(In)seguridad en redes 802.11b

Pau Oliva <pof@eSlack.org>

Feb 2003

Última versión de la presentación y la documentación que la acompaña:

http://pof.eslack.org/wireless/

Tarjetas Wi-Fi: Modos y Chipsets

Chipsets más extendidos

- Hermes (Lucent)
  - Lucent / Agere / **Orinoco**
    - Orinoco, Avaya, Compaq, Lucent
- **Prism** 2 / 2.5 / 3 (Intersil)
    - D-Link, Linksys, Netgear, SMC, USR, Conceptronic
- Airo (Aironet)
  - Cisco
- TI ACX100 (Texas Instruments)
  - 3Com / USR, D-Link, Wisecom, Eusso, Linksys (WAP11 v2.2)

Modos de funcionamiento

- **Ad-Hoc**: conectar dos PC’s sin AP
- **Managed**: Tarjeta asociada con un AP
- **Master**: La tarjeta trabaja como un AP
- **Monitor**: Permite capturar paquetes sin asociarse a un AP o a una red ad-hoc.

Modo Master

- **Master**: La tarjeta trabaja como un AP
  - HostAP: http://hostap.epitest.fi/
  - IEEE 802.1X
  - Dynamic WEP rekeying
  - RADIUS Accounting
  - RADIUS-based ACL for IEEE 802.11 authentication
  - Minimal IAPP (IEEE 802.11f)

SÓLO FUNCIONA CON CHIPSET PRISM

Modo Monitor

- **Monitor**: Permite capturar paquetes sin asociarse a un AP o a una red ad-hoc.
  - Monitoriza un canal especifico sin transmitir paquetes
  - La tarjeta no mira los CRC's de los paquetes
  - **No** es lo mismo que el modo promiscuo

- Chipset PRISM: Sin problemas
- Chipset Orinoco:
  - Parche: http://airsnort.shmoo.com/orinocoinfo.html

Encontrar redes wireless

Material Necesario

- Ordenador portátil o PDA
- Tarjeta Wi-Fi con firmware adecuado
- Programa o driver que permita poner la tarjeta en modo monitor
- Sniffer

Otros materiales adicionales

- Antena direccional o omnidireccional
- GPS
- Equipo electrógeno
- Mochila
- Auriculares
- Medio de transporte (coche, patines, bicicleta…)

Proceso a seguir

- Poner la tarjeta en modo monitor:
  - Linux:
    - Instalar wireless-tools (http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)
    - Instalar pcmcia-cs (http://pcmcia-cs.sourceforge.net/)
      - Chipset Orinoco → parchear pcmcia-cs

```
# iwpriv wlan0 monitor 1 1
```

Proceso a seguir

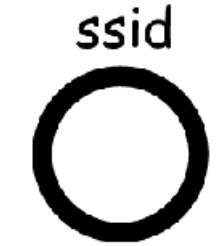
- Instalar un sniffer que nos permita capturar tramas 802.11b en modo monitor:
  - Linux:
    - Kismet: http://www.kismetwireless.net/
    - Airsnort: http://airsnort.shmoo.com/
    - Ethereal: http://www.ethereal.com/
  - Windows:
    - NetStumbler: http://www.netstumbler.com/
    - Airopeek: http://www.wildpackets.com/

Salir a la calle

- Es aconsejable:
  - Desplazarse a poca velocidad
  - Moverse cerca de los edificios
  - Hacerlo preferiblemente en horario laboral
- Según el medio de transporte que utilicemos, esta práctica se denomina:
  - **WarWalking**: Andando
  - **WarSkating**: En patines
  - **WarCycling**: En bicicleta o ciclomotor
  - **WarDriving**: Coche
  - **WarFlying**: Avión

Wardriving

WarChalking

SÍMBOLO	SIGNIFICADO
	Nodo Abierto
	Nodo cerrado
	Nodo con WEP

¿Es ilegal hacer wardriving?

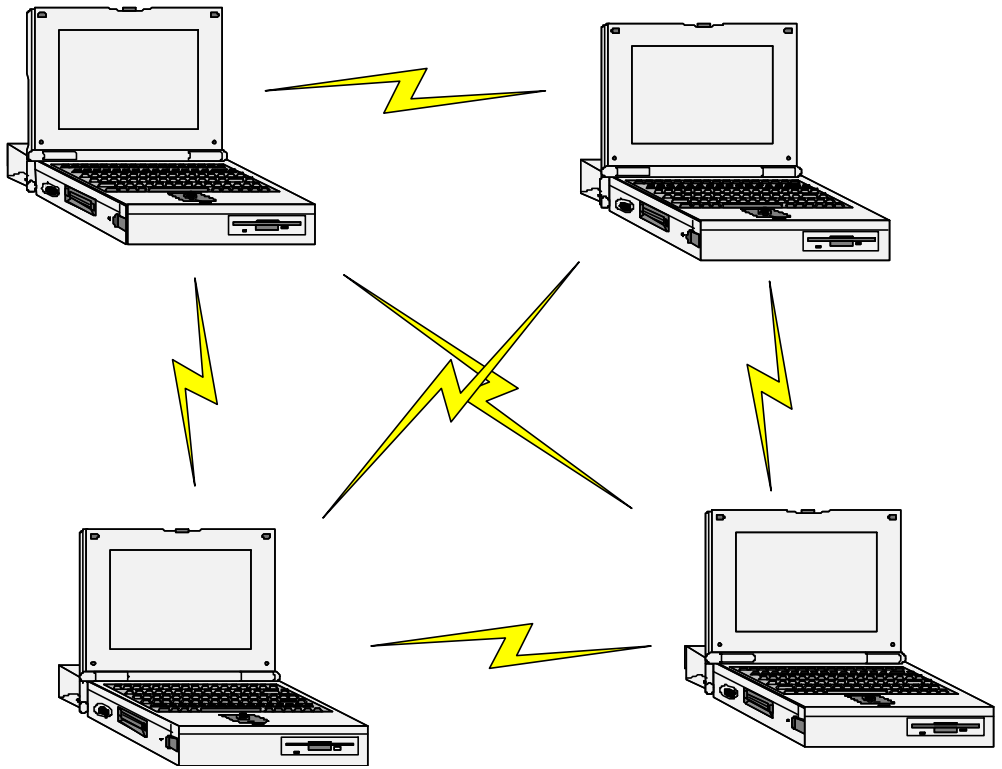
- Las redes Wireless se comunican usando ondas de radio.
- El espectro radioeléctrico se trocea a nivel internacional según REGIONES designadas por la ITU (International Telecomunications Union).
- En cada región son los gobiernos locales los que regulan la administración de cada zona del espectro respetando la normativa de la ITU.
- Cualquier transmisión radioeléctrica está regulada por la Dirección General de Telecomunicaciones.
- La frecuencia utilizada por las redes wireless es 2,4GHz, y es una frecuencia reservada para uso PUBLICO en nuestro país. Esto significa que cualquiera puede emitir lo que quiera en este espectro y los dispositivos que “escuchan” en él deben estar preparados para recibir interferencias inesperadas.
- Es perfectamente legal emitir una petición a través del las ondas de radio diciendo “Quiero ver el contenido de pepe.es” y esperar una respuesta.

Temario

1. 802.11b: Introducción y conceptos básicos
2. Wired Equivalent Privacy (WEP)
3. Proceso de conexión a una WLAN
4. Métodos de Autenticación
5. Vulnerabilidades
    5.1. Deficiencias en la encriptación
    5.2. Deficiencias en la autenticación
6. Ataques
    6.1. Ataques al WEP
    6.2. Ataques a redes wireless
7. Posibles soluciones

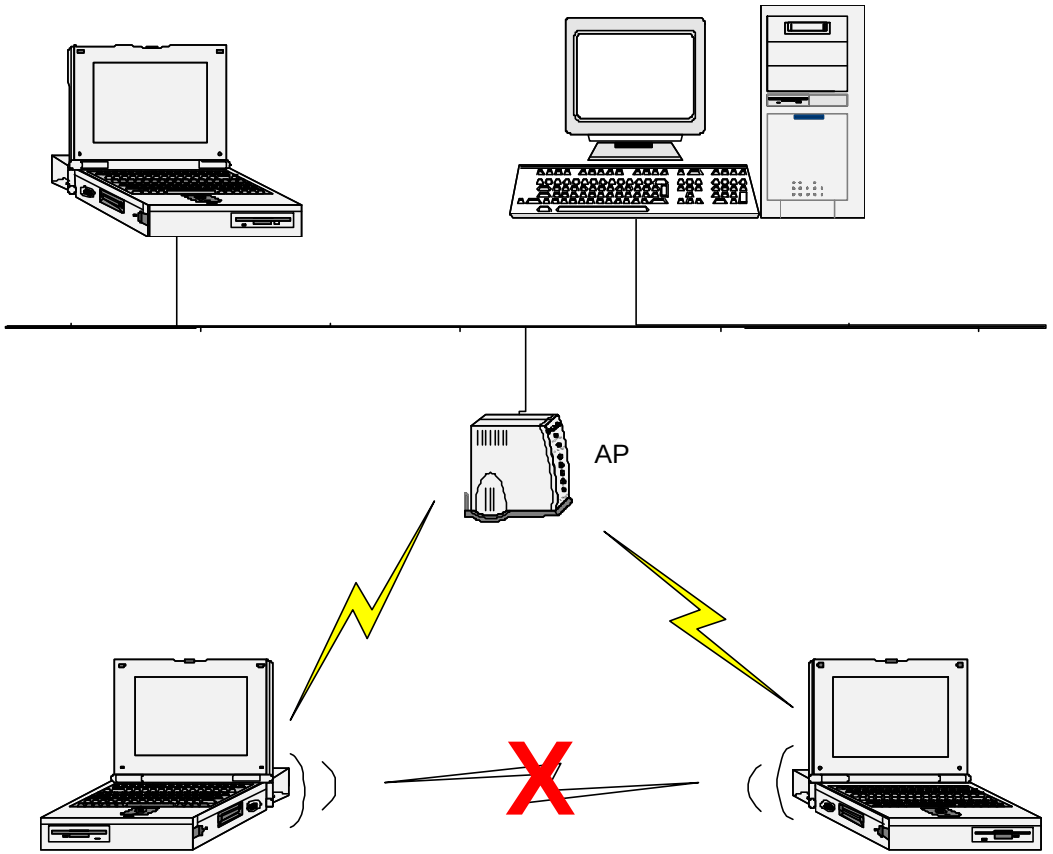
802.11b
Introducción y conceptos básicos

Modo AD-Hoc



- No hay punto de acceso
- Comunicación P2P

Modo Infrastructura



- Como mínimo un Punto de Acceso
- Todo el trafico pasa por el AP
- El AP actúa como un HUB en una LAN

Conceptos básicos

- WEP
  Wired Equivalent Privacy.
  Protocolo de encriptación basado en el algoritmo RC4.
- ESSID
  Extended Service Set Identifier.
  “Nombre” de la red. NO es un password.
- BEACON FRAMES
  “Anuncios” de la red emitidos por el AP.
  Normalmente contienen el ESSID.
- MANAGEMENT FRAMES
  Proceso de autenticación mutua y asociación.

