Seguridad en redes WLAN

PID_00191699

Xavier Perramon Tornil

Universitat Oberta de Catalunya

Índice

Introducción

El problema específico de las redes sin hilo es que, a diferencia de las redes con hilos, el acceso al medio de transmisión es libre, en el sentido de que no es necesario hacer nada especial para conectarse físicamente. Por ejemplo, cualquier usuario que tenga un dispositivo Wi-Fi en modo promiscuo puede ver las tramas que se transmiten a su entorno, sin ninguna otra limitación que la distancia a la estación emisora. Este hecho tiene consecuencias muy importantes, especialmente por lo que se refiere a la seguridad de la información que se transmite en este tipo de redes. Si la información que viaja en este tipo de redes no se protege convenientemente, un atacante puede conseguirla sin demasiado esfuerzo.

En este módulo didáctico veremos qué mecanismos existen para la protección de redes que siguen el estándar IEEE 802.11. En primer lugar, analizaremos el protocolo WEP, que fue el primer protocolo de seguridad para el estándar. Veremos que, a pesar de aportar un nivel de seguridad superior a enviar la información en claro, las vulnerabilidades conocidas de este protocolo hacen que no sea recomendable su utilización. En concreto, veremos diferentes tipos de ataques, y también algunas herramientas que permiten implementarlos.

Finalmente, analizaremos el protocolo WPA en sus diferentes variantes de funcionamiento, que permite superar las limitaciones y los ataques que existen sobre el protocolo WEP.

Objetivos

Los conceptos expuestos en el presente módulo didáctico os van a permitir alcanzar los siguientes objetivos:

1. Conocer los componentes básicos de una red WLAN.
2. Entender los problemas de seguridad y de los ataques a redes WLAN.
3. Comprender el funcionamiento del sistema de protección WEP.
4. Identificar las limitaciones de seguridad del protocolo WEP.
5. Entender el funcionamiento del protocolo WPA, así como de sus diferentes modos de funcionamiento.

1. Conceptos básicos de las redes Wi-Fi

El estándar más utilizado actualmente para las comunicaciones en redes locales inalámbricas (WLAN) es el llamado IEEE 802.11, más conocido como Wi-Fi. La primera versión de la especificación, que data de 1997, permitía comunicaciones de hasta 2 Mbit/s. Desde entonces, se han ido añadiendo extensiones que contemplan velocidades máximas de transmisión cada vez más altas: 11 Mbit/s (802.11b), 54 Mbit/s (802.11a y 802.11g) y 600 Mbit/s (802.11n).

Uno de los criterios de diseño iniciales de este estándar era facilitar la interoperabilidad, cosa que no siempre ha sido del todo compatible con la seguridad. Las primeras versiones basaban la protección de las comunicaciones en el protocolo WEP, que pronto se demostró que era demasiado débil. En el año 2004 se publicó la norma IEEE 802.11i con el objetivo de corregir las deficiencias del sistema de seguridad WEP.

WLAN

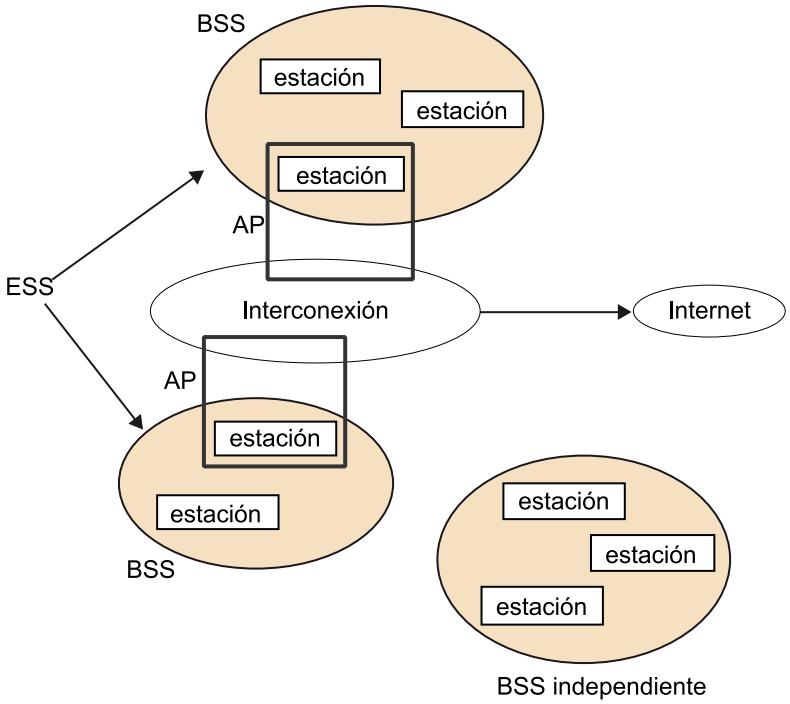
WLAN es la sigla de *wireless local area network*.

IEEE 802.11 permite la comunicación entre dispositivos, denominados **estaciones**, que tengan una interfaz de red inalámbrica. Cada interfaz tiene una dirección MAC de 48 bits con el mismo formato que las direcciones Ethernet. Dos o más estaciones que por su proximidad pueden comunicarse entre sí forman un *basic service set* (BSS). Se distinguen dos tipos de BSS:

- los independientes y
- los infraestructurales.

Un **BSS independiente**, también conocido como red ad-hoc, es una red aislada donde las únicas comunicaciones posibles son las directas de una estación a otra. En **un BSS infraestructural**, en cambio, hay una estación específica llamada **punto de acceso** (AP) que permite la interconexión con otras redes, de cable o inalámbricas. Un *extended service set* (ESS) es un conjunto de uno o más BSS infraestructurales interconectados por medio de sus AP. Desde el punto de vista de las estaciones, la ESS funciona como si fuera un único BSS.

Figura 1. Componentes de las redes Wi-Fi



Configuraciones Wi-Fi

La configuración típica de las redes Wi-Fi domésticas es la de un *router* que, por un lado, da acceso a Internet vía ADSL, y por el otro actúa como AP permitiendo la conexión desde las estaciones que se encuentren en su radio de alcance. En esta configuración hay un ESS formado por un único BSS. En una red Wi-Fi corporativa, en cambio, es habitual tener varios AP en diferentes partes de un edificio: en este caso, todos los BSS normalmente pertenecen a un mismo ESS.

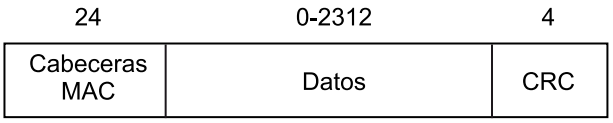
Los BSS se identifican con su BSSID, que en los infraestructurales es la dirección MAC de su AP. Los ESS tienen un identificador de formato libre de hasta 32 bytes. Se utiliza el término genérico *service set identifier* (SSID) para referirse al identificador de un BSS independiente o de un ESS.

Las estaciones pueden entrar y salir de un BSS de forma dinámica. En un BSS infraestructural, el AP anuncia su presencia enviando periódicamente **tramas baliza**, típicamente cada 100 ms. Los diferentes campos de una baliza indican la SSID de la red, las velocidades de transmisión que permite, etc.

Una estación pasa a ser miembro de un BSS infraestructural cuando establece una **asociación** con el AP correspondiente. En cada momento una estación solo puede estar asociada a un AP. Una vez establecida la asociación, la estación normalmente envía y recibe todas sus tramas a través de este AP. Sin embargo, para poder hacer la asociación y entrar en el BSS, hace falta que previamente la estación haya realizado una **autenticación** ante el AP.

Las tramas que pueden enviar y recibir las estaciones Wi-Fi pertenecen a uno de estos tres tipos: tramas de gestión, tramas de datos y tramas de control. Las tramas de gestión incluyen, entre otras, las balizas, las tramas de autenticación y desautenticación, y las de asociación y disociación.

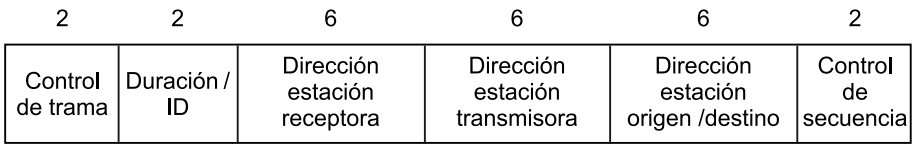
Figura 2. Formato de las tramas de datos



Las tramas de datos que se transmiten en un BSS infraestructural constan de las partes siguientes:

- La cabecera MAC, de 24 bytes, con la estructura que se describe a continuación.
- Los datos que se envían en la trama, con una longitud de hasta 2.304 bytes, o 2.312 si los datos incluyen encapsulamiento WEP.
- Un código de comprobación de errores, que es un código CRC de 32 bits calculado sobre la cabecera y los datos.

Figura 3. Campos de la cabecera MAC de una trama de datos



La cabecera MAC de estas tramas está formada por los campos siguientes:

- Control de trama: incluye varios subcampos como, por ejemplo, la versión del protocolo, tipo y subtipo de trama, un *flag* para indicar si es el último fragmento de una trama fragmentada, etc.
- Duración/ID: en una trama de datos este campo se utiliza para indicar el tiempo en que se tiene que transmitir una trama de control ACK.
- Dirección de la estación receptora: indica la estación a la cual se envía directamente la trama.
- Dirección de la estación transmisora: indica la estación que ha enviado esta trama.
- Dirección de la estación origen/destino. Si es una trama enviada desde una estación al AP, este campo indica el destino final, que puede ser el propio AP (y entonces esta tercera dirección coincide con la primera) o bien otra estación a la que el AP retransmitirá la trama. Por el contrario, si es una trama enviada por el AP a una estación, este campo indica el origen de la trama, que puede ser el mismo AP (y la tercera dirección coincidirá con la segunda) o bien otra estación que ha enviado la trama a través del AP.

- Control de secuencia: incluye un número de secuencia de la trama y un número de fragmento.

2. Métodos de autenticación de las estaciones Wi-Fi

Las primeras versiones del estándar Wi-Fi anteriores al IEEE 802.11i, siguiendo el criterio de simplicidad y de facilitar al máximo la interoperabilidad, preveían dos tipos de autenticación de las estaciones Wi-Fi ante el AP.

- **Autenticación de sistema abierto**. Esta autenticación, si el AP está configurado para permitirla, es muy simple: cada estación que solicita la autenticación recibe automáticamente la confirmación. Por eso también se le conoce como algoritmo de autenticación nulo. La ventaja de esta autenticación es que no hace falta que las estaciones hagan nada especial para completarla. Así se logra el objetivo de facilitar la conexión de las estaciones que se incorporen a la red.

- **Autenticación de clave compartida**. Este método de autenticación se utiliza junto con el sistema de cifrado WEP. En este caso, el AP hace uso de una clave WEP que tiene preconfigurada y que solo habrían de conocer las estaciones que se tuvieran que autenticar. Cuando una estación solicita la autenticación, el AP le manda un mensaje con 128 bytes aleatorios, y la estación tiene que responder con una trama que contenga este mismo mensaje, cifrada con la clave WEP. Se trata, pues, de un protocolo de reto-respuesta con clave simétrica. El problema que presenta es que está basado en el cifrado WEP y, como veremos más adelante, descubrir una clave WEP no es excesivamente complicado para un atacante que pueda capturar una cantidad suficiente de tramas cifradas. Pero además este protocolo de autenticación tiene otro problema, y es que la respuesta cifrada no incluye ningún identificador de la estación que se quiere autenticar. Como veremos también más adelante, esto permite a un atacante que capture un solo intercambio de mensajes utilizar los datos capturados para autenticarse con éxito ante el AP.

A partir de la publicación del estándar IEEE 802.11i, el uso de la autenticación de clave compartida está desaconsejado, y solo se contempla en situaciones donde se quiera mantener la compatibilidad con sistemas anteriores. Para realizar una autenticación más segura, IEEE 802.11i introduce el concepto de *robust security network association* (RSNA), basado en el *extensible authentication protocol* (EAP) de acuerdo con el estándar IEEE 802.1X.

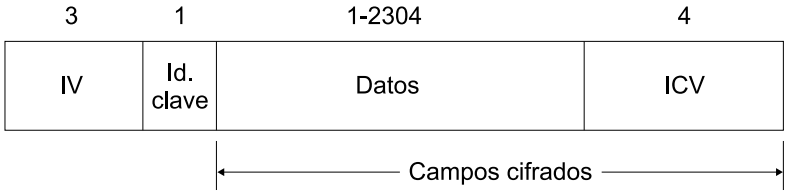
3. Protección de tramas con WEP

La primera versión del estándar IEEE 802.11 definía un mecanismo de seguridad para proteger las tramas enviadas vía radio: se trata del protocolo *wired equivalent privacy* (WEP). Como su nombre indica, este protocolo tiene por objetivo principal conseguir una privacidad de los datos transmitidos ante los simples curiosos que se encuentren por los alrededores, que sea parecida a la de las redes con cable. Del mismo modo que en un lugar donde hay una red con hilo los curiosos no pueden ver nada de la comunicación a menos que dispongan de algún medio para interceptar el cable, en una red inalámbrica con WEP tampoco pueden ver los datos que se transmiten. El método para lograr esta **privacidad** es cifrar los datos. Además del servicio de privacidad, WEP proporciona también el servicio de **integridad** a través de un código de integridad de los datos.

El AP puede tener configuradas hasta 4 claves secretas WEP. Esto permite, por ejemplo, utilizar claves diferentes con grupos de estaciones diferentes, o ir cambiando periódicamente la clave, pero casi siempre únicamente se usa una sola clave WEP.

Cada trama WEP se cifra con una clave de cifrado independiente. Así se dificulta, entre otras cosas, que un atacante pueda detectar datos repetidos. La clave de cifrado utilizada en una trama concreta se obtiene a partir de la clave WEP en uso más un vector de inicialización diferente para cada trama. El valor de este vector de inicialización se tiene que incluir en la propia trama para que el receptor sepa cómo descifrarla.

Figura 4. Datos de una trama WEP



Una trama de datos cifrada con WEP tiene el mismo formato que una trama de datos normal, pero la parte correspondiente a los datos se estructura en cuatro campos.

- El primer campo es el vector de inicialización (IV) utilizado para cifrar la trama.

- El segundo campo es el identificador de clave, que sirve para indicar qué clave WEP de las 4 que puede tener configuradas el AP se ha utilizado en el cifrado.

- En el tercer campo están los datos protegidos.

- El cuarto campo es el *integrity check value* (ICV), que es un CRC de 32 bits calculado sobre el tercer campo (antes de cifrar).

El tercer y cuarto campo, es decir, los datos protegidos y la ICV, se transmiten cifrados.

La inclusión del ICV permite comprobar que la trama cifrada no ha sido manipulada. Si un atacante que no conoce la clave quiere modificar los datos de una trama WEP enviada o inyectar una trama WEP con los datos inventados, muy probablemente cuando el receptor la descifre verá que el ICV no concuerda. De todos modos, un CRC no tiene las propiedades de seguridad que puede tener un código de integridad criptográfico (por ejemplo, basado en funciones *hash*), y por otro lado el resto de campos de la trama no están protegidos, cosa que permite que sí puedan ser manipulados. Otra vez, el criterio de la simplicidad prevaleció sobre la seguridad en el diseño del protocolo (un CRC es mucho más fácil de calcular que un *hash*).

