Seguretat en xarxes WLAN

Xavier Perramon Tornil

PID_00191681

Universitat Oberta de Catalunya

Índex

Introducció

El problema específic de les xarxes sense fils és que, a diferència de les xarxes amb fils, l'accés al mitjà de transmissió és lliure, en el sentit que no cal fer res especial per a connectar-s'hi físicament. Per exemple, qualsevol usuari que tingui un dispositiu Wi-Fi en mode promiscu pot veure les trames que es transmeten al seu entorn, sense cap altra limitació que la distància a l'estació emissora. Aquest fet té conseqüències molt importants, especialment pel que fa a la seguretat de la informació que es tramet en aquest tipus de xarxes. Si la informació que viatja en aquest tipus de xarxes no es protegeix convenientment, un atacant pot aconseguir-la sense gaires esforços.

En aquest mòdul didàctic veurem quins mecanismes existeixen per a la protecció de xarxes que segueixen l'estàndard IEEE 802.11. En primer lloc, analitzarem el protocol WEP que va ser el primer protocol de seguretat per a l'estàndard. Veurem que, malgrat aportar un nivell de seguretat superior a enviar la informació en clar, les vulnerabilitats conegudes d'aquest protocol fan que no sigui recomanable la seva utilització. En concret, veurem diferents tipus d'atacs, i també algunes eines que permeten implementar-los.

Finalment, analitzarem el protocol WPA, en les seves diferents variants de funcionament, que permet superar les limitacions i els atacs que existeixen sobre el protocol WEP.

Objectius

Els materials associats a aquest mòdul us permetran assolir els objectius següents:

1. Conèixer els components bàsics d'una xarxa WLAN.

2. Entendre les problemàtiques de seguretat i els atacs a xarxes WLAN.

3. Comprendre el funcionament de sistema de protecció WEP.

4. Identificar les limitacions de seguretat del protocol WEP.

5. Entendre el funcionament del protocol WPA i també els seus diferents modes de funcionament.

1. Conceptes bàsics de les xarxes Wi-Fi

L'estàndard més utilitzat actualment per a les comunicacions en xarxes locals sense fil (WLAN) és l'anomenat *IEEE 802.11*, més conegut com a *Wi-Fi*. La primera versió de l'especificació, que data del 1997, permetia comunicacions de fins a 2 Mbit/s. Des de llavors s'hi han anat afegint extensions que accepten velocitats màximes de transmissió cada vegada més altes: 11 Mbit/s (802.11b), 54 Mbit/s (802.11a i 802.11g) i 600 Mbit/s (802.11n).

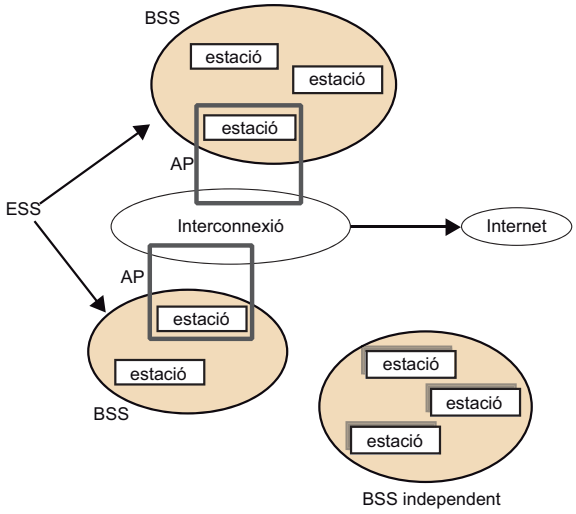
Un dels criteris de disseny inicials d'aquest estàndard era facilitar la interoperabilitat, cosa que no sempre ha estat del tot compatible amb la seguretat. Les primeres versions basaven la protecció de les comunicacions en el protocol WEP, que aviat es va demostrar que era massa feble. L'any 2004 es va publicar la norma IEEE 802.11i amb l'objectiu de corregir les deficiències del sistema de seguretat WEP.

WLAN

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

IEEE 802.11 permet la comunicació entre dispositius, anomenats **estacions**, que tinguin una interfície de xarxa sense fil. Cada interfície té una adreça MAC de 48 bits amb el mateix format que les adreces Ethernet. Dues estacions o més que, per proximitat, poden comunicar-se entre si formen un *basic service set* (BSS). Es distingeixen dos tipus de BSS: els independents i els infraestructurals. Un **BSS independent**, també conegut com a xarxa *ad hoc*, és una xarxa aïllada en què les úniques comunicacions possibles són les directes d'una estació a una altra. En un **BSS infraestructural**, en canvi, hi ha una estació específica anomenada **punt d'accés** (AP) que permet la interconnexió amb altres xarxes, amb fil o sense. Un *extended service set* (ESS) és un conjunt d'un BSS infraestructural, o més, interconnectats mitjançant els seus AP. Des del punt de vista de les estacions, l'ESS funciona com si fos un únic BSS.

Figura 1. Components de les xarxes Wi-Fi



Configuracions Wi-Fi

La configuració típica de les xarxes Wi-Fi domèstiques és la d'un encaminador que d'una banda dóna accés a Internet via ADSL, i d'altra banda actua com a AP permetent la connexió des de les estacions que es trobin en el seu radi d'abast. En aquesta configuració hi ha un ESS format per un únic BSS. En una xarxa Wi-Fi corporativa, en canvi, és habitual tenir diversos AP en diferents parts d'un edifici: en aquest cas tots els BSS normalment pertanyen a un mateix ESS.

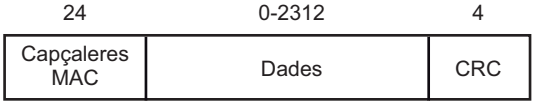
Els BSS s'identifiquen amb un BSSID, que en els infraestructurals és l'adreça MAC del seu AP. Els ESS tenen un identificador de format lliure de fins a 32 bytes. S'utilitza el terme genèric *service set identifier* (SSID) per a referir-se a l'identificador d'un BSS independent o d'un ESS.

Les estacions poden entrar i sortir d'un BSS de manera dinàmica. En un BSS infraestructural l'AP anuncia la seva presència enviant periòdicament **trames balisa**, típicament cada 100 ms. Els diferents camps d'una balisa indiquen l'SSID de la xarxa, les velocitats de transmissió que permet, etc.

Una estació passa a ser membre d'un BSS infraestructural quan estableix una **associació** amb l'AP corresponent. En cada moment una estació només pot estar associada a un AP. Un cop establerta l'associació, l'estació normalment envia i rep totes les seves trames per mitjà d'aquest AP. Abans, però, per a poder fer l'associació i entrar al BSS cal que prèviament l'estació hagi fet una **autenticació** davant l'AP.

Les trames que poden enviar i rebre les estacions Wi-Fi pertanyen a un d'aquests tres tipus: trames de **gestió**, trames de **dades**, i trames de **control**. Les trames de gestió inclouen entre d'altres les balises, les trames d'autenticació i desautenticació, i les d'associació i dissociació.

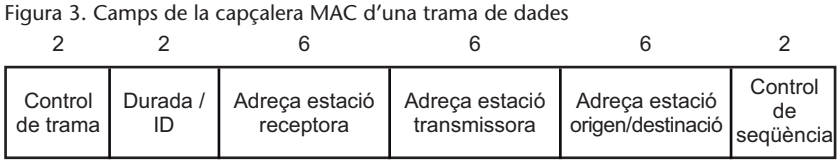
Figura 2. Format de les trames de dades



Les trames de dades que es transmeten en un BSS infraestructural consten de les parts següents:

- La capçalera MAC, de 24 bytes, amb l'estructura que es descriu a continuació.
- Les dades que s'envien en la trama, amb una longitud de fins a 2.304 bytes, o 2.312 si les dades inclouen encapsulament WEP.
- Un codi de comprovació d'errors, que és un codi CRC de 32 bits calculat sobre la capçalera i les dades.

Figura 3. Camps de la capçalera MAC d'una trama de dades



La capçalera MAC d'aquestes trames està formada pels camps següents:

- **Control de trama**: inclou diversos subcamps com ara la versió del protocol, tipus i subtipus de trama, un indicador (*flag*) per a indicar si és l'últim fragment d'una trama fragmentada, etc.

- **Durada/ID**: en una trama de dades aquest camp s'utilitza per a indicar el temps en què s'ha de transmetre una trama de control ACK.

- **Adreça de l'estació receptora**: indica l'estació a la qual s'envia directament la trama.

- **Adreça de l'estació transmissora**: indica l'estació que ha enviat aquesta trama.

- **Adreça de l'estació origen/destinació**. Si és una trama enviada des d'una estació a l'AP, aquest camp indica la destinació final, que pot ser el mateix AP (i llavors aquesta tercera adreça coincideix amb la primera) o bé una altra estació a la qual l'AP retransmetrà la trama. Per contra, si és una trama enviada per l'AP a una estació, aquest camp indica l'origen de la trama, que pot ser el mateix AP (i la tercera adreça coincidirà amb la segona) o bé una altra estació que ha enviat la trama per mitjà de l'AP.

- **Control de seqüència**: inclou un número de seqüència de la trama i un número de fragment.

2. Mètodes d'autenticació de les estacions Wi-Fi

Les primeres versions de l'estàndard Wi-Fi anteriors a l'IEEE 802.11i, seguint el criteri de simplicitat i de facilitar al màxim la interoperabilitat, preveien dos tipus d'autenticació de les estacions Wi-Fi davant l'AP.