Medidas comunes de seguridad

Estas medidas NO son efectivas

- ACL’s basadas en MAC

  El AP sólo permite conectar a los clientes que “conoce”

- No emitir BEACON FRAMES

  o emitirlos sin el ESSID

- Utilizar WEP

  Comunicación cifrada, más difícil de romper.

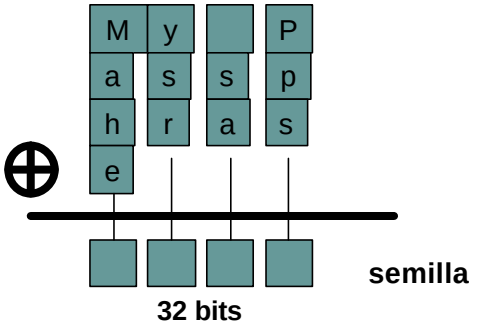
WEP
Wired Equivalent Privacy

WEP: Introducción

- Basado en el algoritmo RC4
- Utiliza llaves de 64 ó 128 bits

  en realidad son 40 / 104 bits (24 == IV)
- La llave se puede generar a partir de una *passphrase* o ser introducida directamente por el usuario
- La llave debe ser conocida por todos los clientes

WEP: Generar la llave



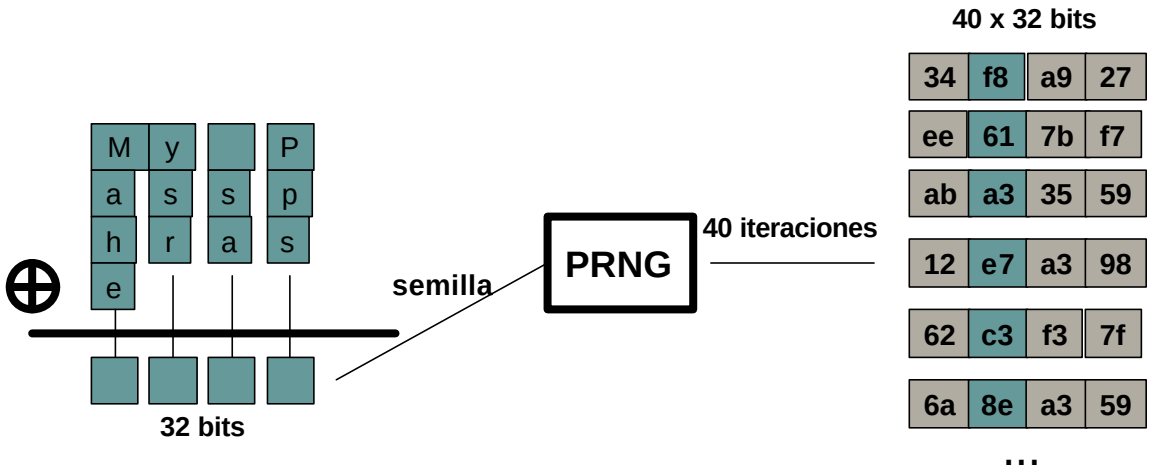
- Se hace una XOR con la cadena ASCII para obtener una semilla de 32 bits

M	y		P	a	s	s	p	h	r	a	s	e
4D	79	20	50	61	73	73	70	68	72	61	73	65

```
4D XOR 61 XOR 68 XOR 65 = 21
79 XOR 73 XOR 72 XOR 0  = 78
20 XOR 73 XOR 61 XOR 0  = 32
50 XOR 70 XOR 73 XOR 0  = 53
```

} SEMILLA

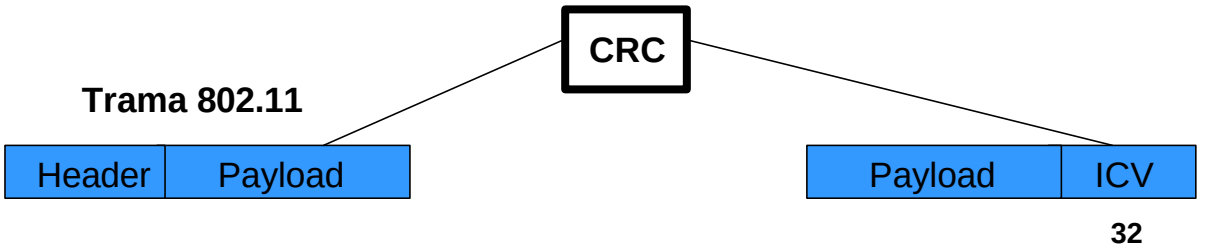
WEP: Generar la llave



- El PRNG utiliza la semilla para generar 40 cadenas de 32 bits cada una.
- Se toma un bit de cada una de las 40 cadenas generadas por el PRNG para construir una llave y se generan 4 llaves de 40 bits.
- Sólo una de las 4 se utilizará para la encriptación WEP

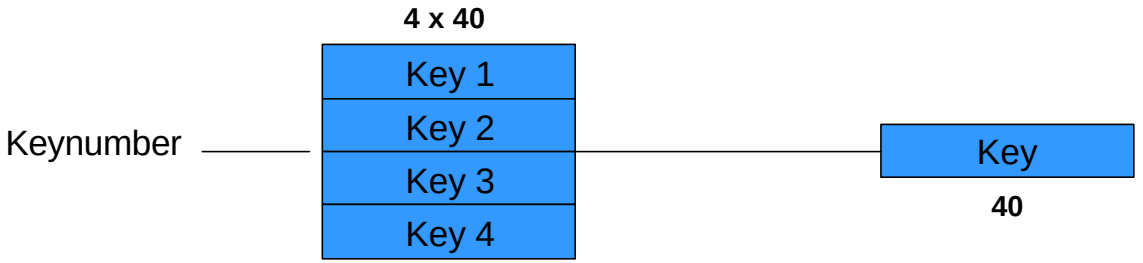
WEP: Generar una trama encriptada

1. Se calcula el CRC de 32 bits del payload de la trama que se quiere enviar.
   Se añade a la trama a encriptar como **valor de chequeo de integridad** (ICV):



WEP: Generar una trama encriptada

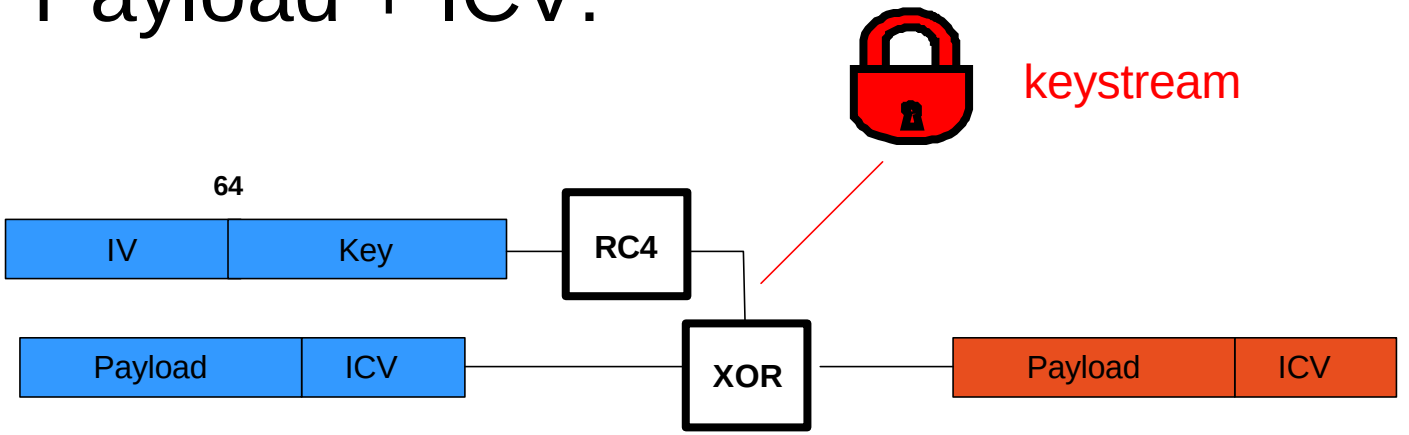
2. Se selecciona una llave de 40 bits, de las 4 llaves posibles:



