

Universidad de Buenos Aires  
Facultad de Ciencias Exactas y Naturales  
Departamento de Computación

Tesis para optar al grado de Licenciado en Ciencias de la Computación

# **Una propuesta para implementar un canal encubierto de red en el protocolo IEEE 802.11**

por

**Leandro Federico Meiners  
(lmeiners@dc.uba.ar)**

Directores:

Lic. Rodolfo Baader (rbaader@dc.uba.ar)

Lic. Ariel Waissbein (ariel.waissbein@coresecurity.com)

2010

---

# Índice general

---

<b>Índice general</b>	<b>1</b>
<b>1 Introducción</b>	<b>2</b>
<b>2 Conceptos previos</b>	<b>5</b>
2.1. Canales encubiertos . . . . .	5
2.2. Protocolo IEEE 802.11 . . . . .	6
2.2.1. Componentes de la arquitectura . . . . .	7
2.2.2. Capa de control de acceso al medio . . . . .	8
2.3. Ataques a WEP . . . . .	13
<b>3 Estado del arte y tecnología</b>	<b>15</b>
3.1. Estado de la tecnología inalámbrica IEEE 802.11 . . . . .	15
3.1.1. Placas de red . . . . .	15
3.1.2. Herramientas de desarrollo . . . . .	17
3.2. Trabajos previos relacionados . . . . .	17
<b>4 Problemática y solución propuesta</b>	<b>20</b>
4.1. Motivación del trabajo . . . . .	20
4.2. Requisitos de un canal encubierto . . . . .	22
4.3. Problemática de las soluciones existentes . . . . .	23
4.4. Canal encubierto propuesto . . . . .	28
4.5. Implementación mediante la modificación de un controlador de red	35
4.5.1. Método de señalización . . . . .	35
4.5.2. Arquitectura de la solución . . . . .	36
4.6. Implementación mediante la inyección de tráfico . . . . .	37
4.6.1. Método de señalización . . . . .	38
4.6.2. Datos de las tramas . . . . .	38

4.7. Implementación en redes protegidas por WPA/WPA2 y redes sin protección . . . . .	40
<b>5 Análisis de resultados y escenarios de uso</b>	<b>41</b>
5.1. Comparación entre las implementaciones . . . . .	41
5.1.1. Medición de ancho de banda . . . . .	43
5.2. Comparación con otras implementaciones de canales encubiertos sobre IEEE 802.11 . . . . .	43
5.3. Ejemplos de uso . . . . .	44
5.4. Escenarios de uso . . . . .	45
<b>6 Conclusiones y trabajo futuro</b>	<b>46</b>
6.1. Conclusiones . . . . .	46
6.2. Trabajo futuro . . . . .	47
<b>Referencias</b>	<b>49</b>
<b>A Mensajes soportados por el protocolo IEEE 802.11</b>	<b>53</b>
<b>B Instalación del controlador</b>	<b>56</b>
<b>C Uso de las herramientas adicionales</b>	<b>57</b>
<b>D Uso de las herramienta de inyección</b>	<b>58</b>
<b>Índice de figuras</b>	<b>59</b>
<b>Índice de cuadros</b>	<b>60</b>

---

# Agradecimientos

---

Este trabajo, que finalmente sale a la luz, no hubiese sido posible sin la guía, comentarios, correcciones y logística aportados por Rodolfo Baader, ni los comentarios y correcciones de Ariel Waissbein. Tampoco sin las discusiones e ideas de Ariel Futoransky. A ellos les agradezco por la posibilidad de hacer este trabajo para poder, finalmente, obtener el título de Licenciado.

Por otro lado quiero agradecer a mis padres y hermanos, por acompañarme todos los años de la carrera, y a mi compañera de ruta, Jimena, por aguantarme todos los días y no presionarme por terminar la tesis (presentada varios largos años después del último final...).

Por último quiero agradecer a mis compañeros de TP, por todas las interminables noches de TP, los fines de semana y los días de estudio en el bar del pabellón uno. Sin ellos dudo que hubiese podido llevar adelante la carrera.

# Capítulo 1

---

## Introducción

---

Las técnicas de ocultación de información pueden ser rastreadas hasta civilizaciones antiguas ([28]). Los canales encubiertos, una forma de ocultar información, fueron definidos en la era digital como “aquellos no intencionados para la transferencia de información” por B. W. Lapmson ([25]). Posteriormente el Departamento de Defensa de Estados Unidos los definió como “cualquier canal de comunicación que puede ser explotado por un proceso para transferir información en violación de la política de seguridad del sistema” ([27]). En lo que respecta a canales encubiertos en redes de comunicaciones, la definición genérica anterior puede ser especializada a “una manipulación de un protocolo de comunicación para transferir información de forma no prevista por la especificación del protocolo” ([35]). Los canales encubiertos son comúnmente utilizados para poder enviar y recibir información de forma anónima (difícil detección del origen) y mediante un canal confidencial<sup>1</sup> (difícil detección de la existencia). En el caso particular de un canal encubierto de red, la difícil detección de existencia se traduce en que no es posible distinguir el uso, o el no uso, del mismo.

Las redes de comunicaciones requieren de un medio físico para la transmisión de información. Históricamente, las redes de área local utilizaban cableado para comunicar a los integrantes de la misma. Actualmente, existe una tendencia a utilizar redes inalámbricas, cuyo medio físico de propagación es el aire, y por lo tanto poseen menores costos de instalación, y son ubicuas (dentro del rango de alcance); permitiendo una mayor flexibilidad respecto a la ubicación y movilidad de sus integrantes.

El protocolo de redes inalámbricas para redes de área local más utilizado actualmente es el protocolo **802.11** estandarizado por **IEEE** ([18]). La primera versión del estándar ([18]) data de 1997 y define al protocolo **WEP** (Wired Equivalent Privacy) para ser utilizado en redes que requieren confidencialidad. En la actualidad la seguridad (confidencialidad, integridad y disponibilidad) de los datos, comunicaciones

---

<sup>1</sup>Cabe aclarar que confidencial refiere al canal en sí mismo y no al carácter de los datos enviados por el mismo ni si los mismos se encuentran cifrados.

y sistemas es una necesidad de las organizaciones y crecientemente también de los individuos.

El protocolo **WEP** se basa en el uso del algoritmo criptográfico **ARC4**[24], para el cual se detectaron debilidades en el algoritmo de generación de sub-claves ([37]). En el año 2002 se publicó un trabajo que explicaba como utilizar las debilidades del algoritmo criptográfico **ARC4** para recuperar la clave **WEP** de una red protegida, efectivamente anulando la protección brindada por el algoritmo. Poco tiempo después surgieron herramientas públicas ([29]) que implementaban los ataques. Como respuesta a este problema de **WEP**, la asociación **Wi-Fi Alliance** (compuesta por las empresas más importantes de la industria de redes inalámbricas) publicó el estándar **WPA** (Wi-Fi Protected Access) ([2]) que sigue utilizando el algoritmo criptográfico **ARC4** pero imposibilita los ataques de **WEP**. Posteriormente, **IEEE** publicó el anexo **IEEE 802.11i** ([17]) al estándar **IEEE 802.11** ([18]) que incorpora **TKIP** (comercialmente conocido como **WPA**) y también agrega otro protocolo de confidencialidad que utiliza el algoritmo de cifrado **AES** ([13]) en modo **CCMP**, comercialmente conocido bajo el nombre **WPA2**.

A pesar de la existencia de herramientas públicas ([39]) para atacar redes **WEP**, sigue habiendo investigación en el tema ([41]) dada la popularidad del protocolo; principalmente relacionada con la gran cantidad de hardware instalado que únicamente soporta dicho protocolo y los costos en recursos y tiempos que implican una migración.

A pesar de la existencia de herramientas públicas ([39]) para atacar redes **WEP**, sigue habiendo investigación en el tema ([41]) dada la popularidad del protocolo; principalmente relacionada con la gran cantidad de hardware instalado que únicamente soporta dicho protocolo<sup>11</sup> y los costos en recursos y tiempos que implican una migración. Cabe resaltar que la última versión del estándar **IEEE 802.11** ([18]), que data del 2007, indica que el protocolo **WEP** se encuentra obsoleto y **TKIP** o **AES-CCMP** deben ser utilizados en su lugar.

Los ataques al protocolo **WEP** conocidos atacan la confidencialidad provista por el protocolo ([37], [6], [10], [3], [42], [41]). Los mismos pueden ser divididos en dos clases: aquellos que permiten recuperar la clave de cifrado utilizada y aquellos que permiten recuperar parte del flujo pseudoaleatorio (*keystream*) utilizado para cifrar. La primera clase de ataques vulnera completamente la confidencialidad brindada por **WEP** mientras que la segunda permite enviar información y leer únicamente una pequeña parte del tráfico. Sin embargo, la primera clase de ataques suele ser más fácil de detectar por sistemas de detección de intrusiones para redes inalámbricas, por ejemplo las herramientas de AirDefense ([19]), AirTight ([23]) o AirMagnet ([22]), ya que son ataques activos que requieren enviar cientos o miles de tramas con propiedades particulares; de forma tal que se pueden construir "*signatures*" que los sistemas

---

<sup>11</sup>A pesar de que **TKIP** fue diseñado para funcionar en el mismo hardware que **WEP** no todos los fabricantes proveen actualizaciones para utilizarlo, siendo necesario adquirir nuevo hardware.

de detección de intrusiones son capaces de detectar.

En este trabajo se muestra la implementación de un canal encubierto de red sobre el protocolo **IEEE 802.11** (en todas sus versiones ya que la especificación de **WEP** no ha sufrido ningún cambio significativo) utilizando las extensiones provistas en el mismo para soportar **WEP**. Este tipo de problemas se ha analizado parcialmente en el protocolo **IEEE 802.11** ([7]), y adicionalmente, en ([14]) y ([38]) se estudian posibles canales encubiertos utilizando las extensiones provistas para **WEP**. Sin embargo, los canales encubiertos propuestos resultan ser triviales de detectar ya que no cumplen con la especificación del protocolo.

El objetivo de implementar un canal encubierto de red que resulte indistinguible, es que permite demostrar que **WEP** no sólo no cumple su objetivo primario (proveer confidencialidad equivalente a la provista por una red cableada), ya demostrado por numerosos estudios, sino que además agrega nuevos problemas de seguridad. Esto aumenta el riesgo de utilizar **WEP**, ya que un atacante podría no sólo obtener acceso a una red protegida por **WEP** (mediante los ataques conocidos) sino que utilizar el canal encubierto para enviar información sin ser detectado. Al no registrarse actividades sospechosas, el atacante podría mantener el control del sistema por más tiempo.

El presente trabajo se organiza de la siguiente manera: en el capítulo siguiente (Capítulo 2) se introducen los canales encubiertos así como al protocolo **IEEE 802.11**, incluyendo **WEP**. En el tercer capítulo (Capítulo 3) se presentan los trabajos similares en este área ([7], [14] y [38]) y el estado actual de la tecnología inalámbrica basada en el estándar **IEEE 802.11**. El cuarto capítulo (Capítulo 4) introduce la problemática que motiva el desarrollo del canal encubierto, presenta una descripción del canal encubierto propuesto y se describe las implementaciones realizadas del canal encubierto propuesto (tanto mediante la modificación a un controlador de red para plataformas **Linux** y la inyección de tramas **IEEE 802.11** en el aire). El quinto capítulo (Capítulo 5) analiza los resultados obtenidos, los posibles escenarios de uso del canal encubierto y compara, principalmente desde un punto de vista de posibilidad de detección, el canal encubierto propuesto con los trabajos relacionados así como las distintas implementaciones desarrolladas. Por último, el sexto capítulo (Capítulo 6) finaliza el presente trabajo, con las conclusiones obtenidas, y reflexionando sobre posibles problemas o áreas interesantes que quedan por explorar. Adicionalmente, se cierra con un análisis de su posible implementación de un canal encubierto similar en redes inalámbricas protegidas por los estándares **WPA** y **WPA2**.

## Capítulo 2

---

# Conceptos previos

---

La presente sección del trabajo describe los conceptos básicos de canales encubiertos (Sección 2.1), el protocolo **IEEE 802.11** (Sección chapter2:ieeeDot11) y el estándar **WEP** (Sección 2.2.2), necesarios para comprender el resto del trabajo.

### 2.1. Canales encubiertos

La seguridad informática está compuesta por tres pilares: la confidencialidad, la integridad y la disponibilidad ([5]). La confidencialidad, el aspecto que nos interesa, se puede definir como “la ocultación de información o recursos” ([5]). Habitualmente este objetivo de la seguridad informática se logra a través del uso de criptografía. Sin embargo, existen otras técnicas para ocultar información como la esteganografía ([28]) o los canales encubiertos.

Como se mencionó previamente, los canales encubiertos fueron definidos en la era digital como “aquellos no intencionados para la transferencia de información” por B. W. Lapmson ([25]) y posteriormente definidos como “cualquier canal de comunicación que puede ser explotado por un proceso para transferir información en violación de la política de seguridad del sistema” ([27]), por el Departamento de Defensa de Estados Unidos.

Los canales encubiertos utilizan recursos compartidos como vía de comunicación ([5]). Los recursos compartidos en un sistema informático son el espacio o el tiempo; esto nos lleva a las siguientes dos clases de canales encubiertos ([5]):

- almacenamiento: utilizan un atributo de un recurso compartido.
- temporales: utilizan una relación temporal o de orden respecto al uso de un recurso compartido.

En lo que respecta a canales encubiertos en redes de comunicaciones, la definición genérica anterior puede ser especializada a “una manipulación de un protocolo de comunicación para transferir información de forma no prevista por la especificación del protocolo” ([35]). Los mismos pueden ser o bien de almacenamiento, utilizando el contenido de las tramas o paquetes del protocolo en sí, para codificar la información del canal, o bien temporales, utilizando el tiempo de envío o el orden de envío de las tramas o paquetes para codificar la información del canal.

El otro aspecto que permite clasificar a los canales encubiertos, ortogonal al anterior, distingue entre un canal al cuál únicamente el emisor y receptor tienen acceso (utilizando un recurso compartido únicamente por ellos) y otro en el que terceras partes también tienen acceso (utilizando un recurso compartido por ellos y por al menos un tercero que no participa del canal). Los primeros se llaman canales encubiertos “sin ruido” mientras que los segundos se llaman canales encubiertos “ruidosos” ([5]).

Por último, cabe destacar que una de las propiedades clave de un canal encubierto es su ancho de banda y es uno de los parámetros más críticos utilizados a la hora de comparar canales encubiertos. Otros criterios de evaluación son:

- Probabilidad de detección
- Grado de anonimidad
- Facilidad de implementación
- Alcance (en el caso particular de canales encubiertos de red)

## 2.2. Protocolo IEEE 802.11

El protocolo de redes inalámbricas más utilizado actualmente es el protocolo **802.11** estandarizado por **IEEE** ([18]). El estándar incluye la capa física (capa uno del modelo **OSI**), dado que es un protocolo inalámbrico esto se traduce en las técnicas de modulación de ondas de radio a emplear, y la capa de *Medium Access Control* (capa de control de acceso al medio) que es parte de la capa de enlace de datos (capa dos) del modelo **OSI**.

El estándar ha atravesado numerosos cambios a lo largo de los años, mayormente relacionados con la capa física. La siguiente tabla resume los cambios incluidos en las distintas enmiendas que modifican la capa física del protocolo.

Las restantes enmiendas que modifican la capa física hacen referencias a cuestiones específicas para ciertas regiones, por mayor información se puede consultar la página oficial del protocolo en <http://www.ieee802.org/11/>.

