidad máxima del bloque de datos adoptado para las pruebas fue de 2312 bytes, por que es el tamaño máximo de los datos de una trama IEEE 802.11. La longitud de la llave privada es de 128 bits siendo establecida de antemano.

El equipo usado para efectuar las pruebas cuenta con las siguientes características:

- Pentium IV.
- A 3 GHz
- 512 RAM
- SUSE LINUX Professional 9.2 OS.

7.3.2. Resultados Obtenidos

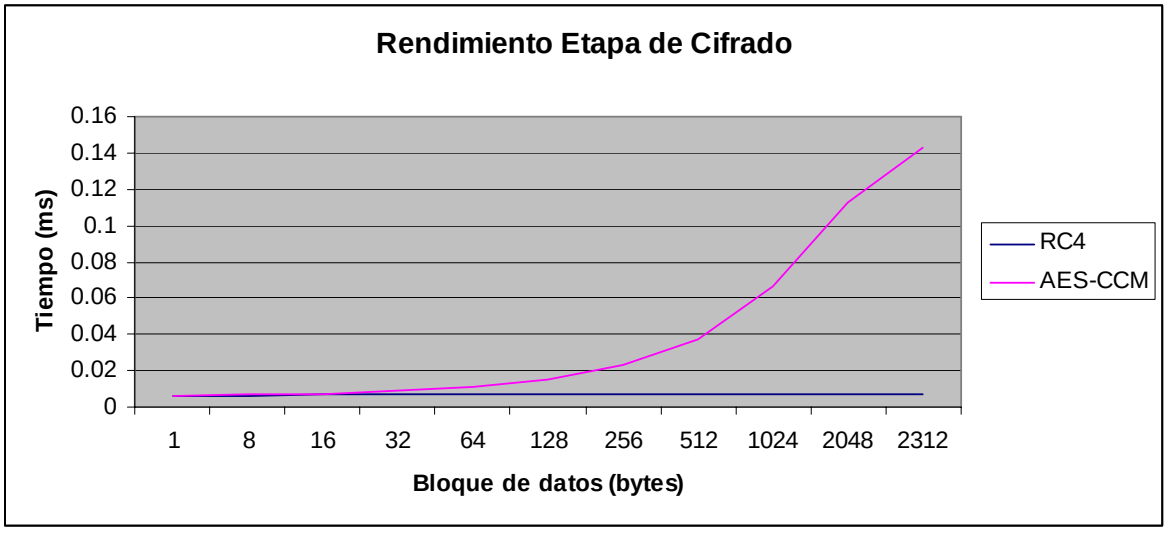
La tabla 7.5 tiene los tiempos obtenidos en la etapa de cifrado del cifrador por flujo RC4 y el cifrador por bloque AES manejando el modo de operación CCM. Las pruebas realizadas para tener estos tiempos fueron a nivel de aplicación. Los tiempos de ejecución están dados en milisegundos.

Tamaño Bytes	RC4	AES-CCM
1	0.006	0.006
8	0.007	0.007
16	0.007	0.007
32	0.007	0.009
64	0.007	0.011
128	0.007	0.015
256	0.007	0.023
512	0.007	0.037
1024	0.007	0.066
2048	0.007	0.113
2312	0.007	0.143

Tabla 7.5. Tiempos de ejecución obtenidos en la etapa de Cifrado de RC4 y AES-CCM.

7.3.3. Análisis de Rendimiento

En la gráfica 7.6 se observa un análisis de rendimiento del proceso de cifrado usando RC4 y AES-CCM.



Gráfica 7.6. Rendimiento en la etapa de Cifrado de AES-CCM.

Siendo RC4 y AES cifradores simétricos, la gráfica ilustra una diferencia mayor en tiempo de ejecución a partir de 128 bytes del tamaño de bloque de datos. El cifrador por flujo RC4 se mantiene casi constante el tiempo en cifrar independientemente del tamaño del

mensaje. Sin embargo el cifrador por bloques AES mantiene un mismo nivel en tiempo de ejecución que el cifrador RC4 en la longitud de bloque de datos de 1 hasta 16 bytes. La causa es la forma en que AES cifra la información mediante bloques de 16 bytes de longitud.

Por lo tanto se recomienda AES-CCM para longitudes de bloques de datos pequeñas de 1 hasta 16 bytes, de esta forma se ofrece el mismo rendimiento que el cifrador RC4.

Mediante el uso de RC4 en el protocolo WEP se ofrece el servicio de confidencialidad. RC4 hace uso de una llave privada de 40 bits concatenada a un vector de inicialización de 24 bits, este vector viaja en el medio inalámbrico en texto claro comprometiendo 24 bits de la llave privada, entre otras vulnerabilidades descritas en la sección 3.6.

Se propone emplear el cifrador por bloques AES mediante el modo de operación CCM para proporcionar el servicio de confidencialidad y autenticación. AES-CCM dispone de una llave privada de 128 bits que se da por hecho el intercambio de llaves, para esto se opta el uso de Criptosistemas de Curvas Elípticas [47].

7.4. AES-CCM (D. Whiting, R. Housley y N. Ferguson) y AES-CCM (S. Merino)

Una vez realizado el programa AES-CCM a nivel aplicación, se planeo efectuar un análisis de rendimiento del tiempo de ejecución obtenido en las pruebas realizadas en la sección 7.1 con la implementación de D. Whiting, R. Housley y N. Ferguson [48]. Esta implementación calcula los vectores de prueba en concreto 24 vectores, en específico efectuá la etapa de autenticación-cifrado a nivel aplicación. La longitud del bloque de datos utilizado es de 23, 24, 25 bits. Por lo tanto se considera este tamaño de bloques de datos para efectuar las pruebas de rendimiento entre los dos programas de AES-CCM a nivel aplicación.

El equipo empleado para generar las pruebas cuenta con las siguientes características:

- Pentium IV.
- A 3 GHz
- 512 RAM
- SUSE LINUX Professional 9.2 OS.

7.4.1. Pruebas Realizadas

El contenido de la tabla 7.6 es el tamaño del bloque de datos aplicado para las pruebas y el número de vector de prueba [35] empleado. Los programas AES-CCM se probaron a nivel aplicación.

Tamaño Bits	Nº Vector de Prueba
23	1
24	2
25	3

Tabla 7.6. Longitud del bloque de datos y Vector de Prueba.