3. Se añade un **Vector de Inicialización** (IV) de 24 bits al principio de la llave seleccionada:

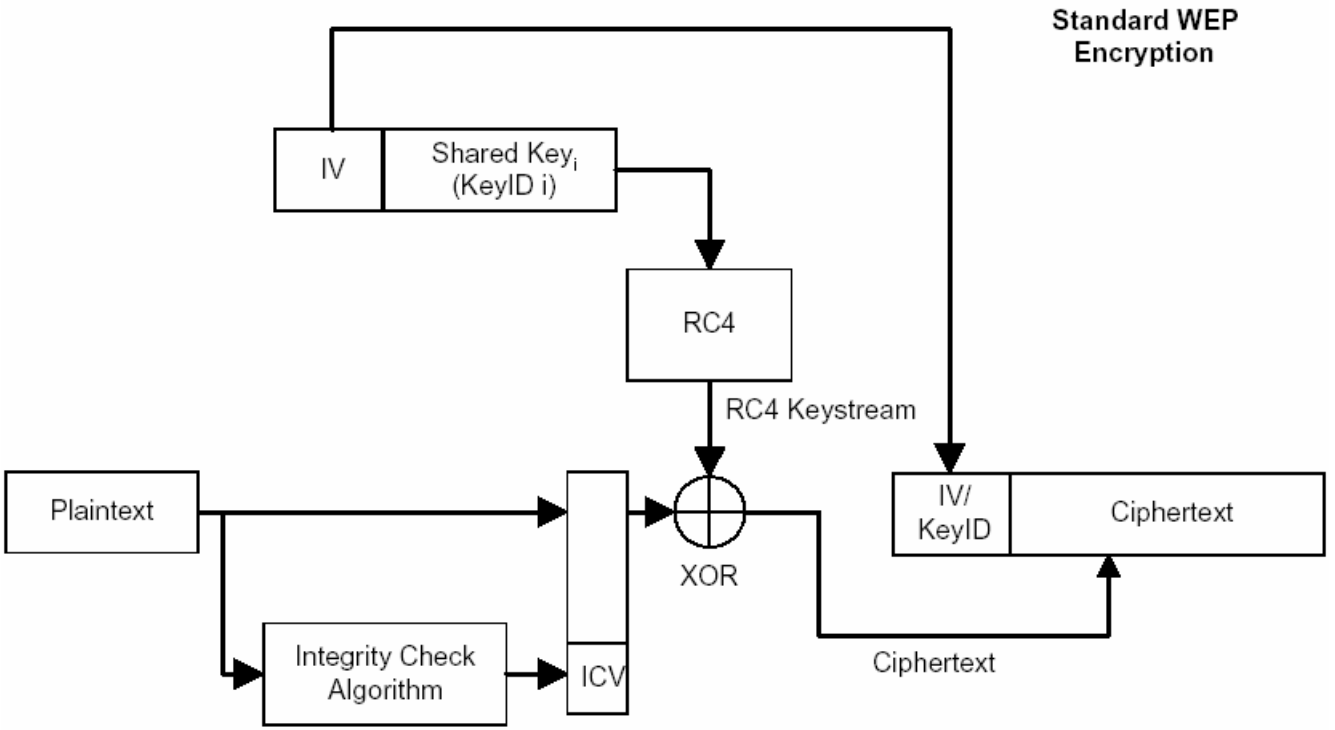
WEP: Generar una trama encriptada

4. Se utiliza el IV y la llave para encriptar el Payload + ICV:



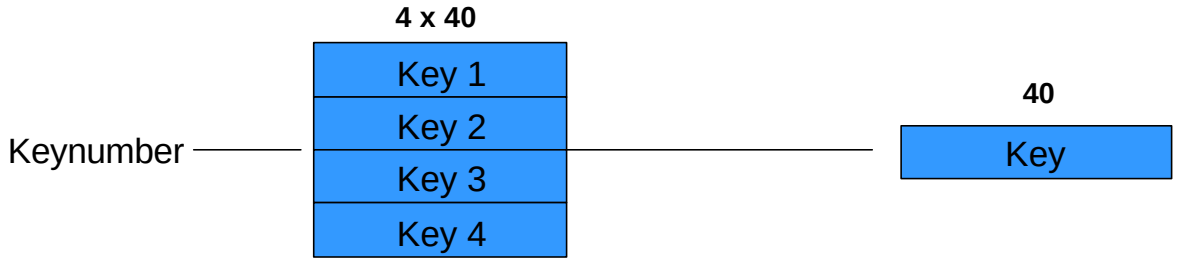
Así queda la trama definitiva:

WEP: Generar una trama encriptada

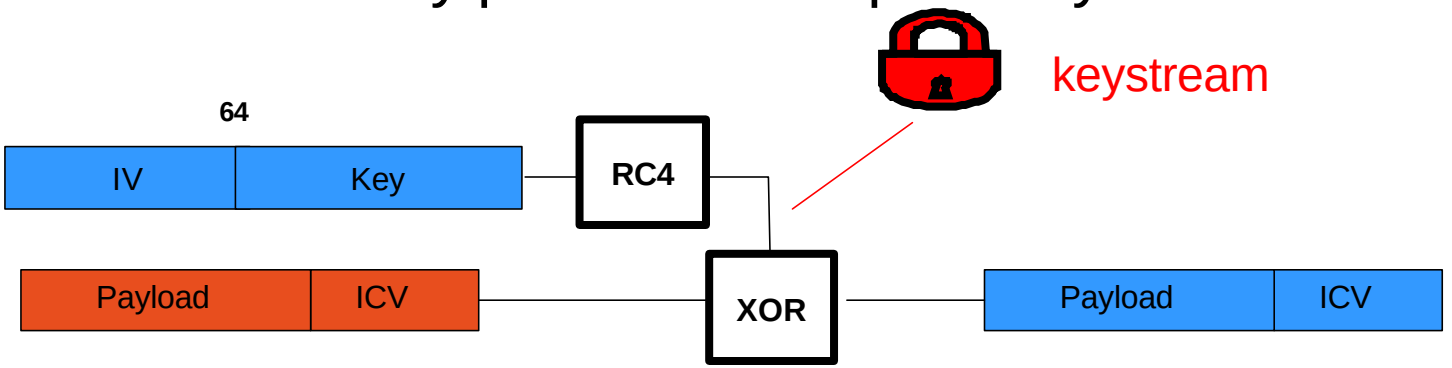


WEP: Desencriptar una trama

1. Se utiliza el numero de llave para seleccionar la llave:

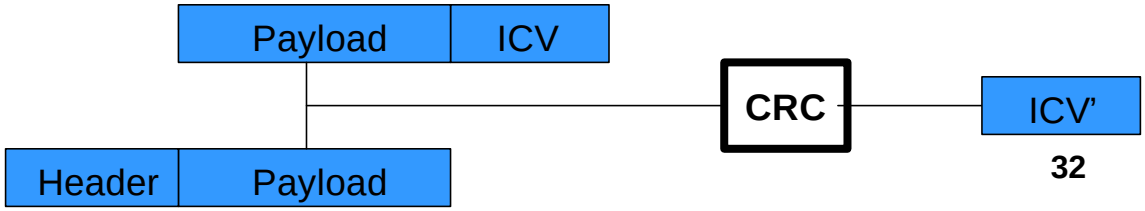


2. Se utiliza IV+Key para desencriptar Payload + ICV:

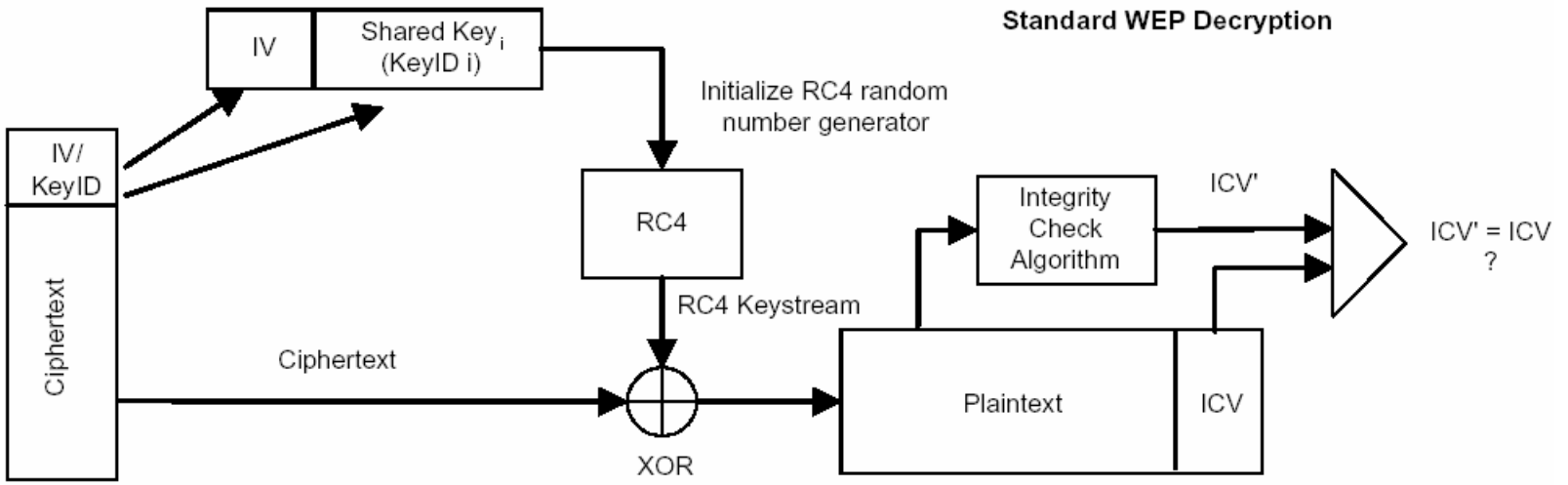


WEP: Desencriptar una trama

3. Se vuelve a recalcular el ICV y se compara con el original:



WEP: Desencriptar una trama



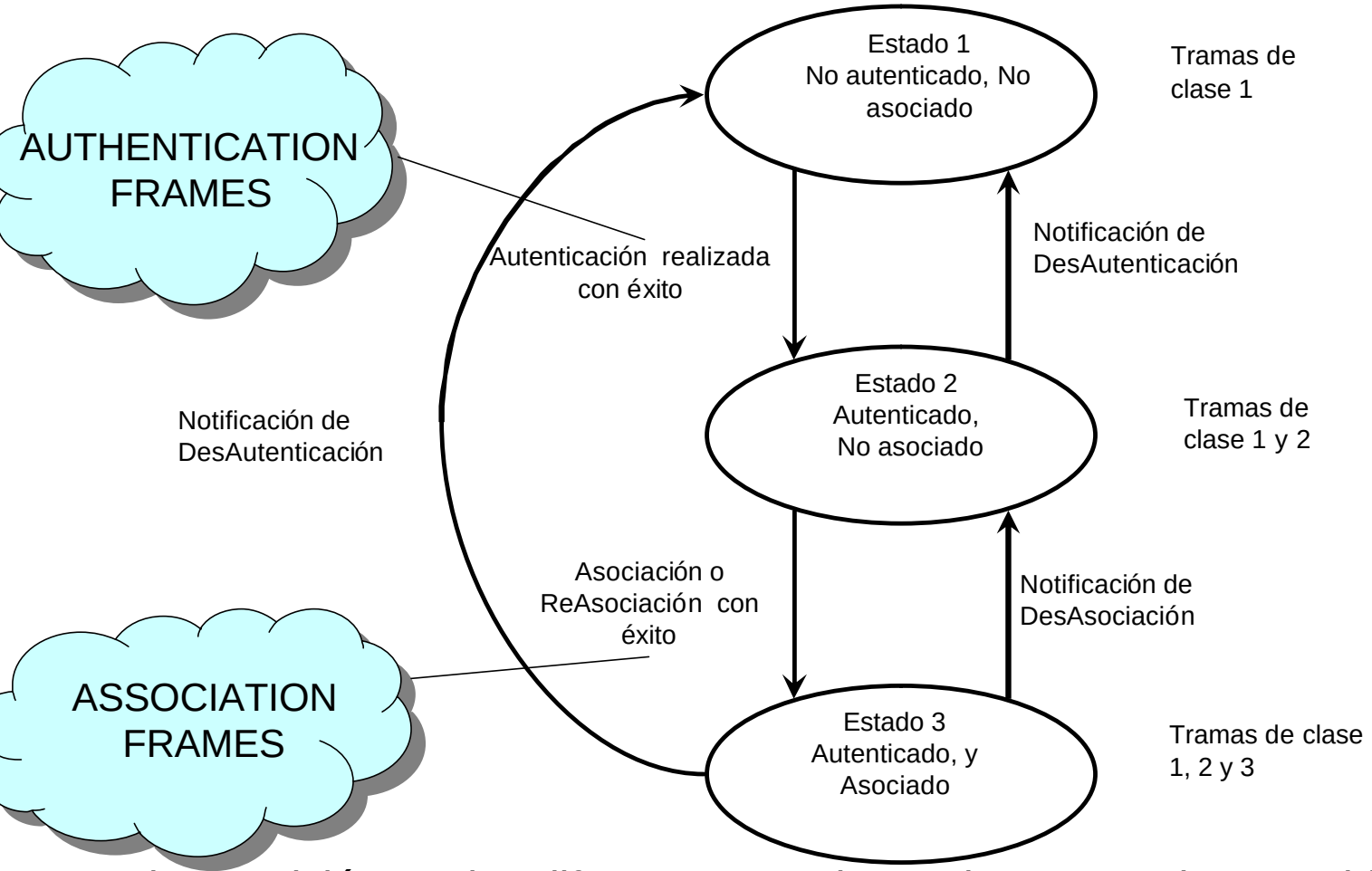
Proceso de conexión a una WLAN

¿Cómo detecta un cliente que hay un AP disponible?