---

<sup>1</sup>Frequency Hopping Spread Spectrum

<sup>2</sup>Direct Sequence Spread Spectrum

<sup>3</sup>Orthogonal Frequency-division Multiplexing

<sup>4</sup>Multiple-input and Multiple-output

Enmienda	Fecha de publicación	Frecuencia de operación	Tasa de transmisión de datos	Técnica de modulación empleada	Alcance aproximado
802.11	1997	2,4Ghz	2 Mbit/s	FHSS <sup>I</sup> o DSSS <sup>II</sup>	100 metros
802.11a	1999	5,0Ghz	54 Mbit/s	OFDM <sup>III</sup>	120 metros
802.11b	1999	2,4Ghz	11 Mbit/s	DSSS <sup>II</sup>	140 metros
802.11g	2003	2,4Ghz	54 Mbit/s	OFDM <sup>III</sup>	140 metros
802.11n	2009	2,4Ghz y 5,0Ghz	100+ Mbit/s	OFDM <sup>III</sup> con MiMo <sup>IV</sup>	250 metros

Cuadro 2.1: Enmiendas de IEEE 802.11 relacionadas con la capa física

En lo que respecta a la capa control de acceso al medio, actualmente hay únicamente dos modificaciones, a saber:

- 802.11e: Agrega **Quality of Service (QoS)**, calidad de servicio) al protocolo.
- 802.11i: Agrega nuevos mecanismos de seguridad al protocolo.

La capa física del protocolo no será descrita en detalle en el presente trabajo. Sin embargo, la siguiente subsección describe los componentes de la arquitectura de una red inalámbrica **IEEE 802.11**, introduciendo la terminología necesaria. Posteriormente, se describe en detalle la capa *Control de Acceso al Medio* del protocolo.

### 2.2.1. Componentes de la arquitectura

Las redes **IEEE 802.11** pueden operar en dos modos diferentes:

- Modo ad hoc: en este modo los clientes se comunican en forma directa.
- Modo Infraestructura: en este modo los clientes se comunican a través de una estación que ha sido asignada este papel.

Las redes *ad hoc* constituyen el tipo de red más básico y pueden estar formadas por tan sólo dos estaciones. En este modo de operación las estaciones se comunican directamente. Generalmente se utiliza este modo por un tiempo breve, con algún fin específico y sin planificación previa. Por ejemplo, dos personas que se encuentran en un aeropuerto y desean intercambiar un archivo.

Las redes que operan en modo infraestructura se distinguen de las anteriores ya que cuentan con un *access point* a través del cual se comunican. Es decir, no existe comunicación directa entre estaciones en este modo de operación. Por lo tanto, un *access point* se puede definir como una estación que provee la funcionalidad de



Nombre del campo	Significado
Frame Control (control de trama)	Posee indicadores de control, como por ejemplo el tipo y el código del mensaje.
Duration/ID (duración/ID)	Puede actuar como un identificador o indicar la duración (tiempo de uso del medio) del mensaje, dependiendo de la clase del mensaje.
Address 1 (Dirección 1)	Dirección <b>MAC</b> .
Address 2 (Dirección 2)	Dirección <b>MAC</b> .
Address 3 (Dirección 3)	Dirección <b>MAC</b> .
Sequence Control (Control de secuencia)	Indica el número de secuencia y el número del fragmento de la trama.
Address 4 (Dirección 4)	Dirección <b>MAC</b> .
QoS Control (Control de QoS)	Identifica a que categoría de tráfico pertenece la trama e información específica de <b>QoS</b> de la trama.
Frame Body (Cuerpo de la trama)	Los datos transportados por la trama.
FCS	Control de integridad de la trama (mediante un <i>CRC-32</i> <sup>1</sup> del encabezado <b>MAC</b> y el cuerpo del mensaje).

Cuadro 2.2: Significado de los campos estándares de una trama

- RA (Receiving station Address): dirección de la estación que recibió la trama, cuya respuesta es la trama actual.

Cabe aclarar que toda trama debe tener al menos una dirección, sin embargo la presencia de las tres restantes dependerá del tipo del mensaje.

### Campo de control de trama (Frame Control)

Como se mencionó anteriormente, el campo de control de trama (*Frame Control*) posee indicadores de control. La siguiente figura detalla los subcampos del campo de control de trama.

Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgt	More Data	Protected Frame	Order
------------------	------	---------	-------	---------	-----------	-------	---------	-----------	-----------------	-------

Figura 2.2: Formato del campo de control de tramas (*Frame Control*)

Los subcampos del campo de control de trama se encuentran descritos en el cuadro 2.3.

Nombre del campo	Significado
Protocol Version (versión del protocolo)	Indica la versión del protocolo, actualmente es cero.
Type (tipo)	Indica el tipo del mensaje.
Subtype (subtipo)	Indica el código del mensaje al ser combinado con el tipo.
To DS (al centro de distribución)	Ver Cuadro 2.4
From DS (del centro de distribución)	Ver Cuadro 2.4
More Flag (indicador de “más”)	Indica la presencia de más fragmentos, que componen la misma trama lógica.
Retry (reintento)	Indica que se está reenviando la trama.
PWR MGT (Manejo de energía)	Indica si la estación se encontrará en modo de ahorro de energía una vez completado el intercambio actual de tramas.
More Data (más datos)	Indica a una estación operando en modo de ahorro de energía que el <i>access point</i> tiene almacenadas tramas destinadas a la misma.
Protected Frame (trama protegida)	Indica que la trama actual se encuentra protegida utilizando el sistema de cifrado <b>WEP</b> o su sucesor <b>WPA/WPA2</b> .
Order (orden):	Utilizado cuando se transmiten tramas operando bajo el modo <i>StrictlyOrdered</i> (“orden estricto”).

Cuadro 2.3: Significado de los subcampos del campo de control de trama

El cuadro siguiente (2.4) explica el significado de los campos *To-DS* and *From-DS* dependiendo de sus valores.

Los distintos mensajes soportados por el protocolo se encuentran descritos en el cuadro A.1 del Apéndice A.

Valor <i>To-DS</i>	Valor <i>From-DS</i>	Significado
0	0	Una trama de datos enviada directamente de una estación a otra en una red <i>ad hoc</i> , o una trama de datos enviada de una estación a un <i>access point</i> (o viceversa) en una red en modo infraestructura, o cualquier trama de control o gestión.
1	0	Una trama enviada, a través del <i>access point</i> , por una estación destinada a otra estación perteneciente a la red.
0	1	Una trama remitida por un <i>access point</i> a una estación de la red.
1	1	Una trama que utiliza las cuatro direcciones disponibles. Actualmente dicha combinación no es utilizada por el estándar.

Cuadro 2.4: Significado de los de los campos *To-DS* and *From-DS*

### WEP - Wired equivalent privacy

Hasta la aprobación de la enmienda **802.11i** el protocolo contaba con un único mecanismo de confidencialidad llamado **WEP**, introducido en la versión original del mismo. La implementación de **WEP** es opcional según el estándar.

La versión original del estándar describe a **WEP** como una forma de proveer un nivel de seguridad “equivalente” para una red inalámbrica al de una red cableada (inherentes a la misma debido a sus atributos físicos). Principalmente, el estándar hace referencia a la posibilidad de un atacante de “escuchar” el tráfico de la red inalámbrica, dado que su medio físico de propagación es compartido y público (a diferencia de lo que sucede en una red cableada).

El estándar original define una clave de cifrado de 40 bits. A pesar de ello, y como paliativo a los primeros fallos de seguridad detectados para **WEP**, muchas implementaciones soportan una clave de 104 bits. La versión actual del estándar menciona este hecho, estandarizando ambas versiones bajo los nombres **WEP-40** y **WEP-104** respectivamente.

La siguiente figura (2.3) detalla los campos de una trama cifrada con **WEP**.

El campo *IV* contiene el vector de inicialización utilizado en el cifrado de la trama (este valor permite al emisor y receptor sincronizar el algoritmo de cifrado), un identificador de la clave utilizada para cifrar (los dispositivos que soportan **WEP** pueden tener configurado hasta cuatro claves diferentes de cifrado), y un relleno. El campo *ICV* es un *CRC-32* de los datos de la trama previo al cifrado. Cabe resaltar que al transmitir la trama los campos de datos e *ICV* se encuentran cifrados.

El procedimiento de cifrado de **WEP**, esquematizado en la Figura 2.4, consta de las siguientes etapas:

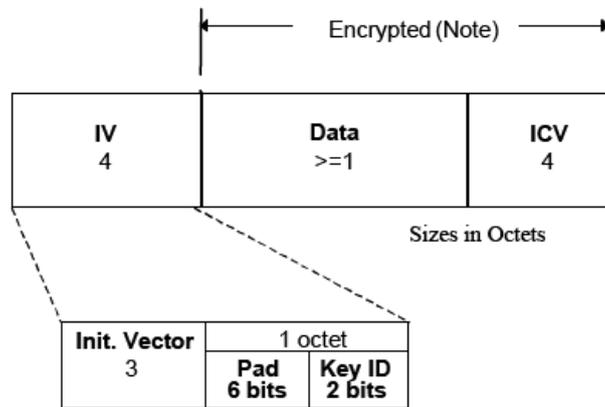


Figura 2.3: Formato de una trama WEP

1. Construcción de la semilla pseudoaleatoria: se concatena el vector de inicialización de 24 bits (para el cual no se especifica un algoritmo de generación) con la clave en uso (de 40 o 104 bits).
2. Calculo del ICV: se debe calcular el CRC-32 del los datos de la trama (en texto claro).
3. Calculo del flujo pseudoaleatorio de cifrado (*keystream*): Se utiliza el algoritmo **ARC4** ([24]) para calcular el flujo pseudoaleatorio de cifrado basado en la semilla pseudoaleatoria construida anteriormente, y con tamaño igual a la longitud de los datos a cifrar.
4. Cifrado: Se aplica la operación binaria **XOR** entre el texto claro y la cadena de cifrado (*keystream*).
5. Paso final: se concatena el *IV* con el resultado del cifrado.

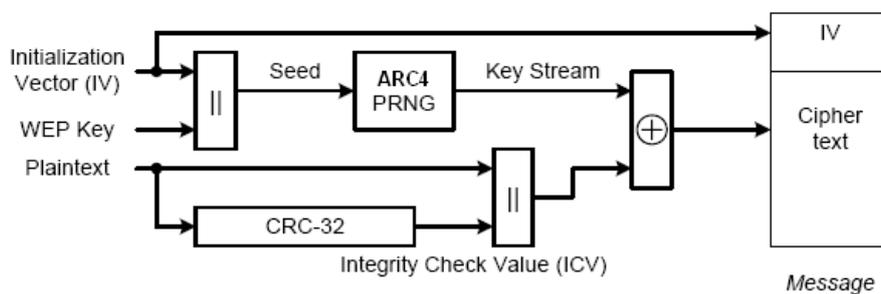


Figura 2.4: Procedimiento de cifrado de WEP

## 2.3. Ataques a WEP

Los ataques al protocolo **WEP** pueden dividirse en dos clases; aquellos que permiten recuperar la clave de cifrado utilizada y aquellos que permiten recuperar parte del flujo pseudoaleatorio utilizado para cifrar (*keystream*), asociado a un vector de inicialización en particular.

La primer clase de ataques se basan en análisis estadísticos de propiedades del algoritmo de cifrado **ARC4** ([37], [10], [41]), mientras que la segunda clase de ataque se basan en fallas en la interacción del protocolo **WEP** y diferentes funcionalidades del protocolo **IEEE 802.11**:

- Con la reutilización de vectores de inicialización ([43]).
- Con la autenticación provista por el mismo ([3]).
- Con la fragmentación ([6]).
- Con el control de integridad ([42]).

La primera clase de ataque vulnera completamente la seguridad brindada por **WEP**, mientras que la segunda permite enviar información, pero leer únicamente una pequeña parte del tráfico (la asociada a los vectores de inicialización para los cuales el *keystream* fue recuperado).

Como se mencionó en la descripción del protocolo **WEP**, el mismo no especifica el algoritmo de generación de vectores de inicialización ni impone restricciones sobre su uso. Por lo tanto, es válido utilizar un único vector de inicialización en todas las tramas enviadas por una estación. Esto aumenta considerablemente la utilidad de los ataques que permiten recuperar parte del flujo pseudoaleatorio utilizado para cifrar, ya que, reutilizando el mismo *keystream* e *IV*, se puede enviar tramas **WEP** válidas que serán interpretadas por el resto de las estaciones pertenecientes a la red. Sin embargo, dado que sólo se pueden descifrar las tramas que utilizan los *IV* para los cuales el ataque recuperó el *keystream*, esta clase de ataques se ve severamente limitada en este aspecto. Considerando que los *IV* son de 24 bits, resulta demasiado costoso efectuar el ataque por cada *IV* (sobre todo considerando que para todos los ataques de esta clase el atacante no puede elegir el *IV*, sino que se usa uno perteneciente a una trama válida enviada por alguna estación de la red). Sin embargo, de poder efectuarlo se podría leer todo el tráfico de la red, obteniendo el mismo nivel de acceso que con los ataques que recuperan la clave de cifrado.

Los ataques criptoanalíticos a **WEP**, dada su naturaleza estadística, requieren desde entre cuatro y cinco millones de tramas ([37]) en sus primeras versiones, luego mejorado a entre quinientos mil y un millón de tramas ([10]), y actualmente entre cuarenta y ochenta mil tramas ([41]). Sin embargo gracias a otra deficiencia que presenta el protocolo **WEP**, que es que no posee protecciones contra ataques de *replay* (reenvío una trama), es decir una trama es válida independientemente de

cuando o cuanto se la transmite, el tiempo de recolección de tramas para llevar a cabo los ataques puede ser disminuido mediante la inyección de una trama válida que genere una respuesta. Dado que la respuesta generada puede ser usada como parte de la muestra, el ataque se ve acelerado. Para aumentar las posibilidades de que la trama seleccionada genere una respuesta conviene utilizar una trama que transmite un paquete de una capa superior que no posee estado, como podría ser un paquete **UDP** o **ARP**. Las herramientas que implementan los ataques anteriores comúnmente utilizan una trama **ARP**, ya que es altamente probable que la misma genere una respuesta y puede ser distinguida (cabe recordar que las tramas están cifradas) por su tamaño ([39]).

## Capítulo 3

---

# Estado del arte y tecnología

---

La presente sección del trabajo describe el estado actual de la tecnología inalámbrica **IEEE 802.11** (Sección 3.1). Además, se describe los trabajos relacionados; una implementación previa de un canal encubierto sobre el protocolo **IEEE 802.11** (Sección 3.2).

### 3.1. Estado de la tecnología inalámbrica IEEE 802.11

La presente sección del informe trata sobre diferentes aspectos técnicos de las placas de red inalámbricas que soportan el protocolo **IEEE 802.11**. También se presenta una biblioteca de software que se utilizó para el desarrollo del presente trabajo.

#### 3.1.1. Placas de red

Las placas de red inalámbricas que soportan el protocolo **IEEE 802.11** se comercializan en diferentes versiones, las más comunes son:

- Placas **b** que soportan la enmienda **b** del estándar.
- Placas **b/g** que soportan las enmiendas **b** y **g** del estándar.
- Placas **a** que soportan dicha enmienda al estándar.
- Placas **a/b/g** que soportan las enmiendas **a**, **b** y **g** del estándar.
- Placas **a/b/g/n** que soportan las enmiendas **a**, **b**, **g** y **n** del estándar.

Las placas que soportan la modulación *FHSS* del estándar original (es decir, sin ninguna enmienda) son poco comunes, dada su baja velocidad (menor a 2 Mbit/s)

y la pronta aparición de la enmienda **b** que utilizaba *DSSS* (la otra modulación soportada por el estándar original) y obtenía velocidades similares a las de las redes cableadas del momento.