- **Autenticació de sistema obert**. Aquesta autenticació, si l'AP està configurat per a permetre-la, és molt simple: cada estació que sol·licita l'autenticació rep automàticament la confirmació. Per això també és conegut com a *algorisme d'autenticació nul*.

  L'avantatge d'aquesta autenticació és que no cal que les estacions facin res especial per a completar-la. Així s'assoleix l'objectiu de facilitar la connexió de les estacions que s'incorporin a la xarxa.

- **Autenticació de clau compartida**. Aquest mètode d'autenticació s'utilitza juntament amb el sistema de xifratge WEP. En aquest cas l'AP fa ús d'una clau WEP que té preconfigurada i que només haurien de conèixer les estacions que s'hi hagin d'autenticar. Quan una estació sol·licita l'autenticació, l'AP li envia un missatge amb 128 bytes aleatoris, i l'estació ha de respondre amb una trama que contingui aquest mateix missatge, encriptada amb la clau WEP.

  Es tracta, doncs, d'un protocol de repte-resposta amb clau simètrica. El problema que presenta és que està basat en el xifratge WEP i, com veurem més endavant, descobrir una clau WEP no és excessivament complicat per a un atacant que pugui capturar una quantitat suficient de trames xifrades.

  Però, a més, aquest protocol d'autenticació té un altre problema, i és que la resposta encriptada no inclou cap identificador de l'estació que es vol autenticar. Com veurem també més endavant, això permet que un atacant que capturi un sol intercanvi de missatges utilitzi les dades capturades per a autenticar-se amb èxit davant l'AP.

A partir de la publicació de l'estàndard IEEE 802.11i, l'ús de l'autenticació de clau compartida està desaconsellat, i només es preveu en situacions en què es vulgui mantenir la compatibilitat amb sistemes anteriors. Per a fer una autenticació més segura IEEE 802.11i introdueix el concepte de *robust security network association* (RSNA), basat en l'*extensible authentication protocol* (EAP) d'acord amb l'estàndard IEEE 802.1X.

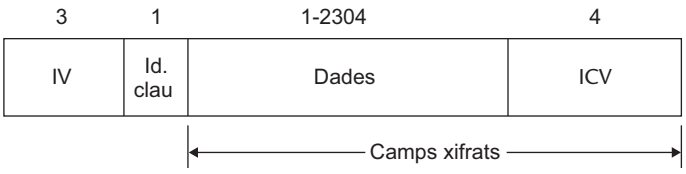
3. Protecció de trames amb WEP

La primera versió de l'estàndard IEEE 802.11 definia un mecanisme de seguretat per a protegir les trames enviades via ràdio: es tracta del protocol *wired equivalent privacy* (WEP). Tal com indica el nom, aquest protocol té per objectiu principal aconseguir una privacitat de les dades transmeses davant dels simples curiosos que hi hagi pels voltants, que sigui semblant a la de les xarxes amb fil. De la mateixa manera que en un lloc amb una xarxa amb fil, els tafaners no poden veure res de la comunicació si no és que disposen d'algun mitjà per a interceptar el cable, en una xarxa sense fil amb WEP tampoc poden veure les dades que es transmeten. El mètode per a assolir aquesta **privacitat** és xifrar les dades. A més del servei de privacitat, el protocol WEP proporciona també el servei d'**integritat** per mitjà d'un codi d'integritat de les dades.

L'AP pot tenir configurades fins a quatre claus secretes WEP. Això permet per exemple utilitzar claus diferents amb grups d'estacions diferents, o anar canviant periòdicament la clau, però la gran majoria de vegades només es fa servir una sola clau WEP.

Cada trama WEP se xifra amb una clau de xifratge independent. Així es dificulta, entre d'altres, que un atacant pugui detectar dades repetides. La clau de xifratge utilitzada en una trama concreta s'obté a partir de la clau WEP en ús més un vector d'inicialització diferent per a cada trama. El valor d'aquest vector d'inicialització s'ha d'incloure en la trama mateix perquè el receptor sàpiga com desxifrar-la.

Figura 4. Dades d'una trama WEP



Una trama de dades xifrada amb WEP té el mateix format que una trama de dades normal, però la part corresponent a les dades s'estructura en quatre camps.

- El primer camp és el vector d'inicialització (IV) utilitzat per a xifrar la trama.

- El segon camp és l'identificador de clau, que serveix per a indicar quina clau WEP de les quatre que pot tenir configurades l'AP s'ha utilitzat en el xifratge.

- Al tercer camp hi ha les dades protegides.

- El quart camp és l'*integrity check value* (ICV), que és un CRC de 32 bits calculat sobre el tercer camp (abans de xifrar).

El tercer i quart camp, és a dir les dades protegides i l'ICV, es transmeten xifrats.

La inclusió de l'ICV permet comprovar que la trama xifrada no ha estat manipulada. Si un atacant que no coneix la clau vol modificar les dades d'una trama WEP enviada o injectar una trama WEP amb les dades inventades, molt probablement quan el receptor la desxifri veurà que l'ICV no concorda. De totes maneres, un CRC no té les propietats de seguretat que pot tenir un codi d'integritat criptogràfic (per exemple basat en funcions resum o *hash*), i d'altra banda la resta de camps de la trama no estan protegits, cosa que permet que sí que puguin ser manipulats. Un altre cop, el criteri de la simplicitat va prevaldre sobre la seguretat en el disseny del protocol (un CRC és molt més fàcil de calcular que un resum).

3.1. El xifratge WEP

L'algorisme criptogràfic utilitzat per a xifrar les trames WEP és una xifra de flux, concretament l'**RC4** (*Ron's code 4*), dissenyat per Ronald Rivest. Va ser escollit per la seva simplicitat i pel nivell de seguretat que proporciona en relació amb la poca complexitat dels càlculs que requereix.

Aquest criteri era especialment important tenint en compte que la majoria de dispositius Wi-Fi poden ser dispositius mòbils en els quals un baix consum d'energia té un paper rellevant. Si s'hagués escollit un algorisme més sofisticat, que comportés més potència de càlcul per a xifrar i desxifrar les mateixes dades, i per tant consumís més energia, l'autonomia de les bateries dels dispositius mòbils es podria veure reduïda sensiblement.

La longitud de les claus RC4 en general no és fixa: hi pot haver claus RC4 de fins a 2.048 bits (encara que una clau de xifratge tan llarga no té gaire sentit per a un xifratge simètric).

El protocol WEP preveu principalment l'ús de dues longituds de clau de xifratge RC4: claus de **64 bits** o claus de **128 bits**.

En les claus de xifratge que s'utilitzen en un BSS per a xifrar cada trama WEP hi ha una part variable i una part fixa:

- La part variable són els primers 24 bits de la clau, i es coneixen com a **vector d'inicialització** (IV). L'IV és diferent per a cada trama que es transmet.

- La part fixa és la resta de la clau: 40 bits si la clau és de 64 en total, o 104 bits si la clau és de 128 en total. Aquesta part fixa es coneix també com a **clau arrel**.

La clau arrel WEP de 40 o 104 bits és, doncs, la clau secreta (o les claus secretes, si se'n fan servir dues, tres o quatre, tot i que la situació més habitual és usar-ne només una) que té configurada l'AP, i que s'ha de configurar en les estacions Wi-Fi que s'hi vulguin associar.

Vector d'inicialització

El nom que se sol donar a la part variable de la clau és el de **vector d'inicialització**, tot i que aquest concepte està relacionat amb les xifres de bloc més que no pas amb les de flux. De fet el concepte de **bits de sal** s'acostaria més a la funció d'aquesta part variable de la clau.

Valor de la clau arrel

El valor de la clau arrel pot ser qualsevol combinació de bits, però per a facilitar la configuració manual de les estacions és habitual que una clau arrel de 104 bits tingui la forma de cadena de 13 caràcters ASCII.

L'algorisme que segueix una estació qualsevol, inclòs l'AP, per a generar una trama xifrada WEP és el següent:

1) Generar una cadena de 24 bits per a utilitzar com a IV, procurant que sigui diferent dels últims IV generats.

2) Concatenar els 24 bits de l'IV amb la clau WEP arrel per a formar la clau RC4 de xifratge de la trama.

3) Calcular el CRC de les dades que cal protegir. Amb aquesta operació s'obté l'ICV.

4) Concatenar les dades amb l'ICV, i xifrar aquesta seqüència amb l'algorisme RC4 utilitzant la clau de xifratge del punt 2. Com que es tracta d'una xifra de flux, aquesta operació consisteix a obtenir tants bits de text de xifratge (*keystream*) com tingui la seqüència, i sumar-los un a un amb els de la seqüència.

5) Omplir la part de dades de la trama WEP amb els camps que la componen: el VI, l'identificador de la clau WEP, i el resultat del xifratge obtingut al punt 4.

L'estació que rep la trama xifrada WEP ha de fer els passos inversos per a desxifrar-la:

1) Llegir el camp IV de la trama WEP.

2) Concatenar els 24 bits de l'IV amb la clau arrel indicada pel camp identificador de clau WEP. Així s'obté la clau RC4 de desxifratge de la trama.