- Los AP transmiten BEACON FRAMES cada cierto intervalo de tiempo fijo.

- Para asociarse con un AP y unirse a una red, un cliente escucha en busca de BEACON FRAMES para identificar AP's.
  - El cliente también puede enviar una trama "PROVE REQUEST" que contenga un ESSID determinado para ver si le responde un AP que tenga el mismo ESSID.

Autenticación y Asociación



- En la transición por los diferentes estados ambas partes intercambian MANAGEMENT FRAMES.

Métodos de autenticación

– Open System Authentication

– Shared Key Authentication

Open System Authentication

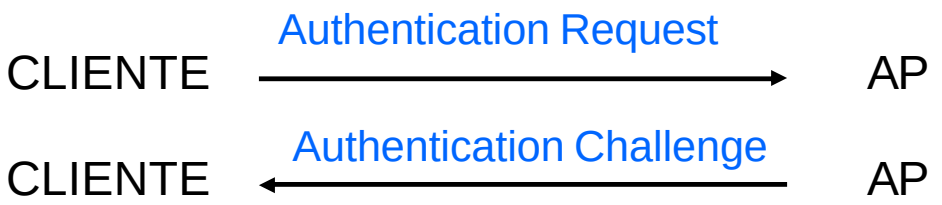
- Protocolo de autenticación por defecto para 802.11b.

- Es un proceso de autenticación NULO:
  - Autentica a cualquier cliente que pide ser autenticado.
  - Las tramas se mandan en texto plano aunque esté activado el cifrado WEP

Shared Key Authentication

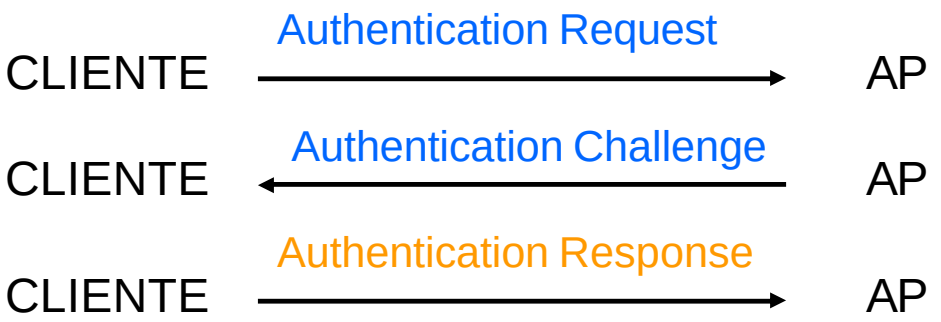
- La estación que quiere autenticarse (cliente), envía una trama **AUTHENTICATION REQUEST** indicando que quiere utilizar una “clave compartida”.

Shared Key Authentication



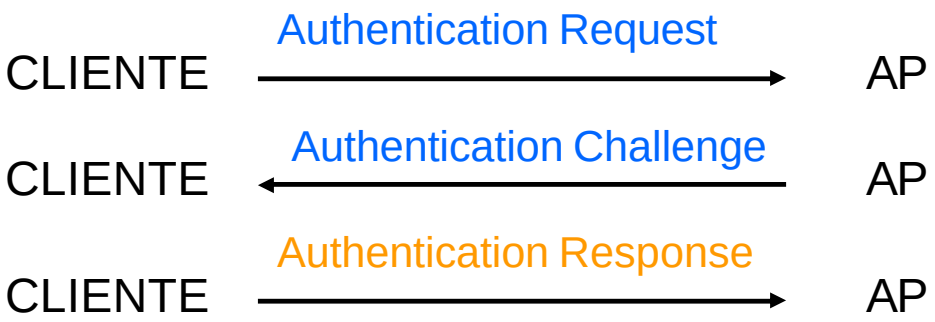
- El destinatario (AP) contesta enviando una trama que contiene 128 octetos de texto (desafío) al cliente.
  - El desafío se genera con la clave compartida y un vector de inicialización (IV) aleatorio utilizando el PRNG.

Shared Key Authentication



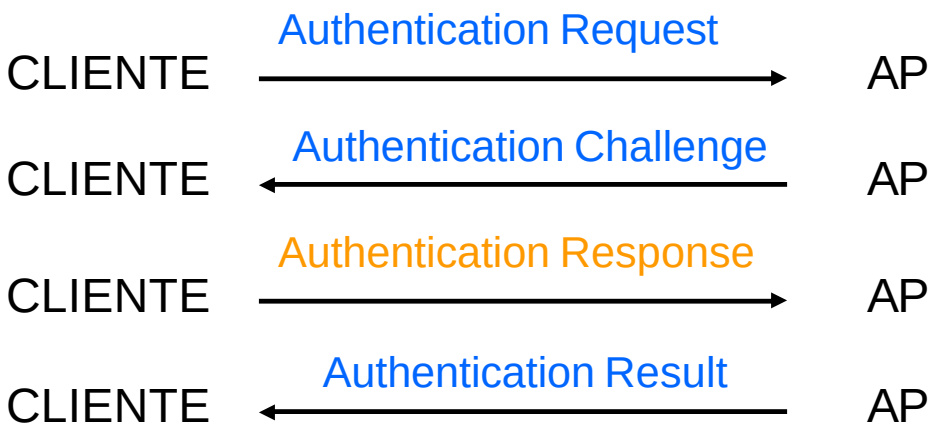
- Una vez el cliente recibe la trama, copia el contenido del texto de desafío en el payload de una nueva trama que encripta con WEP utilizando la *passphrase* y añade un nuevo IV (elegido por el cliente).
- Una vez construida esta nueva trama encriptada, el cliente la envía al AP.

Shared Key Authentication



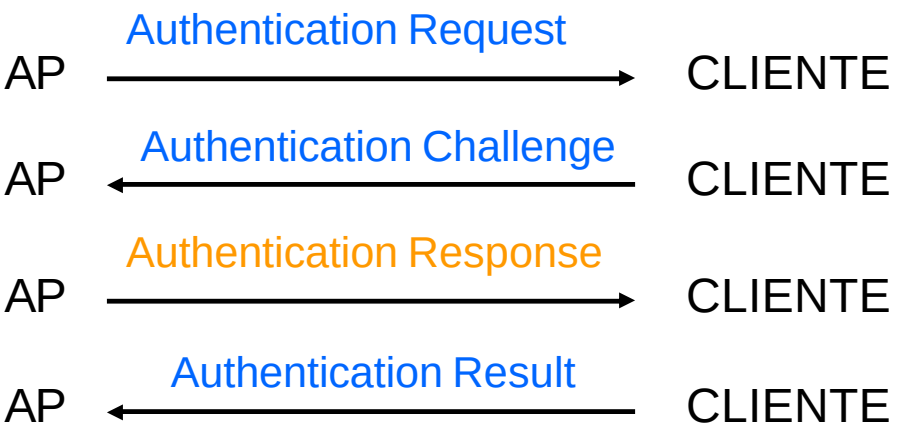
- El AP desencripta la trama recibida y comprueba que:
  - El ICV (Integrity Check Value) sea valido.
  - El texto de desafío concuerde con el enviado en el primer mensaje.

Shared Key Authentication



- Si la comprobación es correcta se produce la autenticación del cliente con el AP

Shared Key Authentication



- Se vuelve a repetir el proceso pero esta vez el primero que manda la trama con el **AUTHENTICATION REQUEST** es el AP, de esta manera se asegura una autenticación mutua.

Vulnerabilidades

- **Deficiencias en la encriptación**
- Deficiencias en la autenticación

Deficiencias en la encriptación

- Características lineales de CRC32
- MIC Independiente de la llave
- Tamaño de IV demasiado corto
- Reutilización de IV

Características lineales del CRC

- El ICV se genera simplemente haciendo un CRC (Cyclic Redundancy Check) de 32 bits del payload de la trama.
- Este mecanismo tiene dos graves problemas:
  - Los CRCs son independientes de la llave utilizada y del IV
  - Los CRCs son lineales: **CRC(m Å k) = CRC(m) Å CRC(k)**
- Debido a que los CRCs son lineales:
  - Se puede generar un ICV válido, ya que el CRC se combina con una operación XOR que también es lineal y esto permite hacer el '*bit flipping*' como veremos a continuación

Características lineales del CRC

- Un atacante intercepta un mensaje m (conocido o no) y lo modifica de forma conocida para producir m':

$$\mathbf{m' = m} \oplus D$$

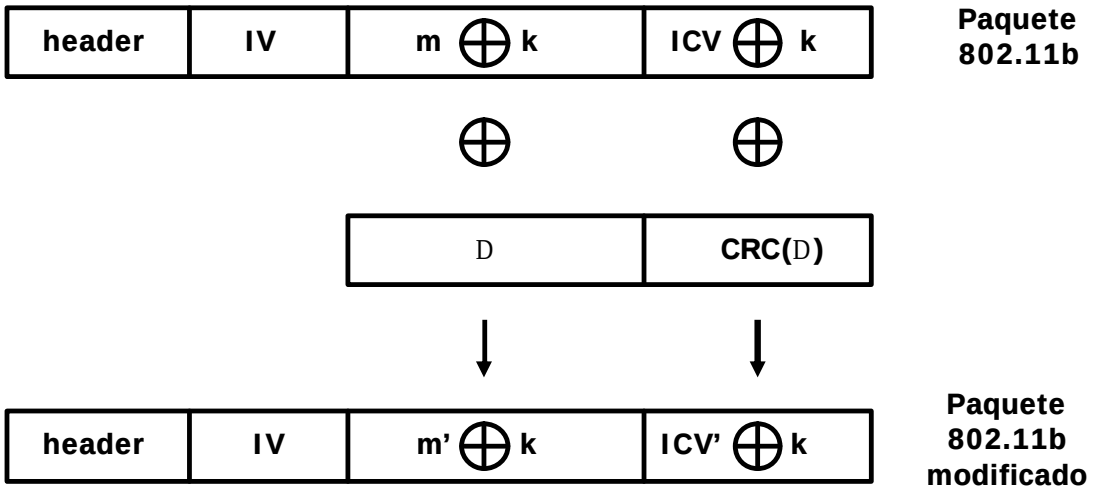
- Como el CRC-32 es lineal, puede generar un nuevo ICV' a partir del ICV de m:

$$\mathbf{ICV' = ICV} \oplus \mathbf{CRC(}D\mathbf{)}$$

- ICV' será valido para el nuevo cyphertext c':

$$\mathbf{c' = c} \oplus D = \mathbf{k} \oplus \mathbf{(m} \oplus D\mathbf{) = k} \oplus \mathbf{m'}$$

Características lineales del CRC



Esta vulnerabilidad fue demostrada teóricamente por
Nikita Borisov, Ian Goldberg y David Wagner (Universidad de Berkeley).