El otro factor importante a tener en cuenta respecto a las placas de red inalámbricas es la existencia de controladores de red (generalmente para el sistema operativo **Linux**) que soportan utilizar la placa en modo “monitor” y/o *access point*.

Las redes **IEEE 802.11** que operan en modo infraestructura funcionan como una red “switchheada”, dado que las estaciones no se comunican directamente sino a través de uno o más *access points*. Por lo tanto, al configurar la placa en modo promiscuo<sup>1</sup> se verán únicamente las tramas destinadas a la máquina y las tramas *broadcast*<sup>11</sup>; de forma análoga a lo que sucede en una red cableada que emplea *switches* en lugar de *hubs*. Cuando una placa se configura en modo “monitor”, procesa todas las tramas que están en el aire independientemente de su dirección destino, de forma análoga a lo que sucede en una red cableada con *hubs* cuando se configura la placa de red en modo promiscuo.

Resumiendo, las placas de redes inalámbricas cuentan con varios modos de operación:

- Cliente: la placa actúa en modo cliente.
- Promiscuo: la placa procesa todo el tráfico *unicast* con destino igual a su dirección **MAC**, o con dirección destino *multicast* o *broadcast*.
- Monitor: la placa procesa todas las tramas en el aire independientemente de su dirección destino y red inalámbrica a la que corresponde.
- Access Point: la placa actúa como un *access point* en lugar de un cliente.

Los modos de operación difieren de las placas de red cableada, ya que en las redes cableadas no hay un equivalente al modo *access point* (son todos clientes), no coexisten varias redes en un medio compartido (esto modifica sutilmente la definición de *broadcast*), y la topología no está dada por el medio (en un red cableada si le llegan físicamente o no las tramas a todos los miembros de la red es una propiedad de la topología y no intrínseca al medio utilizado). El hecho de que el medio es compartido en las redes inalámbricas y no exista posibilidad física de “revertirlo” (por ejemplo, mediante un dispositivo que hiciese lo que hace un *switch*), hace que existan los dos niveles de “sniffing” mencionados anteriormente: promiscuo y monitor.

---

<sup>1</sup>El modo promiscuo configura a una placa de red para que procese todas las tramas y no sólo las tramas *unicast* con destino igual a su dirección **MAC**, o con dirección destino *multicast* o *broadcast*.

<sup>11</sup>Dado que el aire es un medio compartido las tramas *broadcast* a ser procesadas son las que además pertenecen a la red inalámbrica a la que está asociada la estación, indicado mediante el **BSSID** de las tramas.

### 3.1.2. Herramientas de desarrollo

A la hora de desarrollar herramientas que trabajen con el protocolo **IEEE 802.11** el sistema operativo **Linux** se presenta como la alternativa más interesante, por varias razones:

- Existen controladores de código abierto.
- Existen diversos controladores que permiten configurar diferentes placas en modo “monitor”.<sup>1</sup>
- Existen diversos controladores que permiten “inyectar” tramas arbitrarias al aire.<sup>1</sup>

En **Linux** existe una biblioteca de programación llamada *Scapy* ([4]) diseñada para manipular tramas y paquetes de red. La misma permite armar paquetes arbitrarios e interpretar capturas de paquetes. Se encuentra programada en **Python**, por lo cuál es fácil su uso y extensión. Para desarrollar partes del trabajo se utilizó esta herramienta ya que permite un rápido desarrollo de pruebas de concepto al ser utilizada desde un lenguaje *interpretado*.

## 3.2. Trabajos previos relacionados

Los canales encubiertos de red han sido estudiados previamente ([16]). El trabajo ([16]) describe posibles canales encubiertos en protocolos de cada capa del modelo **OSI**; comenzando por la capa física describe un canal encubierto mediante la manipulación del algoritmo **CSMA/CD**, pasando por un canal encubierto a nivel de red que utiliza el campo *type-of-service* del protocolo **IP**, hasta el uso de esteganografía para ocultar datos en la capa de presentación. Los canales encubiertos propuestos varían en su facilidad de detección y dificultad de implementación, pero el objetivo de los autores era demostrar que es viable la implementación de canales encubiertos de red en todas las capas de modelos **OSI**.

Aplicado al protocolo **IEEE 802.11** solamente hemos encontrado una implementación previa ([7]), para la cuál disponemos de código fuente, que será descrita a continuación.

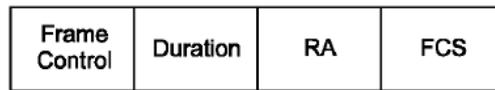
La implementación citada utiliza las tramas *ACK*; trama de control utilizada para confirmar la recepción de una trama. La siguiente figura muestra el formato de una trama *ACK*.

El significado de cada uno de los campos fue detallado en el apartado referente al protocolo **IEEE 802.11**, para mayor información consultar la Sección 2.2.

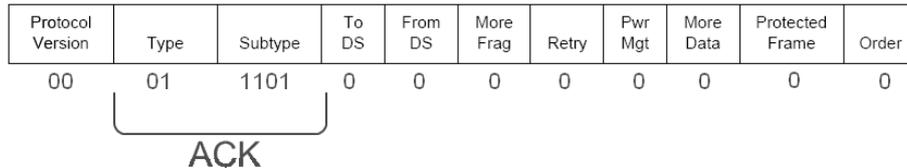
El autor ha desarrollado dos versiones del canal encubierto. Ambas implementan el canal encubierto codificando la información que se desea mandar en el campo

---

<sup>1</sup>Para el sistema operativo Microsoft Windows existe únicamente una placa que permite ser configurada en modo monitor e inyectar tramas, ver [40].

Figura 3.1: Formato de tramas *ACK*

*RA* y construyen una trama *ACK* con los valores indicados en la Figura 3.2, que simplemente indica que la trama es una trama de tipo *ACK*:

Figura 3.2: Formato de tramas *ACK* del canal encubierto [7]

La primera implementación ([9]) simplemente construye el campo *RA* de la siguiente forma:

- Los primeros dos bytes indican que es una trama perteneciente a la comunicación, por defecto su valor es 26214 (*0x6666*, en hexadecimal), aunque éste puede ser modificado por el usuario.
- El tercer byte es el dato que se desea enviar a través del canal encubierto (no puede ser cero, *0x00*, en hexadecimal).
- El resto de los bytes (tres, dado que es una dirección de red) tienen valor uno (*0x01*, en hexadecimal).

No utiliza un mensaje para indicar el inicio de la comunicación mediante el canal encubierto, sin embargo sí utiliza un mensaje para indicar el fin. El mensaje utilizado con dicho propósito es idéntico excepto que el valor del byte de datos es cero (*0x00*, en hexadecimal).

La segunda implementación ([8]), que cambia la forma de señalización, construye el campo *RA* de la siguiente forma:

- Si es la trama que indica el comienzo de la comunicación:
  - Los primeros dos bytes tienen el valor que indica el inicio de comunicación, su valor es 258 (*0x0102*, en hexadecimal).
  - Los siguientes dos bytes indican la longitud de los datos a ser enviados a través del canal.
  - Los últimos dos bytes valen cero (*0x0000*, en hexadecimal).

- Si es una trama de “datos”:
  - Los primeros dos bytes tienen el valor que indica que es una trama de datos, su valor es 772 (0x0304, en hexadecimal).
  - Los próximos cuatro bytes son de datos (completados con cero en caso de que los datos a ser enviados no sean múltiplos de cuatro).
- Si es la trama que indica el fin de la comunicación:
  - Los primeros dos bytes tienen el valor que indica el fin de la comunicación, su valor es 1286 (0x0506, en hexadecimal).
  - Los últimos cuatro bytes valen cero (0x00000000, en hexadecimal).<sup>III</sup>

Es claro que, si bien el autor eligió tramas **ACK**, es posible implementar la misma idea sobre otras tramas de control e incluso sobre tramas de gestión. Únicamente es necesario verificar previamente que la trama elegida no interferirá con el correcto funcionamiento de la red. Para ello basta con verificar que la trama será descartada por el resto de las estaciones de la red.

Adicionalmente, hemos encontrado otras dos propuestas de implementaciones de canales encubiertos sobre redes **IEEE 802.11**; a saber, [14] y [38].

En ([14]) se proponen dos canales encubiertos: uno que utiliza el vector de inicialización para el envío de datos, y el otro los números de secuencia de las tramas de datos. En el primer caso, sólo utilizan el último byte del número de secuencia (con el objetivo de dificultar la detección del canal encubierto, ya que según el estándar el mismo debe ser incrementado linealmente) y para enviar un mensaje primero envían un valor fijo (que indica el comienzo de la transmisión), luego envían la longitud del mensaje y posteriormente tantas tramas como sean necesarias con el mensaje. En el segundo caso, utilizan los tres bytes del vector de inicialización para enviar los datos y en los seis bits de “padding” presentes en el campo *Key ID* de una trama **WEP** un valor fijo que indica que la trama pertenece al canal encubierto. Para enviar un mensaje primero envían una trama que indica la longitud del mensaje y luego envían tantas tramas como sean necesarias con el mensaje.

En ([38]) también se propone utilizar el vector de inicialización para implementar un canal encubierto, pero en conjunto con las direcciones MAC (para lograr un mayor ancho de banda), sin embargo no se detalla cómo se utilizaría los mismos. También se propone enviar datos en tramas corruptas (cuyo CRC-32 falle), pero no se provee ningún detalle de cómo se construirían dichas tramas.

---

<sup>III</sup>El autor indica en el código fuente que estos bytes podrían utilizarse para implementar un **CRC-32** de los datos enviados, con el objetivo de verificar la integridad de los mismos.

## Capítulo 4

---

# Problemática y solución propuesta

---

La presente sección del trabajo introduce la motivación que dio origen al presente trabajo (Sección 4.1). Posteriormente se analiza los requisitos para que un canal encubierto sea “seguro” (Sección 4.2) y se analiza la problemática existente con el tipo de soluciones propuestas anteriormente (Sección 4.3). Luego, describe la solución alcanzada, el canal encubierto propuesto (Sección 4.4), y dos tipos de implementaciones del mismo; mediante la modificación de un controlador de red (Sección 4.5) y mediante la inyección de tramas (4.6).

El capítulo finaliza con un análisis de la factibilidad de implementación de la idea que enmarca el canal encubierto propuesto sobre redes inalámbricas protegidas por los estándares más modernos de seguridad: **WPA** y **WPA2** (Sección 4.7).

### 4.1. Motivación del trabajo

La relevancia de este trabajo surge del hecho que actualmente se sigue utilizando **WEP**, a pesar de los numerosos problemas que se detectaron en el mismo (Sección 2.3). Además, la posibilidad de implementar un canal encubierto utilizando las extensiones del protocolo provistas para soportar **WEP** aumenta el riesgo de utilizarlo, ya que un atacante podría no sólo obtener acceso a una red protegida por **WEP** (mediante los ataques conocidos) sino que utilizar el canal encubierto para enviar información sin ser detectado. Al no registrarse actividades sospechosas, el atacante podría mantener el control del sistema por más tiempo.

El protocolo **WEP** sigue siendo muy popular como lo demuestran diversos estudios; un estudio ([41]) conducido en Alemania en el año 2007 indica que más del 45 % de las redes todavía utilizan **WEP**, estudios similares enfocados a cadenas de

minoristas conducidos en Nueva York (en el 2008) indica que el 29 % de los locales lo utilizan ([21]) y en las ciudades Atlanta, Boston, Chicago, Los Ángeles, Nueva York, San Francisco, Londres y París (realizado a fines del 2007) indica que el 25 % de los locales lo utilizan ([20]). Estudios similares conducidos en la zona céntrica de la ciudad de Buenos Aires en los años 2005, 2006, 2007, 2008 y 2009 ([30], [31], [32], [33], [34]) indica que el porcentaje de redes que utilizan el método de cifrado mencionado ha aumentado junto con la cantidad de redes presentes, de un 27 % hasta el 33 % que se mantuvo hasta el año 2009, decendiendo ligeramente en el mismo. El siguiente gráfico extraído del informe ([34]) exhibe éste punto.

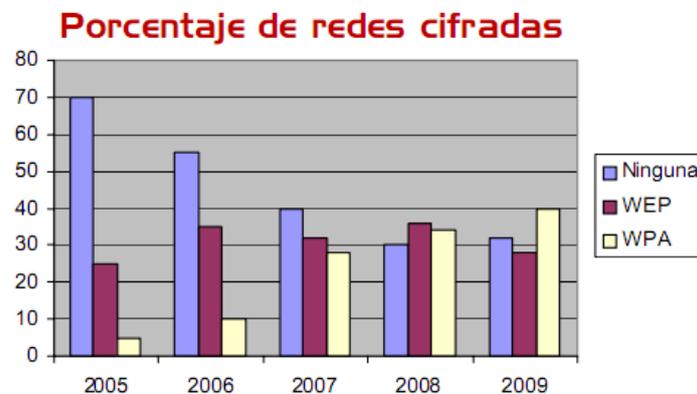


Figura 4.1: Porcentaje de Redes Criptadas en la Zona Céntrica de Buenos Aires

Dada la clara popularidad de **WEP** y los numerosos estudios enfocados en vulnerar la confidencialidad provista por el mismo a una red inalámbrica, decidimos estudiarlo desde otro punto de vista. El objetivo era demostrar que **WEP** no sólo no cumple su objetivo primario, proveer confidencialidad (ya demostrado por numerosos estudios), sino que además agrega problemas de seguridad a una red que supuestamente protege. Para lograr demostrar ésto construimos un canal encubierto, donde su uso es indistinguible del tráfico normal de la red, que existe en el protocolo **IEEE 802.11** debido a las extensiones del mismo provistas para soportar **WEP**. El canal encubierto funciona “dentro” de una comunicación entre dos estaciones que se están comunicando utilizando **WEP**, sin perjudicar su comunicación.

También nos interesó la posibilidad de estudiar canales encubiertos sobre redes inalámbricas ya que este tipo de red presenta una diferencia importante frente a redes cableadas para los mismos: dado que el aire es un medio compartido el tráfico de una red inalámbrica puede ser capturado desde la periferia de la instalación del canal encubierto, a diferencia de protocolos que se usen sobre redes cableadas donde se debe iniciar una conexión que va a ser susceptible a ser identificada y rastreada y que

deberá llevarse a cabo contra un nodo de la red controlado por el usuario del canal encubierto. Esta diferencia presenta una clara ventaja para un atacante que haya instalado el canal encubierto en un sistema comprometido ya que simplemente debe estar en la periferia del sistema comprometido para sustraer información del mismo. En cambio, si se utilizase un canal encubierto sobre un protocolo de red cableado debería realizar una conexión hacia otro sistema bajo su control, potencialmente identificándolo, de ser descubierto el canal ilegítimo.

## 4.2. Requisitos de un canal encubierto

A continuación se detallan los puntos que se definieron como los requisitos que el canal encubierto debía cumplir, con el objetivo de obtener un canal encubierto robusto y de difícil detección.

1. Respete el estándar del protocolo, lo que implica respetar:
  - El formato de las tramas (y sus reglas de construcción)
  - Las secuencias de intercambio de tramas permitidas
2. Tener un ancho de banda estrictamente mayor que cero (el canal debe permitir enviar información).
3. Una trama que pertenece al canal encubierto debe ser “indistinguible”<sup>I</sup> de una trama que no pertenece al mismo.<sup>II</sup>
4. La implementación del protocolo que contenga el canal encubierto debe ser interoperable con implementaciones que no lo contengan.
5. De comprometerse la existencia del canal encubierto, ésta no debe comprometer el contenido del mismo.