7.4.2. Resultados Obtenidos

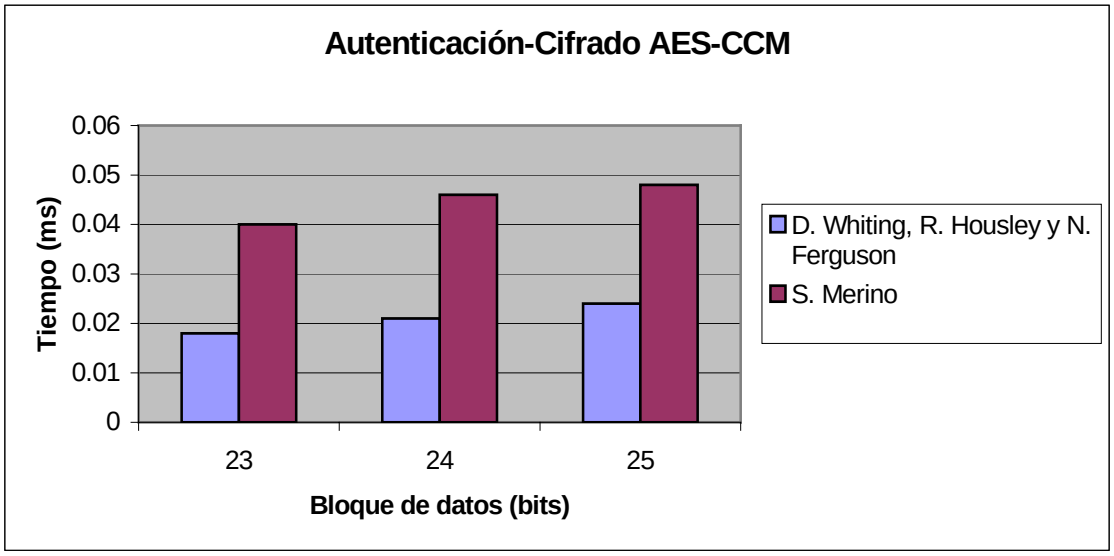
Se exhiben los resultados obtenidos en la tabla 7.7. Esta tabla tiene los tiempos de ejecución obtenidos en milisegundos en realizar la etapa de autenticación-cifrado del programa AES-CCM (D. Whiting, R. Housley y N. Ferguson) y de la implementación AES-CCM (S. Merino).

Tamaño Bits	AES-CCM (D. Whiting, R. Housley y N. Ferguson)	AES-CCM (S. Merino)
23	**0.018**	**0.040**
24	**0.021**	**0.046**
25	**0.024**	**0.048**

Tabla 7.7. Tiempos de ejecución en la etapa de Autenticación-Cifrado de AES-CCM.

7.4.3. Análisis de Rendimiento

La gráfica 7.7 ilustra los tiempos obtenidos en milisegundos respecto al tamaño de bloque de datos en bits. Los tiempos mostrados son de la etapa de autenticación-cifrado del programa AES-CCM (D. Whiting, R. Housley y N. Ferguson) y de la implementación AES-CCM (S. Merino) a nivel aplicación.



Gráfica 7.7. Rendimiento en la etapa de Autenticación-Cifrado de AES-CCM.

Se obseva en la gráfica 7.7 un mejor desempeño por parte del programa AES-CCM (D. Whiting, R. Housley y N. Ferguson) que la implementación AES-CCM (S. Merino) en la etapa de autenticación-cifrado. El tiempo de ejecución obtenido en programa AES-CCM (S. Merino) es el doble que el tiempo obtenido en AES-CCM (D. Whiting, R. Housley y N. Ferguson) respecto a la longitud del bloque de datos.

Por lo tanto se recomienda hacer una optimización de la implementación AES-CCM (S. Merino) para obtener un mejor rendimiento en tiempos de ejecución.

CAPÍTULO 8

8. CONCLUSIONES

Se ha presentado las ventajas que tienen las redes inalámbricas para su aplicación como son la movilidad, escalabilidad, simplicidad, rápidez entre otras. El objetivo fundamental de las redes WLAN es el de proporcionar las facilidades no disponibles en los sistemas cableados y formar una red total donde coexistan los dos tipos de sistemas, siendo motivo de estudio en esta área.

Este trabajo de tesis estudia el estándar IEEE 802.11 en concreto se enfoca a los servicios de seguridad. Los servicios de seguridad definidos por el estándar IEEE 802.11 son: Autenticación, Confidencialidad e Integridad. Estos servicios se proveen al estándar mediante el protocolo WEP (Wired Equivalent Privacy) y CRC32. WEP esta basado en el cifrador simétrico por flujo RC4.

Para el servicio de autenticación, el estándar especifica dos modalidades: *Autenticación de Sistema Abierto*: este tipo de autenticación no garantiza el éxito del proceso de autenticación de una estación a otra, ya que en cualquier momento la estación puede rechazar autenticar a otra estación. *Autenticación de Llave Compartida*: se observa que no existe una autenticación mutua. El cliente se autentica ante el punto de acceso, pero el punto de acceso jamás se autentica ante el cliente. Este esquema abre las puertas a ataques en los cuales, algún intruso se haga pasar por el punto de acceso y pueda redirigir el tráfico de los clientes.

Para la confidencialidad el estándar IEEE 802.11 a nivel de la subcapa MAC define el protocolo *WEP*, la utilización es opcional. El WEP es definido para proteger la información de los usuarios autorizados de una WLAN de escuchas externos. Este servicio provee seguridad en WLAN equivalentes a los de las redes alámbricas. Su principal función es proveer mecanismos de seguridad en el flujo de datos en redes inalámbricas. Sin embargo WEP a desmostradó ser vulnerable como se discutió en la sección 1.2 y 3.6.

Se utiliza el CRC de 32 bits para comprobar la integridad de los datos. Cabe mencionar que CRC-32 sólo detecta alteraciones en la información cuando es accidental (clima, etc.) y no intencional (atacante).

Debido a las debilidades presentadas en cada uno de los servicios de seguridad definidos por el estándar IEEE 802.11, se propone utilizar el cifrador por bloques AES (Estándar Avanzado de Cifrado) empleando el modo de operación CCM (Modo–Contador/CBC-MAC) para ofrecer los servicios de autenticación y confidencialidad.

La razón por la cual se hace uso de AES es por que cuenta con las siguientes características: es un algoritmo público, es cifrador por bloques simétrico, la longitud de la llave es como mínimo de 128 bits, su diseño permite aumentar la longitud de la llave a 192 y 256 bits, es implementable tanto en HW como en SW y fue seleccionado del concurso que el NIST emprendió en 1997.