MIC independiente de la llave

- Esta vulnerabilidad en WEP es conocida en inglés como “Lack of keyed MIC”: Ausencia de mecanismo de chequeo de integridad del mensaje (MIC) dependiente de la llave.
- El MIC que utiliza WEP es un simple CRC-32 calculado a partir del payload, por lo tanto no depende de la llave ni del IV.
- Esto da lugar a que conocido el plaintext de un solo paquete encriptado con WEP sea posible inyectar paquetes a la red.

MIC independiente de la llave

Esta vulnerabilidad fue demostrada teóricamente por David Wagner (Berkeley).

- Un atacante captura un paquete **c = m Å k** donde m es conocido
  (por ejemplo, el atacante envía un e-mail a la victima)

- El atacante recupera el flujo pseudo-aleatorio **k = c Å m** para el IV concreto del paquete

- Supongamos que el atacante quiere inyectar un mensaje m', debe realizar lo siguiente:

  ICV' = CRC32(m')

  – El atacante ya puede ensamblar la parte encriptada del paquete:

  c = (m'|ICV') Å k

- El atacante obtiene un paquete válido y listo para ser inyectado a la red:

Tamaño de IV demasiado corto

- El **Vector de Inicialización** (IV) tiene sólo 24 bits de longitud y aparece en claro (sin encriptar).
- Sólo hay 2^{24} (16.777.216) posibles valores de IV.
- 16M de paquetes pueden generarse en pocas horas en una red wireless con tráfico intenso:
  - Un AP que constantemente envíe paquetes de 1500 bytes a 11Mbps, acabará con todo el espacio de IV disponible después de $(1500 \times 8 / (11 \times 10^6)) \times 2^{24}$ = ~1800 segundos, o 5 horas.
- La corta longitud del IV, hace que éste se repita frecuentemente y dé lugar a la posibilidad de realizar ataques estadísticos para recuperar el plaintext gracias a la reutilización del IV.

Reutilización de IV

- WEP no utiliza el algoritmo RC4 “con cuidado”:
  - El IV se repite frecuentemente → Se pueden hacer ataques estadísticos contra cyphertexts con el mismo IV.

- Si un IV se repite, se pone en riesgo la confidencialidad:
  - Supongamos que *P*, *P’* son dos plaintexts encriptados con el mismo IV.
  - Supongamos *Z = RC4(key, IV)*
  - Los dos ciphertexts son *C = P* ⊕ *Z* y *C’ = P’* ⊕ *Z.*
  - Nótese que

    C Å ***C’*** **= (*P* Å *Z*) Å (*P’* Å *Z*) = (*Z* Å *Z*) Å (*P* Å *P’*) = *P* Å *P’***

    por lo que la XOR de ambos plaintexts es conocida.

Si podemos adivinar un plaintext, el otro puede también ser descubierto estadísticamente.

Reutilización de IV

WEP no usa RC4 con cuidado

- Si RC4 no se usa con cuidado, se vuelve inseguro
- **¡El estándar 802.11b especifica que cambiar el IV en cada paquete es opcional!**
- El IV normalmente es un contador que empieza con valor cero y se va incrementando de uno en uno:
  - Rebotar causa la reutilización de IV's
  - Sólo hay 16M de IV's posibles, así que después de interceptar suficientes paquetes, seguro que hay IV's repetidos
- Un atacante capaz de escuchar el tráfico 802.11 puede descifrar ciphertexts interceptados incluso sin conocer la clave.

Vulnerabilidades

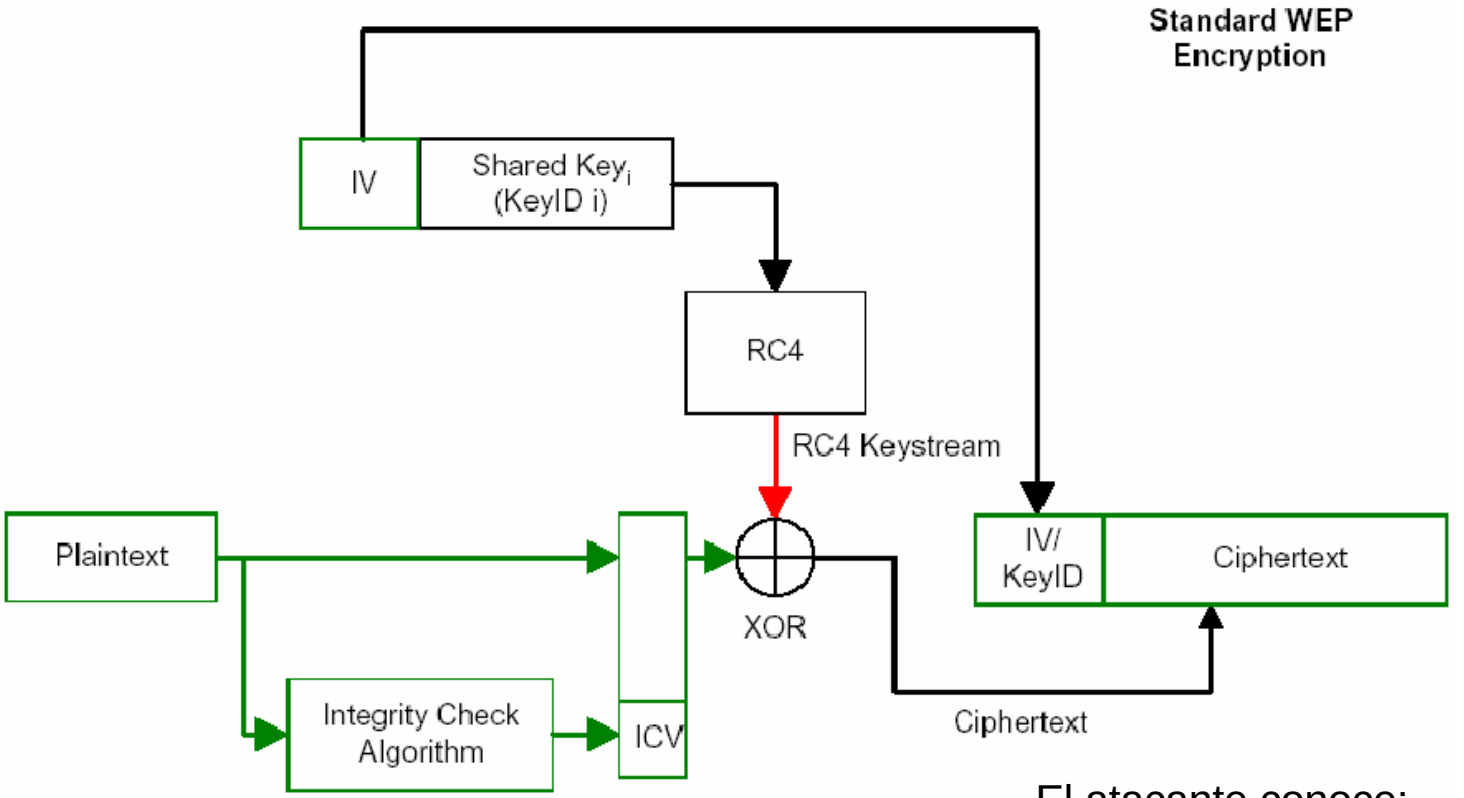
- Deficiencias en la encriptación
- **Deficiencias en la autenticación**

Deficiencias en el método de autenticación *Shared Key*

- El atacante captura el segundo y el tercer *management messages* de una autenticación mutua:

- El segundo mensaje contiene el texto de desafío en claro
- El tercer mensaje contiene el desafío encriptado con la clave compartida.

Deficiencias en el método de autenticación *Shared Key*



El atacante conoce:

- Desafío aleatorio (plaintext, P)
- Desafío encriptado (cyphertext, C)
- IV público

Deficiencias en el método de autenticación *Shared Key*

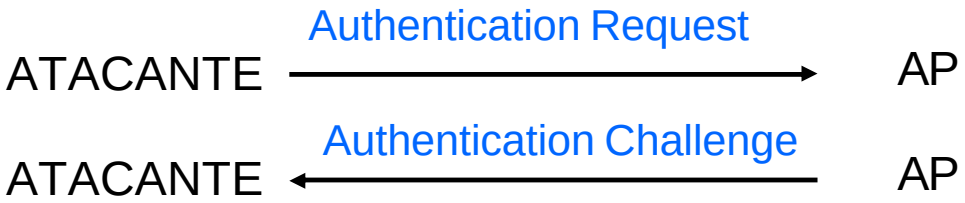
- El atacante puede deducir el flujo pseudo-aleatorio producido usando WEP:

$$WEP_{PR}^{K,IV} = \text{Ciphertext} \otimes \text{Plaintext}$$

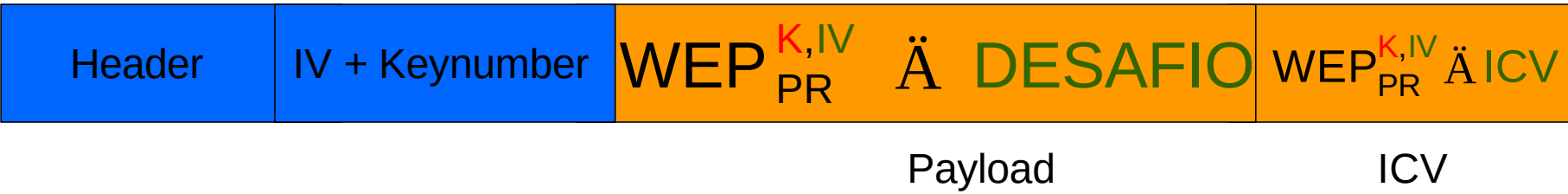
- Todos los elementos excepto el texto de desafío son los mismos para TODAS las *Authentication Responses*.