3.1. El cifrado WEP

El algoritmo criptográfico utilizado para cifrar las tramas WEP es una cifra de flujo, concretamente el **RC4** ("Ron's Code 4"), diseñado por Ronald Rivest. Fue escogido por su simplicidad y por el nivel de seguridad que proporciona en relación con la poca complejidad de los cálculos que requiere.

Este criterio era especialmente importante teniendo en cuenta que la mayoría de dispositivos Wi-Fi pueden ser dispositivos móviles en los que un bajo consumo de energía juega un papel relevante. Si se hubiera escogido un algoritmo más sofisticado, que comportara más potencia de cálculo para cifrar y descifrar los mismos datos, y por lo tanto consumiera más energía, la autonomía de las baterías de los dispositivos móviles se podría ver reducida sensiblemente.

La longitud de las claves RC4 en general no es fija: puede haber claves RC4 de hasta 2.048 bits (aunque una clave de cifrado tan larga no tiene mucho sentido para un cifrado simétrico).

El protocolo WEP prevé principalmente el uso de dos longitudes de clave de cifrado RC4: claves de 64 bits o claves de 128 bits.

En las claves de cifrado que se utilizan en un BSS para cifrar cada trama WEP hay una parte variable y una parte fija:

- La parte variable son los primeros 24 bits de la clave, y se conocen como **vector de inicialización** (IV). El IV es diferente para cada trama que se transmite.

- La parte fija es el resto de la clave: 40 bits si la clave es de 64 en total, o 104 bits si la clave es de 128 en total. Esta parte fija se conoce también como **clave raíz**.

La clave raíz WEP de 40 o 104 bits es, pues, la clave secreta (o las claves secretas si se usan dos, tres o cuatro, aunque la situación más habitual es usar solo una) que tiene configurada el AP, y que se tiene que configurar en las estaciones Wi-Fi que se quieran asociar.

Vector de inicialización

El nombre que se le suele dar a la parte variable de la clave es el de **vector de inicialización**, aunque este concepto está relacionado con las cifras de bloque más que con las de flujo. De hecho, el concepto de **bits de sal** se acercaría más a la función de esta parte variable de la clave.

El algoritmo que sigue una estación cualquiera, incluido el AP, para generar una trama cifrada WEP, es el siguiente:

Valor de la clave raíz

El valor de la clave raíz puede ser cualquier combinación de bits, pero para facilitar la configuración manual de las estaciones es habitual que una clave raíz de 104 bits tenga la forma de cadena de 13 caracteres ASCII.

1) Generar una cadena de 24 bits a utilizar como IV, procurando que sea diferente de los últimos IV generados.

2) Concatenar los 24 bits del IV con la clave WEP raíz para formar la clave RC4 de cifrado de la trama.

3) Calcular el CRC de los datos a proteger. Con esta operación se obtiene el ICV.

4) Concatenar los datos con el ICV y cifrar esta secuencia con el algoritmo RC4 utilizando la clave de cifrado del punto 2. Como se trata de una cifra de flujo, esta operación consiste en obtener tantos bits de texto de cifrado (*keystream*) como tenga la secuencia, y sumarlos uno a uno con los de la secuencia.

5) Llenar la parte de datos de la trama WEP con los campos que la componen: el IV, el identificador de la clave WEP, y el resultado del cifrado obtenido en el punto 4.

La estación que recibe la trama cifrada WEP tiene que realizar los pasos inversos para descifrarla:

1) Leer el campo IV de la trama WEP.

2) Concatenar los 24 bits del IV con la clave raíz indicada por el campo identificador de clave WEP. Así se obtiene la clave RC4 de descifrado de la trama.

3) Descifrar la parte cifrada de la trama con el algoritmo RC4 utilizando la clave de descifrado del punto 2. Como antes, el descifrado consiste en sumar bit a bit los datos cifrados con el texto de cifrado (*keystream*) generado a partir de la clave.

4) Calcular el CRC de los datos descifrados y comprobar que coincide con el campo ICV también acabado de descifrar.

Suma bit a bit

Recordad que la suma bit a bit, que coincide con la operación lógica XOR (*OR* exclusiva), es autocomplementaria y, por lo tanto, el cifrado (suma) y el descifrado (resta) se hacen igual.

El objetivo último de incluir un IV es tener una clave de cifrado diferente por cada trama, y para lo cual cada IV tendría que ser diferente, o al menos que no se repitieran hasta después de un número muy grande de tramas generadas. Las implementaciones Wi-Fi normalmente siguen una de estas dos técnicas para conseguirlo:

IV diferentes

Como máximo, se pueden generar 2^{24} IV diferentes, es decir, unos 16 millones.

- Generar cada IV con un generador de números pseudoaleatorios.
- Generar el primer IV como un número aleatorio de 24 bits y obtener los siguientes sumando 1 cada vez al anterior. Este modo de generar los IV se denomina **modo contador**.

La opción del algoritmo RC4 para el cifrado WEP obedecía, como hemos dicho antes, a la simplicidad de su implementación. A pesar de que era un algoritmo que se consideraba razonablemente seguro, la manera en que se diseñó su uso en el protocolo WEP, especialmente con la introducción de los vectores de inicialización, hace que presente algunas vulnerabilidades importantes.

Para comprender las implicaciones que tienen estas vulnerabilidades, antes veremos las características generales del algoritmo RC4.

3.2. El algoritmo RC4

La simplicidad del algoritmo RC4 viene dada, por un lado, por las operaciones en que se basa, y por el otro, por la poca memoria que requiere para guardar la información de estado del cifrado:

- La única operación aritmética que se necesita para implementar el algoritmo es la suma módulo 256 o, lo que es lo mismo, la suma de 8 bits ignorando el arrastre (*carry*) generado. Esta es una operación sencillísima de implementar en hardware, y muy rápida de calcular. El algoritmo también usa la operación de intercambio (*swap*) de elementos de un vector, pero esta no es una operación aritmética sino simplemente de movimiento de datos en memoria.

- La información de estado con que trabaja el algoritmo es un vector de 256 elementos de 8 bits, más dos contadores o índices también de 8 bits cada uno. En total se necesitan 258 bytes de memoria para guardar esta información de estado (aparte del espacio que ocupe la clave, que solo se necesita en la fase inicial del algoritmo).

Para la descripción del algoritmo, usaremos la notación siguiente:

- $K[0]$, $K[1]$, $K[2]$, etc., son el primer, segundo, tercer, etc., bytes de la clave de cifrado. Si la clave es, por ejemplo, de 128 bits, los elementos que la forman son $K[0]$ hasta $K[15]$.

- $S[0]$, $S[1]$, $S[2]$, etc., son el primer, segundo, tercer, etc., bytes del texto de cifrado (*keystream*) generado.

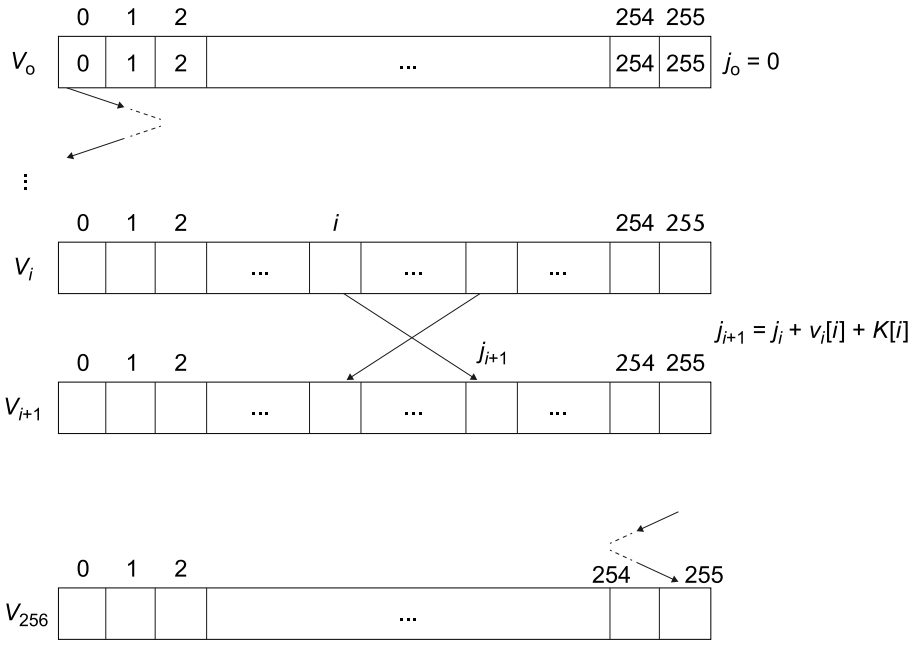
- $V[0]$, ..., $V[255]$ son los elementos del vector de estado del algoritmo.

- i, j son los dos contadores internos con que trabaja el algoritmo.

- El símbolo de suma, +, representa la suma módulo 256, o sea, la suma de 8 bits.

El funcionamiento del algoritmo se divide en dos fases. La primera es el *key schedule algorithm* (KSA) o programación de la clave, y la segunda es el *pseudo-random generation algorithm* (PRGA) o generación del texto de cifrado propiamente dicho.

Figura 5. Esquema del algoritmo KSA del cifrado RC4



El objetivo del algoritmo KSA es obtener, a partir del valor de la clave *K*, un vector de estado *V* que sirva para que el algoritmo PRGA empiece a generar el texto de cifrado *S*. Los pasos que sigue el KSA son estos:

1) El vector *V* parte de un estado inicial (V_0) en que el elemento $V_0[0]$ tiene el valor 0, el elemento $V_0[1]$ tiene el valor 1, ... y el elemento $V_0[255]$ tiene el valor 255. Los contadores *i* y *j* valen 0.

2) Si la clave tiene una longitud de *L* bytes, se concatena consigo misma tantas veces como haga falta hasta que ocupe 256 bytes. (De hecho, no hay que ocupar físicamente estos bytes de memoria; basta con sustituir los accesos a *K*[*i*] por *K*[*i* mod *L*].)

3) Se repite 256 veces la secuencia siguiente:

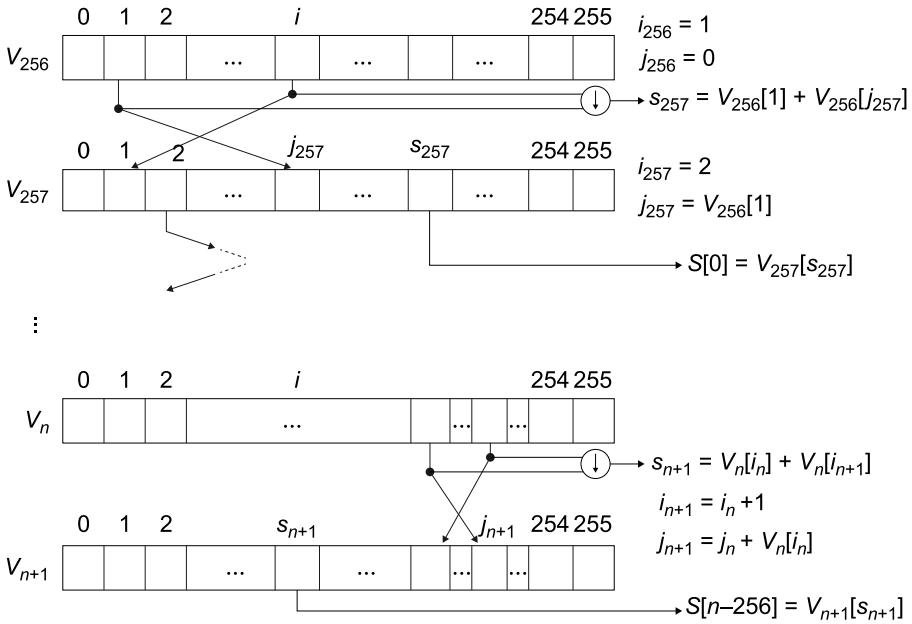
a) Al índice *j* se le suma (módulo 256) $V[i]+K[i]$.

b) Se intercambian los elementos *V*[*i*] y *V*[*j*] del vector de estado.

c) Al índice *i* se le suma 1.

Al final de estos pasos tenemos un vector de estado V_{256} que contiene una permutación aparentemente aleatoria de los elementos 0,...,255.

Figura 6. Esquema del algoritmo PRGA del cifrado RC4



Una vez obtenido el vector V_{256}, que es el vector de estado inicial para empezar la generación del texto de cifrado, se aplica el algoritmo PRGA. En este algoritmo, primero se inicializa el índice *i* a 1 y el índice *j* a 0. A continuación, para cada byte del texto de cifrado *S* que se quiera obtener, se realiza una iteración que consta de los pasos siguientes:

1) Al índice *j* se le suma (módulo 256) $V[i]$.

2) Se intercambian los elementos $V[i]$ y $V[j]$ del vector de estado.

3) Se calcula el índice s como la suma de los dos elementos intercambiados ($s = V[i] + V[j]$).

4) Al índice *i* se le suma (módulo 256) 1.

5) El resultado obtenido en esta iteración, es decir, el siguiente byte del texto de cifrado *S*, es igual al elemento $V[s]$.

4. Vulnerabilidades del protocolo WEP

Como ya hemos dicho antes, el protocolo WEP presenta una serie de vulnerabilidades, algunas de las cuales son independientes de la elección del RC4 como algoritmo de cifrado, y otras que están directamente relacionadas con la manera en que se usa este algoritmo.

4.1. Vulnerabilidades no relacionadas con el algoritmo RC4

Este conjunto de vulnerabilidades es consecuencia de ciertas decisiones de diseño del protocolo WEP independientes del uso del algoritmo RC4.

4.1.1. Inyección de tramas

Un atacante que capture una trama WEP correspondiente a una determinada asociación puede retransmitirla tantas veces como quiera y, si la asociación continúa existiendo, el receptor dará la trama por válida. Si la asociación ya no existe, el atacante puede cambiar las direcciones de la estación transmisora y/o receptora por las de otras estaciones que sí estén asociadas, y el nuevo receptor también dará la trama por válida.

Esto es así, por un lado, porque ni el nivel de enlace IEEE 802.11 ni el protocolo WEP prevén nada para detectar tramas duplicadas. Las tramas WEP inyectadas por el atacante tendrán el IV repetido, pero esto no es problema del receptor. Aunque el uso de los IV tenga por objetivo que las tramas cifradas sean siempre diferentes, nada impide a una estación enviar tramas idénticas cifradas con el mismo IV. Y las estaciones receptoras no suelen comprobar si les llegan tramas con IV repetido, porque esto implicaría tener que recordar los IV de las últimas tramas recibidas y comparar cada trama nueva que llegue, cosa que normalmente no hacen.

Y por otro lado, como los campos de la cabecera MAC no están protegidos por el código de integridad ICV, no hay ningún problema al cambiar las direcciones de esta cabecera.