El modo CCM es el nombre corto de CTR + CBC-MAC. Como su nombre lo indica el CCM combina el modo de cifrado CTR con el modo de autenticación CBC-MAC. El modo de

operación CCM proporciona confidencialidad y la autenticidad de datos. CCM se basa en el cifrador simétrico por bloques AES.

Se ha presentado el modelado en UML (Unified Modeling Language) del estándar IEEE 802.11 en modo DCF (Función de Coordinación Distribuida) empleando el WEP. Se presenta el modelo con los servicios de confidencialidad y autenticación empleando AES-CCM. Este modelo incluye los modelos de análisis y diseño.

Teniendo como base el modelado se realizó la implementación a nivel aplicación del programa AES-CCM efectuando el análisis de desempeño. El equipo utilizado para realizar las pruebas cuenta con las siguientes características: Pentium IV, A 3 GHz, 512 RAM, Suse 9.2 OS.

El análisis de rendimiento a nivel aplicación de AES-CCM muestra un mejor desempeño en la longitud de bloque de datos de 1 a 16 bytes. El tiempo de ejecución obtenido va incrementando parcialmente al tamaño del bloque de datos.

Se anexó a la implementación AES-CCM el uso de Sockets para establecer la comunicación en la red inalámbrica en modo Ad-Hoc. Obteniendo los tiempos de tranferencia de envío y recepción de datos de una PC a otra.

Se realiza la comparación de la etapa de cifrado a nivel aplicación de RC4 y AES-CCM, bajo las mismas condiciones. La longitud máxima del bloque de datos manejado para las pruebas fue de 2312 bytes, por que es el tamaño límite de los datos que puede abarcar una trama IEEE 802.11.

El análisis ilustra que el cifrador por flujo RC4 tiene un mejor desempeño que AES-CCM. Se observa que en un tamaño de bloque de datos de 1 a 16 bytes, RC4 y AES-CCM presentan el mismo rendimiento. Por esta razón se recomienda el uso de AES-CCM para una longitud de bloque hasta de 16 bytes.

Se presenta un análisis de rendimiento entre AES-CCM (D. Whiting, R. Housley y N. Ferguson) y de la implementación AES-CCM (S. Merino). Se tiene un mejor desempeño de AES-CCM (D. Whiting, R. Housley y N. Ferguson), por lo que se propone realizar una optimización de la implementación propuesta AES-CCM (S. Merino).

Este trabajo de tesis ofrece una solución a los servicios de confidencialidad y autenticación mediante AES-CCM debido a las vulnerabilidades del WEP. Sin embargo no se realizó ninguna solución para ofrecer el servicio de integridad en el estándar IEEE 802.11. Se propone utilizar Funciones Hash, MD5 y SHA1. De esta forma se cubren los tres servicios de seguridad definidos por el IEEE 802.11.

La forma en que se maneja la autenticación por el modo de operación CCM es a nivel de mensaje, mientras que la autenticación que se ofrece por el estándar IEEE 802.11 es a nivel de dispositivos que soporten este estándar.

Al utilizar AES-CCM para el servicio de confidencialidad, se define la longitud de la llave privada de 128 bits siendo establecida de antemano. Por lo que surge un nuevo problema la distribución de llaves, para esto se recomienda emplear Criptosistemas de Curvas Elípticas.

Se propone realizar una optimización a la implementación de AES-CCM para tener un mejor desempeño.

En concreto se hizo un programa a nivel aplicación de AES-CCM y AES-CCM empleando sockets, realizando un análisis de rendimiento en cada programa. Sin embargo esta aplicación es una aproximación para que se lleve a nivel protocolario.

Desafortunadamente no se pudo llegar a nivel protocolario debido a la falta de documentación e información para realizarse.

REFERENCIAS

[1]. ANSI/*IEEE Standard 802.11-1999*, IEEE Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, New York: IEEE Press, 1999, págs. 10-84.

[2]. J. Daemen, and V. Rijmen, *The Design of Rijndael, AES-The Advanced Encryption Standard*, Springer-Verlag Berlin Heidelberg, 1st edition (February 15, New York 2002), págs. 31-50.

[3]. D. Whiting, R. Housley, and N. Ferguson, "*Counter with CBC-MAC (CCM) AES Mode of Operation",* Contribution to NIST, June 2002, págs. 1-9. Disponible en: http://csrc.nist.gov/encryption/modes/proposedmodes/

[4]. J. Park and D. Dicoi Syracuse University *"WLAN Security: Currentand Future"*, IEEE Computer Society, SEPTEMBER - OCTOBER 2003, págs. 60-65.

[5]. S. Fluhrer, I. Mantin and A. Shamir, "*Weaknesses in the Key Scheduling Algorithm of RC4"*, 8th Annual Workshop on Selected Areas in Cryptography, August 2001, págs. 1-23.

[6]. Jesse R. Walker, Intel Corporation*, "Unsafe at any key size; An analisys of the WEP encapsulation"*, doc. IEEE 802.11-00/362, October 2000, págs. 1-9.

[7]. A. Stubblefield, J. Ioannidis, A. D. Rubin, *"Using the Fluhrer, Mantin, and Shamir Attack to Break WEP"*, AT&T Labs Technical Report TD–4ZCPZZ, August 2001, págs. 1-11.

[8]. Recomendación UIT-T X.810. Redes de Datos y Comunicación entre Sistemas Abiertos - Seguridad - Tecnología de la Información – Interconexión de Sistemas Abiertos – *Marcos de Seguridad para Sistemas Abiertos: Marco de Confidencialidad.*

[9]. Recomendación UIT-T X.810. Redes de Datos y Comunicación entre Sistemas Abiertos - Seguridad - Tecnología de la Información – Interconexión de Sistemas Abiertos – *Marcos de Seguridad para Sistemas Abiertos: Visión General.*

[10]. W. Trappe and L. Washington. *Introduction to Cryptography with coding theory*. 2nd Edition, Upper Saddle River, NJ: Prentice Hall, 2002.

[11]. IEEE 802.11g-2003 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band

[12]. IEEE Std 802.11i/D7.0, October 2003. (Draft Amendment to ISO/IEC 8802-11/1999(I) ANSI/IEEE Std 802.11, 1999 edition) Draft A STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements.

[13]. David Halasz, *IEEE 802.11i and wireless security*, Agust 2004. Disponible en: http://www.embedded.com

[14]. Haartsen J., Naghshineh M., Inouye J., *"Bluetooth: Vision, Goals, and Architecture"*, 1998, et al.Mobile Computing and Communications Review

[15]. J. Khun-Jush et al. *"HiperLAN2: Broadband Wireless Communications at 5 GHz"*. IEEE Communications Magazine, vol. 40, nº 6, junio 2002.