- El atacante tiene por lo tanto todos los elementos para autenticarse con éxito sin conocer la clave secreta compartida K.

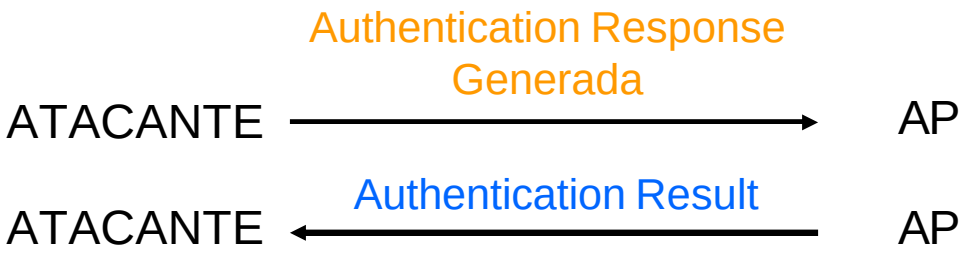
Deficiencias en el método de autenticación *Shared Key*



- El atacante debe generar una trama **AUTHENTICATION RESPONSE** valida:



- Podemos generar el ICV cifrado gracias a que CRC es LINEAL.



Deficiencias en el método de autenticación *Shared Key*

- Una vez finalizado el proceso el atacante sólo esta autenticado, pero todavía no puede utilizar la red ya que no conoce la clave compartida.
- Para poder utilizar la red debe implementar algún ataque al protocolo WEP de los descritos a continuación.

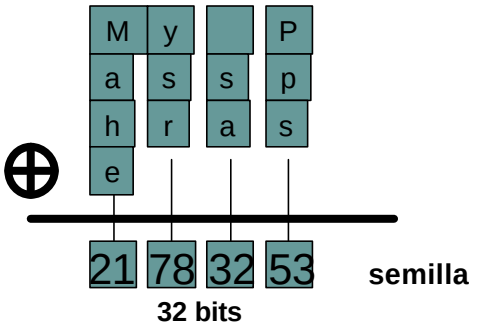
Ataques

- **Ataques al WEP**
- Ataques a redes wireless

Ataques al WEP

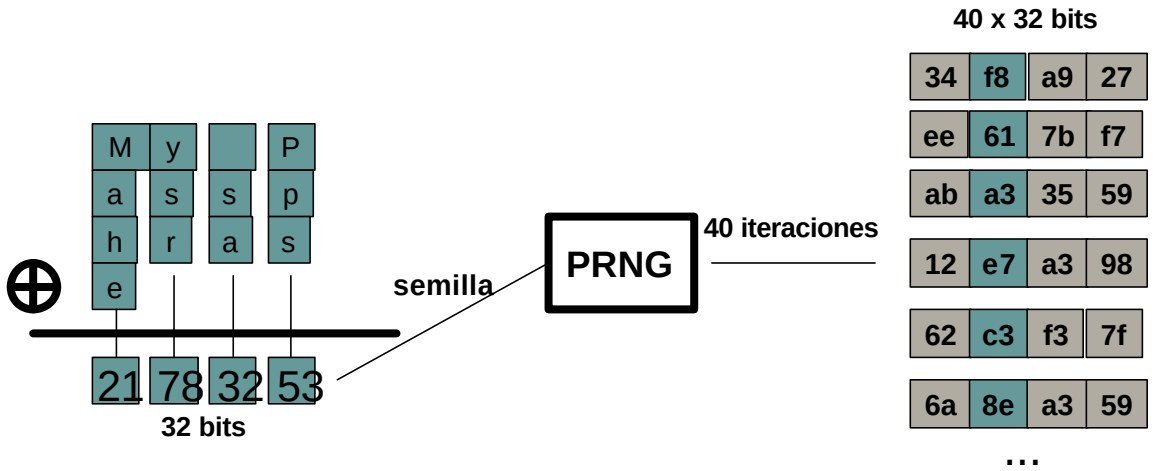
- Ataque de fuerza bruta
- Ataque inductivo Arbaugh
- Debilidades en el algoritmo key Scheduling de RC4

Ataque de fuerza bruta



- La semilla se obtiene a partir de la passphrase
- La passphrase normalmente sólo contiene caracteres ASCII por lo que el bit más alto de cada carácter siempre es ‘0’.
- El resultado de la operación XOR de estos bits también es cero
- Esto provoca una reducción de la entropía de la fuente:
  - Semillas generadas a partir de caracteres ASCII sólo van desde **00:00:00:00** hasta **7F:7F:7F:7F** en lugar de hasta FF:FF:FF:FF.

Ataque de fuerza bruta



- De los 32 bits de la semilla sólo utilizan los bits del 16 al 23
- El PRNG es un generador lineal congruente (LGC: linear congruential generator) de módulo 2^{32}, por lo tanto los bits mas bajos son “menos aleatorios” que los altos:
  - el bit 0 tiene una longitud de ciclo de 2^1, el bit 1 de 2^2, el bit 2 de 2^3, etc.
  - La longitud de ciclo resultante será 2^{24}.
- Con esta longitud de ciclo sólo las semillas que van de **00:00:00:00** a **00:FF:FF:FF** producen llaves únicas.

Ataque de fuerza bruta

- Las semillas sólo llegan hasta 7F:7F:7F:7F
- La última semilla que tiene en cuenta el PRNG es 00:FF:FF:FF
- Sólo necesitamos considerar las semillas desde 00:00:00:00 hasta 00:7F:7F:7F
- La entropía total queda reducida a 21 bits.

Ataque de fuerza bruta

Ataques de fuerza bruta contra la encriptación WEP

- Generar llaves de forma secuencial utilizando semillas de 00:00:00:00 hasta 00:7F:7F:7F.

  Un PIII a 500MHZ tardaría aproximadamente 210 días en encontrar la llave. (Se puede usar computación en paralelo para obtener la llave en un tiempo más razonable)

- Ataque con diccionario:

  Si la passphrase utilizada está en el diccionario conseguimos reducir sustancialmente el tiempo necesario para encontrarla.

Ataque Inductivo Arbaugh

- Permite desencriptar el tráfico de cifrado de una WLAN en tiempo real.
- Se basa en:
  - Características Lineales de CRC
  - MIC independiente de la llave
- Demostrado por William A. Arbaugh (Universidad de Maryland).

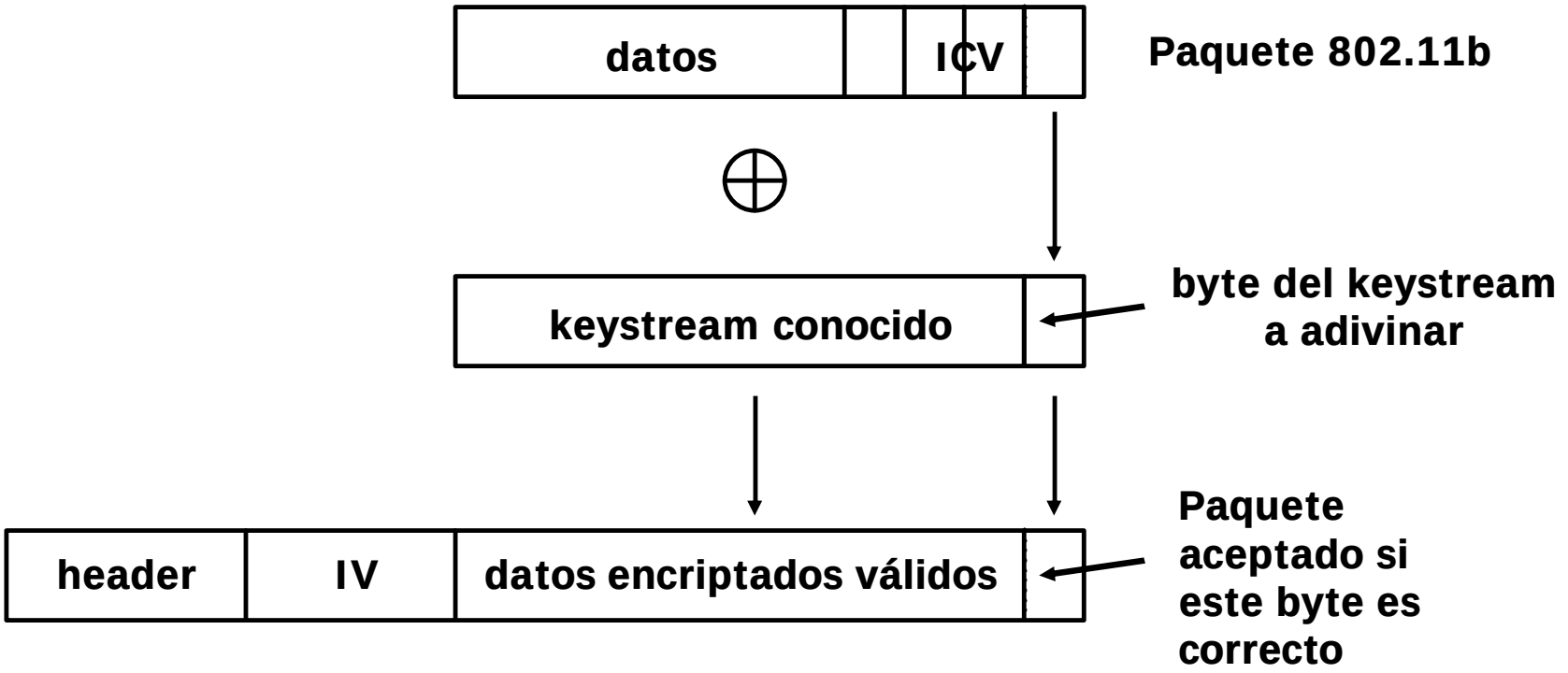
Ataque Inductivo Arbaugh

- Hay que conocer parte del plaintext que viaja encriptado en una trama:
  - Identificando mensajes “*DHCP Discover*” (Origen 0.0.0.0, destino 255.255.255.255)
- Conocemos 24 bytes de keystream para el IV concreto del paquete
  - Keystream = Plaintext Å Cyphertext
- Creamos un paquete (x ej “ping”) que tenga 24 - 3 = 21 bytes de longitud y hacemos una XOR del paquete generado con el keystream que conocemos.