Si bien a primera vista no resulta evidente la necesidad de respetar el estándar del protocolo, basta con tener en cuenta que si el canal encubierto no respeta el mismo se presentará una anomalía que podría conducir a la detección del canal encubierto.

---

<sup>I</sup>Con indistinguible se hace referencia a que la trama se encuentre bien formada (su construcción respete el estándar) y que respete la secuencia de intercambios de tramas a la que pertenece.

<sup>II</sup>La idea detrás del requisito de “indistinguibilidad” proviene de un intento de formalizar el requisito de “difícil detección”, es decir debe ser “difícil” detectar la existencia del canal encubierto observando el tráfico generado.

### 4.3. Problemática de las soluciones existentes

A continuación se presenta un análisis de cualquier implementación que utilice mensajes de gestión y/o de control para enviar los datos del canal encubierto, como es el caso de la implementación previa de canales encubiertos sobre redes **IEEE 802.11** ([7]) de la cuál se dispone de código fuente. Del mismo, se deduce que las implementaciones que utilicen mensajes de gestión y/o de control no cumplen con los requisitos explicitados anteriormente, justificando la decisión de idear e implementar una solución que sí cumpla con los mismos.

En 4.2, se definieron ciertos puntos que un canal encubierto de red “ideal” debería cumplir, por ende la comparación con implementaciones (que utilicen mensajes de gestión y/o de control) se hará con respecto a los mismos. El primer punto enunciado indicaba que se debe respetar el estándar del protocolo, lo cual implicar respetar no sólo el formato de las tramas y sus reglas de construcción, sino que también las secuencias de intercambio de tramas permitidas por el estándar. Por ende, comenzaremos analizando las tramas que se podrían utilizar asumiendo que la estación que está participando del canal encubierto no es el *access point*<sup>I</sup> y que la red está en modo infraestructura. De las tramas de control y gestión existentes, una estación podría enviar las tramas (dependiendo del estado de conexión con una red<sup>II</sup>) descritas en el cuadro 4.1.

Tipo de Trama	Trama	Estado
Gestión	Pedido de asociación	Autenticada y no asociada
Gestión	Pedido de re-asociación	Autenticada y asociada
Gestión	Pedido “Probe”	Cualquiera
Gestión	Desasociación	Autenticada (asociada o no)
Gestión	Deautenticación	Autenticada (asociada o no)
Gestión	Autenticación	No autenticada ni asociada
Control	PS-Poll	Autenticada y asociada
Control	RTS	Cualquiera
Control	CTS	Cualquiera
Control	ACK	Cualquiera

Cuadro 4.1: Tramas de control y gestión que pueden ser enviadas por una estación

Si bien una estación asociada podría enviar cualquiera de las tramas anteriores, excepto un pedido de asociación o autenticación, dos de las tramas se descartan rápidamente como tramas posibles para implementar el canal encubierto puesto que

<sup>I</sup>Si lo fuese tendría algunas opciones adicionales, pero hemos optado por omitir este caso ya que no es el escenario más común y todo lo que se aplica a una estación si se aplica para el caso de un *access point*.

<sup>II</sup>Cabe recordar que una estación puede estar en tres estados diferentes: no autenticada ni asociada, autenticada pero no asociada, autenticada y asociada.

tendrán consecuencias no deseadas: las tramas de *Desasociación* y *Deautenticación* desconectarán a la estación de la red. Del resto de las tramas posibles, el cuadro 4.2 describe la situación en la cuál la estación podría enviar la trama respetando los intercambios de tramas válidos, además indica quién debe comenzar el intercambio de tramas.

Trama	Situación	Quién inicia intercambio de tramas
Pedido de re-asociación	La estación debe haber perdido conectividad con el <i>access point</i> y desea asociarse nuevamente, pero valiéndose de la autenticación previa	Estación
Pedido sonda ( <i>Probe request</i> )	Las estaciones periódicamente emiten “probes” en busca de <i>access points</i> con mejor señal	Estación
PS-Poll	El <i>access point</i> debe indicar en el <i>Beacon</i> que tiene tramas pendientes para la estación y la misma debe estar en modo <i>power-save</i>	<i>Access point</i>
RTS	La estación debe tener datos para enviar antes de mandar un RTS	Estación
CTS	La estación debe recibir una trama RTS para poder responder con un CTS	Cualquier miembro de la red
ACK	La estación debe recibir datos para poder enviar una trama ACK	Cualquier miembro de la red

Cuadro 4.2: Situaciones dónde se utilizan distintos tipos de tramas

El problema con considerar utilizar tramas que pertenezcan a intercambios válidos no iniciados por la estación, es la necesidad de esperar a que esta situación se presente para poder enviar datos del canal encubierto. Por ejemplo, para el caso de *PS-Poll* deberíamos esperar a que la estación entre en modo de ahorro de energía<sup>III</sup>. Es por este motivo que descartamos esta clase de tramas, con excepción de las tramas *ACK*, ya que en una red comúnmente todas las estaciones reciben datos periódicamente. Algo similar es de esperarse con las tramas *CTS*, si bien se supone que una estación debería recibir *RTS* periódicamente (cada vez que otra estación que se encuentre en

<sup>III</sup>También se podría forzar que la estación entre en modo de ahorro de energía, pero comúnmente una estación no entra y sale constantemente de este modo; resultando sospechoso.

el rango de recepción desee enviar datos), en la práctica este tipo de tramas no se utilizan.

Con respecto a tramas que pertenezcan a intercambios válidos iniciados por la estación, el pedido de re-asociación, si bien es factible su uso, será altamente sospechoso ya que una estación no envía este tipo de tramas frecuentemente. Como se mencionó anteriormente tampoco es habitual el uso de tramas *RTS*, por ende tampoco supone una buena elección.

Como conclusión, los siguientes tipos de tramas resultan viables para implementar el canal encubierto, cumpliendo con el requisito de respetar el intercambio de tramas especificado en el estándar del protocolo **IEEE 802.11**:

- Pedido sonda (*Probe request*)
- ACK

Como siguiente paso, se analiza las posibilidades para enviar información en este tipo de tramas. Para el caso de *ACK*, las reglas de construcción son:

- El valor del campo *RA* debe ser el valor del segundo campo de dirección de la trama cuya recepción se está confirmando.
- El valor del campo duración debe ser cero si el bit de "Más fragmentos" se encontraba en cero en la trama que se está confirmando, sino debe ser el valor que se encontraba en el campo duración de dicha trama menos el tiempo necesario para transmitir la trama *ACK*.
- Los valores de los campos del *Frame Control* deben ser:
  - Versión del protocolo: 00
  - Tipo: 01 (Control)
  - Subtipo: 1101 (ACK)
  - To-DS: 0
  - From-DS: 0
  - Más Fragmentos: 0
  - Re-intento: 0
  - Power Management: Depende del estado actual de *power-save* de la estación.
  - Más Datos: 0
  - WEP: 0
  - Orden: 0
- El valor del campo *FCS* se debe calcular como un CRC-32 sobre todos los campos de la trama.

Claramente, todos los valores de una trama *ACK* se encuentran especificados, por ende no se puede implementar un canal encubierto utilizando este tipo de tramas sin violar las reglas de construcción del estándar. Cabe resaltar, que si bien se descartaron las tramas *RTS* y *CTS* por otros motivos, también se encuentra perfectamente establecido el valor de cada uno de sus campos; imposibilitando su uso para implementar un canal encubierto.

Como siguiente paso, se analiza las posibilidades para enviar información en este tipo de tramas. Para el caso de *Probe Request*, las reglas de construcción son:

- Los valores de los campos del *Frame Control* deben ser:
  - Versión del protocolo: 00
  - Tipo: 00 (Gestión)
  - Subtipo: 0100 (Probe request)
  - To-DS: 0
  - From-DS: 0
  - Más Fragmentos: 0
  - Re-intento: 0
  - Power Management: 0
  - Más Datos: 0
  - WEP: 0
  - Orden: 0
- El valor del campo *Duración* es cero a menos que se encuentre un período “libre de contención” en cuyo caso es 32768.
- El valor del campo *DA* puede o bien ser el destino broadcast (*FF:FF:FF:FF:FF:FF*) o la dirección MAC del *access point* que se está buscando.
- El valor del campo *SA* es la dirección de la estación que envía el pedido.
- El valor del campo *BSSID* puede o bien ser el destino broadcast (*FF:FF:FF:FF:FF:FF*) o la dirección MAC del *access point* que se está buscando.
- El valor del campo *Sequence Control* se encuentra especificado por:
  - Número de fragmento: 0000
  - Número de secuencia, que debe ser incrementado por cada trama enviada, por ende será uno más que el valor de la última trama enviada.
- El valor del campo *Frame body* (cuerpo del mensaje) se encuentra especificado por:

- *SSID*: Si no hay valor indica que es el *SSID* broadcast (ante el cuál deben responder todos los *access points*), sino puede ser un *SSID* específico de la red que se está buscando.
  - *Supported Rates*: Indica las velocidades de transmisión soportadas por la estación.
- El valor del campo *FCS* se debe calcular como un CRC-32 sobre todos los campos de la trama.

Cabe resaltar que los campos *DA*, *BSSID* y *SSID* deben ser consistentes: o bien todos son broadcast o, si se busca un *SSID* específico, las direcciones son o bien broadcast o una dirección específica. Por ende, no es posible ni que ambas direcciones sean distintas ni que se busque un *SSID* broadcast pero utilizando una dirección específica para las campos *DA* y *BSSID*.

Dada la especificación de las tramas *Probe Request*, se pueden utilizar el campo *SSID* o los campos *DA* y *BSSID* (haciendo peticiones por un *SSID* particular, para respetar la restricción mencionada anteriormente). Utilizar el campo *SSID* posee la desventaja que comúnmente se utiliza un texto descriptivo (nombre del dueño, empresa, etc.), como se puede apreciar en [12] dónde se detallan los mil *SSID* más comunes. Por ende el uso de este campo para transmitir datos alertará rápidamente a cualquier administrador de red. Sin embargo, una opción interesante a tener en cuenta es que las estaciones *Windows XP* generan periódicamente peticiones por *SSID* aleatorios, ésto ha sido documentado en [11]. Si bien este comportamiento se modificó por cuestiones de seguridad, como se detalla en [26], las estaciones siguen generando periódicamente peticiones por *SSID* aleatorios pero con ciertas restricciones de seguridad. Emulando este comportamiento, pero utilizando el *SSID* aleatorio para enviar datos (al estar cifrados, por las propiedades de un algoritmo criptográfico robusto, los mismos deberían resultar aleatorios), se puede implementar un canal encubierto que, para quién esté analizando el tráfico, resulte indistinguible del tráfico generado por una estación con *Windows XP*.

En cuanto al uso de las direcciones MAC, el comportamiento de enviar sondas por una dirección particular es muy poco común y por ende resultará llamativo. Otro factor importante a tener en cuenta es que las estaciones envían sondas buscando redes que “conocen”, que no suelen ser más de una decena; por ende, enviar peticiones por muchas direcciones MAC distintas resultará inusual.

Por último, es importante resaltar que en las tramas *Probe request* se podría enviar datos en el campo “Last (Vendor Specific) Information Element”, pero dado que su uso no es habitual y su contenido no se encuentra especificado en el estándar, llamaría rápidamente la atención; potencialmente revelando la existencia del canal encubierto.

Hemos visto que las únicas posibilidades para implementar un canal encubierto de red utilizando mensajes de gestión y/o de control del protocolo **IEEE 802.11** que

cumpla con los requisitos expuestos en la Sección 4.2, es utilizar las tramas *ACK* o *Probe request*. Además, hemos demostrado que no es factible con ninguna de las dos tramas; únicamente existe la posibilidad de implementar un canal encubierto que cumpla con los requisitos si consideramos emular un comportamiento, no estándar, que poseen las estaciones *Windows XP* en relación al manejo de tramas *Probe request*. Sin embargo, debido a que hemos decidido concentrarnos en la existencia de canales encubiertos en la especificación del protocolo y no debido a rarezas de implementaciones particulares, no hemos perseguido esta posibilidad.

En ([14]) se proponen dos canal encubiertos: uno que utiliza el vector de inicialización para el envío de datos, y el otro los números de secuencia de las tramas de datos. El uso de los números de secuencia viola el requisito que indica respetar el estándar del protocolo ya que el número de secuencia debe ser incrementado linealmente. El uso del vector de inicialización propuesto es similar al implementado en la versión que inyecta datos, pero a diferencia de ésta, no respeta las secuencias de intercambios de tramas del estándar, utiliza el campo *Key ID* que según el estándar debe permanecer en cero, y tampoco cifra los datos enviados (aunque menciona que esta posibilidad existe).

Adicionalmente, en ([38]) también se propone utilizar el vector de inicialización para implementar un canal encubierto en conjunto con las direcciones MAC (para lograr un mayor ancho de banda); violando el requisito que indica respetar el estándar del protocolo, ya que se enviarán tramas a direcciones MAC que no se corresponden a estaciones asociadas a la red (para las cuales nunca se reciben tramas de confirmaciones). También se propone enviar datos en tramas corruptas (cuyo CRC-32 falle), cuestión que claramente viola el requisito que indica respetar el estándar del protocolo y facilitará la detección del canal encubierto ya que se generaran más fallas de CRC-32 que lo habitual.

Si se considera implementar un canal encubierto temporal, existen otras posibilidades; por ejemplo, alternar entre dos *SSID* para indicar cero o uno. Pero este análisis supera el alcance del presente trabajo.

#### 4.4. Canal encubierto propuesto

Para poder cumplir con los requisitos detallados en 4.2, se ideó un canal encubierto de red que hace “uso y abuso” de las extensiones provistas por el protocolo **IEEE 802.11** para soportar **WEP**, con el objetivo de utilizarlas para enviar información de una forma no prevista por el protocolo pero compatible con el mismo.

El estándar **WEP**, como se describió en el apartado 2.2.2, agrega los siguientes campos a una trama de datos **IEEE 802.11**:

- *IV*: Compuesto por el vector de inicialización (de 24 bits), 6 bits de relleno (que deben estar en cero según el estándar [18]) y 2 bits para indicar la clave de cifrado en uso.

- 
- *ICV* (Integrity Checksum Value): un *CRC-32* de los datos de la trama computado previo al cifrado.

La generación del vector de inicialización no se encuentra especificada en el estándar, por lo tanto cualquier esquema de generación es válido. Sin embargo, en la práctica se comienza con un valor o bien fijo (comúnmente cero), o un valor aleatorio o se retoma el último valor utilizado, y se continúa generando los valores subsiguientes de alguno de los siguientes modos:

- Se incremente el valor actual en modo *little-endian*.
- Se incremente el valor actual en modo *big-endian*.
- Se elige otro valor de forma aleatoria.

También se han observado implementaciones que alternan entre dos valores ([37]). Sin embargo, este tipo de implementaciones es muy poco común.

La idea del canal encubierto propuesto es utilizar el vector de inicialización para transmitir datos. Se eligió este campo ya que su comportamiento no se encuentra especificado, por lo tanto independientemente de los valores que tome este campo al ser utilizado para implementar el canal encubierto, cumplirá con el estándar; siendo inter-operable la implementación con el canal encubierto con otras implementaciones del protocolo.

La pregunta que queda pendiente es cómo utilizar el valor del vector de inicialización para enviar datos. La respuesta es simple; se codifica el valor que se desea enviar en el vector de inicialización y se construye la trama según el proceso explicado en 2.2.2, utilizando el valor resultante de la codificación como el vector de inicialización. La trama construida de esta forma resulta una trama **WEP** válida, que puede ser interpretada por el resto de las estaciones pertenecientes a la red.