4.1.2. Falsificación de la autenticación

La falsificación de la autenticación solo tiene sentido en el método de clave compartida, porque en el método de autenticación de sistema abierto no hay nada que falsificar.

Si un atacante captura las tramas intercambiadas durante una autenticación de clave compartida entre una estación y un AP, puede autenticarse con éxito ante el mismo AP sin necesidad de conocer la clave WEP correspondiente.

En el proceso de autenticación de clave compartida se intercambian 4 tramas: petición de autenticación, reto, respuesta y resultado. La tercera trama está cifrada con WEP, pero el atacante sabe cuál tiene que ser su contenido descifrado porque en la segunda trama está el reto en claro. Por lo tanto, si al contenido cifrado de la tercera trama le resta bit a bit (es decir, le suma) el contenido descifrado, obtiene el texto de cifrado (*keystream*) que se genera con la clave WEP y el IV de la tercera trama.

Entonces, el atacante solo tiene que enviar una petición de autenticación al AP, recibir el reto y construir una trama de respuesta con el IV capturado previamente y la suma bit a bit del reto más el *keystream* calculado. El AP verá que es una respuesta correctamente cifrada porque al descifrarla obtendrá el reto de la segunda trama, y por lo tanto dará por autenticada la estación del atacante.

La captura de las tramas de autenticación, en general, permite obtener una cierta cantidad de *keystream* correspondiente a un determinado IV. En la trama de respuesta hay 140 bytes cifrados de los cuales se conoce el valor descifrado (los 128 del reto más los otros campos), y por lo tanto se obtienen 140 bytes de *keystream*. Esto puede ser útil para generar otras tramas cifradas aparte de la de respuesta a la autenticación. Y además, hay otros ataques que permiten calcular más bytes de *keystream* por si hay que falsificar tramas más largas.

4.1.3. Descifrado de tramas mediante la comprobación de integridad o "ataque *chopchop*"

Un atacante que haya capturado una trama WEP puede descifrar los n últimos bytes de datos cifrados, sin necesidad de conocer la clave de cifrado, mediante el envío al AP de una media de $128 \times n$ tramas.

El código de integridad ICV que incorporan las tramas WEP se calcula con el algoritmo CRC-32, que es un método muy bueno para detectar errores de transmisión pero no es un algoritmo criptográfico. Como el CRC-32 está basado en operaciones aritméticas lineales, como son las divisiones de polinomios módulo 2, es posible hacer cálculos inversos.

A cada secuencia de bits le corresponde un polinomio binario P cuyos coeficientes son los bits de la secuencia. Su CRC es otra secuencia correspondiente al polinomio R tal que la concatenación $P \| R$, que denominaremos X, es múltiple del polinomio generador G, en este caso el polinomio CRC-32. Por lo tanto, para comprobar que el CRC es correcto, solo hay que ver si se cumple

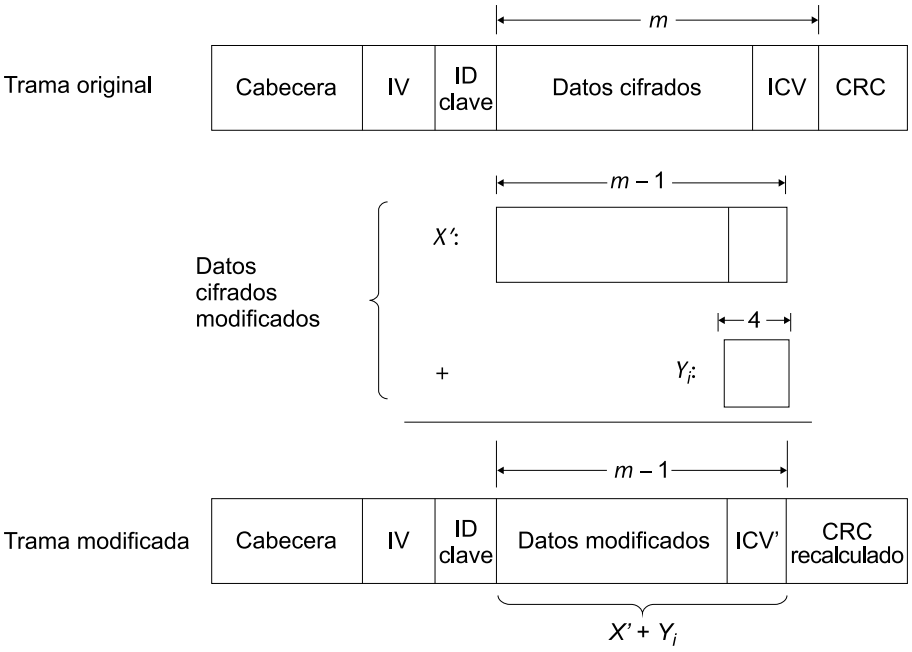
$$X \bmod G = P \| R \bmod G = 0 \qquad 3.1$$

Si consideramos X como secuencia de m bytes $X[0] \| \ldots \| X[m-1]$, y X' es la misma secuencia sin el último byte $(X[0] \| \ldots \| X[m-2])$, es fácil calcular cuál es la secuencia Y que hay que sumar a X' para que continúe siendo divisible por G, es decir, que se cumpla

$$(X' + Y) \bmod G = 0 \qquad 3.2$$

y por lo tanto el CRC continúe siendo correcto. Como resulta que el valor Y solo depende de $X[m-1]$, es decir, del último byte de la secuencia X, podemos calcular los 256 valores $Y_0,\ldots,Y_{255}$ correspondientes a cada uno de los posibles valores del último byte de X.

Figura 7. Modificación de tramas WEP para el ataque *chopchop*



Con esta técnica, el ataque para descifrar una trama WEP consiste en construir nuevas tramas modificadas, hasta 256 en total, siguiendo estos pasos:

- Suprimir el último byte cifrado de la trama original.
- Sumar cada uno de los valores $Y_0,\ldots,Y_{255}$, respectivamente, a los datos cifrados que quedan.

- Recalcular el CRC no cifrado para cada nueva trama.

Entonces, el atacante va enviando estas tramas modificadas al AP para averiguar si el ICV es correcto o no. Cuando la respuesta del AP indique que la trama es correcta, el atacante sabrá cuál es el valor descifrado del último byte cifrado de la trama: el que corresponda al valor Y_i utilizado. Y a partir del valor cifrado y el valor descifrado, haciendo la resta (suma) obtendrá el último byte del *keystream* utilizado para cifrar la trama.

Suma de bits a los datos cifrados

Conceptualmente, la operación que habría que hacer sería descifrar los datos, sumar la secuencia *Y*, y volverlos a cifrar. Pero como el cifrado en realidad también es una suma (de los bits en claro con el *keystream*), el resultado es el mismo si la suma se hace directamente sobre los datos cifrados.

Dado que se pueden probar hasta 256 valores posibles diferentes, el número de tramas que habrá que enviar de media antes de encontrar la buena será de 128.

A partir de la trama buena se puede volver a repetir el ataque para encontrar el penúltimo byte del *keystream*, y así sucesivamente. Con este método se podrían obtener todos los bytes del *keystream* (y por lo tanto descifrar todos los bytes cifrados de la trama original) excepto los 4 primeros, porque los valores *Y* son secuencias de 32 bits. Pero como entre los bytes descifrados estarán los del código ICV, es inmediato deducir los bytes que faltan para que este código sea correcto.

Hay diferentes maneras de hacer que el AP nos diga si el ICV de una trama WEP es correcto o no, entre las cuales podemos mencionar estas dos:

Primeros bytes cifrados de la trama

En la práctica, es posible que el AP descarte las tramas WEP que no tengan una longitud mínima, de forma que hay un punto a partir del cual no se pueden obtener más bytes de *keystream* con el ataque *chopchop*.

- Si disponemos de dos estaciones, desde cada una de ellas podemos hacer una autenticación (falsa, si no conocemos la clave WEP) y una asociación con el AP. Entonces, podemos enviar las tramas desde la primera estación con destino a la segunda a través del AP. Si el ICV de una trama es correcto, el AP la retransmitirá a la segunda estación, y si no, la descartará.

- Otra manera más sencilla es enviar las tramas desde una estación no asociada. Si el ICV de una trama es correcto, el AP responderá con una trama de gestión indicando que la estación no está asociada, y si no, la descartará.

Este ataque que va descabezando byte a byte la trama capturada a medida que la va descifrando se conoce como "ataque *chopchop* de KoreK", o simplemente "ataque *chopchop*". Se basa en la técnica de otro ataque publicado anteriormente, llamado "ataque inductivo de Arbaugh". Este último es de hecho el ataque inverso del *chopchop*: en vez de retroceder byte a byte, va "avanzando" sobre la base de ensayo y error, y así va averiguando los bytes siguientes del

KoreK

KoreK es el pseudónimo del autor que ha publicado varios ataques contra los protocolos Wi-Fi.

keystream. De este modo, con una media de 128 intentos por byte se puede conseguir una longitud arbitraria de *keystream* útil para construir tramas WEP falsificadas más o menos largas sin conocer la clave.

4.1.4. Ataque de fragmentación

Un atacante que haya descubierto n bytes de un *keystream* puede obtener hasta $16 \times n - 60$ bytes de otro *keystream* mediante el envío de hasta 16 tramas fragmentadas al AP.

IEEE 802.11 permite enviar una trama en fragmentos, hasta un máximo de 16. Si una estación envía al AP una trama fragmentada que tenga que ser retransmitida, el AP normalmente recompondrá la trama antes de retransmitirla. Y si los fragmentos están cifrados, la trama recombinada también estará cifrada, posiblemente con un nuevo IV.

Así, un atacante que disponga de n bytes de *keystream* puede construir 16 fragmentos de trama WEP, cada uno con n – 4 bytes de datos más los 4 bytes del ICV, todos ellos cifrados con el mismo IV y *keystream*. Si envía estos fragmentos al AP para que los retransmita, la trama resultante tendrá $16 \times (n-4)$ bytes de datos y 4 bytes de ICV cifrados (siempre que no sobrepase la longitud máxima de los datos de una trama), en total $16 \times n - 60$ bytes de los cuales el valor descifrado será conocido. Por lo tanto, el atacante podrá obtener esta cantidad de bytes de *keystream*.

Si el *keystream* recuperado todavía no llega a la longitud necesaria para cifrar una trama larga, se puede volver a repetir este ataque hasta obtener la longitud máxima posible.

4.2. Vulnerabilidades relacionadas con el algoritmo RC4

Este conjunto de vulnerabilidades es consecuencia del método con que se utiliza el algoritmo RC4 en el protocolo WEP.

4.2.1. El ataque FMS

Los criptoanalistas Scott Fluhrer, Itsik Mantin y Adi Shamir, en un artículo publicado en el año 2001, detallaron las bases teóricas de un ataque que permitía recuperar una clave raíz WEP a partir de las tramas cifradas si se disponía de un número suficiente de estas tramas. Otro equipo de investigadores publicó en el año 2004 los resultados de la primera implementación de este ataque contra una red Wi-Fi real.

Con el ataque FMS, un atacante que conozca el primer byte de *keystream* (S[0]) de aproximadamente entre 4 y 9 millones de tramas WEP cifradas con la misma clave raíz puede recuperar el valor de la clave con una probabilidad de éxito del 50%.

Ataque FMS

El nombre con que se conoce este ataque proviene de las iniciales de los apellidos de sus descubridores.

Este ataque se basa en la observación del funcionamiento del algoritmo RC4. Tal como se utiliza en el protocolo WEP, los 3 primeros bytes de la clave RC4 son siempre conocidos, puesto que forman el vector de inicialización (IV) que se envía en claro prefijado a los datos cifrados. Para cada trama capturada con su IV correspondiente, pues, el atacante dispone de los elementos $K[0]$, $K[1]$ y $K[2]$ de la clave RC4. Con esta información puede reconstruir las 3 primeras iteraciones del algoritmo KSA y obtener el vector de estado V_3 y el valor del índice j_3.

Primer byte de *keystream*

El caso más habitual es que los datos cifrados de una trama no fragmentada, o del primer fragmento de una trama fragmentada, empiecen con el primer byte de la cabecera LLC igual a AA (hexadecimal). Por lo tanto, conocido el primer byte cifrado y el primer byte descifrado, también se conoce el primer byte de *keystream*.

El algoritmo KSA se completa con 253 iteraciones más hasta llegar a obtener el vector V_{256}, a partir del cual se calcula con el algoritmo PRGA el primer byte de *keystream* $S[0]$:

$$S[0] = V_{257}[s_{257}] = V_{257}[V_{256}[1] + V_{256}[j_{257}]] = V_{257}[V_{256}[1] + V_{256}[V_{256}[1]]] \qquad 3.3$$

El contenido del vector V_{256} será en general desconocido, pero se puede seleccionar un subconjunto de las tramas que tengan ciertas propiedades que favorecen el ataque.

Para realizar el ataque, se analiza cada trama y se comprueba si con su IV se cumplen las llamadas **condiciones de resolución**:

1) $V_3[1] < 3$

2) $V_3[1] + V_3[V_3[1]] = 3$

Si la trama no cumple estas condiciones, se descarta y se pasa a la siguiente.

Condiciones de resolución

Estadísticamente, una de cada 21.675 tramas cumplirá las condiciones de resolución para $n = 3$. Si crece n, también crece el número de tramas que cumplen las condiciones.

A continuación, el ataque consiste en ver qué pasaría si se diera la feliz coincidencia de que tres elementos determinados del vector de estado no se intercambiaran con ningún otro en las iteraciones siguientes del KSA. Más concretamente, el caso que se considera es que se den estas tres condiciones a la vez:

1) Que el elemento $V_3[1]$ no cambie de lugar entre las iteraciones 4 y 256.

2) Que el elemento $V_3[V_3[1]]$ no cambie de lugar entre las iteraciones 4 y 256.

3) Que el elemento $V_3[j_4]$ (desconocido, porque si no tenemos el cuarto byte de la clave, $K[3]$, no sabemos qué valor tiene j_4), que en la iteración 4 pasará a ser $V_4[3]$, no cambie de lugar entre las iteraciones 5 y 256.

Como en cada iteración del KSA se intercambian $V[i]$ y $V[j]$, para que se no derroche esta serie de coincidencias el índice j no tendría que pasar por ninguno de los valores "prohibidos" 1, $V_3[1]$ y 3 (este último, a partir de la iteración 5). Con el índice i no hay problema porque, a partir de la iteración 4, valdrá como mínimo 4 y además sabemos, por la primera condición de resolución, que $V_3[1] < 3$. En cuanto a j, se puede calcular que la probabilidad de que este índice no tome ninguno de los 3 valores prohibidos en ninguna de las 253 iteraciones siguientes es $((256-3)/256)^{253}$, aproximadamente un 5%.

Entonces sabemos que, con una probabilidad aproximada del 5%, estos tres elementos del vector de estado no se habrán movido, y por lo tanto $V_{256}[1] = V_3[1]$ y $V_{256}[V_{256}[1]] = V_3[V_3[1]]$. En la primera iteración del PRGA, se intercambian precisamente los elementos $V_{256}[1]$ y $V_{256}[V_{256}[1]]$, y se calcula el índice s_{257} como la suma de estos dos valores. Por la segunda condición de resolución, esta suma es igual a 3, y esto quiere decir que el primer byte de *keystream* generado será igual a $V_{257}[3]$. Si se cumple nuestra hipótesis de inmovilidad, este elemento no se habrá movido entre las iteraciones 5 y 256, y además las condiciones de resolución garantizan que tampoco se habrá movido en la iteración 257.