3) Desxifrar la part xifrada de la trama amb l'algorisme RC4 utilitzant la clau de xifratge del punt 2. Com abans, el xifratge consisteix a sumar bit a bit les dades xifrades amb el text de xifratge (*keystream*) generat a partir de la clau.

4) Calcular el CRC de les dades desxifrades i comprovar que coincideix amb el camp ICV també acabat de desxifrar.

L'objectiu últim d'incloure un IV és tenir una clau de xifratge diferent per a cada trama, i per a això cada IV hauria de ser diferent, o almenys que no es repeteixin fins al cap d'un nombre molt gran de trames generades. Les implementacions Wi-Fi normalment segueixen una d'aquestes dues tècniques per a aconseguir-ho:

- Generar cada IV amb un generador de nombres pseudoaleatoris.

- Generar el primer IV com un nombre aleatori de 24 bits i obtenir els següents sumant 1 cada vegada a l'anterior. Aquest mode de generar els VI s'anomena **mode comptador**.

Suma bit a bit

Recordeu que la suma bit a bit, que coincideix amb l'operació lògica XOR (*OR* exclusiva), és autocomplementària i que, per tant, el xifratge (suma) i el desxifratge (resta) es fan igual.

IV diferents

Com a màxim es poden generar 2^{24} IV diferents, és a dir uns 16 milions.

L'opció de l'algorisme RC4 per al xifratge WEP obeïa, com hem comentat abans, a la simplicitat de la seva implementació. Tot i que era un algorisme que es considerava raonablement segur, la manera com es va dissenyar el seu ús en el protocol WEP, especialment amb la introducció dels vectors d'inicialització, fa que presenti algunes vulnerabilitats importants.

Per comprendre les implicacions que tenen aquestes vulnerabilitats, abans veurem les característiques generals de l'algorisme RC4.

3.2. L'algorisme RC4

La simplicitat de l'algorisme RC4 ve donada d'una banda per les operacions en què es basa, i de l'altra per la poca memòria que requereix per a guardar la informació d'estat del xifratge:

- L'única operació aritmètica que es necessita per a implementar l'algorisme és la suma mòdul 256 o, el que és el mateix, la suma de 8 bits ignorant l'arrossegament (*carry*) generat. Aquesta és una operació senzillíssima d'implementar en maquinari, i molt ràpida de calcular. L'algorisme també fa servir l'operació d'intercanvi (*swap*) d'elements d'un vector, però no és una operació aritmètica sinó simplement de moviment de dades en memòria.

- La informació d'estat amb què treballa l'algorisme és un vector de 256 elements de 8 bits, més dos comptadors o índexs també de 8 bits cadascun. En total es necessiten 258 bytes de memòria per a guardar aquesta informació d'estat (a part de l'espai que ocupi la clau, que només es necessita en la fase inicial de l'algorisme).

Per a la descripció de l'algorisme farem servir la notació següent:

- $K[0]$, $K[1]$, $K[2]$, etc., són el primer, segon, tercer, etc., bytes de la clau de xifratge. Si la clau és, per exemple, de 128 bits els elements que la formen són $K[0]$ fins a $K[15]$.

- $S[0]$, $S[1]$, $S[2]$, etc., són el primer, segon, tercer, etc., bytes del text de xifratge (*keystream*) generat.

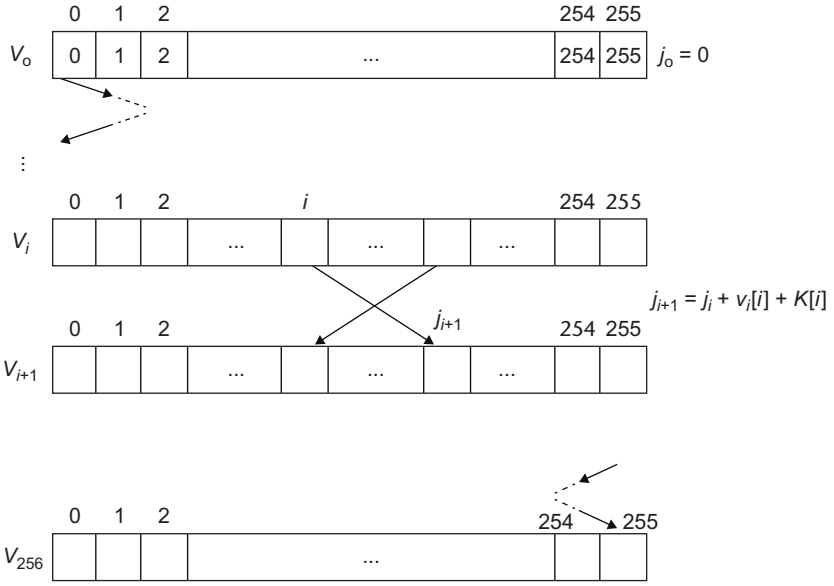
- $V[0], \ldots, V[255]$ són els elements del vector d'estat de l'algorisme.

- i, j són els dos comptadors interns amb què treballa l'algorisme.

- El símbol de suma, +, representa la suma mòdul 256, és a dir la suma de 8 bits.

El funcionament de l'algorisme es divideix en dues fases. La primera és el *key schedule algorithm* (KSA) o programació de la clau, i la segona és el *pseudo-random generation algorithm* (PRGA) o generació del text de xifratge pròpiament dit.

Figura 5. Esquema de l'algorisme KSA de xifratge RC4



L'objectiu de l'algorisme KSA és obtenir, a partir del valor de la clau K, un vector d'estat V que serveixi perquè l'algorisme PRGA comenci a generar el text de xifratge S. Els passos que segueix el KSA són aquests:

1) El vector V parteix d'un estat inicial (V_0) en què l'element $V_0[0]$ té el valor 0, l'element $V_0[1]$ té el valor 1, ...i l'element $V_0[255]$ té el valor 255. Els comptadors i i j valen 0.

2) Si la clau té una longitud de L bytes, es concatena amb si mateixa tantes vegades com calgui fins que ocupi 256 bytes. (De fet no cal ocupar físicament aquests bytes de memòria, n'hi ha prou de substituir els accessos a $K[i]$ per $K[i \bmod L]$.)

3) Es repeteix 256 vegades la seqüència següent:

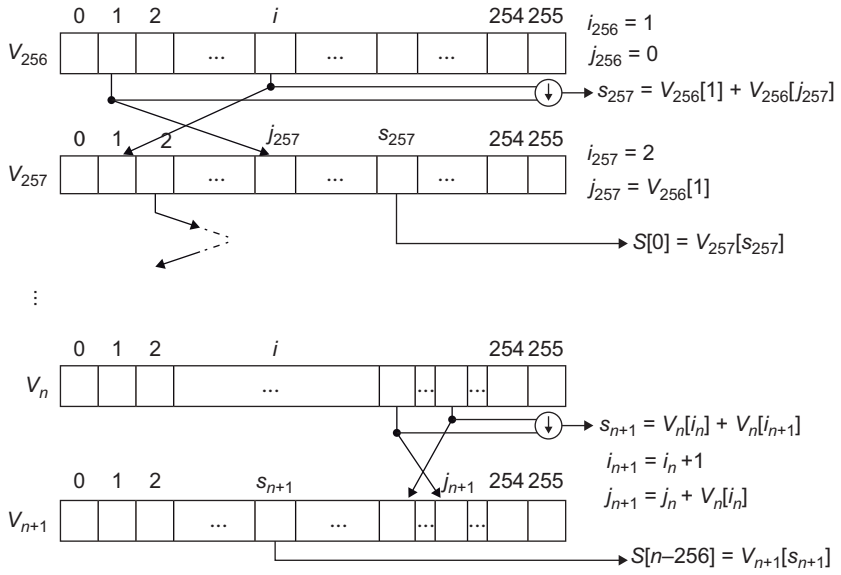
a) A l'índex j s'hi suma (mòdul 256) $V[i] + K[i]$.

b) S'intercanvien els elements $V[i]$ i $V[j]$ del vector d'estat.

c) A l'índex i s'hi suma 1.

Al final d'aquests passos tenim un vector d'estat V_{256} que conté una permutació aparentment aleatòria dels elements 0, ..., 255.

Figura 6. Esquema de l'algorisme PRGA de xifratge RC4



Un cop obtingut el vector V_{256}, que és el vector d'estat inicial per a començar la generació del text de xifratge, s'aplica l'algorisme PRGA. En aquest algorisme primer s'inicialitza l'índex i a 1 i l'índex j a 0. A continuació, per a cada byte del text de xifratge S que es vulgui obtenir, es fa una iteració que consta dels passos següents:

1) A l'índex j s'hi suma (mòdul 256) $V[i]$.

2) S'intercanvien els elements $V[i]$ i $V[j]$ del vector d'estat.

3) Es calcula l'índex s com la suma dels dos elements intercanviats ($s = V[i] + V[j]$).

4) A l'índex i s'hi suma (mòdul 256) 1.

5) El resultat obtingut en aquesta iteració, és a dir el següent byte del text de xifratge S, és igual a l'element $V[s]$.

4. Vulnerabilitats del protocol WEP

Com ja hem comentat abans, el protocol WEP té una sèrie de vulnerabilitats, algunes de les quals són independents de l'elecció de l'RC4 com a algorisme de xifratge, i d'altres que estan directament relacionades amb la manera com s'usa aquest algorisme.

4.1. Vulnerabilitats no relacionades amb l'algorisme RC4

Aquest conjunt de vulnerabilitats és conseqüència de certes decisions de disseny del protocol WEP independents de l'ús de l'algorisme RC4.