[16]. Chinitz, Leigh, *"HomeRF Technical Overview"*, May 9, 2001, aviable in the official site: www.homerf.org/data/events/past/pubseminar_0501/tech_overview.pdf

[17]. IEEE Std 802.1X-2001, "Port-Based Network Access Control", junio de 2001.

[18]. Francisco Javier Zuluaga Ramírez y Juan Manuel Cruz Alcaraz, *Servicios de Seguridad en el protocolo para redes inalámbricas IEEE 802.11x*, W. Trrappe and L. Washington, págs. 1-12.

[19]. WAP Forum. *WAP-261-WTLS-20010406-a*, WTLS Specification. 2001. Versión 06-Abril-2001, Disponible en http://www.wapforum.com.

[20]. N. Borisov, I. Goldberg, and D. Wagner, *"Intercepting Mobile Communications: The Insecurity of 802.11"*, Disponible en: http://www.isaac.cs.berkeley.edu/isaac/wep-draft.pdf

[21]. William A. Arbaugh, Narendar Shankar, and Y.C. Justin Wan, *"Your 802.11 Wireless Network Has No Clothes"*, Disponible en: http://www.cs.umd.edu/~waa/wireless.pdf

[22]. Francisco López Ortiz, *El estándar IEEE 802.11 Wireless LAN*. 2000.

[23]. Qiang Ni, Lamia Romdhani, Thierry Turletti, and Imad Aad. *QoS Issues and Enhancements for IEEE 802.11 Wireless LAN,* Théme 1-Reseaux et systémes, Projet Planete, Rappor de recherche N° 4612-November 2002, ISSN 0249-6399, ISRN INRIA/RR--4612--FR+ENG, págs. 1-36.

[24]. *IEEE Standard 1363-2000.* IEEE Standard Specifications for Public-Key Cryptography, New York: IEEE Press, 2000.

[25]. Diffie W. and Hellman M. New directions in cryptography. *IEEE Trans. Information Theory*, pages 644–654, 1976. IT-22(6).November 1976.

[26]. W. Stallings*, Cryptography & Network Security: Principles and Practice*, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, 1998, págs. 72-74, 121-130, 215-218.

[27]. Manuel J. Lucena López, *Criptografía y Seguridad en Computadores,* Tercera Edición (Versión 2.15), Marzo del 2004, págs. 72-74, 121-126, 130.

[28]. Joan Daemen and Vincent Rijmen, *AES submission document on Rijndael*, Version 2, September 1999. Disponible en: http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf

[29]. C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. Matyas, L. O'Connor, M. Peyravian, D. Safford and Nevenko Zunic, *MARS - a candidate cipher for AES, IBM Corporation*, Revised, September, 22 1999.

[30]. FIPS PUB 197, *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology, U.S. Department of Commerce, November 2001. Disponible en: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[31]. M. Bellare, T. Kohno, and C. Namprempre, *"Authenticated encryption in SSH: provably fixing the SSH binary packet protocol"*, Proceedings of the 9th ACM conference on Computer and Communications Security (CCS-02), ACM Press, 2002.

[32]. M. Bellare and C. Namprempre, *"Authenticated encryption: Relations among notions and analysis of the generic composition paradigm"*, Advances in Cryptology - ASIACRYPT '00, Lecture Notes in Computer Science, vol. 1976, T. Okamoto, ed., Springer-Verlag, 2000.

[33]. J. Jonsson, *"On the Security of CTR + CBC-MAC NIST Modes of Operation –Additional CCM Documentation.Proceedings from SAC (Selected Areas of Cryptography) 2002"*, St John's, Newfoundland, August 2002.

[34]. D. Whiting, R. Housley, and N. Ferguson, "*AES Encryption & Authentication Using CTR Mode &CBC-MAC"*, IEEE P802.11 doc 02/001r2, May 2002.

[35]. D. Whiting, R. Housley, and N. Ferguson, "*Counter with CBC-MAC (CCM)",* IETF, September 2003. Disponible en: ftp://ftp.rfc-editor.org/in-notes/rfc3610.txt

[36]. NIST Special Publication 800-38C, "*Recommendation for Block Cipher Modes of Operation: the CCM mode for Authentication and Confidentiality",* U.S. DoC/NIST, May 2004.

[37]. D. Johnston and J Walker, *802.16 Security Enhancements*, IEEE 802.16 Presentation Submission, 2003.

[38]. P. Rogaway, M. Bellare, J. Black, and T. Krovetz, *"OCB: A block-cipher mode of operation for efficient authenticated encryption"*, Eighth ACM Conference on Computer and Communications Security (CCS-8), ACM Press, 2001.

[39]. I. Jacobson, G. Booch y J. Rumbaugh. *El proceso Unificado de Desarrollo de Software.* Pearson Education. Madrid 2000.

[40]. I. Jacobson, G. Booch y J. Rumbaugh. *El Lenguaje de Modelado Unificado, Guía de Usuario.* Pearson Education. Madrid 2000.

[41]. S. Merino y M. León, *"Modelado en UML del servicio de Privacidad usando AES (Estándar Avanzado de Cifrado) para el estándar IEEE 802.11"*, Segundo Congreso Nacional de Ciencias de la Computación "Retos en las Tecnologías de Información", Noviembre 2004, págs. 198-203, ISSN: 968 863 798 X.

[42]. R. Aldeco, M. A: León, *"Modelling with UML of the Authentication Services using Elliptic Curves for IEEE 802.11"*, 2nd.Computer Science National Congress, BUAP, Puebla, Mexico, November 2004.

[43]. S. Merino and M. León, *"UML Model of the IEEE 802.11 Privacy Service using AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC)"*, XV Congreso Interuniversitario Electrónica, Computación y Eléctrica, Marzo 2005.

[44]. S. Merino and M. León, *"UML Model of the IEEE 802.11 Privacy Service using AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC) and AES Implementation"*, *Research on Computing Science*, vol. 13, págs. 137-147, Mayo 2005, ISSN: 1665-9899.

[45]. Michael Welschenbach, *Cryptography in C and C++* Apress; 2001.

[46]. M. León, R. Aldeco and S. Merino *"Performance Analysis of the Confidentiality Security Service in the IEEE 802.11 using WEP, AES-CCM, and ECC"*, 2nd International Conference on Electrical and Electronics Engineering and XI Conference on Electrical Engineering, Septiembre 2005.