Desde el punto de vista del receptor de los datos transmitidos a través del canal encubierto resulta que, debido a que el vector de inicialización se envía como parte del campo *IV* de la trama, el receptor puede identificar, debido a la codificación, que es una trama que contiene parte de los datos enviados a través del canal encubierto, y decodificar el valor codificado en el vector de inicialización; obteniendo el dato enviado a través del canal encubierto.

Por último, resta la descripción del proceso de codificación empleado. Dados los mecanismos de funcionamiento del canal encubierto, resulta necesario que el proceso de codificación le permita al receptor identificar que una trama es parte de la comunicación del canal encubierto. Además, uno de los requisitos impuestos es que de detectarse el canal encubierto no se comprometa los datos enviados a través del mismo. Para cumplir con este requisito se recurrió a la criptografía; los datos codificados en el vector de inicialización estarán cifrados. Por lo tanto, únicamente

resta determinar cómo el receptor puede identificar la trama, para ello se analizaron varias alternativas:

1. Enviar valores fijos de señalización en el vector de inicialización, por ejemplo un valor indica que las tramas subsiguientes serán parte del canal encubierto y otro valor que indique el fin de la secuencia de tramas (señalización de inicio y fin).
2. Un prefijo o sufijo en el vector de inicialización que indique que la trama pertenece al canal encubierto (señalización en banda).
3. Utilizar otro campo de la trama para indicar la pertenencia al canal encubierto.

La tercer opción se descarto rápidamente porque todos los campos del protocolo poseen sus valores especificados (excepto el campo *IV*), de forma tal que al utilizar uno de indicador se podrían presentar las siguientes situaciones:

- Falso positivo: una trama que no es parte del canal encubierto posee el valor utilizado como indicador debido a requisitos funcionales.
- Violación del estándar: se puede utilizar un campo de forma no prevista por el estándar, produciendo una incompatibilidad con otras implementaciones o simplemente una anomalía (que facilitará la detección del canal encubierto). Esto puede producirse por no respetar la sintaxis (construcción per se de la trama) o la semántica que asociada a un indicador.
- Necesidad de envío adicional de tramas: según qué valor se utilice para indicar la presencia del canal encubierto puede ser necesario enviar tramas adicionales para respetar la semántica asociada con el indicador<sup>1</sup>, agregando complejidad al canal encubierto y reduciendo el ancho de banda de la comunicación.

Las restantes dos alternativas no presentan los mismos problemas. En ambos casos las tramas utilizadas para la señalización son “indistinguibles” ya que se encuentran bien formadas, y respetan los intercambios de tramas estipulados por el protocolo. A modo de desempate se considero la facilidad de detectar el canal encubierto utilizando ambas alternativas.

Los siguientes gráficos muestran una secuencia de valores de vectores de inicialización simulando el canal encubierto y el comportamiento estándar de las tres clases de controladores de red mencionados anteriormente. Los gráficos se construyeron bajo los siguientes supuestos:

- Los datos enviados por el canal encubierto son aleatorios: dado que los mismos se encuentran cifrados, y considerando que el algoritmo criptográfico es de alta

---

<sup>1</sup>Por ejemplo, si se utiliza el indicador de *Retry* se debería respetar que la trama haya sido enviada previamente sin el indicador, obligando a que la trama se envíe dos veces.

calidad, los mismos deberían tener sesgos despreciables. De no ser así, habría un potencial problema en el algoritmo de cifrado.

- Se fijó el valor  $0xEEEEEE$  para indicar el inicio del canal encubierto y  $0xFFFF$  para indicar el fin de la comunicación. Esta decisión es completamente arbitraria, sin embargo las conclusiones obtenidas no dependen de los valores en sí, sino de su existencia.
- Al utilizar señalización en banda (con respecto al campo), es decir que se utiliza un byte del campo *IV* para indicar que los restantes bytes del campo pertenecen al canal encubierto, se fijó el byte indicador en  $0xFF$ . Nuevamente, esta elección es completamente arbitraria, sin embargo las conclusiones obtenidas no dependen del valor en sí, sino de su existencia.

Primero analizaremos el uso de señalización en banda para los tres casos de controladores de red. Cabe aclarar que para el caso del controlador que incrementa en modo *big-endian* se utilizó el primer byte del campo como indicador, dado que de esta forma se demoraría la mayor cantidad posible de tramas hasta que el incremento lineal del controlador colisione con un valor que podría pertenecer al canal encubierto. En el caso donde el controlador incrementa en modo *little-endian* se utilizó, por la misma razón, el valor del último byte.

Las Figuras 4.2, 4.3 y 4.4 muestran los resultados con señalización en banda para cada tipo de controlador.

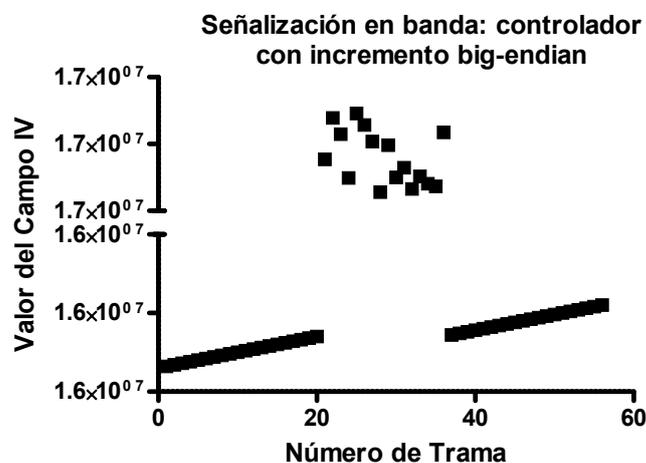


Figura 4.2: Señalización en banda: controlador con incremento big-endian

Como se puede apreciar en los gráficos, resulta trivial discernir entre dos flujos de tramas, aquellas que pertenecen al canal encubierto y las que pertenecen al tráfico normal del controlador (en los tres casos). Incluso, sabiendo que la mayoría de los

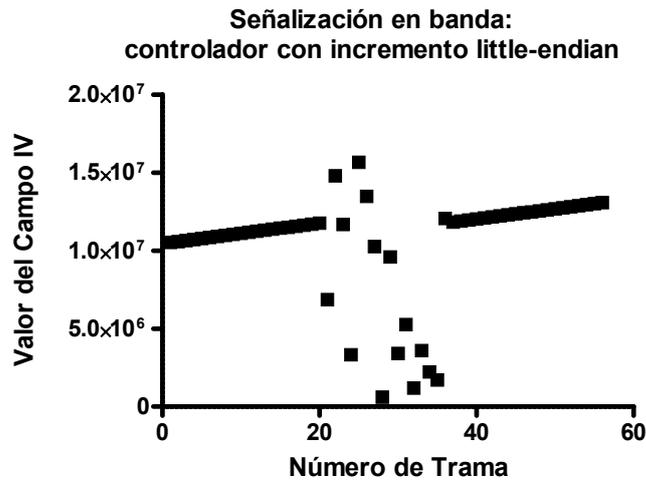


Figura 4.3: Señalización en banda: controlador con incremento little-endian

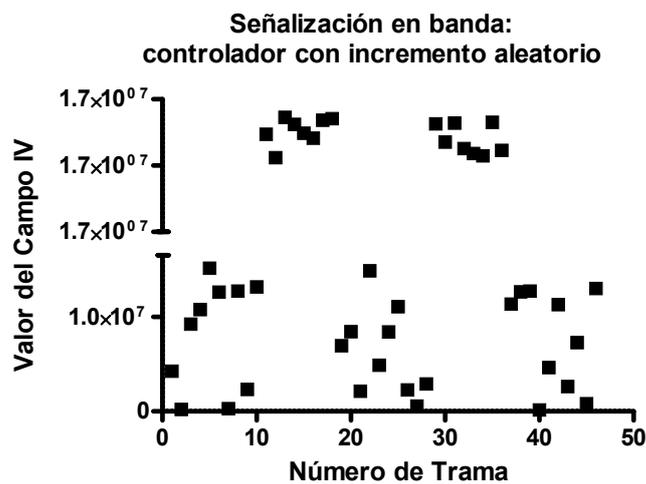


Figura 4.4: Señalización en banda: controlador con incremento aleatorio

controladores se comportan de alguna de las tres formas mencionadas anteriormente es posible deducir cuál de los flujos es cuál.

Las Figuras 4.5, 4.6 y 4.7 muestran los resultados de utilizar tramas que indican el comienzo y el fin para cada tipo de controlador.

Para los casos en que el controlador incrementa en forma lineal (ya sea *big-endian* o *little-endian*), nuevamente, es trivial advertir la presencia de dos flujos distintos. Al igual que para el caso de señalización en banda, si se sabe que mayoritariamente los controladores incrementan de forma lineal es posible identificar cual es el flujo que pertenece al canal encubierto y cual pertenece al tráfico normal producido por el

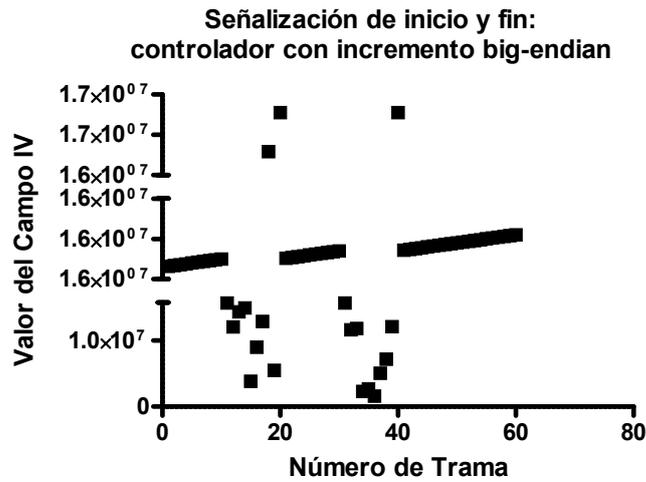


Figura 4.5: Señalización de inicio y fin: controlador con incremento big-endian

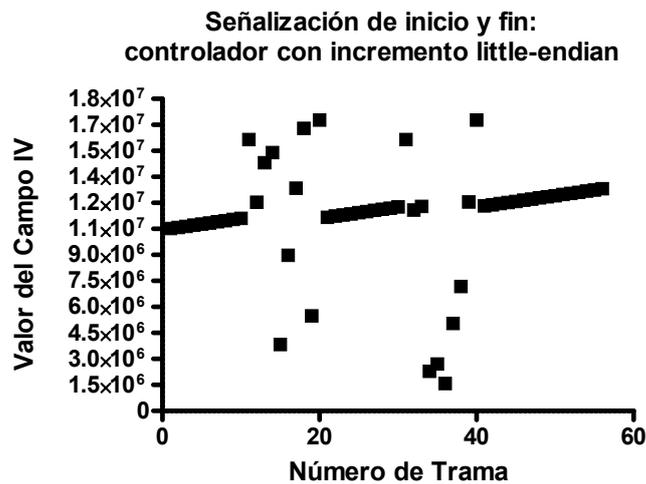


Figura 4.6: Señalización de inicio y fin: controlador con incremento little-endian

controlador.

En el caso de saltos aleatorios en la elección de los valores de  $IV$  no resulta trivial distinguir ambos flujos. Sin embargo, de continuar los flujos en el tiempo, sobre todo si ocurren intercalados, los valores de inicio y fin serán más frecuentes que el resto indicando una tendencia a éstos valores (de hecho, si se observa el gráfico con atención es posible distinguir que éstos son los únicos valores repetidos). Este comportamiento no indica claramente ni qué flujo es el canal encubierto (es imposible distinguir que valor indica comienzo y cuál indica fin) ni la existencia de un canal encubierto; podría ser un sesgo de la implementación del algoritmo de elección del

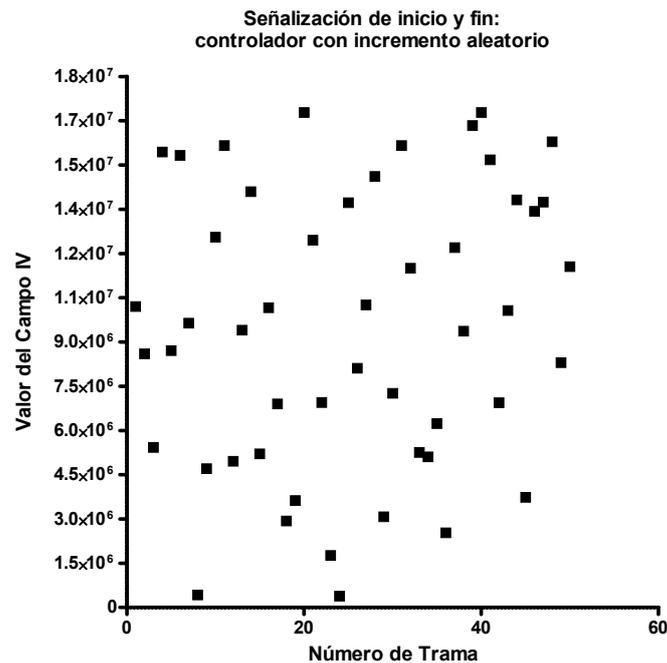


Figura 4.7: Señalización de inicio y fin: controlador con incremento aleatorio

vector de inicialización.

En el caso de que se hubiese empleado un único valor para señalar el inicio o el fin, lo mismo sucedería con la diferencia que el valor elegido resaltaría aun más (dado que tendría el doble de apariciones). Si, por el contrario, utilizamos un conjunto mayor de valores para señalar el inicio y/o fin la probabilidad de repetición se verá disminuida, resaltando menos los valores que actúan como indicador respecto al resto. Este razonamiento, llevado al infinito, conduce a que si fuesen equiprobables (es decir, todos los valores actúan como indicadores) no se podría distinguir la existencia de más de un flujo. Sin embargo, en este caso el receptor no tendría forma de distinguir que la trama pertenece al canal encubierto a primera vista, lo cuál resulta lógico: si el receptor tuviese como distinguirla el atacante debería (dada una muestra suficientemente grande) poder hacerlo. Por lo tanto, la forma de implementarlo sería separando un conjunto de valores para actuar como indicadores suficientemente grande para que resulte poco probable que se repitan y elegimos al azar dentro del conjunto cada vez que se necesita enviar un indicador.

Del razonamiento anterior, surge una forma adicional de señalización, que llamaremos *encapsulada*, la cual consiste en agregar una capa de control (encapsulado en el campo *IV*) mediante la cuál el receptor puede distinguir si la trama pertenece al canal encubierto. De esta forma ningún valor tiene un significado especial; es decir, que son todos equiprobables. Sin embargo, al descifrar un *IV*, de tener la estructura correcta (asegurada mediante un código de detección de errores, por ejemplo un

**CRC**) será considerada una trama de datos perteneciente al canal.

Cada uno de los métodos anteriores tiene sus ventajas y desventajas. Desde el punto de detectabilidad hay un claro ganador, sin embargo también resulta ser un método más complejo y costoso (el receptor debe descifrar todas las tramas y verificar su código de detección de errores).

La elección del método a utilizar depende de cómo se implemente el canal encubierto (modificación de un controlador ó inyección de tráfico), por lo tanto se explicara la elección del método de señalización para cada caso en su apartado correspondiente.