En definitiva, si no se han movido los elementos mencionados tendremos que:

$$S[0] = V_{257}[3] = V_4[3] = V_3[j_4] = V_3[j_3 + V_3[3] + K[3]] \qquad 3.4$$

A partir de aquí, como j_3 y el vector V_3 son conocidos y la hipótesis inicial del ataque es que $S[0]$ también es conocido, es inmediato encontrar el valor $K[3] = V_3^{-1}[S[0]] - j_3 - V_3[3]$ (haciendo las operaciones en módulo 256). Este será el valor correcto del cuarto byte de la clave siempre que sea verdad que los tres elementos en cuestión no se han movido de lugar. Si no, el resultado obtenido con esta fórmula será un valor que podemos considerar aleatorio.

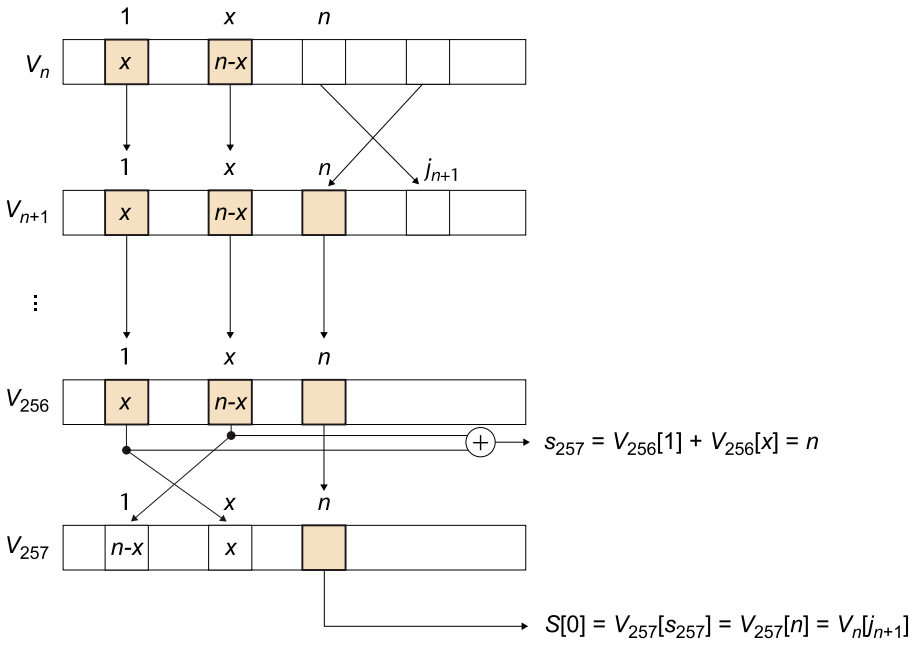
Es decir, si calculamos el posible valor de $K[3]$ a partir de un número de tramas suficiente para que la distribución de los valores incorrectos sea uniforme, encontraremos que aproximadamente 15 de cada 270 tramas darán un mismo valor, y las otras 255 darán valores diferentes entre sí. Empíricamente se puede comprobar que a partir de unas 60 tramas analizadas ya hay un valor que

Inversión de la transformación *V*[*x*]

Dado $y = V[x]$, siempre es posible encontrar el valor único $x = V^{-1}[y]$ porque V es una permutación de los elementos 0, ..., 255: cada elemento aparece una y solo una vez en la permutación.

destaca sobre los otros porque se repite al menos 3 o 4 veces. Este valor candidato que tiene mayoría de "votos" es el que en principio se puede considerar como correcto.

Figura 8. Elementos del vector de estado que no se tendrían que modificar para que sea cierta la hipótesis del ataque FMS



Una vez determinado el valor candidato a ser el cuarto byte de la clave, $K[3]$, se puede volver a empezar el ataque para averiguar el byte siguiente, $K[4]$. En general, las condiciones de resolución para intentar obtener el valor del byte $K[n]$ son:

1) $V_n[1] < n$

Probabilidad de no intercambio

A medida que avanza n también crece la probabilidad de que los elementos no sean intercambiados porque quedan menos iteraciones hasta el final del KSA, pero la variación no es muy grande: desde el 5,07% para $n = 3$ hasta el 5,84% para $n = 15$.

2) $V_n[1] + V_n[V_n[1]] = n$

y el valor se calcula con la fórmula

$$K[n] = V_n^{-1}[S[0]] - j_n - V_n[n] \qquad 3.5$$

En la obtención de cada byte de la clave se pueden encontrar falsos positivos. Esto pasará si entre las tramas que no cumplen la condición de inmovilidad hay varias que coinciden en dar un mismo valor (incorrecto) de $K[n]$, y superan en número a las tramas que dan el valor correcto, de forma que el valor falso tiene más votos que el bueno. Cuando un byte averiguado $K[n]$ es inco-

rrecto, todos los bytes siguientes $K[m]$ con $m > n$ estarán mal calculados porque el algoritmo KSA para obtener el vector V_m se estará aplicando con valores erróneos.

Obtención del último byte de la clave

Si hay muchas tramas por procesar, en vez de obtener el último byte de la clave $K[15]$ por el sistema de votación, puede ser más eficiente obtener solo hasta el penúltimo byte y hacer las comprobaciones por "fuerza bruta" con cada uno de los 256 posibles valores del último byte. En algunos casos, incluso se puede aplicar la fuerza bruta a los dos últimos bytes de la clave.

La comprobación para saber si los bytes de la clave son correctos solo se puede hacer cuando se han averiguado todos, desde $K[3]$ hasta $K[15]$ en el caso de una clave WEP-104. Entonces se pueden utilizar unos cuantos IV para ver si la clave calculada da el byte de *keystream* $S[0]$ correspondiente a cada uno de los IV. Si no es así, hay que volver atrás y rehacer los cálculos. Por ejemplo, se puede mirar cuál de los bytes de la clave ha ganado por una mayoría más estrecha, sustituirlo por el segundo valor que haya tenido más votos, y recalcular todos los bytes posteriores. Y si la comprobación tampoco es satisfactoria, ir repitiendo estas sustituciones hasta encontrar el valor correcto o hasta que se haya ultrapasado un número máximo de intentos.

El hecho de que el modo contador requiera más tramas viene dado por la distribución de los IV que cumplen las condiciones de resolución. En el modo aleatorio estarán repartidos uniformemente entre las tramas capturadas, mientras que en el modo contador estarán concentrados en series de tramas consecutivas o muy próximas. Si el atacante tiene la suerte de topar pronto con una de estas series, puede obtener rápidamente los IV necesarios, pero si no, necesitará de media muchas más tramas.

Éxito del ataque FMS

Un criterio para considerar que el ataque no ha tenido éxito es que el método de prueba y error para encontrar la clave buena no dé ningún resultado al cabo de 2 o 3 minutos, con la potencia de cálculo media de los ordenadores actuales.

4.2.2. El conjunto de ataques KoreK

Se conoce con el nombre de "ataques KoreK" una serie de ataques al protocolo WEP que explotan determinadas correlaciones entre la clave raíz y los primeros bytes de texto de cifrado o *keystream*. En el año 2004, una persona que usaba el pseudónimo "KoreK" publicó una herramienta que integraba todos estos ataques, que en total eran 17. Algunos de ellos ya se conocían previamente, como el caso del ataque FMS, y los otros fueron descubiertos por KoreK.

Con los ataques KoreK, un atacante que conozca los dos primeros bytes de *keystream* ($S[0]$, $S[1]$) de aproximadamente entre 150.000 y 700.000 tramas WEP cifradas con la misma clave raíz puede recuperar el valor de la clave con una probabilidad de éxito del 50%.

Segundo byte de *keystream*

Así como en la gran mayoría de casos el primer byte de datos cifrados de una trama WEP es el primer byte de la cabecera LLC, el segundo byte cifrado será el segundo byte de la misma cabecera, que también es igual a AA (hexadecimal), y a partir de este valor se puede saber el segundo byte de *keystream* $S[1]$.

Los ataques KoreK se pueden dividir en tres grupos:

- Ataques que permiten averiguar $K[n]$ a partir de $K[0],...,K[n-1]$ y $S[0]$. El ataque FMS pertenece a este grupo.
- Ataques que permiten averiguar $K[n]$ a partir de $K[0], ..., K[n-1]$, $S[0]$ y $S[1]$.
- Ataques "negativos" que, si V_n cumple ciertas condiciones y $S[0]$ toma ciertos valores, permiten descartar determinados valores de $K[n]$.

Muchos de los ataques se basan, igual que el FMS, en la probabilidad de que determinados elementos del vector de estado no sean intercambiados a partir de la iteración n del algoritmo KSA. Cada uno de los ataques individuales tiene sus propias condiciones de resolución. La herramienta que se publicó iba comprobando trama por trama si se cumplían las condiciones de algún ataque, y si era así lo llevaba a cabo y obtenía un voto a favor de un candidato a $K[n]$, o un voto en contra en caso de que se tratara de un ataque negativo. Los votos se ponderaban según la probabilidad de éxito del ataque realizado.

El hecho de implementar varios ataques diferentes en paralelo facilita el descubrimiento de la clave con un número menor de tramas analizadas. Experimentalmente se ha encontrado que a partir de 150.000 tramas los ataques KoreK permiten obtener el valor correcto de la clave WEP con una probabilidad de éxito del 50% si los IV están generados aleatoriamente. En cambio, si los IV se generan en modo contador, el número de tramas necesarias crece hasta 700.000 para lograr el mismo 50% de éxito. A partir de 270.000 tramas en modo aleatorio, y 1.700.000 en modo contador, la tasa de éxito es del 90%.

4.2.3. El ataque PTW

En el año 2007 se publicó un nuevo ataque, conocido como PTW, que es una variante mejorada de otro que fue descubierto en el 2005, llamado **ataque Klein**.

Con el ataque PTW, un atacante que conozca los bytes de *keystream* del tercero al decimoquinto (*S*[2], ..., *S*[14]) de aproximadamente 35.000 tramas WEP cifradas con la misma clave raíz puede recuperar el valor de la clave con una probabilidad de éxito del 50%.

Ataque PTW

El nombre con que se conoce este ataque proviene de las iniciales de los apellidos de los autores que lo publicaron: Andrei Pyshkin, Erik Tews y Ralf-Philipp Weinmann.

El ataque Klein se basa en una anomalía de las propiedades estadísticas de la programación de claves RC4. El algoritmo KSA genera un vector de estado aparentemente aleatorio, y como cada elemento puede tener uno de 256 valores posibles, es de esperar que la probabilidad de que un elemento $V[n]$ tenga un determinado valor x sea 1/256, es decir, aproximadamente un 0,4%. Una combinación de elementos del vector, como por ejemplo $V[V[i]+V[j]]+V[j]$, en principio también tendría que ser igual a cualquier valor $0 \leq x \leq 255$ de manera equiprobable. Pero la llamada **correlación de Jenkins** demuestra que hay un valor de esta expresión que es más probable que los otros. Concretamente:

Primeros 15 bytes de *keystream*

Hay algunas tramas, como es el caso de las que contienen paquetes ARP, en que los 15 primeros bytes de datos corresponden a campos de cabeceras con valores constantes. Por lo tanto, si estas tramas están cifradas se pueden obtener 15 bytes de *keystream*.

$$Prob(V[V[i]+V[j]]+V[j]=i)=2/256 \qquad 3.6$$

Así, la combinación de elementos anterior puede tomar el valor *i* con una probabilidad aproximada del 0,8%, en vez del 0,4% que cabría esperar.

Además, la correlación de Jenkins también demuestra que el resto de valores $x \neq i$ son equiprobables, de forma que la probabilidad de cada uno de ellos es $(1-2/256)/255 = 127/32.640$.

Igual que en el ataque FMS, en el ataque Klein inicialmente hay que hacer las 3 primeras iteraciones del algoritmo KSA para obtener V_3, y en la iteración 4 sabemos que el elemento $V_3[j_4]$ pasará a ser $V_4[3]$. En alguna de las iteraciones siguientes el índice *j* puede tomar el valor 3, y entonces este elemento cambiará de lugar. Pero el índice *i* no volverá a valer 3 hasta la iteración 258, ya dentro del algoritmo PRGA, es decir, hasta después de 254 iteraciones.

Si consideramos que las variaciones de *j* tienen un comportamiento aleatorio, la probabilidad de que el índice *j* no tome el valor 3 en ninguna de estas 254 iteraciones es $(255/256)^{254}$, es decir, un 37%. Como el índice *i* tampoco valdrá 3 en ninguna de las 254 iteraciones, esta es la probabilidad de que el elemento $V_4[3]$ no se haya movido de su lugar hasta la iteración 258. En el otro 63% de los casos *j* habrá valido 3 en algún momento y el elemento $V_{258}[3]$ ya no será el mismo que estaba en $V_4[3]$.

En la iteración siguiente, la 259, se obtendrá el tercer byte de *keystream* *S*[2]:

$$S[2] = V_{259}[s_{259}] = V_{259}[V_{258}[i_{258}] + V_{258}[j_{259}]] = V_{259}[V_{258}[3] + V_{258}[j_{259}]] \qquad 3.7$$

Como en esta iteración se habrán intercambiado los elementos de las posiciones 3 y j_{259}, la suma $V_{258}[3] + V_{258}[j_{259}]$ será la misma que $V_{259}[j_{259}] + V_{259}[3]$, y por lo tanto:

$$S[2] = V_{259}[V_{259}[3] + V_{259}[j_{259}]] \qquad 3.8$$

Sumando $V_{259}[j_{259}]$:

$$S[2] + V_{259}[j_{259}] = V_{259}[V_{259}[3] + V_{259}[j_{259}]] + V_{259}[j_{259}] \qquad 3.9$$

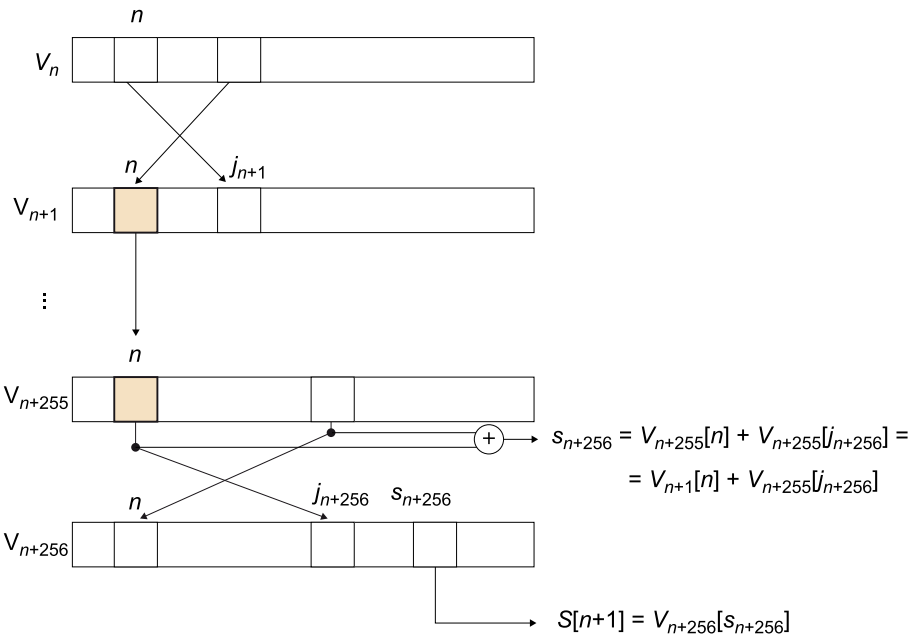
Y por la correlación de Jenkins sabemos que esta expresión tiene más probabilidad de valer 3 que cualquier otro valor. Así, tenemos que hay una probabilidad de 2/256 que se cumpla $S[2] + V_{259}[j_{259}] = 3$ o, puesto que $V_{259}[j_{259}]$ es el elemento que estaba en $V_{258}[3]$ antes del último intercambio, que se cumpla $S[2] + V_{258}[3] = 3$.