Ataque Inductivo Arbaugh

- Calculamos el ICV del paquete generado y añadimos sólo los primeros 3 bytes
- Realizamos una XOR con el resto del keystream y añadimos el último byte del ICV al final del paquete tratando de adivinar el siguiente byte del keystream
- Enviamos el paquete:
  - Si recibimos respuesta (“echo reply” en el ejemplo) conocemos un byte más del keystream para el IV concreto
  - Si no recibimos respuesta, iteramos con las 255 posibilidades restantes hasta encontrarlo.

Ataque Inductivo Arbaugh



Ataque Inductivo Arbaugh

- 100 pruebas/segundo = 32 minutos → keystream de 1500 bytes valido para un IV determinado.
- Con un keystream entero, los otros son fáciles de obtener:
  - Generar un paquete del cual se devuelva respuesta (broadcast pings)
  - Conocemos el plaintext de la respuesta y recibimos diferentes IV's
- Tabla de <IV – keystream> para desencriptar en tiempo real.
  - La tabla ocupará 2^{24} x 1500 = 24GB.
  - Un solo atacante tardará una media de 30 horas en construir la tabla.
- Cuando el atacante recibe un paquete mira en la tabla a que keystream corresponde el IV recibido y hace una XOR del keystream con el cyphertext del paquete para obtener el plaintext.

Debilidades en el algoritmo key Scheduling de RC4

- Permite adivinar la llave WEP
- Se basa en:
  - Monitorización **pasiva** de la transmisión
  - Recolecta paquetes “debiles”
- Una vez se han recolectado suficientes paquetes, es posible *adivinar* la llave utilizada para realizar la encriptación
- Publicado en Agosto del 2001 por:

  Scott Fluhrer, Itsik Mantin y Adi Shamir

Debilidades en el algoritmo key Scheduling de RC4

- IV's débiles: (weak IV's)
  - IV's en los que no hay información de la llave en el keystream
  - Los autores lo llaman "*resolved condition*" o condición resuelta.
- Cada uno de estos *paquetes resueltos* sólo tiene ausencia de información de un byte de la llave.
- Este byte debe ser adivinado para que el siguiente paquete pueda ofrecer información del siguiente byte de la llave.
- Se buscan los IVs débiles que cumplen esta condición.
- Hay una posibilidad del 5% de adivinar el byte de la llave correctamente cuando encontramos un paquete con un IV débil (*paquete resuelto*)

Debilidades en el algoritmo key Scheduling de RC4

- Cuando se han recolectado suficientes IVs débiles para un valor concreto de un byte de la llave, el análisis estadístico muestra una tendencia hacia un valor en particular para ese byte de la llave.

- Se le da una puntuación a cada una de las 256 posibilidades según la probabilidad de ser el valor correcto.

- La llave se intenta adivinar a partir de los valores con mayor puntuación en el análisis estadístico
  - Hay un 95% de posibilidades de que un IV no revele información sobre un byte de la llave

Debilidades en el algoritmo key Scheduling de RC4

- Los IV's debiles no estan distribuidos de forma lineal a través del espacio de IV's.
- El número de paquetes que necesitamos recolectar antes de descubrir un byte de la llave varía en función de en que valor se encuentre el contador de IV's de las tarjetas que estemos monitorizando.

3er byte del IV	IV's débiles	Num. de IV's
00 – 0C	3.000	852.000
0D – EE	3.000	14.800.000
EF – FF	3.000	1.100.000

Hay 9K IV's débiles en los 16M de IV's posibles

Debilidades en el algoritmo key Scheduling de RC4

- ¿Cuántos paquetes encriptados necesitamos recolectar para crackear la llave WEP?
  - La mayoría de las llaves pueden ser adivinadas después de encontrar aproximadamente 2000 paquetes resueltos
  - Algunas llaves requieren que capturemos incluso más de 4000 paquetes resueltos
- Podremos adivinar la llave después de recolectar de 5 a 10 millones de paquetes encriptados

Debilidades en el algoritmo key Scheduling de RC4

- ¿Cuánto tiempo tardaremos en recolectar todos estos paquetes?

Para hacernos una idea:

- Una situación normal:
  - 4 PC's → navegando continuamente → 1 millon de paquetes/día → aprox. 120 paquetes resueltos/día → tardaremos aprox. 16 dias
- En una red saturada tardaríamos aprox. 1 día
- En una red MUY saturada de 2 a 6 horas.

Conclusiones

	Fuerza bruta	**Ataque inductivo Arbaugh**	**Debilidades en el key sheduling de RC4**
Ventajas	Conviene utilizarlo con diccionario cuando: El atacante conoce varias passphrases posibles: el nombre de la empresa, el nombre del perro del admin… El atacante conoce las letras de la passphrase pero no el orden	Si el IV siempre es 0 es el metodo más rápido (~32 min) y no necesita espacio en el HD (~1500bytes)	Permite adivinar la llave WEP en un tiempo relativamente pequeño Al ser pasivo es imposible detectar que estamos realizando el ataque
Inconvenientes	Se necesita mucho tiempo para encontrar la llave	Hay 16Millones de IV's posibles → necesita mucho espacio en el HD (24 Gb) y mucho tiempo	No puede realizarse si el IV siempre es 0. Si la red no está saturada tarda mucho tiempo en encontrar la llave
Se puede realizar de forma pasiva:	Si → MUY lento No → Un poco más rápido	No	Si

Ataques

– Ataques al WEP

– **Ataques a redes wireless**

Ataques a redes wireless

- Romper ACL’s basados en MAC
- Ataque de Denegación de Servicio (DoS)
- Descubrir ESSID ocultados
- Ataque *Man in the middle*
- Ataque *ARP Poisoning*

Romper ACL’s basados en MAC

- Una de las medidas más comunes que se utiliza para securizar una red wireless es restringir las máquinas que podrán comunicarse con el AP haciendo filtrado por dirección MAC
- Para llevar a cabo el ataque:
  - Esnifamos y vemos la MAC de cualquiera de los clientes
  - Nos ponemos su MAC y ya habremos saltado la restricción

```
# ifconfig <interface> hw ether <mac address>
# setmac <interface> <mac address>
```

- Si queremos podemos “anular” a la máquina que le hemos “robado” la dirección MAC con un ataque DoS.

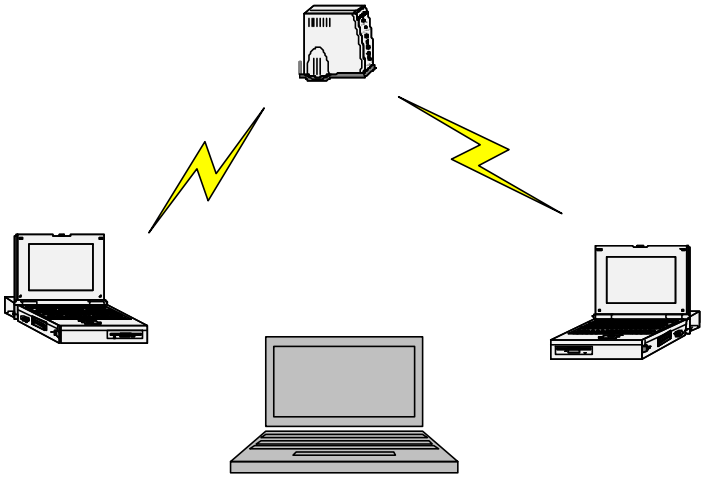
Ataque de Denegación de Servicio (DoS)

- Para realizar el ataque:
  - Esnifar y ver cual es la dirección MAC del AP
  - Nos ponemos la MAC del AP, es decir nos hacemos pasar por AP.
  - Para denegarle el servicio a un cliente mandamos continuamente notificaciones de desasociación o desautenticación (*management frames*).
  - Si en lugar de a un solo cliente queremos denegar el servicio a todos los clientes de la WLAN, mandamos estas tramas a la dirección MAC de broadcast.

Descubrir ESSID ocultados

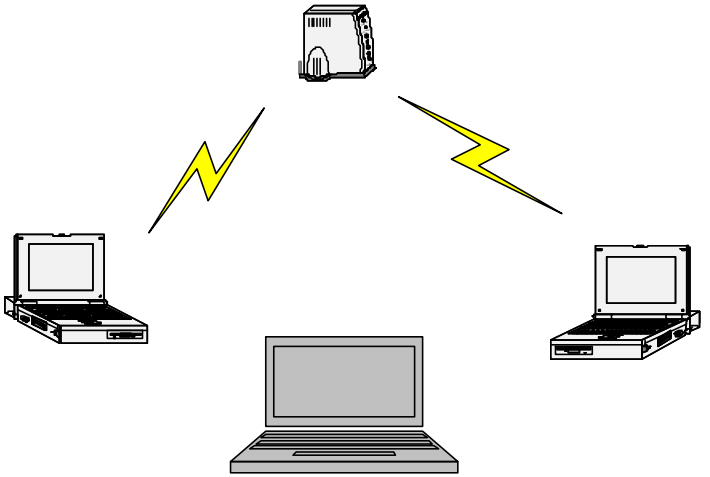
- Una medida de seguridad bastante común es “ocultar” el ESID:
  - el AP no manda BEACON FRAMES o los manda sin ESSID
- Para descubrir el ESSID:
  - Esnifar y esperar a que un cliente se conecte → veríamos el ESSID en la trama PROVE REQUEST del cliente
  - Podemos “provocar” la desconexión de un cliente:
    - Nos ponemos la dirección física del AP y mandamos una trama DISASSOC a la dirección MAC del cliente (o a la de broadcast) → el cliente intentará volver a asociarse con lo que podremos ver el ESSID en los *management frames*.

Ataque Man in the middle



- Consiste en convencer al cliente (la victima) de que el host que hay en el medio (el atacante) es el AP, y hacer lo contrario con el AP, es decir, hacerle creer al AP que el atacante es el cliente.

Ataque Man in the middle



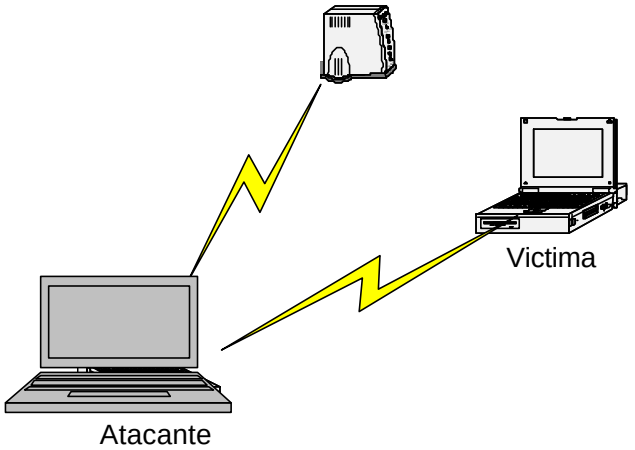
- Esnifamos para obtener:
  - El ESSID de la red (si esta ocultado, usaremos el método anterior)
  - La dirección MAC del AP
  - La dirección MAC de la victima

Ataque Man in the middle

- Mandamos tramas DEAUTH a la victima → La tarjeta wi-fi de la victima empezará a escanear canales en busca de un AP para poderse autenticar.