4.1.1. Injecció de trames

Un atacant que capturi una trama WEP corresponent a una associació determinada pot retransmetre-la tantes vegades com vulgui i, si l'associació continua existint, el receptor donarà la trama per vàlida. I si l'associació ja no existeix, l'atacant pot canviar les adreces de l'estació transmissora i/o receptora per les d'altres estacions que sí que estiguin associades, i el nou receptor també donarà la trama per vàlida.

Això és així, d'una banda, perquè ni el nivell d'enllaç IEEE 802.11 ni el protocol WEP preveuen res per a detectar trames duplicades. Les trames WEP injectades per l'atacant tindran l'IV repetit, però això no és problema del receptor. Encara que l'ús dels IV tingui per objectiu que les trames xifrades siguin sempre diferents, res no impedeix a una estació enviar trames idèntiques xifrades amb el mateix IV. I les estacions receptores no solen comprovar si els arriben trames amb IV repetit, perquè això implicaria haver de recordar els IV de les últimes trames rebudes i comparar cada trama nova que arribi, cosa que normalment no fan.

I d'altra banda, com que els camps de la capçalera MAC no estan protegits pel codi d'integritat ICV, no hi ha cap problema a canviar les adreces d'aquesta capçalera.

4.1.2. Falsificació de l'autenticació

La falsificació de l'autenticació només té sentit en el mètode de clau compartida, perquè en el mètode d'autenticació de sistema obert no hi ha res per falsificar.

Si un atacant captura les trames intercanviades durant una autenticació de clau compartida entre una estació i un AP, pot autenticar-se amb èxit davant el mateix AP sense necessitat de conèixer la clau WEP corresponent.

En el procés d'autenticació de clau compartida s'intercanvien quatre trames: petició d'autenticació, repte, resposta, i resultat. La tercera trama està xifrada amb WEP, però l'atacant sap quin ha de ser el contingut desxifrat perquè en la segona trama hi ha el repte en clar. Per tant, si del contingut xifrat de la tercera trama en resta bit a bit (és a dir, hi suma) el contingut desxifrat, obté el text de xifratge (*keystream*) que es genera amb la clau WEP i l'IV de la tercera trama.

Llavors l'atacant només ha d'enviar una petició d'autenticació a l'AP, rebre el repte, i construir una trama de resposta amb l'IV capturat prèviament i la suma bit a bit del repte més el *keystream* calculat. L'AP veurà que és una resposta correctament xifrada perquè en desxifrar-la obtindrà el repte de la segona trama, i per tant donarà per autenticada l'estació de l'atacant.

La captura de les trames d'autenticació, en general, permet obtenir una certa quantitat de *keystream* corresponent a un determinat IV. A la trama de resposta hi ha 140 bytes xifrats dels quals es coneix el valor desxifrat (els 128 del repte més els d'altres camps), i per tant s'obtenen 140 bytes de *keystream*. Això pot ser útil per a generar altres trames xifrades a part de la de resposta a l'autenticació. I a més hi ha altres atacs que permeten calcular més bytes de *keystream*, per si cal falsificar trames més llargues.

4.1.3. Desxifratge de trames mitjançant la comprovació d'integritat o atac *chopchop*

Un atacant que hagi capturat una trama WEP pot desxifrar els n últims bytes de dades xifrades, sense necessitat de saber la clau de xifratge, enviant a l'AP una mitjana de $128 \times n$ trames.

El codi d'integritat ICV que incorporen les trames WEP es calcula amb l'algorisme CRC-32, que és un mètode molt bo per a detectar errors de transmissió, però no és un algorisme criptogràfic. Com que el CRC-32 està basat en operacions aritmètiques lineals com són les divisions de polinomis mòdul 2, és possible fer càlculs inversos.

A cada seqüència de bits li correspon un polinomi binari P els coeficients del qual són els bits de la seqüència. El seu CRC és una altra seqüència corres-

ponent al polinomi R tal que la concatenació $P \parallel R$, que anomenarem X, és múltipla del polinomi generador G, en aquest cas el polinomi CRC-32. Per tant, per a comprovar que el CRC és correcte només cal veure si es compleix

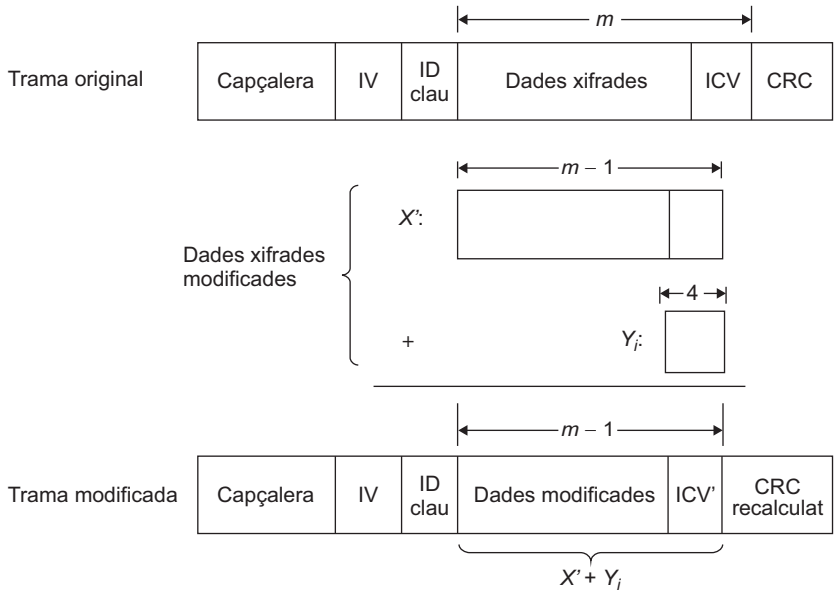
$$X \bmod G = P \parallel R \bmod G = 0$$

Si considerem X com a seqüència de m bytes $X[0] \parallel \cdots \parallel X[m-1]$, i X' és la mateixa seqüència sense l'últim byte $(X[0] \parallel \cdots \parallel X[m-2])$, és fàcil calcular quina és la seqüència Y que cal sumar a X' perquè continuï essent divisible per G, és a dir que es compleixi

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

i que, per tant, el CRC continuï essent correcte. Com que resulta que el valor Y només depèn de $X[m-1]$, és a dir de l'últim byte de la seqüència X, podem calcular els 256 valors $Y_0, \ldots, Y_{255}$ corresponents a cadascun dels possibles valors de l'últim byte de X.

Figura 7. Modificació de trames WEP per a l'atac *chopchop*



Amb aquesta tècnica, l'atac per a desxifrar una trama WEP consisteix a construir noves trames modificades, fins a 256 en total, seguint aquests passos:

- Suprimir l'últim byte xifrat de la trama original.
- Sumar cadascun dels valors $Y_0 \ldots, Y_{255}$, respectivament, a les dades xifrades que queden.
- Recalcular el CRC no xifrat per a cada nova trama.

Llavors l'atacant va enviant aquestes trames modificades a l'AP per tal d'esbrinar si l'ICV és correcte o no. Quan la resposta de l'AP indiqui que la trama és correcta, l'atacant sabrà quin és el valor desxifrat de l'últim byte xifrat de la trama: el que correspongui al valor Y_i utilitzat. I a partir del valor xifrat i el valor desxifrat, fent la resta (suma) obtindrà l'últim byte del *keystream* utilitzat per a xifrar la trama.

Suma de bits a les dades xifrades

Conceptualment, l'operació que caldria fer seria desxifrar les dades, sumar-hi la seqüència Y, i tornar-les a encriptar. Però com que el xifratge en realitat també és una suma (dels bits en clar amb el *keystream*), el resultat és el mateix si la suma es fa directament sobre les dades xifrades.

Com que es poden provar fins a 256 valors possibles diferents, el nombre de trames que caldrà enviar de mitjana abans de trobar la bona serà de 128.

A partir de la trama bona es pot tornar a repetir l'atac per a trobar el penúltim byte del *keystream*, i així successivament. Amb aquest mètode es podrien obtenir tots els bytes del *keystream* (i per tant desxifrar tots els bytes xifrats de la trama original) excepte els 4 primers, perquè els valors Y són seqüències de 32 bits. Però com que entre els bytes desxifrats hi haurà els del codi ICV, és immediat deduir els bytes que falten perquè aquest codi sigui correcte.

Hi ha diferents maneres de fer que l'AP ens digui si l'ICV d'una trama WEP és correcte o no, entre les quals podem esmentar aquestes dues:

- Si disposem de dues estacions, des de cadascuna podem fer una autenticació (falsa, si no coneixem la clau WEP) i una associació amb l'AP. Llavors podem enviar les trames des de la primera estació amb destinació la segona per mitjà de l'AP. Si l'ICV d'una trama és correcte l'AP la retransmetrà a la segona estació, i si no, la descartarà.

- Una altra manera més senzilla és enviar les trames des d'una estació no associada. Si l'ICV d'una trama és correcte l'AP respondrà amb una trama de gestió que indicarà que l'estació no està associada, i si no, la descartarà.

Primers bytes xifrats de la trama

En la pràctica, és possible que l'AP descarti les trames WEP que no tinguin una longitud mínima, de manera que hi ha un punt a partir del qual no es poden obtenir més bytes de *keystream* amb l'atac *chopchop*.