Para obtener el valor del byte $K[3]$, el ataque Klein se basa en la probabilidad de que la hipótesis $S[2] + V_4[3] = 3$ sea cierta. Entonces, como:

$$V_4[3] = V_3[j_4] = V_3[j_3 + V_3[3] + K[3]] \qquad 3.10$$

despejando $K[3]$ tenemos que el valor candidato es $K[3] = V_3^{-1}[3 - S[2]] - j_3 - V_3[3]$ (con todas las operaciones en módulo 256).

Figura 9. Elemento del vector de estado que no se tendría que modificar para que fuera cierta la hipótesis del ataque Klein



Como en el ataque FMS, para cada trama se obtiene un voto de un candidato a $K[3]$, y después de analizar todas las tramas disponibles se determina el candidato más probable. Una vez decidido el valor de $K[3]$ se repite el proceso para el resto de bytes $K[n]$. La fórmula general para obtener cada byte es:

$$K[n] = V_n^{-1}[n - S[n-1]] - j_n - V_n[n] \qquad 3.11$$

Vamos a analizar ahora la probabilidad de que los valores candidatos encontrados sean correctos, centrándonos en el caso $n = 3$. Hay dos combinaciones que hacen que la hipótesis del ataque Klein ($S[2] + V_4[3] = 3$) sea cierta:

1) Que se den simultáneamente estas dos condiciones:

- El elemento $V_{258}[3]$ es el mismo que $V_4[3]$. Esto pasa, como hemos visto antes, con una probabilidad del 37%.

- Se cumple $S[2] + V_{258}[3] = 3$. Según la correlación de Jenkins, la probabilidad es 2/256. La probabilidad total de esta primera combinación es

$$0{,}37 \times 2/256 = 0{,}74/256. \qquad 3.12$$

2) Que se den simultáneamente estas otras condiciones:

- El elemento $V_{258}[3]$ es diferente de $V_4[3]$. La probabilidad es del 63%.

- Coincide que $V_4[3]$ es tal que $S[2] + V_4[3] = 3$. Según la correlación de Jenkins, teniendo en cuenta que ahora no estamos en el caso más probable, la probabilidad es 127/32.640. La probabilidad total de esta otra combinación es $0{,}63 \times 127/32.640 = 0{,}63/256$.

Como son combinaciones disjuntas, podemos sumar las probabilidades parciales y tenemos que la probabilidad total de que sea cierta la hipótesis es 1,37/256. Dicho de otro modo, la probabilidad de que se cumpla la hipótesis del ataque Klein es 1,37 veces la "normal". Este resultado no es muy espectacular si lo comparamos, por ejemplo, con la hipótesis del ataque FMS, que se cumplía con una probabilidad aproximada del 5% (unas 14 veces la normal).

Sin embargo, la gran diferencia entre el ataque FMS y el ataque Klein es que en este último no hay condiciones de resolución y todas las tramas se pueden usar para obtener un valor candidato para cada $K[n]$. Así, aunque la probabilidad de cumplirse la hipótesis del ataque sea 10 veces menor, en el ataque Klein se necesitan muchas menos tramas para recuperar la clave WEP.

Probabilidad de la hipótesis del ataque Klein

Mientras que la hipótesis del ataque FMS tiene una probabilidad que varía con *n*, en el ataque Klein es la misma para cualquier *n* porque el número de iteraciones consideradas es constante (254).

Efecto de la correlación de Jenkins

Observad que sin la correlación de Jenkins la probabilidad total de la hipótesis sería $0{,}37 \times 1/256 + 0{,}63 \times 1/256 = 1/256$. Es decir, el caso $S[2] + V_4[3] = 3$ se daría con la misma probabilidad que cualquier otro.

Experimentalmente se ha comprobado que a partir de 43.000 tramas el ataque Klein permite obtener el valor correcto de la clave WEP con una probabilidad de éxito del 50%. Además, a partir de 60.000 tramas la probabilidad de éxito ya es del 90%. Por otro lado, el número de tramas necesarias en el ataque Klein es independiente de cómo se generan los IV (modo aleatorio o modo contador), puesto que no hay condiciones de resolución y se aprovechan todas las tramas.

El ataque PTW propiamente dicho consiste en añadir una serie de mejoras al ataque Klein que permiten aumentar la eficiencia. Estos son algunos de los cambios introducidos en el ataque PTW:

1) Mientras que en el ataque Klein, una vez determinado el valor de $K[3]$, se busca el de $K[4]$, el de $K[5]$, etc., en el ataque PTW se buscan las sumas acumuladas de los bytes de la clave raíz: $\sigma_3 = K[3], \sigma_4 = K[3] + K[4]$, $\sigma_5 = K[3] + K[4] + K[5]$, y así sucesivamente. Estas sumas se pueden obtener si, en vez de trabajar por ejemplo con $j_5 = j_4 + V_4[4] + K[4]$, se continúa desarrollando la expresión y se trabaja con $j_5 = j_3 + V_3[3] + V_4[4] + K[3] + K[4]$.

Con esta modificación no se reduce el número de tramas necesarias para encontrar la clave y además las probabilidades de que se cumplan las hipótesis sobre las sumas de bytes son inferiores a las de las hipótesis de Klein. Pero trabajar con las sumas tiene la ventaja de hacer mucho más rápida la busca de claves alternativas cuando se descubre que una clave candidata es incorrecta. Esto se debe al hecho de que, a diferencia de los bytes $K[i]$, que se tienen que calcular secuencialmente porque cada uno depende de los anteriores, cada suma σ_i se puede obtener de manera independiente de las otras.

A partir de las sumas σ_3, σ_4, ... σ_{15} es inmediato obtener $K[3]$, $K[4]$, ... $K[15]$ haciendo unas simples restas. Si con las sumas más votadas se obtiene una clave que no es la correcta, se cambia una de las sumas por la siguiente más votada y solo hay que volver a restar las σ_i necesarias para obtener una clave nueva. Esto es mucho más rápido que recalcular los vectores de estado para volver a generar los nuevos valores de los bytes $K[i]$, como se hace en el ataque Klein.

2) Esta rapidez en la obtención de nuevas claves permite implementar algoritmos más exhaustivos y eficientes para encontrar la clave correcta. Por ejemplo, si ordenamos los candidatos a cada σ_i de más a menos votado, podemos seleccionar un número máximo m de candidatos más votados para utilizarlos en las diferentes combinaciones de claves posibles. Trabajando con $m = 2$, puede haber hasta $2^{13} = 8.192$ combinaciones diferentes de claves a probar. Con $m = 3$ el número de combinaciones ya asciende a $3^{13} = 1.594.323$, que puede ser muy elevado si en cada intento se tienen que recalcular los vectores de estado como en el ataque Klein.

En cambio, el ataque PTW permite utilizar valores m mayores, e incluso valores m_i que sean diferentes para cada σ_i y que varíen dinámicamente. La técnica de los números dinámicos de candidatos consiste en poner inicialmente todos los valores m_i a 1, lo cual da una única combinación de bytes de la clave. Si esta clave no es la correcta, se incrementa en 1 el máximo m_i correspondiente a la suma σ_i en que el candidato de la posición $m_i + 1$ tenga menos diferencia de votos respecto al de la posición m_i y se vuelven a probar las combinaciones de claves en que intervenga el nuevo candidato. Si tampoco se encuentra la correcta, se vuelve a incrementar otro m_i, y así sucesivamente.

3) El uso de sumas de bytes σ_n en vez de bytes $K[n]$ introduce un problema que no existe en el ataque Klein. Si en algún byte $K[n]$ se da el caso de que a partir de cierta posición $4 \leq p \leq n$ la suma $K[p]+K[p+1]+\ldots+K[n]+p+(p+1)+\ldots+n$ es igual a 0, es muy probable que el índice $j_p$sea igual a j_{n+1}. Esto significa que en la iteración p del KSA se producirá un intercambio que deshará la hipótesis de trabajo del ataque PTW, y no será aplicable la correlación de Jenkins. Como esta condición es independiente del IV, en este caso todos los valores de la suma σ_n serán más o menos equiprobables y será mucho más difícil acertar el valor correcto. Entonces se dice que $K[n]$ es un **byte fuerte** de la clave.

Una solución consiste en detectar que un byte es fuerte cuando la distribución de los votos de las sumas candidatas se asemeja a una distribución uniforme. Entonces se trata de deducir para cada posible valor de p cuáles son los valores de $K[n]$ que hacen que se cumpla la condición de byte fuerte $\Sigma_p^n K[i]+i=0)$, y considerar los resultados obtenidos como candidatos a $K[n]$. Otras soluciones son probar por fuerza bruta todos los valores de $K[n]$ o, si hay muchos bytes fuertes en la clave y la fuerza bruta no es viable, realizar el ataque Klein en vez del PTW.

La introducción de estas mejoras permite al ataque PTW rebajar el número de tramas necesarias para tener un 50% de probabilidad de éxito hasta 35.000, y hasta 47.000 para un 90% de éxito.

4.3. Herramientas para explotar las vulnerabilidades WEP

Después de publicarse los diferentes ataques contra el protocolo WEP, se desarrollaron herramientas que implementaban estos ataques, muchas de ellas de código abierto o software libre. Una de las primeras fue AirSnort (2001). Luego, la publicación de los ataques FMS y KoreK dio lugar al proyecto Aircrack (2004). A partir de este y otros desarrollos, como por ejemplo la herramienta `wesside` (2004), se creó Aircrack Next Generation o Aircrack-ng (2006). Desde la versión 0.9 (2007), Aircrack-ng implementa el ataque PTW.

Aircrack-ng es de hecho un paquete que incorpora varias herramientas, entre ellas `airmon-ng`, `aireplay-ng`, `airodump-ng` y la propia herramienta `aircrack-ng`.

La herramienta `airmon-ng`

Esta utilidad sirve para poner la tarjeta Wi-Fi en modo monitor, y de esta manera poder capturar tramas enviadas por otras estaciones. El modo monitor es equivalente al modo promiscuo sin la necesidad de establecer una asociación con ninguna otra estación o AP.

La herramienta `aireplay-ng`

Esta herramienta permite inyectar tramas nuevas o previamente capturadas. Así se puede forzar la generación de tramas WEP de respuesta en caso de que no haya estaciones Wi-Fi activas por los alrededores.

La herramienta `aireplay-ng` puede trabajar en diferentes modos, entre los cuales podemos destacar los siguientes:

1) Modo desautenticación. En este modo se generan tramas falsas de desautenticación destinadas a una estación autenticada con el AP. El objetivo es provocar que esta estación inicie una nueva autenticación y, dependiendo del sistema operativo con que trabaje, envíe una petición ARP para averiguar la dirección IP del AP que hace de *router*.

2) Modo autenticación falsa. En este modo se realiza un ataque de falsificación de autenticación como el que hemos visto anteriormente. Esto puede ser necesario para realizar posteriormente otros ataques si no hay ninguna otra estación asociada.

Longitud de los paquetes ARP

En vez de buscar solo paquetes con 40 bytes de datos cifrados, `aireplay-ng` busca paquetes que tengan 40 o 58 bytes. El motivo es que si el origen es un ordenador de una red por cable, habrá añadido bytes de relleno hasta llegar a la longitud mínima de los datos de una trama Ethernet (46 bytes).

3) Modo inyección de paquetes ARP. Este es probablemente el modo más efectivo para forzar el envío de tramas WEP y poder capturar así los IV necesarios para obtener la clave. En este modo, la herramienta escucha el medio hasta que detecta una petición ARP.

El paquete ARP estará cifrado, pero es relativamente fácil detectar que una trama contiene una petición ARP. En primer lugar, porque la longitud de los datos cifrados será de 40 bytes: 8 de cabecera LLC, 28 del propio paquete ARP, y 4 de ICV. Y después porque la dirección de destino, que no está cifrada, será la dirección de difusión (*broadcast*).

Cuando se ha capturado la petición ARP, la herramienta *aireplay-ng* empieza a retransmitirla una vez tras otra, aprovechando la vulnerabilidad que ya hemos visto de inyección de tramas. Si la dirección IP solicitada es la del AP, este enviará tantas tramas WEP con paquetes ARP de respuesta como peticiones reciba y cada respuesta estará cifrada con un IV diferente. Si la petición original la había enviado una estación solicitando la dirección de otra estación, el AP retransmitirá la petición, la estación solicitada enviará la respuesta al AP y el AP retransmitirá la respuesta al solicitante original, de forma que por cada trama inyectada se generarán 3 nuevas, cada una con su propio IV.

Este ataque suele ser muy productivo, porque normalmente los filtros de paquetes dejan pasar sin restricciones el tráfico del protocolo ARP y los sistemas de detección de intrusiones no toman ninguna acción especial con este tipo de paquetes. La estación que había generado la petición original puede recibir muchísimas respuestas, pero lo más habitual es que no les haga caso. Con esta técnica, pues, es fácil obtener unos cuantos centenares de IV por segundo, y conseguir en menos de un minuto los necesarios para hacer un ataque PTW.

La herramienta `airodump-ng`

Esta herramienta es el equivalente de la utilidad *tcpdump* de las redes por cable: captura las tramas que detecta y permite guardarlas en un fichero. Las tramas pueden provenir de un ataque activo provocado con `aireplay-ng`, o bien ser capturadas en un ataque pasivo simplemente escuchando el medio. En este último caso es muy probable que la mayoría de tramas capturadas correspondan a paquetes IP.

Cuando guarda las tramas capturadas en fichero, `airodump-ng` tiene la opción de guardar solo la información útil para la herramienta *aircrack-ng*: el IV y los primeros bytes de *keystream* de cada trama.