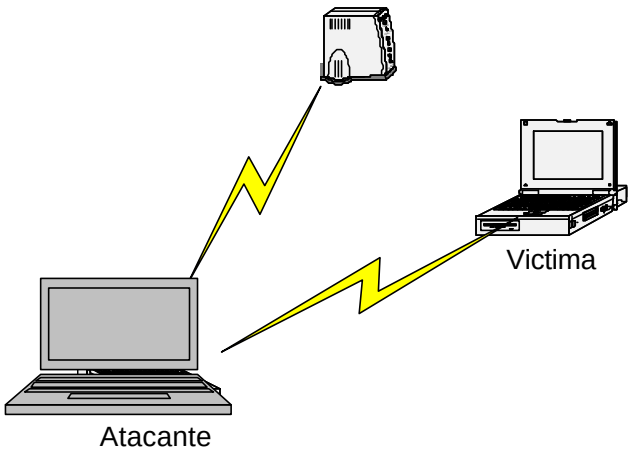
- El atacante hace creer a la victima que él es el AP real, utilizando la misma MAC y el mismo ESSID que el AP al que la victima estaba asociado, pero operando por un canal distinto (modo master).

Ataque Man in the middle



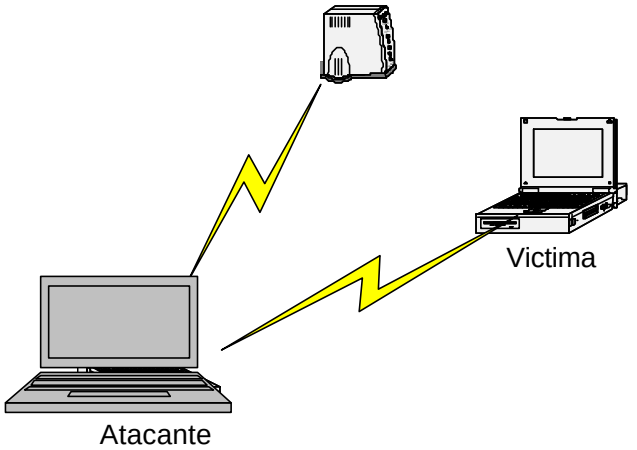
- Por otra parte, el atacante se asocia con el AP real, utilizando la dirección MAC de la victima
- De esta manera hemos conseguido insertar al atacante entre la victima y el AP

Ataque Man in the middle



- Todos los datos que viajan entre la victima y el AP pasan a través del atacante
- El ataque ha sido realizado a nivel de enlace → el atacante puede ver, capturar e incluso modificar las tramas en los niveles superiores del modelo OSI

Ataque Man in the middle



- Muchas soluciones de seguridad están pensadas asumiendo que las capas 1 y 2 son seguras, esto es incierto para las redes wireless
- Hay que ir con mucho cuidado en implementaciones de VPN que no realizan las comprobaciones necesarias de autenticación para protegerse de ataques *Man in the middle* en redes wireless.

Ataque ARP Poisoning

- **Objetivo**: Envenenar la cache de ARP's de un sistema situado en la LAN cableada detrás de la WLAN para realizar un ataque Man In The Middle.
- Sólo puede llevarse a cabo cuando el atacante está "*conectado a la misma LAN lógica*" que las victimas:
  - Efectividad limitada a redes conectadas con switches, hubs y bridges, pero no routers.
  - La mayoría de los AP actúan como bridges transparentes de capa 2 → permite que los paquetes ARP pasen de la red wireless hacia la LAN donde está conectado el AP y viceversa.

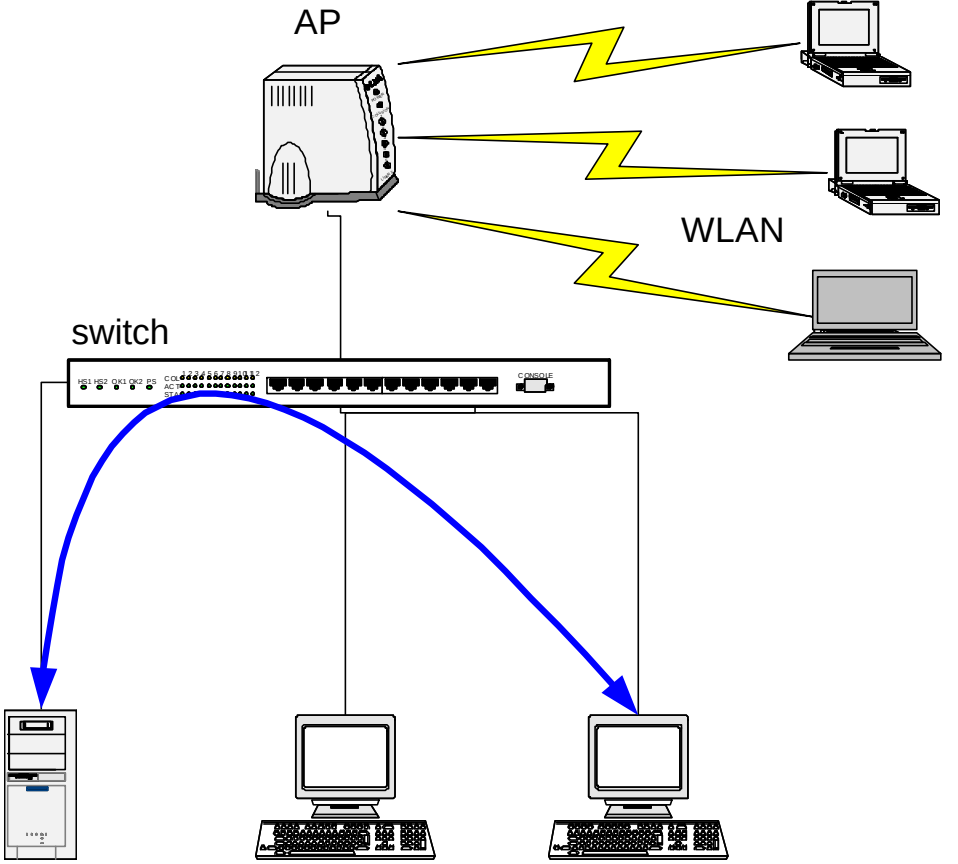


Tabla ARP – PC 1

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.3	3C:2A:4A:59:1A:7A

Tabla ARP – PC 3

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.1	6A:8F:5E:99:2B:78

Tabla ARP – PC 2

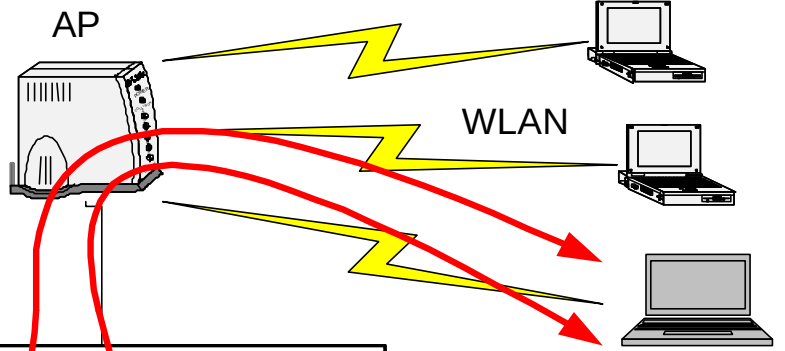
Dirección IP	Dirección MAC
192.168.1.1	6A:8F:5E:99:2B:78
192.168.1.3	3C:2A:4A:59:1A:7A

(In)seguridad en redes 802.11b

Pau Oliva <pof@eSlack.org>

Ataque ARP Poisoning

- El atacante manda paquetes *ARP REPLY* a PC 2 diciendo que la dirección IP de PC 1 la tiene la MAC del atacante → de esta manera consigue “envenenar” la caché de ARP’s de PC 2.
- Luego realiza la misma operación atacando a PC 1 y haciéndole creer que la dirección IP de PC 2 la tiene también su propia MAC.
- PC 1 y PC 2 actualizan su caché de acuerdo a la información que el atacante ha inyectado a la red.
- Como el switch y el AP forman parte del mismo dominio de broadcast, los paquetes ARP pasan de la WLAN a la LAN con cables sin ningún problema.



PC 1
192.168.1.1

PC 2
192.168.1.2

PC 3
192.168.1.3

Tabla ARP – PC 1

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.3	F0:F0:F0:F0:F0:F0

Tabla ARP – PC 3

Dirección IP	Dirección MAC
192.168.1.2	3B:AF:44:F8:2C:B3
192.168.1.1	F0:F0:F0:F0:F0:F0

Tabla ARP – PC 2

Dirección IP	Dirección MAC
192.168.1.1	6A:8F:5E:99:2B:78
192.168.1.3	3C:2A:4A:59:1A:7A

Ataque ARP Poisoning

- Podríamos frenar este ataque creando dos VLAN's en el switch:
  - Una para la boca a la que está conectado el AP
  - Otra para el resto de máquinas

- Otra forma de frenarlo sería utilizando tablas de ARP estáticas.

Conclusiones

- **Confidencialidad**
  - Tu red es vulnerable desde una distancia de 10 kilómetros.
  - Todo tu tráfico puede ser desencriptado fácilmente.
- **Control de Acceso**
  - Cualquier persona puede unirse a tu red cuando quiera.
  - Seguramente incluso a tu red interna.
- **Integridad**
  - Todo tu tráfico es vulnerable a ser modificado y re-enviado.
  - Hackeo tu servidor DHCP → todo tu tráfico se enruta ahora a través de mi portátil.
- **Fiabilidad**
  - Tu red puede venirse abajo en cualquier momento.

Posibles soluciones

Posibles soluciones

ATAQUE	WEP	WEP 802.1x con EAP TLS	AES 802.1x con EAP TLS
Unintentional IV reuse	SI	SI	NO
Intentional IV reuse	SI	SI	NO
Realtime decryption	SI	NO	NO
Known plaintext	SI	SI	NO
Partial known plaintext	SI	SI	NO
Authentication forging	SI	NO	NO
Denial of Service	SI	NO*	NO*
Dictionary attack	SI	SI	NO

Preguntas?