## 4.5. Implementación mediante la modificación de un controlador de red

La presente sección describe como se modificó la pila (“stack”) del protocolo **IEEE 802.11** incluido en el núcleo *Linux* desde la versión 2.6.22, para implementar el canal propuesto. La pila del protocolo **IEEE 802.11** llamada *mac80211*<sup>IV</sup> (incluida desde la versión 2.6.22) es utilizada por diferentes controladores, por ejemplo por el controlador incluido en el núcleo *Linux* desde la versión 2.6.25 para las placas **Zydas ZD1211**.

### 4.5.1. Método de señalización

Se decidió implementar dos controladores diferentes que utilicen dos mecanismos distintos de señalización (que conformaban dos ejemplares representativos de cada clase de señalización):

- Señalización en banda, sobre un controlador que incrementa el *IV* en modo *big-endian*.
- Señalización encapsulada, sobre un controlador que incrementa el *IV* de forma aleatoria.

Para el caso de señalización en banda se fijó utilizar el primer byte como indicador, utilizando el valor (*0xFF*) para indicar que la trama pertenece al canal encubierto. Se utilizó el primer byte para señalar ya que el controlador incrementa de forma *big-endian* dado que, como se explicó anteriormente, de esta forma se demoraría la mayor cantidad posible de tramas hasta que el incremento lineal del controlador colisione con un valor que podría pertenecer al canal encubierto.

Para el segundo tipo de señalización se definió el siguiente formato para el campo *IV*:

- Primer byte: valor aleatorio utilizado como vector de inicialización para cifrar, mediante un cifrador de flujo, los datos del canal encubierto.

---

<sup>IV</sup> *mac80211* reemplaza a la pila anterior llamada *ieee80211*.

- Segundo byte: dato del canal encubierto (cifrado).
- Tercer byte: código de detección de errores (cifrado), basado en *CRC8-SMBUS*<sup>V</sup>.

En la señalización encapsulada implementada, la verificación de la pertenencia de una trama al canal encubierto se realizará de la siguiente manera:

1. Se concatena al valor del primer byte la contraseña que protegerá las comunicaciones del canal encubierto, para construir la clave utilizada para cifrar los datos del canal y el *CRC8-SMBUS*.
2. Se descifran el segundo y tercer byte del campo *IV*, utilizando la clave construida en el paso anterior y el algoritmo de cifrado de flujo *ARC4*.
3. Se calcula el valor *CRC8-SMBUS* del segundo byte descifrado.
4. Se compara el valor calculado con el valor del tercer byte descifrado, de coincidir la trama pertenece al canal encubierto.

Cabe resaltar que la elección de *CRC8-SMBUS* fue arbitraria pudiéndose utilizar cualquier otro código de detección de errores de un byte.

Debido a la presencia del *CRC8-SMBUS*, es posible para el que conoce la clave del canal encubierto verificar la pertenencia de un *IV* al canal encubierto, ya que los datos poseen una estructura interna. La probabilidad de que un *IV* aleatorio sea reconocido como del canal es:  $1/256$  (ya que cada dato enviado tiene un único *CRC8-SMBUS*).

Se decidió utilizar un vector de inicialización para cifrar ya que de esta forma es posible sincronizar al emisor y receptor y permitir que el cifrado del mismo dato del canal encubierto resulte en un campo *IV* distinto, siempre y cuando el vector de inicialización elegido haya variado (razón por la cuál se lo elige aleatoriamente). De esta forma nos aseguramos que la distribución de la fuente que está generando los datos del canal encubierto no se traduzca en una distribución análoga del campo *IV*. Esto se debe a que si se cifra con la misma clave todos los datos, valores iguales resultarían en el mismo valor cifrado; resultando en una mera traslación del sesgo de la fuente emisora.

#### 4.5.2. Arquitectura de la solución

Dada la intención de implementar dos variaciones del canal encubierto se modificó el controlador de red de forma tal que cree una entrada en el sistema de archivos **procfs**<sup>VI</sup> del núcleo de **Linux**. Al escribir a éste pseudo-archivo desde un programa

<sup>V</sup> Ver <http://smbus.org/faq/crc8Applet.htm> para la descripción técnica del código de detección de errores.

<sup>VI</sup>Un pseudo-sistema de archivos utilizado en los sistemas operativos tipo *Unix* para comunicarse con el núcleo

de usuario, el controlador de red utilizará éstos datos como vectores de inicialización de **WEP**.

De esta forma se consiguió reducir la complejidad de implementar dos variantes, ya que las únicas diferencias entre ambas variantes del controlador son:

- Incrementar de forma lineal el *IV* de **WEP**<sup>VII</sup>, versus hacerlo de forma aleatoria.
- Una verificación adicional en el controlador que incrementa de forma lineal de forma tal de saltarse los *IV* que comiencen con la secuencia que indica que la trama pertenece al canal encubierto.

Otra ventaja de este mecanismo es que luego de instalar el canal el mismo se puede seguir utilizando desde un proceso de usuario, es decir no es necesario mantener privilegios de administrador para utilizarlo.

Adicionalmente, se crearon las siguientes herramientas adicionales para cada controlador:

- **gerenate\_ivs.py**: A partir de un archivo especificado como parámetro se cifra el mismo utilizando el algoritmo de cifrado simétrico *Blowfish*[36] con la clave provista por el usuario, y se le indica la secuencia de *IV* al controlador mediante la escritura en la entrada de **procfs** creada por el controlador.
- **sniff\_ivs.py**: Escucha en la interfaz indicada como parámetro (que debe estar en modo monitor) por tramas **WEP** emitidas por la estación especificada (mediante su dirección *MAC*) que pertenezcan al canal encubierto, descifrándolas (con la clave provista por el usuario) y guardando los datos transmitidos por el canal encubierto al archivo especificado como parámetro.
- **read\_ivs.py**: Realiza lo mismo que la anterior excepto que a partir de una captura de red en formato *PCAP*<sup>VIII</sup>.

La instalación del controlador se encuentra detallada en el apéndice B y el uso de las herramientas en el apéndice C.

## 4.6. Implementación mediante la inyección de tráfico

En la presente sección se describe el diseño de la herramienta desarrollada para implementar el canal encubierto mediante la inyección de tramas **IEEE 802.11**; ya que de ésta forma no es necesario modificar el controlador de red. La solución se programó en **Python** utilizando la biblioteca **Scapy**[4] para el manejo del envío de tramas.

---

<sup>VII</sup> Como consecuencia de éste requisito el controlador debe recordar el último valor del canal encubierto utilizado.

<sup>VIII</sup> Puede ser visualizada con programas como WireShark (<http://www.wireshark.org/>)

#### 4.6.1. Método de señalización

Debido al interés de poder comparar entre los dos tipos de soluciones, se estudió la posibilidad de implementar los mismos métodos de señalización que para las implementaciones que modifican el controlador de red, a saber: señalización en banda y señalización encapsulada.

Para el caso de señalización en banda rápidamente se presenta un problema ya que al no poder indicarle al controlador que no debe utilizar ciertos vectores de inicialización (aquellos que indiquen pertenencia al canal encubierto) se interpretarán datos como pertenecientes al canal encubierto cuando no lo son. Un problema similar ocurre con la señalización de inicio y fin: tramas legítimas se pueden intercalar ya que no hay forma de indicarle al controlador que no envíe datos por un cierto periodo de tiempo. Por este motivo se decidió implementar únicamente la versión que utiliza señalización encapsulada.

#### 4.6.2. Datos de las tramas

A la hora de implementar la herramienta surgió el siguiente problema: ¿qué datos enviar en las tramas cifradas? En el caso de la implementación modificando el controlador, este problema no se presentó ya que se enviaban en las tramas del canal encubierto los datos indicados por el sistema operativo.

Se analizaron las siguientes alternativas al problema:

- Enviar paquetes de datos vacíos<sup>IX</sup>.
- Enviar datos inválidos.
- Enviar datos válidos.

Con respecto a la primer alternativa, cabe recordar que el estándar **IEEE 802.11** posee una trama especial que es una trama de datos vacía, por ende no se puede utilizar la trama de datos habitual y enviarla vacía. Tampoco se puede utilizar la trama especial de datos vacíos ya que no posee la encapsulación provista por **WEP** y por ende no posee un campo de vector de inicialización. Ésto se debe a que al no tener datos, no hay nada que cifrar, por ende no tiene los campos adicionales que tiene una trama cifrada.

Por otro lado, la segunda alternativa es viable, pero presenta dos variantes: o bien los datos basura se cifran de forma correcta para que la estación receptora los envíe a la pila de red y sean descartados por la misma (por ser datos basura) o no se cifran correctamente y son descartados por la pila **IEEE 802.11** (dado que el **ICV** va a ser incorrecto). En el primer caso es difícil prever si se presentará un problema en la estación receptora y en el segundo caso será fácil detectar que hay un problema, ya

---

<sup>IX</sup>Normalmente sólo son utilizados por los controladores de red para indicar el inicio de modo de ahorro de energía por parte de la estación o si se está utilizando el modo de operación **PCF**.

que una estación generará más tramas malformadas que lo usual<sup>x</sup>; conduciendo a la posible detección del canal encubierto.

Por lo mencionado anteriormente, se decidió implementar la tercera opción: enviar datos válidos. Ya que no presenta ninguno de los problemas exhibidos anteriormente. A la hora de definir qué datos enviar se decidió enviar paquetes pertenecientes a un protocolo de capa superior que no tuviese estado para evitar cualquier conflicto en la red (producto de no cumplir con la secuencia de estados impuesta por el protocolo de capa superior). Se analizaron varias alternativas: paquetes *ICMP*, paquetes *ARP* y paquetes *UDP*. Se optó por enviar un *gratuitous ARP* ya que este paquete no produce una respuesta por parte de la estación receptora y tampoco generará conflictos en la red. Cabe resaltar que el envío reiterado de cualquier tipo de trama puede resultar sospechoso y conducir a la detección del canal encubierto, aunque nuestro análisis toma como restricción realizar solamente un análisis de la trama en sí misma y no de su contenido.

El análisis de la factibilidad de detección de las diferentes implementaciones fue tratada en 4.4. Sin embargo, el análisis que se realizó asumía que las tramas del canal encubierto eran indistinguibles de las tramas que no pertenecían al mismo. Este supuesto es cierto para el caso de las implementaciones que modifican el controlador, ya que lo único que se modifica es el algoritmo de elección de *IV*. Por el contrario, esto no es cierto para el caso de la implementación que inyecta tramas, ya que la misma no respeta la asignación de números de secuencia (el número de secuencia es un campo que se incrementa linealmente y está presente en todas las tramas de datos). Dado que el mismo debe ser incrementado de a uno no es posible hacer que las tramas del canal encubierto sean indistinguibles de las tramas legítimas.

Para evitar que se pueda detectar por el incremento no lineal del número de secuencia a una trama del canal encubierto, la primera implementación (llamada "inject\_ivs.py") que se realizó emula una estación que se conecta a la red (se autentica y asocia) y envía las tramas incrementando el número de secuencia linealmente; en lugar de enviar las tramas como si fuese una estación que ya se encontraba utilizando la red.

La segunda implementación (llamada "reinject\_ivs.py"), que emula una estación que ya pertenece a la red (copiando su dirección MAC), utiliza en cada trama el último número de secuencia utilizado por la estación real, pero indicando que es una retransmisión (así la estación receptora ignora la trama). De esta forma se logra emular una estación que ya pertenece a la red y respetar el incremento lineal del número de secuencia.

Sin embargo, aún se presentaba un problema a resolver al implementar la segunda versión, a saber: qué datos cifrar. Si se cifran datos distintos, se puede producir un error en el receptor y además, si son de distinta longitud ésta diferencia podrá ser percibida por un sistema de detección de intrusiones; la retransmisión de la trama

---

<sup>x</sup>Al descifrar las tramas el *ICV* no será correcto.

anterior no puede ser de distinta longitud, ya que justamente es una retransmisión de la misma trama. Por este motivo, se optó por utilizar los mismos datos contenidos en la trama, lo que implica descifrar la trama y volverla a cifrar con el valor de *IV* a utilizar para el canal encubierto.

El uso de las herramientas se detalla en el apéndice D.

#### 4.7. Implementación en redes protegidas por WPA/WPA2 y redes sin protección

En las redes protegidas por **WPA/WPA2** no es posible implementar el mismo canal encubierto. Esto se debe a que a la hora de diseñar **WPA/WPA2** se buscó un mecanismo que permitiese prevenir contra ataques de repetición (“replay attacks”<sup>XI</sup>).

En el caso de **WPA** o **TKIP** (nombre no comercial utilizado en el estándar), que sigue utilizando el algoritmo **ARC4** para el cifrado, el vector de inicialización se construye a partir de un contador que es incrementado en uno por cada trama enviada. Por lo tanto el valor del vector de inicialización no puede ser aleatorio como en el caso de **WEP** sino que está perfectamente definido, por ende no pudiendo ser utilizado para implementar un canal encubierto.

En el caso de **WPA2** o **CCMP** (nombre no comercial utilizado en el estándar), se presenta una situación similar. A diferencia de **WPA** se utiliza el algoritmo de cifrado **AES**, sin embargo, de forma similar el vector de inicialización utilizado es un contador que debe ser incrementado en uno por cada trama enviada.

En redes sin cifrado claramente no es posible implementar el canal encubierto de una forma análoga, ya que el formato de trama utilizado no contempla el envío de un vector de inicialización.

El canal encubierto propuesto no es viable de ser implementado en redes inalámbricas sin protección o con otro tipo de protección, sin embargo existe un campo llamado *Duration/ID* (presente en todas las tramas), cuyo valor podría ser manipulado para implementar un canal encubierto de forma similar a lo realizado para los *timestamp* del protocolo *TCP* en ([15]). Además, en todos los casos es posible implementar canales encubierto utilizando las tramas de control o gestión, similares a la descrita en la Sección de *Trabajos previos* (Sección 3.2).

En resumen, el canal encubierto propuesto es posible debido a la especificación del protocolo **WEP**, dando otra razón más para dejar de utilizar el mismo como mecanismo de protección.

---

<sup>XI</sup>Los protocolos vulnerables a este ataque son aquellos donde es posible reenviar una trama, sin modificación, y la misma sigue siendo válida.

## Capítulo 5

---

# Análisis de resultados y escenarios de uso

---

El presente capítulo del trabajo compara la implementación propuesta con las existentes ([7], [14] y [38]), en la Sección 5.2, y las distintas implementaciones desarrolladas entre sí, en la Sección 5.1. Luego, resume los resultados obtenidos (Sección 5.1.1) y describe posibles escenarios de uso (Sección 5.4).

### 5.1. Comparación entre las implementaciones

En la presente sección se resumen las diferencias entre las cuatro implementaciones desarrolladas.

Comenzaremos por analizar la facilidad de instalación del canal encubierto. Ambos tipos de implementaciones requieren privilegios de administrador para funcionar; ya que una requiere modificar el controlador de red, mientras que la otra requiere enviar tramas en modo *raw* a la interfaz de red. Por otro lado, los requisitos de software y hardware para ambas implementaciones son diferentes. Las implementaciones que modifican el controlador requieren que se esté utilizando un controlador que utilice la pila **IEEE 802.11** llamada *mac80211*. Este requisito no está presente en la versión que inyecta, sin embargo ésta tiene como requisito que el sistema tenga instalado *Python*.

El análisis de la factibilidad de detección de las diferentes implementaciones fue tratada en 4.4, motivo por el cuál no se reiterará el análisis sino que simplemente se presentarán sus conclusiones. Como primer paso del análisis, se analizaron distintos esquemas de señalización para “marcar” las tramas pertenecientes al canal encubierto, independientemente de cualquier implementación. La conclusión del análisis dio que la única manera, entre las presentadas, de ocultar el *IV* perteneciente al canal encubierto

de los que no lo son es utilizar señalización encapsulada sobre un controlador que envíe el tráfico de forma aleatoria.