Figura 10. Estructura de un paquete ARP en una trama Wi-Fi

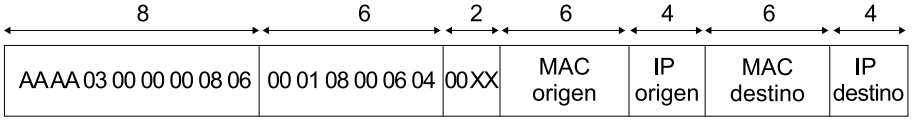
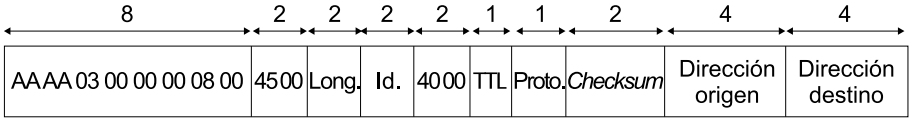


Figura 11. Estructura de una cabecera IPv4 en una trama Wi-Fi



El *keystream* se obtiene deduciendo el tipo de trama de que se trata. Si por su longitud se infiere que contiene un paquete ARP, se pueden conocer como mínimo los primeros 22 bytes de datos descifrados, y restando los datos cifrados se obtiene el *keystream*:

- Los primeros 8 bytes son la cabecera LLC/SNAP, con valor fijo: en hexadecimal, AA:AA (puntos de acceso origen y destino), 03 (código de control), 00:00:00 (código de organización) y 08:06 (tipo Ethernet: ARP).

- Los 6 bytes siguientes son la cabecera ARP, también con valor fijo: en hexadecimal, 00:01 (tipo de protocolo: Ethernet), 08:00 (protocolo de red: IPv4), 06 (longitud de dirección MAC), 04 (longitud de dirección IP).

- Los 2 bytes siguientes son el código de operación ARP: 00:01 (petición) o 00:02 (respuesta).

- Los 6 bytes siguientes son la dirección MAC de origen, que tiene que coincidir con la que está en la cabecera MAC 802.11 (no cifrada).

LLC/SNAP

LLC es la sigla de *logical link control*, y SNAP es la sigla de *sub-network access protocol*.

A continuación está la dirección IP de origen, que no se puede deducir a partir de la misma trama pero sí del contexto observando otras tramas. La dirección MAC de destino, o bien es cero en las peticiones o se encuentra descifrada en la cabecera 802.11 en las respuestas. Finalmente, está la dirección IP de destino, que también se puede deducir del contexto.

Otros tipos de tramas pueden contener paquetes del protocolo STP o, por defecto, se asume que contienen paquetes IPv4. En este último caso, se pueden obtener los primeros 12 bytes de *keystream* a partir de los valores de los bytes descifrados, haciendo ciertas suposiciones que se cumplen en la gran mayoría de los casos, como por ejemplo que el paquete no está fragmentado:

STP

STP es la sigla de *spanning tree protocol*.

- Los 8 primeros bytes son la cabecera LLC/SNAP, igual que la de las tramas con paquetes ARP pero cambiando el tipo Ethernet a 08:00 (IPv4).

- En los 2 bytes siguientes está la versión del protocolo, la longitud de la cabecera y el campo de servicios diferenciados. En la gran mayoría de paquetes IPv4 estos dos bytes son, en hexadecimal, 45:00.

- Los 2 bytes siguientes son la longitud del paquete IP, que se puede deducir por la longitud de la trama.

A continuación están los 2 bytes del identificador de datagrama, que en general tendrán un valor desconocido. A la hora de encontrar la clave WEP con el ataque PTW, Aircrack-ng hace una busca de todos los valores posibles de estos dos bytes si solo dispone de paquetes IPv4. En los 2 bytes siguientes, si el paquete no está fragmentado, estarán los valores 40:00 o 00:00, dependiendo de si está activado el *flag* DF (*don't fragment*) o no. El primero de estos dos bytes es el último que se utiliza para obtener el *keystream* que necesita el ataque PTW. Aircrack-ng asume que un 85% de los paquetes tendrán el *flag* activado y

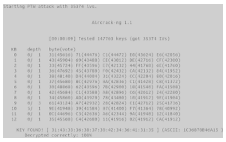
asigna el peso correspondiente a los votos que obtenga con esta suposición. El resto de campos de la cabecera IPv4 pueden tener valores más indeterminados, pero ya no se utilizan en el ataque PTW.

La herramienta `aircrack-ng`

Esta es la herramienta que a partir de los IV obtenidos implementa el ataque para recuperar la clave WEP. Por defecto aplica el ataque PTW y los ataques KoreK en paralelo, que incluyen el ataque FMS, pero tiene opciones para deshabilitar los ataques que no interese realizar. También puede ejecutar automáticamente un ataque Klein cuando encuentra que en la clave hay muchos bytes fuertes resistentes al ataque PTW.

La figura siguiente muestra un ejemplo del resultado obtenido con una ejecución de la herramienta `aircrack-ng`. En este ejemplo, el ataque ha tardado 9 segundos en encontrar la clave correcta a partir de poco más de 35.000 IV. Mientras prueba posibles valores de la clave, la herramienta va mostrando por pantalla el número de votos ponderados que obtienen los valores candidatos de cada byte de la clave.

Figura 12. Ejemplo de ejecución de la herramienta `aircrack-ng`



5. Soluciones a las vulnerabilidades WEP

Cuando se vio que el diseño inicial de la seguridad en el estándar IEEE 802.11 tenía deficiencias importantes, se propusieron varias soluciones para intentar corregir estos problemas.

Entre las propuestas que se hicieron, podemos destacar las siguientes:

- En el artículo que describía el ataque FMS, del año 2001, sus autores sugerían aplicarlo a las tramas en las cuales el IV empieza por $K[0] = n$ y $K[1] = 255$ cuando se está buscando el valor $K[n] (3 \leq n < 8)$, porque en estos casos es más fácil que se cumpla la hipótesis del ataque. Por lo tanto, una primera solución era no enviar tramas WEP cifradas con estos IV, llamados **vectores de inicialización débiles**. La realidad es que esta medida solo incrementa muy ligeramente el número de tramas que necesita el atacante para descubrir la clave, pero aun así, la mayoría de sistemas operativos la incorporaron a su núcleo, y a las tarjetas Wi-Fi que la implementaban en su hardware se les dio la etiqueta comercial **WEPplus**.

- Otra solución propuesta en el año 2001, llamada **WEP2**, se basaba en el uso de vectores de inicialización de 128 bits y claves raíz secretas también de 128 bits, es decir, claves RC4 de 256 bits en total. Esto, de hecho, no impide realizar los ataques diseñados para el protocolo WEP original, pero sí que alarga el tiempo necesario para completarlos con éxito, a pesar de que el crecimiento no es exponencial sino solo lineal.

- Algunos fabricantes optaron por una solución más efectiva, conocida como **Dynamic WEP**. Como su nombre indica, esta técnica consiste en ir cambiando dinámicamente las claves WEP, cosa que complica considerablemente los ataques respecto a las claves estáticas. Las diferentes implementaciones, sin embargo, no eran interoperables entre sí porque cada fabricante seguía sus propias especificaciones.

A partir de la propuesta de las claves dinámicas empezaron los trabajos de estandarización que darían lugar a la publicación de la especificación **IEEE 802.11i** en el 2004. En la edición del año 2007, esta extensión dejó de ser una especificación separada y se incorporó como un capítulo del estándar base IEEE 802.11.

Mientras se estaba elaborando el texto de esta especificación, y dada la urgencia para resolver los problemas que presentaba el protocolo WEP, la asociación de fabricantes Wi-Fi Alliance desarrolló una solución intermedia llamada **WPA**, con la intención de que se pudiera utilizar con el mismo hardware de las tarjetas Wi-Fi existentes, o introduciendo solo pocas modificaciones al microsoftware (*firmware*). Esta solución estaba basada en los borradores que iba publicando el grupo de trabajo IEEE 802.11i. Cuando se aprobó la versión oficial en el año 2004, el estándar IEEE 802.11i fue incorporado a las especificaciones de la Wi-Fi Alliance con el nombre de WPA2.

WPA

WPA es la sigla de Wi-Fi *protected Access*

5.1. WPA

El estándar WPA introduce cambios fundamentales tanto en el método de autenticación de las estaciones como en el algoritmo de cifrado de las tramas.

A diferencia del protocolo WEP, en que normalmente hay una sola clave secreta compartida por el AP y las estaciones, WPA prevé el uso de claves diferentes en cada **asociación segura**, es decir, en cada RSNA, y define los mecanismos para establecer estas claves dinámicamente.

RSNA

RSNA es la sigla de *robust security network association*.

El uso de una clave única compartida entre el AP y las estaciones, como prevé el protocolo WEP, puede ser apropiada para una red inalámbrica doméstica, pero es más problemática en una red corporativa media o grande. Cuando hay decenas o centenares de estaciones con la misma clave, si un atacante accede a la clave en una de las estaciones, automáticamente las comunicaciones de todas las otras quedan comprometidas. Además, cambiar la clave puede requerir actualizaciones manuales en cada una de las estaciones, cosa que puede ser poco práctica.

Por eso, WPA prevé el uso del método de control de acceso a la red definido en otro estándar de la serie IEEE 802, concretamente el **IEEE 802.1X**. Este estándar facilita el intercambio seguro de claves de sesión entre dos nodos de la red, previa autenticación mutua entre ellos. Que la autenticación sea mutua en WPA implica que la estación se autentica ante el AP, pero el AP también se autentica ante la estación, para que esta se pueda asegurar de que no está hablando con un AP falsificado.

El estándar IEEE 802.1X se basa a su vez en el protocolo EAP, que permite llevar a cabo una autenticación trabajando al nivel de enlace, es decir, sin necesidad de tener asignada todavía una dirección de red (IP). EAP prevé el uso de varios métodos de autenticación y, como su nombre indica, se pueden añadir otros definidos en otras especificaciones. Así, por medio del EAP se puede realizar

EAP

EAP es la sigla de *extensible authentication protocol*. Este protocolo está definido en la especificación RFC 3748.

una autenticación basada, por ejemplo, en nombres de usuario y contraseñas, en claves públicas y certificados X.509, en dispositivos físicos como tarjetas con chip, etc.

Lo que hace IEEE 802.1X es definir un formato de tramas denominado EAPOL para enviar los mensajes del protocolo EAP por una red local. Por otro lado, en la terminología IEEE 802.1X el extremo de la comunicación EAP que solicita la autenticación se denomina **suplicante**, y el otro extremo, el que la concede, se denomina **autenticador**. El autenticador puede conceder la autenticación por sí mismo, o bien puede comunicarse con un **servidor de autenticación** que toma la decisión final. Típicamente el servidor utilizará un protocolo, como por ejemplo RADIUS o Diameter, para llevar a cabo la autenticación.

EAPOL

EAPOL es la sigla de *EAP over LANs*.

WPA también continúa permitiendo el uso de una clave compartida o PSK, por simplicidad en el caso de redes pequeñas como suelen ser las redes domésticas. Pero en este caso la clave de cifrado no es directamente la clave compartida más un IV, como en el protocolo WEP, sino que la clave compartida se utiliza para derivar las correspondientes claves de sesión para cada asociación.

PSK

PSK es la sigla de *pre-shared key*.

En cualquier caso, cada pareja estación - AP utiliza sus propias claves para proteger sus comunicaciones. Así se consigue que una estación no pueda espiar las tramas enviadas entre el AP y otra estación del mismo BSS.

Este esquema, no obstante, tiene un inconveniente: mientras que con una clave compartida es fácil enviar una trama cifrada simultáneamente además de un nodo, como en el caso del tráfico de difusión (*broadcast*) o de difusión selectiva (*multicast*), con claves independientes sería necesario enviar tantas tramas como destinatarios, cada una cifrada con la clave correspondiente. Para evitar esta ineficiencia, en WPA se trabaja con dos tipos de claves:

- Las **claves entre parejas** son las que se utilizan para las tramas entre cada par de nodos, es decir entre el AP y cada estación.

- Las **claves de grupo** son conocidas por todos los miembros del BSS y se utilizan para las tramas de difusión (*broadcast*) o de difusión selectiva (multicast). Se puede generar una nueva clave de grupo cada vez que una estación abandona el BSS y se disocia del AP, para evitar que pueda continuar descifrando el tráfico del grupo.

5.1.1. Autenticación WPA y gestión de claves

WPA define dos modos de autenticación:

- El **modo WPA-PSK**. Es el que trabaja con una clave maestra predefinida, compartida entre el AP y las estaciones. Como hemos dicho antes, suele

usarse solo en redes con pocas estaciones. La Wi-Fi Alliance también dio a este modo el nombre **WPA-Personal**.

- El **modo WPA-802.1X**. Es el que utiliza el control de acceso basado en IEEE 802.1X más EAP, junto con un servidor de autenticación. La Wi-Fi Alliance también dio a este modo el nombre **WPA-Enterprise**.

Tanto si se utiliza un modo como el otro, una estación que quiere entrar en un ESS tiene que seguir estos pasos:

1) La estación tiene que identificar el ESS al que quiere acceder. Mediante las tramas baliza descubre la información que necesita sobre el ESS y el AP que lo gestiona, como por ejemplo el BSSID (es decir, la dirección MAC del AP), las velocidades de transmisión soportadas, etc.

Entre los campos de la trama baliza, también llamados IE, puede haber uno de tipo RSN (*robust security network*). Si este IE está presente, quiere decir que el AP soporta el establecimiento de asociaciones seguras WPA. Los diferentes subcampos del IE RSN indican los algoritmos de autenticación y de cifrado soportados.

IE

IE es la sigla de *information element*.

2) Por compatibilidad con los sistemas que implementan la máquina de estados 802.11, la estación primero tiene que hacer una autenticación de sistema abierto, como hemos visto en el apartado 2, seguida de una asociación 802.11.

La trama de gestión que contiene la solicitud de asociación incluye un IE de tipo RSN donde se especifican el algoritmo de autenticación y el de cifrado que la estación está dispuesta a utilizar, de entre los anunciados por el AP en las tramas baliza. El algoritmo de autenticación escogido determina si esta se realizará en modo WPA-PSK o en modo WPA-802.1X.

Autenticación de clave compartida

Recordad que la autenticación de clave compartida es totalmente insegura, y por eso no se contempla en el estándar WPA.

3) La estación y el AP establecen de manera segura una clave maestra entre parejas o PMK de 256 bits.

- Si se trabaja en modo WPA-PSK, la clave maestra PMK es directamente la clave compartida PSK previamente configurada. Muchas veces, para facilitar la configuración de las estaciones, la PSK no se especifica directamente sino como una frase de paso (*passphrase*). En estos casos, los bits de la PSK son el resultado de aplicar una función de generación de claves, definida en el estándar PKCS #5 y basada en funciones hash, a partir de la frase de paso y el SSID.

PMK y MSK

PMK es la sigla de *pairwise master key*, y MSK es la sigla de *master session key*.