Aquest atac que va escapçant byte a byte la trama capturada a mesura que la va desxifrant es coneix com a *atac chopchop de KoreK*, o simplement *atac chopchop*. Es basa en la tècnica d'un altre atac publicat anteriorment, anomenat *atac inductiu d'Arbaugh*. Aquest últim és de fet l'atac invers del *chopchop*: en comptes de retrocedir byte a byte, va "avançant" a còpia d'assaig i error i així va esbrinant els bytes següents del *keystream*. D'aquesta manera, amb una mitjana de 128 intents per byte es pot aconseguir una longitud arbitrària de *keystream*, útil per a construir trames WEP falsificades més o menys llargues sense saber la clau.

4.1.4. Atac de fragmentació

KoreK

KoreK és el pseudònim de l'autor que ha publicat diversos atacs contra els protocols Wi-Fi.

Un atacant que hagi descobert n bytes d'un *keystream* pot obtenir fins a $16 \times n - 60$ bytes d'un altre enviant fins a 16 trames fragmentades a l'AP.

IEEE 802.11 permet enviar una trama en fragments, fins a un màxim de 16. Si una estació envia a l'AP una trama fragmentada que hagi de ser retransmesa, l'AP normalment recompondrà la trama abans de retransmetre-la. I, si els fragments estan xifrats, la trama recombinada també estarà xifrada, possiblement amb un IV nou.

Així, un atacant que disposi de n bytes de *keystream* pot construir 16 fragments de trama WEP, cadascun amb $n-4$ bytes de dades més els 4 bytes de l'ICV, tots xifrats amb el mateix IV i *keystream*. Si envia aquests fragments a l'AP perquè els retransmeti, la trama resultant tindrà $16 \times (n-4)$ bytes de dades i 4 bytes d'ICV encriptats (sempre que no sobrepassi la longitud màxima de les dades d'una trama), en total $16 \times n - 60$ bytes dels quals el valor desxifrat serà conegut. Per tant, l'atacant podrà obtenir aquesta quantitat de bytes de *keystream*.

Si el *keystream* recuperat encara no arriba a la longitud necessària per a xifrar una trama llarga, es pot tornar a repetir aquest atac fins a obtenir la longitud màxima possible.

4.2. Vulnerabilitats relacionades amb l'algorisme RC4

Aquest conjunt de vulnerabilitats és conseqüència del mètode amb què s'utilitza l'algorisme RC4 en el protocol WEP.

4.2.1. L'atac FMS

Els criptoanalistes Scott Fluhrer, Itsik Mantin i Adi Shamir, en un article publicat l'any 2001, van detallar les bases teòriques d'un atac que permetia recuperar una clau arrel WEP a partir de les trames xifrades si es disposava d'un nombre suficient d'aquestes trames. Un altre equip d'investigadors va publicar l'any 2004 els resultats de la primera implementació d'aquest atac contra una xarxa Wi-Fi real.

Amb l'atac FMS, un atacant que conegui el primer byte de *keystream* ($S[0]$) d'aproximadament entre 4 i 9 milions de trames WEP encriptades amb la mateixa clau arrel pot recuperar el valor de la clau amb una probabilitat d'èxit del 50%.

Atac FMS

El nom amb què es coneix aquest atac prové de les inicials dels cognoms dels seus descobridors.

Aquest atac es basa en l'observació del funcionament de l'algorisme RC4. Tal com s'utilitza en el protocol WEP els 3 primers bytes de la clau RC4 són sempre coneguts, ja que formen el vector d'inicialització (IV) que s'envia en clar prefixat a les dades xifrades. Per a cada trama capturada amb l'IV corresponent, doncs, l'atacant disposa dels elements $K[0]$, $K[1]$ i $K[2]$ de la clau RC4. Amb aquesta informació pot reconstruir les 3 primeres iteracions de l'algorisme KSA i obtenir el vector d'estat V_3 i el valor de l'índex j_3.

Primer byte del *keystream*

El cas més habitual és que les dades xifrades d'una trama no fragmentada, o del primer fragment d'una trama fragmentada, comencin amb el primer byte de la capçalera LLC igual a AA (hexadecimal). Per tant, conegut el primer byte xifrat i el primer byte desxifrat, també es coneix el primer byte de *keystream*.

L'algorisme KSA es completa amb 253 iteracions més fins a arribar a obtenir el vector V_{256}, a partir del qual es calcula amb l'algorisme PRGA el primer byte de *keystream* $S[0]$:

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

El contingut del vector V_{256} serà en general desconegut, però es pot seleccionar un subconjunt de les trames que tinguin certes propietats que afavoreixen l'atac. Per a fer l'atac, s'analitza cada trama i es comprova si amb el seu IV es compleixen les anomenades **condicions de resolució**:

1) $V_3[1] < 3$

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

Condicions de resolució

Estadísticament, una de cada 21675 trames complirà les condicions de resolució per a $n = 3$. Si creix n, també creix el nombre de trames que compleixen les condicions.

Si la trama no compleix aquestes condicions, es descarta i es passa a la següent.

A continuació, l'atac consisteix a veure què passaria si es donés la feliç coincidència que tres elements determinats del vector d'estat no s'intercanviessin amb cap altre en les iteracions següents del KSA. Més concretament, el cas que es considera és que es donin aquestes tres condicions alhora:

1) Que l'element $V_3[1]$ no canviï de lloc entre les iteracions 4 i 256.

2) Que l'element $V_3[V_3[1]]$ no canviï de lloc entre les iteracions 4 i 256.

3) Que l'element $V_3[j_4]$ (desconegut, perquè si no tenim el quart byte de la clau, $K[3]$, no sabem quant val j_4), que a la iteració 4 passarà a ser $V_4[3]$, no canviï de lloc entre les iteracions 5 i 256.

Com que a cada iteració del KSA s'intercanvien $V[i]$ i $V[j]$, perquè es no malbarati aquesta sèrie de coincidències l'índex j no hauria de passar per cap dels valors "prohibits" 1, $V_3[1]$ i 3 (aquest últim, a partir de la iteració 5). Amb l'índex i no hi ha problema perquè a partir de la iteració 4 valdrà com a mínim 4 i a més sabem, per la primera condició de resolució, que $V_3[1] < 3$. Pel que fa a j, es pot calcular que la probabilitat que aquest índex no prengui cap dels 3 valors prohibits en cap de les 253 iteracions següents és $((256 - 3)/256)^{253}$, és a dir aproximadament un 5%.

Llavors sabem que, amb una probabilitat aproximada del 5%, aquests tres elements del vector d'estat no s'hauran mogut, i per tant $V_{256}[1] = V_3[1]$ i $V_{256}[V_{256}[1]] = V_3[V_3[1]]$. A la primera iteració del PRGA, s'intercanvien precisament els elements $V_{256}[1]$ i $V_{256}[V_{256}[1]]$, i es calcula l'índex s_{257} com la suma d'aquests dos valors. Per la segona condició de resolució aquesta suma és igual a 3, i això vol dir que el primer byte de *keystream* generat serà igual

a $V_{257}[3]$. Si es compleix la nostra hipòtesi d'immobilitat, aquest element no s'haurà mogut entre les iteracions 5 i 256, i a més les condicions de resolució garanteixen que tampoc s'haurà mogut a la iteració 257.

En definitiva, si no s'han mogut els elements esmentats tindrem que:

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

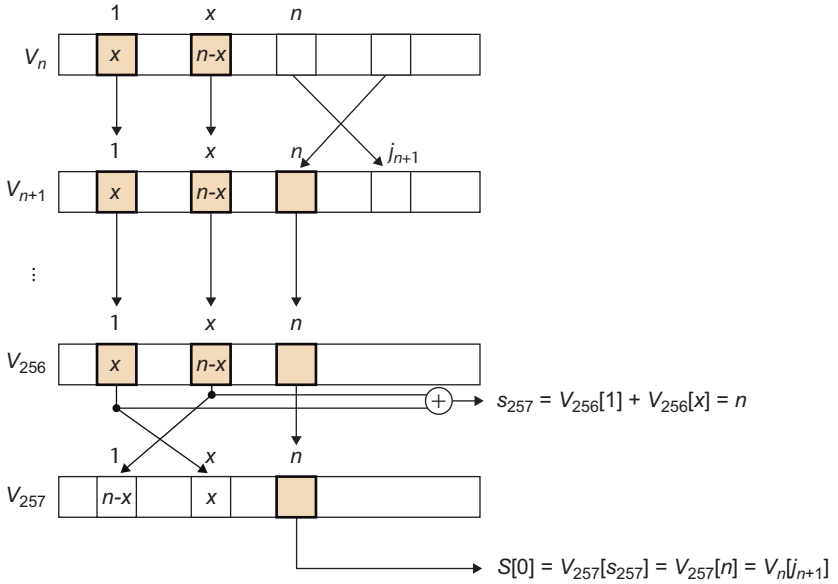
A partir d'aquí, com que j_3 i el vector V_3 són coneguts i la hipòtesi inicial de l'atac és que $S[0]$ també és conegut, és immediat trobar el valor $K[3] = V_3^{-1}[S[0]] - j_3 - V_3[3]$ (fent les operacions en mòdul 256). Aquest serà el valor correcte del quart byte de la clau sempre que sigui veritat que els tres elements en qüestió no s'han mogut de lloc. Si no, el resultat obtingut amb aquesta fórmula serà un valor que podem considerar aleatori.