Como se explicó anteriormente, este análisis es suficiente para las implementaciones que modifican el controlador, pero no es lo es para las versiones que inyectan tramas. Para esta clase de implementaciones la siguiente tabla resume las conclusiones, para dos casos: cuando el controlador de red de la estación es incremental (independientemente del modo que incremente) o cuando es aleatorio.

Implementación	Controlador Incremental	Controlador Aleatorio
Emula una estación nueva que se une a la red (inject_ivs.py)	Detección en base a los números de secuencia	
Emula una estación existente en la red (reinject_ivs.py)	Detección en base al <i>IV</i>	Indetectable

Cuadro 5.1: Análisis de detectabilidad para implementaciones que inyectan tramas

En el caso “indetectable”, el único comportamiento anómalo que se podría detectar es que el controlador debe retransmitir tramas periódicamente; cuestión que ocurre usualmente en redes inalámbricas debido al uso de un medio “ruidoso” como es el aire.

Con respecto a la confiabilidad de la solución propuesta, y debido a que el canal como está planteado no es bidireccional, no existe ningún mecanismo que garantice que el receptor recibió los datos. De implementarse un canal bidireccional, por ejemplo, utilizando el mismo canal encubierto en el sentido inverso o un canal alternativo, es posible implementar un protocolo que utilice como medio de transporte al canal encubierto y que garantice el envío de los datos. Además, cabe mencionar que en el caso del canal encubierto utilizando señalización encapsulada, independiente de qué implementación, se puede incorrectamente catalogar tramas como pertenecientes al canal que en realidad no lo son (aunque con muy baja probabilidad). Esto se podría solucionar utilizando un protocolo superior que ayude a detectar este tipo de errores.

Por último, resta analizar si la solución propuesta es un canal encubierto con o sin “ruido”. En 2.1 se definió lo que se entiende por canal encubierto con y sin “ruido”: un canal sin ruido es aquel que utiliza un recurso disponible únicamente a los usuarios del canal, mientras que un canal con ruido utiliza un recurso que también está disponible a terceros que no participan del canal. En el caso particular de nuestra propuesta, nos encontramos ante un canal sin “ruido” ya que si bien los datos (cifrados) enviados a través del canal encubierto pueden ser leídos por cualquiera dentro del rango de alcance de la señal inalámbrica, el campo *IV* utilizado como medio para enviar los

datos del canal sólo puede ser “escrito” por el emisor del canal.

### 5.1.1. Medición de ancho de banda

Calculamos el ancho de banda de un canal encubierto de red como la cantidad de datos (bytes) del canal que se pueden enviar en cada trama del protocolo. El ancho de banda de cada implementación depende fuertemente del método de señalización utilizado, como se puede observar de la siguiente tabla:

Método de señalización	Ancho de banda disponible
FIXME: En banda	N bytes por trama
Valor de inicio y fin	$(N \text{ bytes} \times \# \text{tramas} - 2 \times N) \div (\# \text{tramas} - 2)$
Encapsulado	N bytes por trama

Cuadro 5.2: Ancho de banda disponible según el método de señalización

Donde **N** indica la cantidad de bytes pertenecientes al canal encubierto que una implementación en particular envía en cada trama. La siguiente tabla muestra los valores de **N** en el caso particular de las implementaciones descritas en el presente trabajo:

Método de señalización de la implementación	Valor de N
Valor de inicio y fin	2
Encapsulado	1

Cuadro 5.3: Valor de **N** según el tipo de implementación

## 5.2. Comparación con otras implementaciones de canales encubiertos sobre IEEE 802.11

En cuanto a la implementación [7], ya hemos analizado que desde un punto de detectabilidad ofrece una protección baja en 4.3. Adicionalmente, al no cifrar el contenido de los datos del canal encubierto de detectarse el canal encubierto resulta trivial deducir su contenido. El ancho de banda es de 3 bytes por cada trama enviada en el caso de la primera implementación y 4 bytes por cada trama enviada más dos tramas adicionales que se envían siempre para indicar inicio y fin (ya que utiliza señalización de inicio y fin).

En cuanto a la interoperabilidad es posible decir que en principio no debería interferir ya que las tramas *ACK* no serán interpretadas por ninguna estación a menos que coincida el valor del campo *RA* con la dirección *MAC* de una estación de la red, en cuyo caso la estación podría interpretar que a una estación le llegó una trama pero

la confirmación se debe al canal encubierto y no a la correcta recepción de la trama por la estación destinataria. Debido a la selección de valores realizada por la primera implementación esto no debería suceder (según [1] el valor 0x6666 para los primeros dos bytes de la dirección MAC no se encuentran asignados a ningún fabricante<sup>1</sup>). En cuanto a la segunda implementación, sucede lo mismo: los valores de los distintos tipos de tramas (inicio, fin y datos) utilizan valores no asignados. Por otro lado, esto también afecta a la detectabilidad ya que se está utilizando valores de direcciones MAC que no debería tener asignados ninguna estación.

Por último, podemos decir que el canal no es confiable (ya que no posee ningún mecanismo para cerciorarse la recepción de datos) y que la complejidad de instalación es similar a las soluciones implementadas donde se inyecta tráfico.

En cuanto a las propuestas de [14], ya hemos analizado que desde un punto de detectabilidad ofrecen una protección baja en 4.3. Adicionalmente, ya que no cifran el contenido de los datos del canal encubierto (aunque menciona que esta posibilidad existe) de detectarse el canal encubierto resulta trivial deducir su contenido. El ancho de banda será de tres bytes por trama para la propuesta de utilizar el vector de inicialización y de un byte para la propuesta de utilizar el número de secuencia, más dos tramas adicionales (una para indicar que el inicio de datos pertenecientes al canal encubierto y otra para indicar la cantidad de datos a ser enviados).

Con respecto a la interoperabilidad, la propuesta de utilizar el número de secuencia puede traer problemas ya que es posible que se envíen datos con números de secuencias menores a los ya utilizados; resultando en que el *access point* o la estación receptora descarte las tramas. La otra propuesta también puede ocasionar problemas ya que el “padding” del campo *Key ID* debería permanecer en cero (según el estándar); esultando en que el *access point* o la estación receptora descarte las tramas.

Por último, podemos decir que el canal no es confiable (ya que no posee ningún mecanismo para cerciorarse la recepción de datos), pero al no presentarse una implementación no es posible analizar la complejidad de instalación.

En cuanto a las propuestas de [14], ya hemos analizado que desde un punto de detectabilidad ofrecen una protección baja en 4.3. Dado que no se cuenta con una implementación de las propuestas ni un detalle de las mismas, resulta imposible analizar el ancho de banda que poseen, su interoperabilidad y confiabilidad.

### 5.3. Ejemplos de uso

Los archivos “controlador\_flag.cap” y “controlador\_no\_flag.cap”, son capturas de tramas (en formato *PCAP*<sup>II</sup>.) del envío del texto “mensaje de prueba” (en ASCII),

---

<sup>1</sup>Los primeros tres bytes de la dirección MAC indican el OUI (Organizationally Unique Identifier), que son asignados a los fabricantes por **IEEE**.

<sup>II</sup>Puede ser visualizada con programas como WireShark (<http://www.wireshark.org/>)

a través de del canal encubierto utilizando las implementaciones que modifican el controlador de red y el método de señalización de inicio y fin y de señalización encapsulada, respectivamente.

Los archivos “reinyeccion.cap” y “inyeccion.cap”, son capturas de tramas (en formato *PCAP*) del envío del texto “mensaje de prueba” (en ASCII), a través de del canal encubierto utilizando las implementaciones que inyectan tráfico, señalización encapsulada y respetan los números de secuencia o no, respectivamente.

La clave **WEP** utilizada en la red de pruebas fue: “AABBCCDDEE” y la contraseña del canal encubierto fue “testtest”.

## 5.4. Escenarios de uso

Los canales encubiertos generalmente son utilizados con el objetivo de sustraer información de forma sigilosa. Además, si la comunicación es bidireccional se puede utilizar el canal para enviar información o ejecutar comandos (y ver sus resultados) en el sistema remoto.

La sustracción de información de forma desapercibida de un sistema puede tener tanto usos legítimos como ilegítimos, por ejemplo:

- Robo de información: un atacante que ha comprometido un sistema puede utilizar un canal encubierto para copiarse información sin ser detectado.
- Monitoreo de un sistema: un atacante puede observar un sistema comprometido sin riesgo de ser detectado, o un administrador puede observar el comportamiento de un atacante en un sistema instalado para ser comprometido (comúnmente llamado *honeypot*) sin ser detectado por el atacante (cuestión que probablemente modifique drásticamente su comportamiento).
- Para poder enviar y/o recibir información a través de un canal altamente observado, por ejemplo por un empleador o gobierno, sin “hacer sonar las alarmas”.

Es importante aclarar que un canal encubierto no es una herramienta de ataque; no puede ser utilizado para comprometer sistemas. Sólo permite mantener una conversación de difícil detección, para que se utiliza es lo que pone en tela de juicio su legalidad.

## Capítulo 6

---

# Conclusiones y trabajo futuro

---

La presente sección del trabajo presenta las conclusiones (6.1) resultantes de la realización del mismo. Por último, cierra el trabajo dejando ideas para posibles trabajos futuros en el área (6.2).

### 6.1. Conclusiones

La conclusión fundamental del presente trabajo es que es posible construir un canal encubierto sobre el protocolo **IEEE 802.11** (utilizando las extensiones provistas en el mismo para soportar **WEP**), que cumple con los requisitos planteados en la Sección 4.2, que esencialmente definen un canal encubierto cuyo uso es “indistinguible” de su no uso.

La existencia de canales encubiertos en un protocolo de red es un problema de seguridad ya que aumenta el riesgo de utilizar el mismo. Esto se debe a que utilizar el protocolo introduce un mecanismo oculto de comunicación que puede ser aprovechado por un atacante con resultados perjudiciales para quien emplea el protocolo.

La idea fundamental presentada en el presente trabajo es la de utilizar campos con valores pseudo-aleatorios especificados en un protocolo para el envío de datos pertenecientes a un canal encubierto. Como se mostró en el caso específico de **WEP** del protocolo para redes inalámbricas **IEEE 802.11**, permitió desarrollar un controlador de red donde no es posible dirimir cuáles de las tramas generadas por el controlador de red contienen información que pertenece al canal encubierto y cuáles no. Este concepto puede permitir construir canal encubiertos de difícil detección en una amplia gama de protocolos; generalmente los protocolos que utilicen criptografía utilizan valores pseudo-aleatorios que podrían ser aprovechados.

Adicionalmente, se presentó una idea para implementar un canal encubierto que funcionaría en redes sin protección o protegidas por **WPA/WPA2**, que aprovecha

una peculiaridad de la forma en que las estaciones *Windows XP* construyen las tramas *Probe Request*. Debido a esa peculiaridad, las *Windows XP* envían datos aleatorios, lo que permitiría utilizar el mismo concepto que el aplicado al canal encubierto propuesto en el presente trabajo.

Uno de los objetivos del presente trabajo era mostrar que **WEP** no sólo no cumple su objetivo primario, proveer confidencialidad (ya demostrado por numerosos estudios), sino que además agrega otros problemas de seguridad a una red que supuestamente protege. El mismo se cumplió al demostrar que **WEP** permite implementar un canal encubierto de muy difícil detección que no es posible implementar en redes sin protección ni en redes con **WPA/WPA2** (según el análisis presentado en 4.7).

## 6.2. Trabajo futuro

Es posible extender las ideas presentadas en este trabajo en diferentes formas. A continuación se presentan las líneas de acción que consideramos más interesantes y fructuosas:

- Extender el concepto de utilizar un valor pseudo-aleatorio de un protocolo de comunicación para implementar un canal encubierto a otros protocolos de red.
- Analizar el uso del campo *Duration/ID* del protocolo **IEEE 802.11** para implementar un canal encubierto “temporal”, basado en diferencias entre el tiempo de transmisión de una trama y el tiempo indicado, para los casos de tramas en que la interpretación del campo es un tiempo de duración.
- Analizar el uso del campo *Duration/ID* del protocolo **IEEE 802.11** para implementar un canal encubierto de “almacenamiento”, utilizando los bits menos significativos para implementar el canal encubierto de forma similar a lo realizado para los *timestamp* del protocolo *TCP* en ([15]).
- Mejorar la implementación del canal encubierto para que sea bidireccional, con el objetivo de crear una interfaz de red virtual que permita utilizar protocolos como *TCP/IP* a través del canal encubierto.
- Estudiar la posibilidad de implementar un canal encubierto sobre el protocolo para redes inalámbricas de largo alcance **IEEE 801.16**.
- Estudiar otras alternativas para implementar un canal encubierto en redes inalámbricas protegidas por **WPA/WPA2**, ya que no es posible extrapolar la idea presentada a éste tipo de redes (como se demostró en la Sección 4.7).
- Estudiar la viabilidad de implementar el canal encubierto propuesto en la Sección 4.3, basado en las particularidades del manejo de tramas *Probe Request* por parte de las estaciones *Windows XP*. Debido a que las estaciones *Windows*

*XP* envían datos aleatorios periódicamente, es posible extrapolar el concepto presentado en este trabajo, potencialmente obteniendo un canal encubierto “indistinguible”.

---

# Referencias

---

- [1] Public oui listing. URL <http://standards.ieee.org/regauth/oui/oui.txt>.
- [2] Wi-Fi Alliance. *Wi-Fi Protected Access (WPA)*. URL <http://www.wi-fi.org>.
- [3] William A. Arbaugh, Narendar Shankar, and Y. C. Justin Wan. Your 802.11 wireless network has no clothes, May 15 2001. URL <http://citeseer.ist.psu.edu/472552.html>; <http://www.drizzle.com/~aboba/IEEE/wireless.pdf>.
- [4] Philippe Biondi. Scapy. URL <http://www.secdev.org/projects/scapy/>.
- [5] M. Bishop. *Computer Security: Art and Science*. Addison-Wesley, Boston, USA, 2003.
- [6] Andrea Bittau, Mark Handley, and Joshua Lackey. The final nail in WEP's coffin. In *IEEE Symposium on Security and Privacy*, pages 386–400. IEEE Computer Society, 2006. ISBN 0-7695-2574-1. URL <http://doi.ieeecomputersociety.org/10.1109/SP.2006.40>.
- [7] L. Butti and F. Veysset. Wi-fi advanced stealth. Black Hat Briefings USA, 2006. URL <http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Veyssett.pdf>.
- [8] Laurent Butti. Python raw covert, . URL <http://rfakeap.tuxfamily.org/pyrawcovert-0.1.tar.gz>.
- [9] Laurent Butti. Raw covert, . URL <http://rfakeap.tuxfamily.org/rcovert-0.1.tar.gz>.
- [10] Rafik Chaabouni. Break wep faster with statistical analysis. Technical report, EPFL, LASEC, June 2006.
- [11] Shane A. Macaulay Dino A. Dai Zovi. Attacking automatic wireless network selection, 2005. URL <http://www.theta44.org/karma/aawns.pdf>.