- Si se trabaja en modo WPA-802.1X, se inicia el protocolo EAP para llevar a cabo el autenticación. La estación se pone de acuerdo con el autenticador, que puede ser el AP o un servidor de autenticación, sobre el método EAP a utilizar. Este método tiene que garantizar que un espía que observe la comunicación no pueda obtener ninguna contraseña u otra información se-

creta que le permita realizar una autenticación fraudulenta. Entonces suplicante y autenticador se intercambian los mensajes EAP necesarios hasta completar la autenticación. Si el proceso acaba con éxito, el resultado es que la estación y el AP se han autenticado mutuamente de manera satisfactoria, y además el método EAP utilizado también tiene que proporcionar un valor de 512 bits que se utilizará como clave maestra de sesión o MSK. Finalmente se obtiene la PMK, que es igual a los primeros 256 bits de la MSK.

4) La autenticación se completa ejecutando una **negociación en 4 pasos** o *4-way handshake*. Por un lado, esta negociación permite verificar que tanto el AP como la estación han obtenido correctamente la clave maestra PMK, y de este modo comprobar que son los auténticos. Y por otro lado, como resultado de la negociación se obtienen también las claves necesarias para proteger las tramas WPA, tanto entre parejas como de grupo.

Los mensajes de la *4-way handshake* se envían en un tipo especial de trama, denominado EAPOL-Key, definido en el estándar IEEE 802.1X. Durante la negociación se obtienen una clave de cifrado y una clave de autenticación de mensaje, llamadas KEK y KCK respectivamente, para ser utilizadas exclusivamente en la propia negociación. Los mensajes de la negociación se envían cifrados con RC4, excepto los dos primeros porque la clave KEK todavía no está disponible, y autenticados con HMAC-MD5, excepto el primero porque la clave KCK tampoco está disponible. Además, uno de los campos de las tramas EAPOL-Key es un contador para detectar ataques de repetición.

KEK, KCK, PTK y GTK

KEK es la sigla de *key encryption key*, KCK es la sigla de *key confirmation key*, PTK es la sigla de *pairwise transient key*, y GTK es la sigla de *group temporal key*.

El intercambio de mensajes en la *4-way handshake* es el siguiente:

a) El autenticador (AP) envía al suplicante (estación) un valor aleatorio N_A.

b) El suplicante genera otro valor aleatorio N_S y calcula la clave **transitoria entre parejas**o PTK, de 512 bits. El cálculo se hace aplicando una función unidireccional a la clave maestra PMK, los valores aleatorios N_A y N_S, y las direcciones MAC de autenticador y suplicante. Entonces se obtienen las claves KCK y KEK tomando los primeros 128 bits de la PTK y los 128 bits siguientes, respectivamente. Una vez obtenidas estas claves, el suplicante envía el valor N_S al autenticador.

c) El autenticador hace los mismos cálculos para obtener la clave PTK, y a partir de esta, la KCK y la KEK. Entonces envía la clave **temporal de grupo** o GTK al suplicante, cifrada con la KEK.

d) El suplicante comprueba que el mensaje anterior es correcto. Si es así, la autenticidad del autenticador (AP) habrá quedado confirmada. El suplicante envía entonces al autenticador un mensaje que no hace falta que contenga nada en su campo de datos. Si el autenticador ve que es correcto, la autenticidad del suplicante (estación) también habrá quedado confirmada.

Como resultado del proceso anterior, el AP y la estación han acordado una clave PTK para utilizar entre ellos. De esta PTK, tomando los bits 256-511, se obtiene una **clave temporal** o TK, que será la que se usará para el cifrado y la autenticación de las tramas WPA.

TK

TK es la sigla de *temporal key*.

En la negociación, además, el AP envía a la estación la clave de grupo GTK. Esta clave de grupo la determina unilateralmente el AP. La manera de obtenerla es una cuestión interna del AP, pero, por analogía con la PTK, se puede obtener por ejemplo aplicando una función unidireccional a una clave maestra de grupo o GMK más un valor aleatorio N_G.

Una vez establecida la sesión, el protocolo *4-way handshake* se puede volver a iniciar en cualquier momento para renegociar la clave transitoria PTK, por ejemplo cuando la sesión es larga y ya hace cierto tiempo que se está utilizando la misma clave. En este caso, el envío de la clave GTK es opcional si no ha cambiado.

Y en el momento en que cambie la GTK, por ejemplo porque una estación ha salido del BSS y ya no tiene que continuar recibiendo tráfico de difusión, se realiza otro tipo de negociación denominado *group key handshake* entre el AP y cada estación. Esta negociación tiene dos pasos porque solo hay que enviar la nueva clave GTK cifrada a la estación, y que esta confirme que lo ha recibido.

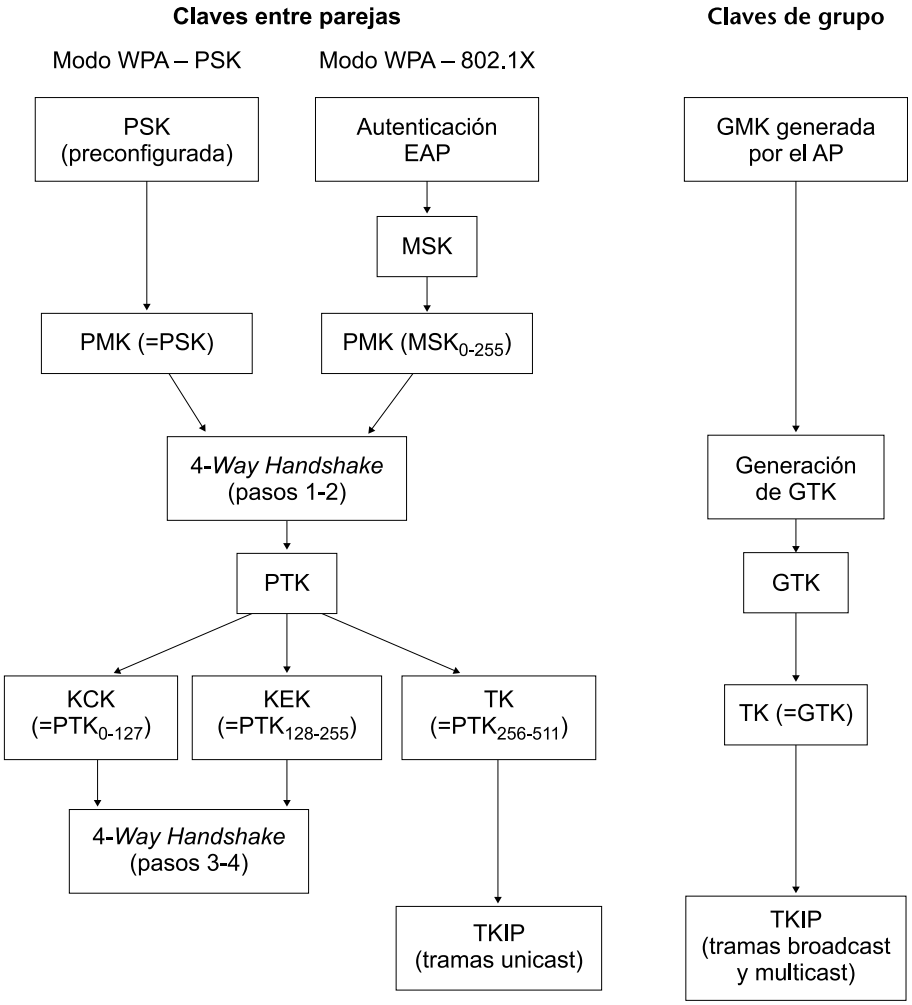
Como se puede comprobar, la autenticación WPA introduce fuertes medidas de seguridad para evitar cualquier tipo de ataque: un mecanismo seguro para derivar una clave maestra PMK que es diferente para cada sesión con cada estación (excepto en el modo WPA-PSK), una clave transitoria PTK que se puede ir cambiando periódicamente, y un protocolo de generación de claves temporales en cuatro pasos añadido al método de autenticación, con el uso de claves criptográficas independientes de las de la comunicación normal, derivadas de las direcciones MAC y con contadores para evitar ataques de repetición.

Una debilidad de este esquema es el uso de los algoritmos RC4 y MD5 para el cifrado y la autenticación de la negociación en cuatro pasos, que no son tan seguros como otros algoritmos que se han desarrollado posteriormente. Pero el objetivo inicial del estándar WPA era que se pudiera utilizar con el hardware disponible, y esta era una solución de compromiso mientras no se extendía la implementación del WPA2. Por otro lado, en el modo WPA-PSK una estación

que capture los valores N_A y N_s de la negociación de otra estación, que no se envían cifrados, inmediatamente sabrá cuál es su clave PTK y podrá descifrar su tráfico.

A modo de resumen, el diagrama siguiente muestra las relaciones entre las diferentes claves que forman la llamada jerarquía de claves WPA.

Figura 13. Jerarquía de claves WPA



Métodos de autenticación EAP usados en WPA-802.1X

Actualmente existen decenas de métodos EAP, entre los estandarizados por el IETF y los definidos por varios fabricantes. Algunos de los usados más habitualmente en el modo WPA-802.1X son:

- **EAP-TLS**. En este método, la comunicación con el servidor de autenticación, por ejemplo un servidor RADIUS, se protege mediante el protocolo TLS con autenticación mutua basada en certificados de servidor y de cliente.

TLS

El protocolo de seguridad TLS (*transport layer security*) se estudia en el apartado 5 de este módulo.

- **EAP-TTLS** (*EAP-tunneled TLS*). Es una variante simplificada del método anterior en que no son necesarios los certificados de cliente, lo cual lo hace

mucho más práctico. Se utiliza el protocolo TLS para crear un canal seguro o "túnel" solo con certificado de servidor. Entonces, se realiza la autenticación del cliente con otro método, que puede ser por ejemplo basado en contraseña, a través de este canal seguro.

- **PEAP** (*protected EAP*). Es un método genérico para encapsular la autenticación de cliente dentro de otro método con autenticación de servidor, por ejemplo basado en TLS.

Además de los pasos para realizar la autenticación, cada uno de estos métodos tiene que definir también cómo se genera el valor a utilizar como clave maestra de sesión (MSK).

Los métodos como EAP-TTLS o PEAP establecen la autenticación del servidor, pero entonces hay que usar otro método para la autenticación del cliente. Este otro método puede ser, por ejemplo:

Generación de la MSK

En los métodos EAP basados en TLS, el MSK se genera normalmente aplicando una función unidireccional a las cadenas de bits aleatorias utilizadas en la fase de negociación TLS (*handshake protocol*) y el secreto maestro que se obtiene de ello.

- **EAP-MD5**. Es un método de reto-respuesta. La respuesta es un hash MD5 de una cadena formada por la contraseña del cliente más el reto.

- **EAP-MSCHAPv2** (*EAP-Microsoft challenge handshake authentication protocol version 2*). Utiliza el protocolo MSCHAPv2, definido en la especificación RFC 2759.

- **EAP-GTC** (*EAP-generic token card*). También es un método de reto-respuesta en el cual la respuesta es generada por un dispositivo físico como puede ser una tarjeta con chip.

5.1.2. El cifrado TKIP

Además del método de autenticación, el otro cambio fundamental introducido en WPA respecto a WEP es el algoritmo de cifrado. O más exactamente, la generación de las claves de cifrado, puesto que el algoritmo propiamente dicho es el mismo: RC4. Esto se decidió, como ya hemos visto, para intentar aprovechar el hardware de las tarjetas de red que existían entonces.

El esquema de cifrado que se utiliza en el estándar WPA se denomina TKIP. Las principales diferencias que presenta respecto al esquema WEP son:

TKIP

TKIP es la sigla de temporal key integrity protocol.

- La clave con que se cifran los datos de cada trama no se obtiene a partir de un vector de inicialización variable y una parte fija, sino que todos los bits de la clave RC4 se recalculan en cada trama.

- Las tramas TKIP incorporan un código MIC calculado a partir de una clave secreta, como prevención contra los ataques de modificación o truncamiento como el *chopchop*. El código MIC no sustituye sino que complementa el campo ICV. Por otro lado, cuando se produce fragmentación este código se calcula sobre la trama original antes de fragmentar, en vez de haber un código MIC por cada fragmento.

- Para evitar ataques de inyección, el código MIC no se calcula solo sobre los datos cifrados sino que también se añaden las direcciones MAC de las estaciones origen y destino.

- Cada trama incluye un contador de secuencia de 48 bits, denominado TSC, como medida contra los ataques de repetición. Este contador se reinicializa a 1 cada vez que se usa una clave temporal TK nueva. El contador N de una trama enviada después de otra con contador M tiene que cumplir $N > M$ (si son tramas consecutivas, puede ser por ejemplo $N = M + 1$, pero no necesariamente). Las tramas recibidas que no sigan esta regla son descartadas. Algunas tramas pueden tener asignada una prioridad, y puede suceder que tramas de prioridades diferentes sean recibidas en orden diferente al de envío. Por lo tanto, emisor y receptor tienen que mantener un contador TSC independiente por cada prioridad utilizada (puede haber como máximo 8 diferentes).

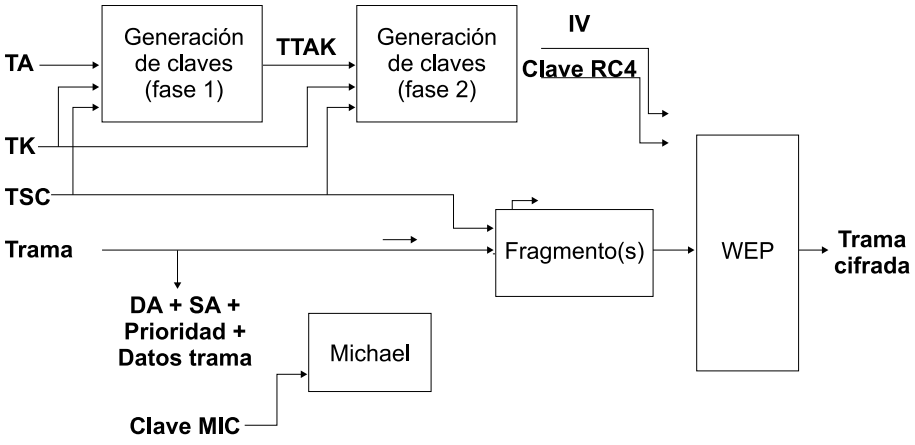
TSC

TSC es la sigla de *TKIP sequence counter*.

MIC

MIC es la sigla de *message integrity code*, que es la nomenclatura con que IEEE 802.11 se refiere al código de autenticación de mensaje, para evitar confusiones con *medium access control* (MAC).

Figura 14. Generación de tramas cifradas TKIP



El proceso que se sigue en el algoritmo TKIP para generar una trama cifrada incluye los pasos siguientes:

1) Se genera el código MIC aplicando un algoritmo llamado Michael a las entradas siguientes:

- La información siguiente de la trama: la dirección MAC de destino (DA), la dirección MAC de origen (SA), la prioridad y el campo de datos.

- La clave MIC de 64 bits. Para evitar ataques de repetición en sentido contrario se utilizan dos claves MIC diferentes para las tramas del AP a la estación y para las tramas de la estación al AP. La primera se obtiene de los bits 128-191 de la clave TK, y la segunda de los bits 192-255 de la misma clave.

El algoritmo Michael es una sencilla función *hash* con operaciones simples que se puede calcular muy rápidamente. No es, sin embargo, una función *hash* segura porque no tiene la propiedad de la unidireccionalidad.