És a dir, si calculem el possible valor de $K[3]$ a partir d'un nombre de trames suficient perquè la distribució dels valors incorrectes sigui uniforme, trobarem que aproximadament 15 de cada 270 trames donaran un mateix valor, i les altres 255 donaran valors diferents entre si. Empíricament es pot comprovar que a partir d'unes 60 trames analitzades ja hi ha un valor que destaca sobre els altres perquè es repeteix almenys 3 o 4 vegades. Aquest valor candidat que té majoria de "vots" és el que en principi es pot considerar com a correcte.

Inversió de la transformació $V[x]$

A partir de $y = V[x]$, sempre és possible trobar el valor únic $x = V^{-1}[y]$ perquè V és una permutació dels elements 0, . . . , 255: cada element apareix una i només una vegada en la permutació.

Figura 8. Elements del vector d'estat que no s'haurien de modificar perquè sigui certa la hipòtesi de l'atac FMS



Un cop determinat el valor candidat a ser el quart byte de la clau, $K[3]$, es pot tornar a començar l'atac per tal d'esbrinar el byte següent, $K[4]$. En general, les condicions de resolució per a intentar obtenir el valor del byte $K[n]$ són les següents:

1) $V_n[1] < n$

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

i el valor es calcula amb la fórmula

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

En l'obtenció de cada byte de la clau es poden trobar falsos positius. Això passarà si entre les trames que no compleixen la condició d'immobilitat n'hi ha diverses que coincideixen a donar un mateix valor (incorrecte) de $K[n]$, i superen en nombre les trames que donen el valor correcte, de manera que el valor fals té més vots que el bo. Quan un byte esbrinat $K[n]$ és incorrecte, tots els bytes següents $K[m]$ amb $m > n$ estaran mal calculats perquè l'algorisme KSA per a obtenir el vector V_m s'estarà aplicant amb valors erronis.

Probabilitat de no intercanvi

A mesura que avança n també creix la probabilitat que els elements no siguin intercanviats perquè queden menys iteracions fins al final del KSA, però la variació no és gaire gran: des del 5,07% per a $n = 3$ fins al 5,84% per a $n = 15$.

La comprovació per a saber si els bytes de la clau són correctes només es pot fer quan s'han esbrinat tots, des de $K[3]$ fins a $K[15]$ en el cas d'una clau WEP-104. Llavors es poden utilitzar uns quants VI per veure si la clau calculada dóna el byte de *keystream* $S[0]$ corresponent a cadascun dels VI. Si no és així, cal tornar enrere i refer els càlculs. Per exemple, es pot mirar quin dels bytes de la clau ha guanyat per una majoria més estreta, substituir-lo pel segon valor que hagi tingut més vots, i recalcular tots els bytes posteriors. I si la comprovació tampoc és satisfactòria, es poden repetir aquestes substitucions fins a trobar el valor correcte o fins que s'hagi ultrapassat un nombre màxim d'intents.

Obtenció de l'últim byte de la clau

Si hi ha moltes trames per processar, en comptes d'obtenir l'últim byte de la clau $K[15]$ pel sistema de votació, pot ser més eficient obtenir només fins al penúltim byte i fer les comprovacions per força bruta amb cadascun dels 256 possibles valors de l'últim byte. En alguns casos fins i tot es pot aplicar la força bruta als dos últims bytes de la clau.

Experimentalment s'ha comprovat que a partir de 4 milions de trames, l'atac FMS permet obtenir el valor correcte de la clau WEP amb una probabilitat d'èxit del 50%. Aquest nombre de trames, però, puja fins a 9 milions si els IV estan generats en mode comptador en comptes d'haver-se generat aleatòriament. També s'ha comprovat que, a partir d'un cert punt, per molt que s'augmenti el nombre de trames processades difícilment s'ultrapassa el 75% de probabilitat d'èxit.

Èxit de l'atac FMS

Un criteri per a considerar que l'atac no ha tingut èxit és que el mètode d'assaig i error per a trobar la clau bona no doni cap resultat al cap de 2 o 3 minuts, amb la potència de càlcul mitjana dels ordinadors actuals.

El fet que el mode comptador requereixi més trames ve donat per la distribució dels IV que compleixen les condicions de resolució. En el mode aleatori estaran repartits uniformement entre les trames capturades, mentre que en el mode comptador estaran concentrades en sèries de trames consecutives o molt pròximes. Si l'atacant té la sort de topar aviat amb una d'aquestes sèries pot obtenir ràpidament els IV necessaris, però si no, necessitarà de mitjana moltes més trames.

4.2.2. El conjunt d'atacs KoreK

Es coneix amb el nom d'*atacs KoreK* una sèrie d'atacs al protocol WEP que exploten determinades correlacions entre la clau arrel i els primers bytes de text d'encriptació o *keystream*. L'any 2004 una persona que feia servir el pseudònim

"KoreK" va publicar una eina que integrava tots aquests atacs, que en total eren 17. Alguns ja es coneixien prèviament, com és el cas de l'atac FMS, i els altres van ser descoberts per KoreK.

Amb els atacs KoreK, un atacant que conegui els dos primers bytes de *keystream* ($S[0]$,$S[1]$) d'aproximadament entre 150.000 i 700.000 trames WEP encriptades amb la mateixa clau arrel pot recuperar el valor de la clau amb una probabilitat d'èxit del 50%.

Els atacs KoreK es poden dividir en tres grups:

- Atacs que permeten esbrinar $K[n]$ a partir de $K[0], \ldots, K[n-1]$ i $S[0]$. L'atac FMS pertany a aquest grup.
- Atacs que permeten esbrinar $K[n]$ a partir de $K[0], \ldots, K[n-1]$, $S[0]$ i $S[1]$.
- Atacs "negatius" que, si V_n compleix certes condicions i $S[0]$ pren certs valors, permeten descartar determinats valors de $K[n]$.

Segon byte de *keystream*

Així com en la gran majoria de casos el primer byte de dades encriptades d'una trama WEP és el primer byte de la capçalera LLC, el segon byte encriptat serà el segon byte de la mateixa capçalera, que també és igual a AA (hexadecimal), i a partir d'aquest valor es pot saber el segon byte de *keystream* $S[1]$.

Molts dels atacs es basen, igual que l'FMS, en la probabilitat que determinats elements del vector d'estat no siguin intercanviats a partir de la iteració n de l'algorisme KSA. Cadascun dels atacs individuals té condicions de resolució pròpies. L'eina que es va publicar anava comprovant trama per trama si es complien les condicions d'algun atac, i si era així el duia a terme i obtenia un vot a favor d'un candidat a $K[n]$, o un vot en contra, en cas que es tractés d'un atac negatiu. Els vots es ponderaven segons la probabilitat d'èxit de l'atac realitzat.

El fet d'implementar diversos atacs diferents en paral·lel facilita la tasca de descobrir la clau amb un nombre menor de trames analitzades. Experimentalment s'ha trobat que a partir de 150.000 trames els atacs KoreK permeten obtenir el valor correcte de la clau WEP amb una probabilitat d'èxit del 50% si els VI estan generats aleatòriament. En canvi, si els VI es generen en mode comptador el nombre de trames necessàries creix fins a 700.000 per a assolir el mateix 50% d'èxit. A partir de 270.000 trames en mode aleatori, i 1.700.000 en mode comptador, la taxa d'èxit és del 90%.

4.2.3. L'atac PTW

L'any 2007 es va publicar un nou atac, conegut com a PTW, que és una variant millorada d'un altre que va ser descobert el 2005, anomenat **atac Klein**.

Amb l'atac PTW, un atacant que conegui els bytes de *keystream* del tercer al quinzè ($S[2], \ldots, S[14]$) d'aproximadament 35.000 trames WEP encriptades amb la mateixa clau arrel pot recuperar el valor de la clau amb una probabilitat d'èxit del 50%.

Atac PTW

El nom amb què es coneix aquest atac prové de les inicials dels cognoms dels autors que el van publicar: Andrei Pyshkin, Erik Tews i Ralf-Philipp Weinmann.

L'atac Klein es basa en una anomalia de les propietats estadístiques de la programació de claus RC4. L'algorisme KSA genera un vector d'estat aparentment aleatori, i com que cada element pot tenir un de 256 valors possibles, és d'esperar que la probabilitat que un element $V[n]$ tingui un determinat valor x sigui 1/256, és a dir aproximadament un 0,4%. Una combinació d'elements del vector, com ara $V[V[i] + V[j]] + V[j]$, en principi també hauria de ser igual a qualsevol valor $0 \leq x \leq 255$ de manera equiprobable. Però l'anomenada **correlació de Jenkins** demostra que hi ha un valor d'aquesta expressió que és més probable que els altres. Concretament:

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

Així, la combinació d'elements anterior pot prendre el valor i amb una probabilitat aproximada del 0,8%, en comptes del 0,4% que s'esperaria.

A més, la correlació de Jenkins també demostra que la resta de valors $x \neq i$ són equiprobables, de manera que la probabilitat de cadascun és $(1 - 2/256)/255 = 127/32.640$.

Igual que en l'atac FMS, en l'atac Klein inicialment cal fer les 3 primeres iteracions de l'algorisme KSA per a obtenir V_3, i a la iteració 4 sabem que l'element $V_3[j_4]$ passarà a ser $V_4[3]$. En alguna de les iteracions següents l'índex j pot prendre el valor 3, i llavors aquest element canviarà de lloc. Però l'índex i no tornarà a valer 3 fins a la iteració 258, ja dins de l'algorisme PRGA, és a dir fins al cap de 254 iteracions.