- [12] Wireless Geographic Logging Engine. Ssid stats (top 1000). URL <http://wigle.net/gps/gps/main/ssidstats>.
- [13] FIPS. *Advanced Encryption Standard (AES)*. National Institute for Standards and Technology, pub-NIST:adr, November 2001. URL <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
- [14] L. Frikha and Z. Trabelsi. A new covert channel in wifi networks. In *Risks and Security of Internet and Systems, 2008. CRISIS '08. Third International Conference on*, pages 255 –260, 28-30 2008. doi: 10.1109/CRISIS.2008.4757487.
- [15] Giffin, Greenstadt, Litwack, and Tibbetts. Covert messaging through TCP timestamps. In *International Workshop on Privacy Enhancing Technologies (PET), LNCS*, volume 2, 2002.
- [16] Handel and Sandford. Hiding data in the OSI network model. In *IWIH: International Workshop on Information Hiding*, 1996.
- [17] IEEE. *IEEE Std 802.11i, Amendment to IEEE Std 802.11 - Amendment 6: Medium Access Control (MAC) Security Enhancements*. IEEE, 2004.
- [18] IEEE. *IEEE Std 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE, Edition 2007. URL <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>.
- [19] AirDefense Inc. Motorola airdefense - security and compliance solutions, . URL <http://www.airdefense.net/products/servicesplatform/securitycompliance/index.php>.
- [20] AirDefense Inc. Airdefense's comprehensive survey of 3,000 retail stores finds many wireless data security vulnerabilities as holiday shopping season nears, . URL [http://www.airdefense.net/newsandpress/11\\_15\\_07.php](http://www.airdefense.net/newsandpress/11_15_07.php).
- [21] AirDefense Inc. Airdefense's survey of retailers across new york city discovers wireless security vulnerabilities in brooklyn, the bronx, manhattan, queens and staten island, . URL [http://www.airdefense.net/newsandpress/01\\_14\\_08.php](http://www.airdefense.net/newsandpress/01_14_08.php).
- [22] AirMagnet Inc. Airmagnet enterprise, . URL <http://www.airmagnet.com/products/enterprise/>.
- [23] AirTight Networks Inc. Spectraguard enterprise, . URL <http://www.airtightnetworks.com/home/solutions/wireless-intrusion-prevention.html>.
- [24] K.Kaukonen and R.Thayer. A stream cipher encryption algorithm "arcfour", 1999.

- [25] B. W. Lampson. A note on the confinement problem. *ACM*, 16(10):613–615, October 1973.
- [26] Microsoft. Description of the wireless client update for windows xp with service pack 2, 2007. URL <http://support.microsoft.com/kb/917021>.
- [27] Department of Defense Standard. *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985.
- [28] Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding – a survey. *Proceedings of the IEEE (USA)*, 87(7):1062–1078, July 1999.
- [29] Anton T. Rager. Wepcrack. URL <http://wepcrack.sourceforge.net>.
- [30] Cybsec S.A. Wardriving buenos aires 2005, . URL [http://www.cybsec.com/upload/TadeoCwierz\\_Buenos\\_Aires\\_Wardriving2005\\_1.pdf](http://www.cybsec.com/upload/TadeoCwierz_Buenos_Aires_Wardriving2005_1.pdf).
- [31] Cybsec S.A. Wardriving buenos aires 2006, . URL [http://www.cybsec.com/upload/Tendencias06\\_WardrivingBsAs2006.pdf](http://www.cybsec.com/upload/Tendencias06_WardrivingBsAs2006.pdf).
- [32] Cybsec S.A. Wardriving buenos aires 2007, . URL [http://www.cybsec.com/upload/cybsec\\_Tendencias07\\_Wardriving\\_BsAs2007.pdf](http://www.cybsec.com/upload/cybsec_Tendencias07_Wardriving_BsAs2007.pdf).
- [33] Cybsec S.A. Wardriving buenos aires 2008, . URL [http://www.cybsec.com/upload/Estadisticas\\_WarDriving\\_Wireless.pdf](http://www.cybsec.com/upload/Estadisticas_WarDriving_Wireless.pdf).
- [34] Cybsec S.A. Tendencias en seguridad de la información, . URL [http://www.cybsec.com/upload/tendencias\\_Arg\\_2009\\_v1\\_Pmilano.pdf](http://www.cybsec.com/upload/tendencias_Arg_2009_v1_Pmilano.pdf).
- [35] R. Sbrusch. Network covert channels: Subversive secrecy. Technical report, SANS Institute, October 2006. URL [http://www.sans.org/reading\\_room/whitepapers/covert/1660.php](http://www.sans.org/reading_room/whitepapers/covert/1660.php).
- [36] Bruce Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *Fast Software Encryption, Cambridge Security Workshop*, pages 191–204, London, UK, 1994. Springer-Verlag. ISBN 3-540-58108-1.
- [37] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. Using the fluhrer, mantin, and shamir attack to break WEP. In *NDSS. The Internet Society*, 2002. ISBN 1-891562-14-2; 1-891562-13-4. URL <http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/stubbl.pdf>.
- [38] Krzysztof Szczypiorski. Hiccups: Hidden communication system for corrupted networks. In *In Proc. of: The Tenth International Multi-Conference on Advanced Computer Systems ACS'2003, October 22-24, 2003 Mie;dzydroje*, pages 31–40, 2003.

- [39] The Aircrack-NG team. Aircrack-ng suite. URL <http://www.aircrack-ng.org>.
- [40] CACE Technologies. Aircap. URL <http://www.cacetech.com/products/airpcap.htm>.
- [41] Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin. Breaking 104 bit wep in less than 60 seconds. Cryptology ePrint Archive, Report 2007/120, 2007. URL <http://eprint.iacr.org/2007/120.pdf>.
- [42] Y.C.J. Wav W.A. Arbaugh, N. Shankar. An inductive chosen plaintext attack against wep/wep2, 2001.
- [43] R. Walker. IEEE P802.11 wireless LANs unsafe at any key size; an analysis of the WEP encapsulation, November 16 2001. URL <http://citeseer.ist.psu.edu/558358.html>; <http://md.hudora.de/archiv/wireless/unsafew.pdf>.

## Apéndice A

---

# Mensajes soportados por el protocolo IEEE 802.11

---

Los distintos mensajes soportados por el protocolo se encuentran descritos en el cuadro A.1.

Tipo de trama	Subtipo	Descripción
Gestión	Association request	Pedido de asociación a la red
Gestión	Association response	Respuesta al pedido de asociación
Gestión	Reassociation request	Pedido de reasociación a la red
Gestión	Reassociation response	Respuesta al pedido de reasociación
Gestión	Probe request	Consulta por disponibilidad de una red
Gestión	Probe response	Respuesta a la consulta de disponibilidad
Gestión	Beacon	Indica la presencia de un <i>access point</i>
Gestión	ATIM	Utilizado para el manejar el ahorro de energía en redes <i>ad hoc</i>
Gestión	Disassociation	Indicación de desasociación de una red
Gestión	Authentication	Pedido/respuesta de autenticación
Gestión	Deauthentication	Indicación de deautenticación de una red
Gestión	Action	Utilizada para implementar mensajes de gestión específicos a un vendedor
Control	Block Ack Request (BlockAckReq)	Pedido para anular el uso de tramas de confirmación (ACK)
Control	Block Ack (BlockAck)	Respuesta al pedido de anular el uso de tramas de confirmación (ACK)

continúa en la próxima página

continuado de la página previa		
Tipo de trama	Subtipo	Descripción
Control	PS-Poll	Utilizado por una estación en modo de ahorro de energía para solicitar las tramas pendientes almacenadas por el <i>access point</i>
Control	RTS	Pedido de envío de tramas (reserva del medio físico)
Control	CTS	Respuesta afirmativo a la solicitud de envío de tramas
Control	ACK	Confirmación de recepción
Control	CF-End	Indica el fin del período de contención
Control	CF-End + CF-Ack	Indica el fin del periodo de contención y confirma la recepción de una trama CF-Poll (trama que le indica a una estación que puede transmitir durante un periodo libre de contención)
Datos	Datos	Trama de datos
Datos	Datos + CF-Ack	Trama de datos con CF-Ack
Datos	Datos + CF-Poll	Trama de datos con CF-Poll
Datos	Datos + CF-Ack + CF-Poll	Trama de datos con CF-Ack y CF-Poll
Datos	Null (no data)	Trama de datos vacía
Datos	CF-Ack (no data)	Trama de datos vacía y indica CF-Ack
Datos	CF-Poll (no data)	Trama de datos vacía y indica CF-Poll
Datos	CF-Ack + CF-Poll (no data)	Trama de datos vacía y indica CF-Ack y CF-Poll
Datos	QoS Datos	Trama de datos que utiliza funciones de QoS
Datos	QoS Datos + CF-Ack	Trama de datos que utiliza funciones de QoS e indica CF-Ack
Datos	QoS Datos + CF-Poll	Trama de datos que utiliza funciones de QoS e indica CF-Poll
Datos	QoS Datos + CF-Ack + CF-Poll	Trama de datos que utiliza funciones de QoS e indica CF-Ack y CF-Poll
Datos	QoS Null (no data)	Trama de datos vacía que utiliza funciones de QoS e indica CF-Ack
Datos	QoS CF-Poll (no data)	Trama de datos vacía que utiliza funciones de QoS e indica CF-Poll

continúa en la próxima página

APÉNDICE A. MENSAJES SOPORTADOS POR EL PROTOCOLO **IEEE 802.115**

continuado de la página previa		
Tipo de trama	Subtipo	Descripción
Datos	QoS CF-Ack + CF-Poll (no data)	Trama de datos vacía que utiliza funciones de QoS e indica CF-Ack y CF-Poll

Cuadro A.1: Significado de los diferentes tipos y subtipos de tramas

## Apéndice B

---

# Instalación del controlador

---

La instalación del controlador de red depende de la distribución **Linux** que se esté empleando, pero se deben realizar los siguientes pasos:

1. Obtener los fuentes del núcleo de **Linux** utilizado por la distribución. También es posible utilizar la versión “vanilla” del núcleo <sup>1</sup>.
2. Copiar los archivos *ieee80211.i.h*, *main.c* y *wep.c* al directorio *net/mac80211* del núcleo descargado en el paso anterior.
3. En el directorio *net/mac80211* ejecutar el comando:

```
make -C /lib/modules/`uname -r`/build M=`pwd` modules
```

4. Copiar el archivo *mac80211.ko* al directorio:  
*/lib/modules/`uname -r`/kernel/net/mac80211*.
5. Listar los módulos que dependen de *mac80211*:

```
lsmod | egrep '^mac80211' | awk '{print $4}'
```

6. Eliminar los módulos mediante el comando: `rmmmod`.
7. Eliminar el módulo *mac80211*: `rmmmod mac80211`.
8. Cargar los módulos que fueron eliminados mediante el comando *rmmmod* (incluido *mac80211*) con el comando: `modprobe`.

---

<sup>1</sup>La versión del núcleo distribuidas en [www.kernel.org](http://www.kernel.org) que no tiene ninguno de los parches aplicados por la distribución

## Apéndice C

---

# Uso de las herramientas adicionales

---

El comando *generate\_ivs.py* tiene el siguiente modo de uso:

```
$ python generate_ivs.py -i ARCHIVO_A_ENVIAR
```

El comando *sniff\_ivs.py* tiene el siguiente modo de uso:

```
$/sniff_ivs.py -i INTERFAZ -o ARCHIVO_SALIDA -m DIRECCION_MAC
```

Donde *INTERFAZ* indica el nombre de la interfaz que se encuentra en modo monitor (y en el canal correcto), para recibir las tramas enviadas por la estación que está haciendo uso del canal encubierto, y *DIRECCION\_MAC* indica su dirección MAC.

En el caso de la versión que para el controlador que realiza señalización encapsulada el comando es:

```
$/sniff_ivs.py -i INTERFAZ -o ARCHIVO_SALIDA -m DIRECCION_MAC -c #_BYTES_ESPERAR
```

Donde *#\_BYTES\_ESPERAR* indica la cantidad de bytes a recibir mediante el canal encubierto y *INTERFAZ* e *DIRECCION\_MAC* indica lo mismo que en el caso anterior.

El comando *read\_ivs.py* tiene el siguiente modo de uso:

```
$/read_ivs.py -i ARCHIVO_ENTRADA -o ARCHIVO_SALIDA -m DIRECCION_MAC
```

## Apéndice D

---

# Uso de las herramienta de inyección

---

El comando *inject\_ivs.py* tiene el siguiente modo de uso:

```
$/inject_ivs.py -i ARCHIVO_ENTRADA -f INTERFAZ  
                -w CLAVE_WEP -k ID_CLAVE_WEP -b BSSID [-s SSID]
```

Donde *INTERFAZ* indica el nombre de la interfaz que se encuentra en modo monitor (y en el canal correcto), para enviar las tramas enviadas por la estación que está haciendo uso del canal encubierto, *CLAVE\_WEP* la clave **WEP** a utilizar para cifrar las tramas, *ID\_CLAVE\_WEP* indica el identificador de la clave **WEP**<sup>1</sup>. El *BSSID* es la dirección MAC del access point de la red. Por último, el *SSID* es el *SSID* de la red, que es un parámetro opcional ya que solamente es necesario cuando el access point está “escondiendo” el mismo.<sup>11</sup>

El comando *reinject\_ivs.py* tiene el siguiente modo de uso:

```
$/reinject_ivs.py -i ARCHIVO_ENTRADA -f INTERFAZ  
                 -w CLAVE_WEP -b BSSID -m DIRECCION_MAC
```

Donde *INTERFAZ*, *CLAVE\_WEP* y *BSSID* tienen el mismo significado que en el caso anterior, y *DIRECCION\_MAC* es la dirección MAC de la estación que se desea utilizar como dirección origen en las tramas que se envían; para imitar una estación que pertenezca a la red.

---

<sup>1</sup>Cabe recordar que según el estándar IEEE 802.11 se pueden configurar hasta cuatro claves **WEP**, y cada trama indica cuál es la que está en uso. Comúnmente se utiliza primer clave (con identificador cero)

<sup>11</sup>La mayoría de los access point soportan no enviar el *SSID* en las tramas *beacon*, como una medida de seguridad, si bien dicha funcionalidad no está contemplada en el estándar.

---

# Índice de figuras

---

2.1. Formato de tramas MAC . . . . .	8
2.2. Formato del campo de control de tramas ( <i>Frame Control</i> ) . . . . .	10
2.3. Formato de una trama WEP . . . . .	12
2.4. Procedimiento de cifrado de WEP . . . . .	12
3.1. Formato de tramas <i>ACK</i> . . . . .	18
3.2. Formato de tramas <i>ACK</i> del canal encubierto [7] . . . . .	18
4.1. Porcentaje de Redes Criptadas en la Zona Céntrica de Buenos Aires . . . . .	21
4.2. Señalización en banda: controlador con incremento big-endian . . . . .	31
4.3. Señalización en banda: controlador con incremento little-endian . . . . .	32
4.4. Señalización en banda: controlador con incremento aleatorio . . . . .	32
4.5. Señalización de inicio y fin: controlador con incremento big-endian . . . . .	33
4.6. Señalización de inicio y fin: controlador con incremento little-endian . . . . .	33
4.7. Señalización de inicio y fin: controlador con incremento aleatorio . . . . .	34

---

# Índice de cuadros

---

2.1. Enmiendas de IEEE 802.11 relacionadas con la capa física . . . . .	7
2.2. Significado de los campos estándares de una trama . . . . .	9
2.3. Significado de los subcampos del campo de control de trama . . . . .	10
2.4. Significado de los de los campos <i>To-DS</i> and <i>From-DS</i> . . . . .	11
4.1. Tramas de control y gestión que pueden ser enviadas por una estación	23
4.2. Situaciones dónde se utilizan distintos tipos de tramas . . . . .	24
5.1. Análisis de detectabilidad para implementaciones que inyectan tramas	42
5.2. Ancho de banda disponible según el método de señalización . . . . .	43
5.3. Valor de <b>N</b> según el tipo de implementación . . . . .	43
A.1. Significado de los diferentes tipos y subtipos de tramas . . . . .	55