[47]. Hankerson, D., Menezes, A., and Vanstone, S. "Guide to Elliptic Curve Cryptography", Springer-Verlag, New York, 2004

[48]. D. Whiting, R. Housley, and N. Ferguson, "*Test Vectors - Counter with CBC-MAC (CCM) AES Mode of Operation",* Contribution to NIST, June 2002, págs. 1-9. Disponible en: http://csrc.nist.gov/encryption/modes/proposedmodes/

[49]. Pagina oficial de Suse Linux en español http://www.novell.com/es-es/linux/suse/

[50]. Linux driver for the 802.11 g Prism GT, Prism Duette, Prism Indigo ChipSets htttp://www.prism54.org

ÁPENDICE A

A. INSTALACIÓN SUSE 9.2

Introduzca el primer CD-ROM o el DVD de SUSE LINUX en el lector correspondiente. Después de reiniciar el ordenador, SUSE LINUX arranca desde el medio que se encuentra dentro del lector y se inicia el proceso de instalación.

Una vez instalado el sistema podemos completarlo con las instalaciones vía red [49].

La pantalla de bienvenida

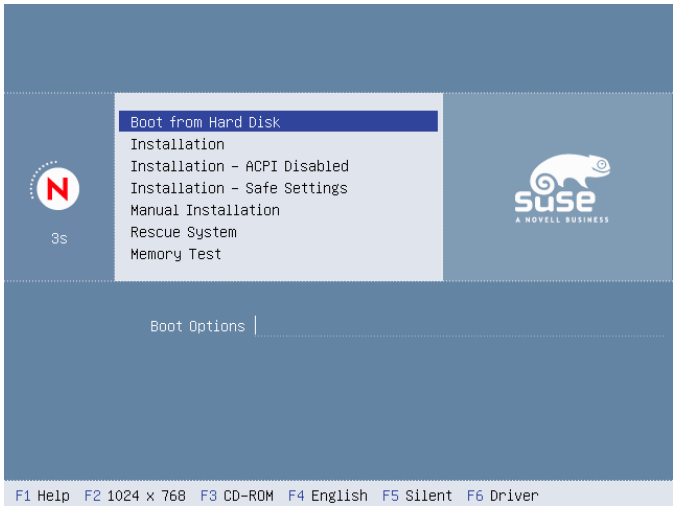


Figura 1. La pantalla de bienvenida.

La pantalla de inicio muestra varias posibilidades para el desarrollo posterior del proceso de instalación. En la parte superior se encuentra la opción 'Boot from Harddisk', que arranca el sistema ya instalado. Debido a que una vez realizada la instalación a menudo se introduce el CD para instalar otros componentes de software, esta opción está preseleccionada. No obstante, seleccione para la instalación la opción 'Installation' con las teclas de cursor (flechas). A continuación se cargará YaST y comenzará la instalación.

Selección del idioma

Es posible seleccionar el idioma deseado para SUSE LINUX y YaST. El idioma elegido se aplica también a la configuración del teclado y YaST define además una zona horaria estándar que es la más apropiada para su configuración de idioma. Estas opciones pueden modificarse posteriormente. Si contra toda previsión el ratón todavía no funciona, utilice las flechas del teclado hasta llegar al idioma deseado, a continuación pulse **Tab** hasta que el botón 'Siguiente' esté activado y finalmente pulse la tecla **Intro**.

Figura 2. Selección del idioma.

Modo de instalación

El usuario puede decidir si quiere realizar una 'Nueva Instalación' o 'Actualizar un sistema existente'. Evidentemente sólo puede realizar una actualización si ya tiene SUSE LINUX instalado. Este sistema ya instalado se puede arrancar con la opción 'Arrancar el sistema instalado'. Si en algún caso el sistema SUSE LINUX dejara de arrancar (por ejemplo porque se ha borrado accidentalmente una parte importante del sistema), puede utilizar la opción 'Reparar el sistema instalado' para intentar que el sistema pueda arrancarse de nuevo. Si hasta ahora no ha instalado ningún SUSE LINUX, sólo puede realizar una instalación nueva.

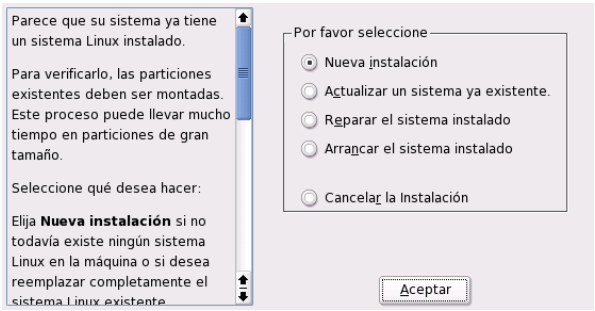


Figura 3. Selección del modo de instalación

Propuesta para la instalación

Después de la detección del hardware, aparecerá el diálogo de propuestas (ver figura 4) con información sobre el hardware detectado, las propuestas de instalación y de particiones. Si pulsa sobre una de las opciones y después la configura, al acabar siempre volverá a aparecer con los nuevos valores en el mismo diálogo de propuestas.

Figura 4. Ventana de diálogo de propuestas.

Configuración del teclado

Seleccione en este diálogo la distribución del teclado deseada. Generalmente coincide con el idioma seleccionado. Compruebe la configuración pulsando algunas teclas, sobre todo y/z y los caracteres acentuados. Si no aparecen los caracteres esperados, es porque la distribución del teclado aún no es la correcta. Con 'Siguiente' puede volver a las propuestas.

Ratón

En caso de que YaST no haya detectado automáticamente el ratón, muévase con la tecla **Tab** hasta que esté activado el botón 'Cambiar'. Pulse entonces **Espacio** y después las teclas de dirección hasta llegar al punto 'Ratón'. Pulsando **Intro** aparece el diálogo para la selección del tipo de ratón.

Utilice las teclas ↑ y ↓ para seleccionar el ratón. Si conserva la documentación del ratón, encontrará allí una descripción del tipo de ratón. Con la combinación de teclas **Alt** + **T** puede seleccionar el ratón temporalmente para probarlo. Si el ratón no reacciona como se espera, seleccione un nuevo tipo con el teclado y compruébelo. Pulse **Tab** e **Intro** para hacer la selección permanente.

Particionar

Si ha seleccionado la partición en la ventana de diálogo de propuestas, aparecerá el diálogo de particiones de YaST con la configuración actual. Puede aceptar, cambiar o eliminar las opciones de configuración en caso de que quiera realizar una nueva distribución del espacio.