Si considerem que les variacions de j tenen un comportament aleatori, la probabilitat que l'índex j no prengui el valor 3 en cap d'aquestes 254 iteracions és $(255/256)^{254}$, és a dir un 37%. Com que l'índex i tampoc valdrà 3 en cap de les 254 iteracions, aquesta és la probabilitat que l'element $V_4[3]$ no s'hagi mogut del seu lloc fins a la iteració 258. En l'altre 63% dels casos j haurà valgut 3 en algun moment i l'element $V_{258}[3]$ ja no serà el mateix que hi havia a $V_4[3]$.

A la iteració següent, la 259, s'obtindrà el tercer byte de *keystream* $S[2]$:

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

Com que en aquesta iteració s'hauran intercanviat els elements de les posicions 3 i j_{259}, la suma $V_{258}[3] + V_{258}[j_{259}]$ serà la mateixa que $V_{259}[j_{259}] + V_{259}[3]$, i per tant:

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

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

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

Primers 15 bytes de *keystream*

Hi ha algunes trames, com és el cas de les que contenen paquets ARP, en què els 15 primers bytes de dades corresponen a camps de capçaleres amb valors constants. Per tant, si aquestes trames estan encriptades es poden obtenir 15 bytes de *keystream*.

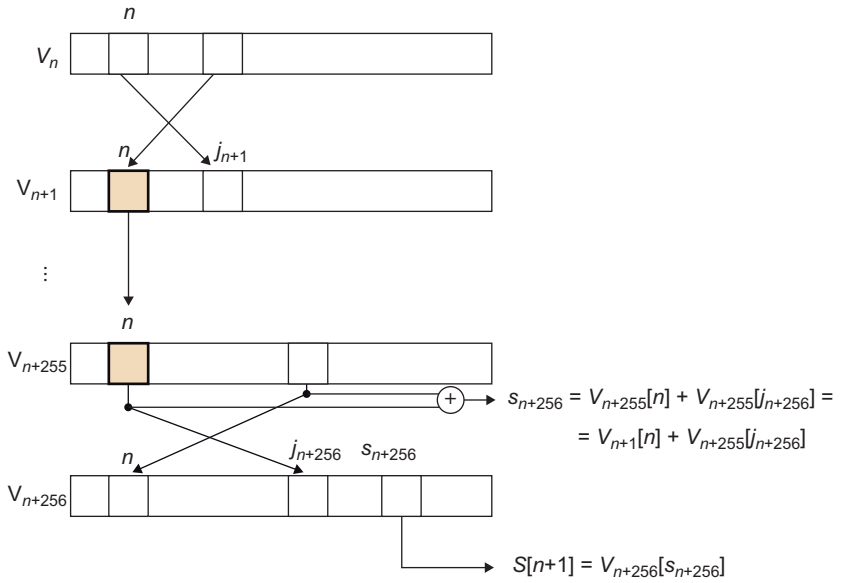
I per la correlació de Jenkins sabem que aquesta expressió té més probabilitat de valer 3 que qualsevol altre valor. Així tenim que hi ha una probabilitat de 2/256 que es compleixi $S[2] + V_{259}[j_{259}] = 3$ o, ja que $V_{259}[j_{259}]$ és l'element que hi havia a $V_{258}[3]$ abans de l'últim intercanvi, que es compleixi $S[2] + V_{258}[3] = 3$.

Per a obtenir el valor del byte $K[3]$, l'atac Klein es basa en la probabilitat que la hipòtesi $S[2] + V_4[3] = 3$ sigui certa. Llavors, com que:

$$V_4[3] = V_3[j_4] = V_3[j_3 + V_3[3] + K[3]]$$

aïllant $K[3]$ tenim que el valor candidat és $K[3] = V_3^{-1}[3 - S[2]] - j_3 - V_3[3]$ (amb totes les operacions en mòdul 256).

Figura 9. Element del vector d'estat que no s'hauria de modificar perquè sigui certa la hipòtesi de l'atac Klein



Com en l'atac FMS, per a cada trama s'obté un vot a un candidat a $K[3]$, i després d'analitzar totes les trames disponibles es determina el candidat més probable. Un cop decidit el valor de $K[3]$ es repeteix el procés per a la resta de bytes $K[n]$. La fórmula general per a obtenir cada byte és:

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

Analitzem ara la probabilitat que els valors candidats trobats siguin correctes, i centrem-nos en el cas $n = 3$. Hi ha dues combinacions que fan que la hipòtesi de l'atac Klein ($S[2] + V_4[3] = 3$) sigui certa:

- Que es donin simultàniament aquestes dues condicions:

– L'element $V_{258}[3]$ és el mateix que $V_4[3]$. Això passa, com hem vist abans, amb una probabilitat del 37%.

– Es compleix $S[2]+V_{258}[3] = 3$. Segons la correlació de Jenkins, la probabilitat és 2/256.

La probabilitat total d'aquesta primera combinació és

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

- Que es donin simultàniament aquestes altres condicions:

– L'element $V_{258}[3]$ és diferent de $V_4[3]$. La probabilitat és del 63%.

– Coincideix que $V_4[3]$ és tal que $S[2] + V_4[3] = 3$. Segons la correlació de Jenkins, tenint en compte que ara no estem en el cas més probable, la probabilitat és 127/32.640.

La probabilitat total d'aquesta altra combinació és $0{,}63 \times 127/32.640 = 0{,}63/256$.

Com que són combinacions disjuntes, podem sumar les probabilitats parcials i tenim que la probabilitat total que sigui certa la hipòtesi és 1,37/256. Dit d'una altra manera, la probabilitat que es compleixi la hipòtesi de l'atac Klein és 1,37 vegades la "normal". Aquest resultat no és gaire espectacular, si el comparem per exemple amb la hipòtesi de l'atac FMS, que es complia amb una probabilitat aproximada del 5% (unes 14 vegades la normal).

Probabilitat de la hipòtesi de l'atac Klein

Mentre que la hipòtesi de l'atac FMS té una probabilitat que varia amb n, en l'atac Klein és la mateixa per a qualsevol n perquè el nombre d'iteracions considerades és constant (254).

La gran diferència, però, entre l'atac FMS i l'atac Klein és que en aquest últim no hi ha condicions de resolució i totes les trames es poden usar per a obtenir un valor candidat per a cada $K[n]$. Així, encara que la probabilitat de complir-se la hipòtesi de l'atac **sigui 10 vegades menor**, en l'atac Klein es necessiten moltes menys trames per a recuperar la clau WEP.

Efecte de la correlació de Jenkins

Observeu que sense la correlació de Jenkins la probabilitat total de la hipòtesi seria $0{,}37 \times 1/256 + 0{,}63 \times 1/256 = 1/256$. És a dir, el cas $S[2] + V_4[3] = 3$ es donaria amb la mateixa probabilitat que qualsevol altre.

Experimentalment s'ha comprovat que a partir de 43.000 trames l'atac Klein permet obtenir el valor correcte de la clau WEP amb una probabilitat d'èxit del 50%. A més, a partir de 60.000 trames la probabilitat d'èxit ja és del 90%. D'altra banda, el nombre de trames necessàries en l'atac Klein és independent de com es generen els VI (mode aleatori o mode comptador), ja que no hi ha condicions de resolució i s'aprofiten totes les trames.

L'atac PTW pròpiament dit consisteix a afegir una sèrie de millores a l'atac Klein que permeten augmentar-ne l'eficiència. Aquests són alguns dels canvis introduïts en l'atac PTW:

- Mentre que en l'atac Klein, un cop determinat el valor de $K[3]$, es busca el de $K[4]$, el de $K[5]$, etc., en l'atac PTW es busquen les sumes acumulades dels bytes de la clau arrel: $\sigma_3 = K[3], \sigma_4 = K[3] + K[4], \sigma_5 = K[3] + K[4] + K[5]$, i així successivament. Aquestes sumes es poden obtenir si, en comptes de treballar per exemple amb $j_5 = j_4 + V_4[4] + K[4]$, es continua desenvolupant l'expressió i es treballa amb $j_5 = j_3 + V_3[3] + V_4[4] + K[3] + K[4]$.

  Amb aquesta modificació no es redueix el nombre de trames necessàries per a trobar la clau i a més les probabilitats que es compleixin les hipòtesis sobre les sumes de bytes són inferiors a les de les hipòtesis de Klein. Però treballar amb les sumes té l'avantatge de fer molt més ràpida la cerca de claus alternatives quan es descobreix que una clau candidata és incorrecta. Això es deu al fet que, a diferència dels bytes $K[i]$, que s'han de calcular seqüencialment perquè cadascun depèn dels anteriors, cada suma σ_i es pot obtenir de manera independent de les altres.

  A partir de les sumes $\sigma_3, \sigma_4, \ldots \sigma_{15}$ és immediat obtenir $K[3], K[4], \ldots K[15]$ fent unes simples restes. Si amb les sumes més votades s'obté una clau que no és la correcta, es canvia una de les sumes per la següent més votada i només cal tornar a restar les σ_i necessàries per a obtenir una clau nova. Això és molt més ràpid que recalcular els vectors d'estat per a tornar a generar els valors nous dels bytes $K[i]$, com es fa en l'atac Klein.