2) En caso de que sea necesario, se aplica la fragmentación a la trama más el código MIC. A cada fragmento se le asigna un contador TSC diferente, siempre respetando el orden creciente.

3) Se aplica una función criptográfica, llamada **fase 1**, a las entradas siguientes:

- La clave temporal TK obtenida en la *4-way handshake*. Como clave para el cifrado, se utilizan los primeros 128 bits (0-127) de la clave TK.

- La dirección MAC de la estación transmisora, TA.

- El contador TSC. Para la fase 1 se utilizan los 24 bits de más peso del contador.

El resultado de la fase 1 es un valor TTAK (*TKIP-mixed transmit address and key*) de 80 bits. Este valor será el mismo para todas las tramas que tengan los mismos 24 bits de más peso del contador TSC, y por lo tanto no hará falta recalcularlo cada vez.

4) A continuación se aplica otra función criptográfica, llamada **fase 2**, a las entradas siguientes:

- El resultado TTAK de la fase 1.

- La misma clave de cifrado que en la fase 1 (los bits 0-127 de la clave TK).

- El contador TSC. Para la fase 2 se utilizan los 24 bits de menos peso del contador.

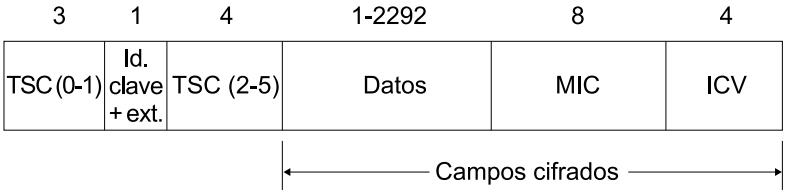
El resultado de la fase 2 es una clave de cifrado RC4 de 128 bits con 24 bits de IV y 104 bits de clave raíz.

La propiedad principal del cifrado TKIP es que los 104 bits de clave raíz son diferentes para cada trama, con lo cual los ataques estadísticos contra el cifrado WEP no son aplicables.

El IV se construye de forma que los bytes primero y tercero se copian de los 16 bits de menos peso del TSC, y el segundo byte se deriva del primero con la precaución de que el resultado no sea un IV débil, es decir, que no tenga el segundo byte igual a 255.

5) Finalmente, la trama, o cada fragmento de trama si es el caso, se cifra igual que en el protocolo WEP.

Figura 15. Datos de una trama TKIP



La estructura de la trama cifrada TKIP que se genera es ligeramente diferente de la de las tramas WEP normales.

- El primer campo contiene el IV, igual que en las tramas WEP, en este caso obtenido a partir de los dos bytes de menos peso del TSC.

- El siguiente campo tiene activado un bit de extensión para indicar que, a continuación, hay un campo adicional.

- El campo adicional de extensión se aprovecha para incluir el resto de bytes del TSC (2-5).

- A continuación de los datos de la trama, y antes del campo ICV, se insertan los 64 bits del código MIC.

5.1.3. Vulnerabilidades y contramedidas

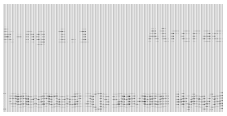
La principal vulnerabilidad del sistema WPA está en el uso del modo de autenticación de clave compartida, WPA-PSK. Aunque se trabaje con una clave maestra PMK de 256 bits, si esta clave proviene exclusivamente de una palabra más o menos fácil de recordar, el espacio de posibles claves queda muy reducido y hace viable un ataque por fuerza bruta. Por ejemplo, la herramienta `aircrack-ng` permite realizar un ataque de diccionario sobre los paquetes de la negociación *4-way handshake* entre una estación y el AP. A diferencia de los ataques WEP, que cuantas más tramas tengan disponibles más rápidamente pueden encontrar la clave, el ataque de diccionario WPA solo necesita las

tramas de una sola negociación. De hecho, bastan dos de las cuatro tramas. Estas tramas se pueden obtener con `airodump-ng`, esperando de manera pasiva que alguna estación establezca una asociación con el AP o bien provocando de manera activa la asociación con el modo desautenticación de la herramienta `aireplay-ng`. El ataque consiste en ir probando, para cada palabra del diccionario, si las claves que se derivan cuadran con el contenido cifrado y autenticado de las tramas capturadas.

La conclusión es que si se utiliza el modo WPA-PSK hay que escoger una clave que no sea ninguna palabra de diccionario en ningún idioma ni una combinación trivial de palabras (por ejemplo, una palabra escrita del revés). Se aconseja utilizar frases largas, de al menos 20 caracteres, que no contengan palabras de diccionario.

La figura siguiente muestra el ejemplo de una ejecución de la herramienta `aircrack-ng` para descubrir una clave WPA-PSK con el ataque de diccionario. En menos de un minuto y medio, y después de probar poco más de 28.000 palabras de diccionario, en este ejemplo la herramienta ha encontrado que la clave es la palabra "funicular".

Figura 16. Ejemplo de ejecución de un ataque de diccionario WPA



En cuanto al protocolo TKIP, como hemos visto antes, su diseño incluye una serie de protecciones criptográficas para evitar los ataques de inyección, de repetición, de modificación, de truncamiento, etc. Pero además la especificación incluye también una medida en el funcionamiento del protocolo para intentar contrarrestar los ataques contra el código MIC. El objetivo es evitar ataques de ensayo y error, del estilo del ataque *chopchop* sobre el ICV. Si un atacante consiguiera romper el código MIC podría inyectar tramas correctas.

Para evitarlo, el protocolo TKIP está diseñado para retardar la velocidad a la cual se pueden hacer los intentos de ataque. Concretamente, la especificación establece que las tramas con los campos CRC, ICV o TSC incorrectas tienen que ser ignoradas. Pero si estos campos son correctos y el código MIC es erróneo, esto tiene que ser considerado como un posible ataque y señalizado como tal en los registros (*logs*) de seguridad. La estación que detecte el error tiene que enviar al AP un tipo especial de trama EAPOL-Key, denominado *Michael MIC failure report*. Y si se detectan dos tramas erróneas en menos de 60 segundos, se tiene que deshabilitar la recepción de tramas TKIP durante un minuto. Cuando se restablezca, las claves tendrían que ser renegociadas.

Esto implica que un atacante no podrá hacer más de dos intentos por minuto. Aun así se han propuesto ataques, como el llamado ataque Beck-Tews, que en ciertas condiciones teóricamente permitirían a un atacante descifrar los últimos 12 bytes de una trama (MIC e ICV) en poco más de 12 minutos. Si es una trama ARP con contenido conocido, se puede obtener fácilmente la clave MIC puesto que el algoritmo Michael no está diseñado para ser unidireccional. Y en 4 o 5 minutos más se podría obtener suficiente *keystream* como para poder inyectar determinados tipos de tramas.

5.2. WPA2

La especificación WPA2 incorpora toda la funcionalidad del estándar IEEE 802.11i. Los cambios que introduce respecto a WPA son de dos tipos:

- Por un lado, define mecanismos como la preautenticación y el almacenamiento de claves maestras (*PMK caching*) que hacen más rápida y eficiente la reautenticación de una estación móvil cuando sale de un BSS y entra en un BSS adyacente del mismo ESS (*roaming*).

- Por otro lado, introduce también un nuevo algoritmo de cifrado, denominado **CCMP**, que no está basado en el RC4 sino en la cifra AES-128. Este método de cifrado es mucho más seguro, puesto que actualmente no se conocen vulnerabilidades significativas, y aunque no es tan sencillo de implementar como el RC4, es bastante más eficiente que la mayoría de otras cifras de bloque existentes.

CCMP

CCMP es la sigla de *CTR with CBC-MAC protocol*.

Los sistemas WPA2 tienen que soportar obligatoriamente el cifrado CCMP. El uso del cifrado TKIP es opcional, por compatibilidad con los sistemas WPA. Por otro lado, cuando se trabaja con CCMP los algoritmos criptográficos utilizados en la *4-way handshake* son AES (según el estándar RFC 3394) para el cifrado y HMAC-SHA1 para la autenticación de mensaje, en vez de RC4 y HMAC-MD5, respectivamente.

El cifrado CCMP consiste en aplicar el modo CCM definido en la especificación RFC 3610 a la cifra de bloque AES con clave de 128 bits. El modo CCM proporciona a la vez autenticación de mensaje y confidencialidad, todo con la misma clave. La clave CCMP es de 128 bits y se obtiene, como en TKIP, de los bits 0-127 de la clave temporal TK.

Bits de la clave TK

Los primeros 128 bits de la clave TK son los únicos que se necesitan en el protocolo CCMP porque se utilizan tanto para cifrar como para autenticar. Cuando se trabaja con CCMP, pues, no hay que generar 512 bits de PTK en los pasos 2 y 3 de la *4-way handshake*, sino que basta con 384.

1) El código de autenticación MAC se genera realizando un cifrado AES-128 en modo CBC con la técnica conocida como CBC-MAC. El vector de inicialización, según la especificación CCM, tiene 1 byte de *flags*, 2 bytes que indican la longitud de los datos, y los otros 13 bytes tienen que ser únicos para cada trama. Para conseguirlo, estos 13 bytes se construyen de la manera siguiente:

- 1 byte codifica la prioridad.

- Los 6 bytes siguientes son la dirección MAC (*medium access control*) de la estación transmisora.

- Los últimos 6 bytes son un **número de paquete** o PN (*packet number*) de 48 bits que se incrementa en cada trama. Después de este vector de inicialización, los 2 bloques siguientes que se cifran contienen una combinación de los campos invariantes de la cabecera MAC (*medium access control*) de la trama, que son básicamente todos excepto el campo Duración/ID. A continuación de estos 2 bloques se cifran los datos de la trama, completadas al final con bytes iguales a 0 si su longitud no es múltiple de 16. Del resultado de cifrar el último bloque se toman los primeros 64 bits, y este será el código MAC que autenticará la trama.

2) Los datos cifrados se obtienen aplicando un cifrado en el "modo contador" (*CTR mode*) según la terminología CCM. Para cada bloque de datos, se construye un bloque auxiliar que tiene 1 byte de *flags*, 2 bytes que contienen un contador igual a 1 para el primer bloque y que se incrementa en cada uno de los bloques siguientes, y los otros bytes son iguales a los 13 bytes únicos que se utilizan para generar el código MAC.

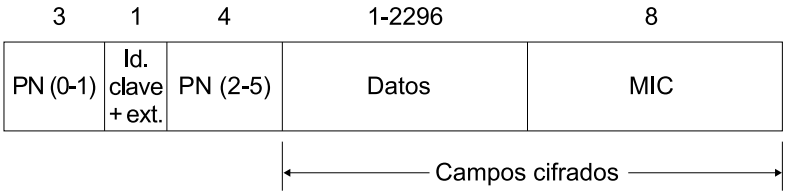
Terminología CCM

En el contexto de la especificación CCM, "MAC"significa *message authentication code*, y "vector de inicialización"tiene el sentido que se le da normalmente en las cifras de bloque, es decir, el bloque aleatorio que se utiliza como si fuera el anterior al primer bloque.

Entonces se cifra este bloque auxiliar con AES-128, y el resultado se suma bit a bit (con la operación XOR) con el correspondiente bloque, como si fuera una cifra de flujo. Si la longitud del último bloque es menor de 16 bytes, solo se utilizan los bytes del bloque auxiliar cifrado que hagan falta.

El código MAC obtenido en el primer paso también se cifra sumándolo con otro bloque auxiliar cifrado, en el que el contador es igual a 0. El resultado es el código MIC, según la terminología WPA.

Figura 17. Datos de una trama CCMP



Después de aplicar la autenticación de mensaje y el cifrado ya se puede generar la trama cifrada CCMP. Su estructura es parecida a la de las tramas TKIP, sustituyendo los bytes del contador TSC por los del número de paquete PN, y con la diferencia principal de que en CCMP no se incluye el campo ICV. Este campo, que en TKIP sirve para reforzar el código MIC basado en el algoritmo Michael, no se considera necesario en CCMP dada la fortaleza de la autenticación CBC-MAC con la cifra AES.

Los únicos ataques prácticos que se conocen sobre el sistema WPA2 son los mismos que afectan al sistema WPA cuando se utiliza el modo PSK. Es decir, WPA2-PSK es exactamente igual de vulnerable a ataques de diccionario que WPA-PSK, puesto que estos ataques actúan sobre la negociación *4-way handshake* y son independientes del algoritmo de cifrado de las tramas.

Resumen

En este módulo didáctico hemos estudiado los diferentes mecanismos que existen para la protección de la información en las redes WLAN.

En particular hemos visto cómo el protocolo WEP, por medio del criptosistema en flujo RC4, permite cifrar la información que viaja por la red para dificultar su intercepción por parte de un atacante. Aun con la mejora que supone el protocolo WEP respecto a enviar la información en claro, también hemos podido ver que este protocolo presenta una serie de debilidades que hacen que sea atacable de diferentes maneras. Además, hay una serie de herramientas específicas, como el Aireplay-ng o el Aircrack-ng, que permiten explotar las vulnerabilidades del protocolo WEP.

Finalmente, hemos estudiado que el protocolo WPA protege las redes WLAN de manera mucho más efectiva. En concreto, hemos visto los diferentes modos de funcionamiento del WPA y cómo gestiona de manera más segura tanto la fase de autenticación de los usuarios como el sistema de cifrado de la información.

Glosario

AP (*Access Point*) Estación específica que permite la interconexión con otras redes, con hilo o sin.

EAP (*Extensible authentication Protocol*) Protocolo que permite llevar a cabo una autenticación trabajando al nivel de enlace, es decir, sin necesidad de tener asignada aún una dirección de red (IP).

IEEE 802.11 Es el estándar definido por el Institute of Electrical and Electronics Engineers para las comunicaciones en redes locales sin hilos, también conocido como Wi-Fi.

PSK Pre-Shared Key. Clave compartida.

RC4 Algoritmo criptográfico de cifrado en flujo diseñado por Ronald Rivest (de aquí el acrónimo Ron's code 4), que se utiliza para cifrar las tramas WEP.

service set identifier *m* Identificador de formato libre de hasta 32 bytes que hace referencia a una estación sin hilos.

SSID *m* Véase ***service set identifier***.

TKIP Temporal Key Integrity protocol. Esquema de cifrado que se utiliza en el estándar WPA basado en el algoritmo de cifrado en flujo RC4.

WEP *m* Véase ***wired equivalent privacy***.

wired equivalent privacy *m* Sistema de protección que incorpora el estándar IEEE 802.11 para tecnología LAN sin hilos.

WLAN *f* Véase ***wireless local area network***.

Bibliografía

Borisov, N.; Goldberg, I.; Wagner, D. (2001). "Intercepting Mobile Communications: The Insecurity of 802.11". En: *Proceedings of Mobicom 2001*. ACM Press.

IEEE Computer Society (1999). *Std. 802.11. Parte 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Nueva York: IEEE Press.

Ma, Jianfeng, Ma, Zhuo, Wang, Changguang (2009). *Security Access in Wireless Local Area Networks: From Architecture and Protocols to Realization*. Springer-Verlag Berlín.