Al seleccionar 'Aceptar la propuesta tal y como está', no se efectuará ninguna modificación y el diálogo de propuesta se quedará como está. Al seleccionar 'Particionar basándose en esta propuesta', aparecerá directamente el diálogo para expertos que permite definir opciones de configuración muy detalladas.

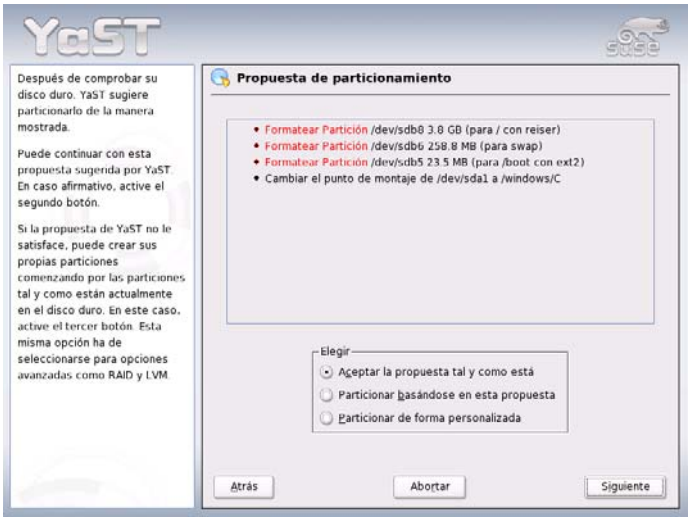


Figura 5. Editar propuesta de particiones.

Al escoger 'Particionar de forma personalizada', aparecerá un diálogo en el que se puede seleccionar el disco duro (ver figura 6). Aquí verá una lista de todos los discos duros disponibles en el sistema. Escoja aquel en el que quiera instalar SUSE LINUX.

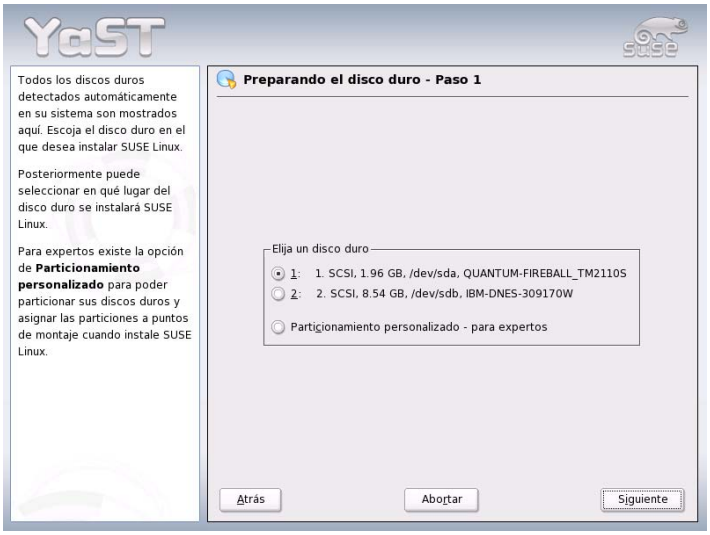


Figura 6. Selección del disco duro.

En el diálogo de expertos (ver figura 7) puede modificar manualmente el particionamiento de uno o varios discos duros así como añadir, eliminar o editar particiones.

La lista del diálogo de experto muestra todos los discos duros y todas las particiones (ya sean existentes o propuestas). Los discos duros se visualizan como dispositivos sin números (por ejemplo /dev/hda o /dev/sda) mientras que las distintas particiones se representan como partes de estos dispositivos (por ejemplo /dev/hda1 o /dev/sda1). También se muestra el tamaño, tipo, sistema de archivos y punto de montaje de todos los discos duros y particiones. El punto de montaje determina el directorio que se usa para integrar una partición en el árbol de archivos de Linux.

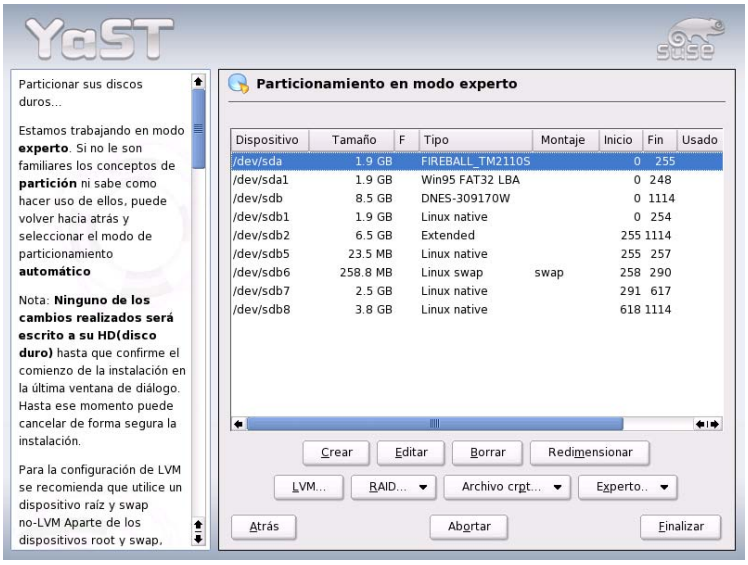


Figura 7. El particionador de YaST en modo experto.

Asimismo se muestra el espacio libre del disco duro y se selecciona de forma automática. Si quiere disponer de más espacio para Linux, puede liberar un disco duro para esta función seleccionando dicho disco duro de la lista, comenzando desde abajo hacia arriba, o sea en la secuencia de la última a la primera partición. Sin embargo, no se puede por ejemplo escoger la segunda de tres particiones para Linux y dejar la primera y tercera para otro sistema operativo.

Software

SUSE LINUX incluye una gran cantidad de software que se instala según el perfil del usuario. Seleccionar por separado los paquetes de software del gran conjunto disponible sería muy tedioso. Por este motivo, SUSE LINUX ofrece varios subconjuntos preconfigurados. De acuerdo al espacio de disco disponible, YaST selecciona automáticamente uno de estos subconjuntos y muestra esta propuesta.

Mínima (recomendada sólo para aplicaciones especiales)

Sólo se instala el sistema operativo con diferentes servicios. No hay entorno gráfico y el control del ordenador se realiza por medio de consolas de texto. Este tipo de sistema es ideal para aplicaciones de servidor que requieren poca o ninguna interacción con el usuario.

Sistema gráfico mínimo (sin KDE)

Si le falta espacio de disco y no le gusta el escritorio KDE, instale este conjunto de software. El sistema dispone de un entorno gráfico básico con ventanas de terminal, pero le faltan las habituales funciones de arrastrar y soltar. Sin embargo, pueden utilizarse todos los programas que cuentan con una interfaz gráfica propia (ej. Netscape). No se instala ningún programa ofimático.

Sistema estándar (con GNOME y paquete ofimático)

Este es el sistema estándar más grande disponible. Contiene el escritorio GNOME con la mayoría de sus programas y los paquetes ofimáticos. Este es el tipo de instalación idóneo para estaciones de trabajo. YaST lo selecciona si encuentra suficientes recursos para ello.

Sistema estándar (con KDE y paquete ofimático)

Este es el sistema estándar más grande disponible. Contiene el escritorio KDE con la mayoría de sus programas y los paquetes ofimáticos. Este es el tipo de instalación idóneo para estaciones de trabajo. YaST lo selecciona si encuentra suficientes recursos para ello.

Al pulsar 'Software' en el apartado de propuestas puede seleccionar uno de los sistemas básicos. Además puede iniciar el módulo de selección de software (es decir, el administrador de paquetes), pulsando en 'Selección detallada' para modificar individualmente la selección de software instalada (ver figura 8).

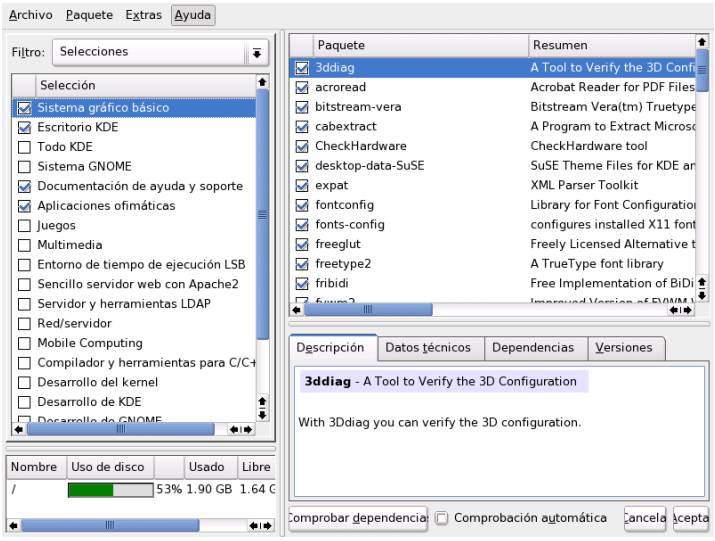


Figura 8. YaST: instalar y eliminar software (administrador de paquetes).

Cargador de arranque

YaST determina correctamente el modo de arranque durante la instalación por lo que, en circunstancias normales, puede adoptar estas configuraciones sin necesidad de modificarlas. No obstante, si necesita cambiar la configuración predeterminada debido a requisitos especiales del sistema, también podrá hacerlo.

Configuración de la zona horaria

En este diálogo, en el campo 'Reloj de hardware configurado para', puede elegir entre las opciones 'Hora local' y 'GMT'. Su selección depende de la configuración del reloj en la BIOS del ordenador. Si está configurado con el valor GMT, SUSE LINUX se encarga de cambiar automáticamente entre horario de verano y de invierno.

Realizar la instalación

Al pulsar 'Siguiente' acepta la propuesta con todos los cambios realizados y llega al diálogo verde de confirmación. Si elige 'Sí, instalar' la instalación se inicia con las opciones seleccionadas. Dependiendo de la capacidad de la CPU y la selección de software, la instalación dura generalmente entre 15 y 30 minutos. Después de la instalación de paquetes, YaST inicia el sistema instalado antes de continuar con la configuración del hardware y los servicios.

Completar la instalación

Una vez que el sistema y el software seleccionado han sido instalados, deberá especificar una contraseña para el administrador del sistema (usuario root). A continuación tendrá la oportunidad de configurar el acceso a Internet y la conexión de red. De esta forma podrá instalar actualizaciones de software para SUSE LINUX durante la instalación y configurar servicios de DNS para la gestión central de usuarios en la red. Finalmente, podrá configurar el hardware conectado.

Contraseña de root

Figura 9. Definir la contraseña para el usuario root.

Root es el nombre del superusuario o administrador del sistema que tiene todos los permisos de los que carece un usuario normal. Para definir la contraseña de root tiene que seguir el mismo proceso que para definir la contraseña de un login normal. Hay que introducir la contraseña dos veces para su comprobación (ver figura 9).

Configuración de red

En el siguiente paso tiene la oportunidad de conectar su sistema al resto del mundo. Puede configurar la tarjeta de red, RDSI, módem y DSL. Si el sistema está equipado con este tipo de hardware, aproveche esta ocasión. En ejecuciones posteriores de YaST se pueden descargar actualizaciones de Internet para SUSE LINUX que se tendrán en cuenta durante la instalación.

Crear usuarios locales

Para crear cuentas de usuario se utiliza el diálogo de la figura 10. Debe indicar su nombre y apellidos y elegir también un nombre de usuario. Si no se le ocurre ningún nombre de usuario adecuado, puede crearlo automáticamente pulsando el botón 'Sugerencia'.

Por último hay que definir una contraseña para el usuario. Tiene que introducirla dos veces para su comprobación.

Figura 10. Indicar nombre de usuario y contraseña.

Configuración de hardware

Después de haber completado la instalación se mostrará un diálogo en el que puede configurar la tarjeta gráfica junto con diversos componentes de hardware conectados al sistema como impresoras o tarjetas de sonido. Si pulsa sobre los diferentes componentes puede iniciar la configuración del hardware. YaST detecta y configura el hardware de forma automática.

Login gráfico

Ahora SUSE LINUX está instalado. Si el login automático está activado, puede utilizarlo directamente sin pasos adicionales. En caso contrario, aparece en el monitor el login gráfico que puede ver en la figura 11. Introduzca el nombre de usuario definido anteriormente y su contraseña para entrar al sistema.

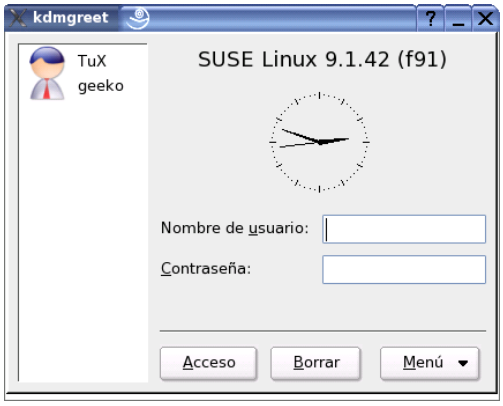


Figura 11. Entrar al sistema (KDE).

ÁPENDICE B

B. INSTALACIÓN TARJETA INALÁMBRICA SMC 2802W V.2

Suse soporta las tarjetas que funcionan bajo las especificaciones 802.11a, b y/o g. Las tarjetas actuales se basan, por lo general, en el estándar 802.11g, aunque aún existen tarjetas 802.11b. Principalmente, se soportan tarjetas con los siguientes chips:

- Lucent/Agere Hermes™
- Intel PRO/Wireless 2100™
- Intersil Prism2/2.5/3™
- Intersil PrismGT™
- Atheros 5210, 5211, 5212™
- Atmel at76c502, at76c503, at76c504, at76c506™
- Texas Instruments ACX100™

Considerando la información anterior para facilitar el reconocimiento de la tarjeta sin ningún problema en el sistema operativo se cuenta con la tarjeta inalámbrica SMC2802W, que tiene el chip PRISM Nitro. La tarjeta inalámbrica EZ ConnectTMg PCI a 54 Mbps se utiliza para las pruebas y la programación de este documento.

En la instalación de Suse 9.2, fue detectada la tarjeta SMC2802W asignando automáticamente los controladores para su funcionamiento [50], desafortunadamente los controladores asignados (de una tarjeta alámbrica) provoca conflictos con la configuración.

Debido al problema anteriormente planteado, se realizó lo siguiente para poner a funcionar la tarjeta inalámbrica, cabe mencionar que para poder dar esta solución se probaron otras soluciones las cuales fueron un rotundo fracaso.

Primero se obtiene el software ndiswrapper de: http://ndiswrapper.sourceforge.net, este archivo se baja en la carpeta */root*. El software ndiswrapper toma los controladores del dispositivo para Windows y los adapta para que funcionen en Linux.

Como segundo paso indispensable es tener los controladores de la tarjeta inalámbrica, la liga es: http://www.smc-europe.com/english/support/2802W_V2.html, se crea una carpeta llamada *drivers* en el directorio de *root*, entonces se guarda en la carpeta */root/drivers*. Recuerde poner el archivo con extensión *.inf* y el *.sys* dentro de esta misma carpeta.

Ahora se realiza la instalación del ndiswrapper: se descomprime el archivo y en una consola en línea de comando se pone *make install*.

Una vez terminada la instalación se pone el siguiente comando: *ndiswrapper –i /root/drivers/2802w.inf*

Como siguiente paso es la configuración de la tarjeta inalámbrica. Se accede al *YAST* al menú *Network Devices>Network Card* y se elimina la configuración que por defauld creó el YAST para la tarjeta inalámbrica, por ejemplo: *eth0*.

Una vez eliminada la instalación anterior para la tarjeta inalámbrica. Se elije *other no detected* para crear una nueva instalación, dar click en *Configurar*. Se selecciona de tipo *wireless*, en nombre se pone *0* y *static-1*. Las opciones siguientes de configuración se seleccionan de acuerdo a las necesidades del usuario.

Ahora se escribe *prism54* en el archivo */etc/hotplug/blacklist*.

Nuevamente se accede al *YAST* en el icono de *Sistema* > *etc/sysconfig editor*. En el lado izquierdo se accede al directorio *Hardware/Hotplug*. Se selecciona la opción de HOTPLUG_PCI_DRIVERTYPE_BLACKLIS y en el lado derecho se anexa *net/wireless*.

Nuevamente en la parte izquierda se selecciona la opción de COLDPLUG_PCI_CLASSES_BLACKLIST y se anexa *0d* en el lado derecho. Se da guardar los cambios. Por lo que se procede a reiniciar la computadora.

Se abre una consola y se pone el comando: *modprobe ndiswrapper*. Tarda unos minutos sin retornar ningún valor, en caso contrario de que haya marcado error, se repite el proceso desde el inicio.

En caso que todo sea un exitó se escribe el comando *dmesg* y se despliega el siguiente mensaje:

ndiswrapper: no versión for “struct_module” found:kernel tainted
ndiswrapper: unsopported module, taining kernel
ndiswrapper:versión 0.10 loaded (preempt=yes, smp=no)
ndiswrapper:using irq 11
ndiswrapper:Windows driver trying to use uninitialized lock cf5d9f74, fixing it.
wlan0: ndiswrapper ethernet device 00:04:e2:b4:58:09 using driver 2802w2.sys
ndiswrapper:device wlan0 supports WPA with AES/CCMP and TKIP ciphers
ndiswrapper: driver 2802w.sys (SMC, 04/29/2004, 3.0.11.1) added
wlan0: no IPv6 routers present

Ahora mediante el comando *iwconfig* se despliega la información acerca de la configuración inalámbrica detectada, como se muestra a continuación:

lo no wireless extensions

sit0 no wireless extensions

wlan0 IEEE 802.11g ESSID:off/any Nicknam:”linux”
Mode:Ad-Hoc Frecuency:2.462GHz Cell:00:00:00:00:00:00
Bit Rate=2Mb/s Tx-Power:32 dBm
RTS thr=2347 B Fragment thr=2346 B
Encription Key:off Power Management:off
Link Quality:100 Signal level:0 Noise level:0
Rx invalid nwid:0 Rx invalid crypt:0 Rxinvalid frag:0
Tx excesive retries:0 Invalid misc:0 Missed beacon:0

Se puede utilizar el comando *ifconfig* para tener conocimiento de las configuraciones de todos los dispositivos de redes.

wlan0 Link encap: Ethernet Hwaddr 00:04:e2:b4:58:09
inet addr:192.168.100.6 Bcast:192.168.100.255 Mask 255.255.255.0
inet6 addr:fe80::204:e2ff:feb4:5809/64 Scope:link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Rx paquets:0 errors:0 dropped:0 overruns:0 frame:0
Tx paquets:6 errors:0 dropped:0 overruns:0 frame:0
Collisions:0 TxQueueLen:1000 Rx bytes:0 (0.0b)
Tx bytes:460 (460.0b) Interrupt:11 Memory:ed000000-ed001fff

Si ha podido obtener todo como se mostro en la parte anterior entonces la tarjeta esta configurada y funcionando de manera adecuada. Puede utilizar las wireless-tools incluidas en el kernel. Por ejemplo:

Apagado: ifdown wlan 0
Preder: ifup wlan0
Checar estado: ifstatus wlan0

Entre otras opciones más que puedes manipular accediendo al manual de las wireless-tools (man wireless-tools).