- Aquesta rapidesa en l'obtenció de claus noves permet implementar algorismes més exhaustius i eficients per a trobar la clau correcta. Per exemple, si ordenem els candidats a cada σ_i de més a menys votat, podem seleccionar un nombre màxim m de candidats més votats per a utilitzar-los en les diferents combinacions de claus possibles. Treballant amb $m = 2$, hi pot haver fins a $2^{13} = 8.192$ combinacions diferents de claus per provar. Amb $m = 3$ el nombre de combinacions ja puja a $3^{13} = 1.594.323$, que pot ser molt elevat si a cada intent s'han de recalcular els vectors d'estat com en l'atac Klein.

  En canvi, l'atac PTW permet utilitzar valors m més grans, i fins i tot valors m_i que siguin diferents per a cada σ_i i que variïn dinàmicament. La tècnica dels nombres dinàmics de candidats consisteix a posar inicialment tots els valors m_i a 1, la qual cosa dóna una única combinació de bytes de la clau. Si aquesta clau no és la correcta, s'incrementa en 1 el màxim m_i corresponent a la suma σ_i en què el candidat de la posició $m_i + 1$ tingui menys diferència de vots respecte al de la posició m_i, i es tornen a provar les combinacions de claus en què intervingui el nou candidat. Si tampoc es troba la correcta, es torna a incrementar un altre m_i, i així successivament.

- L'ús de sumes de bytes σ_n en comptes de bytes $K[n]$ introdueix un problema que no existeix en l'atac Klein. Si en algun byte $K[n]$ es dóna el cas que a partir d'una certa posició $4 \leq p \leq n$ la suma $K[p] + K[p+1] + \cdots + K[n] + p + (p+1) + \cdots + n$ és igual a 0, és molt probable que l'índex j_p sigui igual a j_{n+1}. Això vol dir que a la iteració p del KSA es produirà un intercanvi que desfarà la hipòtesi de treball de l'atac PTW, i no serà aplicable la correlació de Jenkins. Com que aquesta condició és independent del VI, en aquest cas tots els valors de la suma σ_n seran més o menys equiprobables i serà molt més difícil encertar el valor correcte. Llavors es diu que $K[n]$ és un **byte fort** de la clau.

  Una solució consisteix a detectar que un byte és fort quan la distribució dels vots de les sumes candidates s'assembla a una distribució uniforme. Llavors es tracta de deduir per a cada possible valor de p quins són els valors de $K[n]$ que fan que es compleixi la condició de byte fort ($\sum_p^n K[i] + i = 0$), i considerar els resultats obtinguts com a candidats a $K[n]$. Altres solucions són provar per força bruta tots els valors de $K[n]$ o, si hi ha molts bytes forts a la clau i la força bruta no és viable, fer l'atac Klein en comptes del PTW.

La introducció d'aquestes millores permet a l'atac PTW rebaixar el nombre de trames necessàries per a tenir un 50% de probabilitat d'èxit fins a 35.000, i fins a 47.000 per a un 90% d'èxit.

4.3. Eines per a explotar les vulnerabilitats WEP

Després que es van publicar els diferents atacs contra el protocol WEP, es van desenvolupar eines que implementaven aquests atacs, moltes de codi obert o programari lliure. Una de les primeres va ser AirSnort (2001). Després, la publicació dels atacs FMS i KoreK va donar lloc al projecte Aircrack (2004). A partir d'aquest desenvolupament i d'altres, com ara l'eina Wesside (2004), es va crear Aircrack Next Generation o Aircrack-ng (2006). Des de la versió 0.9 (2007), Aircrack-ng implementa l'atac PTW.

Aircrack-ng és de fet un paquet que incorpora diverses eines, entre les quals hi ha `airmon-ng`, `aireplay-ng`, `airodump-ng` i `aircrack-ng` mateix.

L'eina `airmon-ng`

Aquesta utilitat serveix per a posar la targeta Wi-Fi en mode monitor, i d'aquesta manera poder capturar trames enviades per altres estacions. El mode monitor és equivalent al mode promiscu sense la necessitat d'establir una associació amb cap altra estació o AP.

L'eina `aireplay-ng`

Aquesta eina permet injectar trames noves o prèviament capturades. D'aquesta manera es pot forçar la generació de trames WEP de resposta en cas que no hi hagi estacions Wi-Fi actives pels voltants.

L'eina `aireplay-ng` pot treballar en diferents modes, entre els quals podem destacar els següents:

- **Mode desautenticació**. En aquest mode es generen trames falses de desautenticació destinades a una estació autenticada amb l'AP. L'objectiu és provocar que aquesta estació iniciï una nova autenticació i, depenent del sistema operatiu amb què treballi, enviï una petició ARP per tal d'esbrinar l'adreça IP de l'AP que fa d'encaminador.

- **Mode autenticació falsa**. En aquest mode es fa un atac de falsificació d'autenticació com el que hem vist anteriorment. Això pot ser necessari per a fer posteriorment altres atacs si no hi ha cap altra estació associada.

- **Mode injecció de paquets ARP**. Aquest és probablement el mode més efectiu per a forçar l'enviament de trames WEP i poder capturar així els VI necessaris per a obtenir la clau. En aquest mode, l'eina escolta el medi fins que detecta una petició ARP.

  El paquet ARP estarà xifrat, però és relativament fàcil detectar que una trama conté una petició ARP. En primer lloc perquè la longitud de les dades xifrades serà de 40 bytes: 8 de capçalera LLC, 28 del paquet ARP mateix, i 4 d'ICV. I després perquè l'adreça de destinació, que no està xifrada, serà l'adreça de difusió (*broadcast*).

  Un cop capturada la petició ARP, l'eina `aireplay-ng` comença a retransmetre-la una vegada darrere l'altra, aprofitant la vulnerabilitat que ja hem vist d'injecció de trames. Si l'adreça IP sol·licitada és la de l'AP, aquest enviarà tantes trames WEP amb paquets ARP de resposta com peticions rebi, i cada resposta estarà xifrada amb un IV diferent. Si la petició original l'havia enviat una estació sol·licitant l'adreça d'una altra estació, l'AP retransmetrà la petició, l'estació sol·licitada enviarà la resposta a l'AP i l'AP retransmetrà la resposta al sol·licitant original, de manera que per cada trama injectada se'n generaran 3 de noves, cadascuna amb IV propi.

  Aquest atac sol ser molt productiu, perquè normalment els filtres de paquets deixen passar sense restriccions el trànsit del protocol ARP i els sistemes de detecció d'intrusions no prenen cap acció especial amb aquest tipus de paquets. L'estació que havia generat la petició original pot rebre moltíssimes respostes, però el més habitual és que les ignori. Amb aquesta tècnica, doncs, és fàcil obtenir uns quants centenars d'IV per segon, i aconseguir en menys d'un minut els necessaris per a fer un atac PTW.

Longitud dels paquets ARP

En comptes de buscar només paquets amb 40 bytes de dades xifrades, `aireplay-ng` busca paquets que en tinguin 40 o 58 bytes. El motiu és que si l'origen és un ordinador d'una xarxa amb fil, haurà afegit bytes de farciment fins a arribar a la longitud mínima de les dades d'una trama Ethernet (46 bytes).

L'eina `airodump-ng`

Aquesta eina és l'equivalent de la utilitat `Tcpdump` de les xarxes amb fil: captura les trames que detecta i permet guardar-les en un fitxer. Les trames poden provenir d'un atac actiu provocat amb `aireplay-ng`, o bé ser capturades en un atac passiu simplement escoltant el medi. En aquest últim cas és molt probable que la majoria de trames capturades corresponguin a paquets IP.

Quan guarda les trames capturades en fitxer, `airodump-ng` té l'opció de guardar-hi només la informació útil per a l'eina `aircrack-ng`: l'IV i els primers bytes de *keystream* de cada trama.

Figura 10. Estructura d'un paquet ARP en una trama Wi-Fi

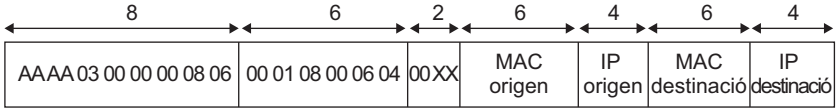
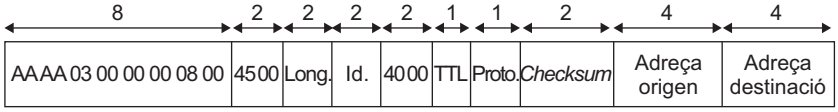


Figura 11. Estructura d'una capçalera IPv4 en una trama Wi-Fi



El *keystream* s'obté deduint el tipus de trama de què es tracta. Si per la seva longitud s'infereix que conté un paquet ARP, es poden conèixer com a mínim els primers 22 bytes de dades desxifrades, i restant les dades xifrades s'obté el *keystream*:

LLC/SNAP

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

- Els primers 8 bytes són la capçalera LLC/SNAP, amb valor fix: en hexadecimal, AA:AA (punts d'accés origen i destinació), 03 (codi de control), 00:00:00 (codi d'organització) i 08:06 (tipus Ethernet: ARP).

- Els 6 bytes següents són la capçalera ARP, també amb valor fix: en hexadecimal, 00:01 (tipus de protocol: Ethernet), 08:00 (protocol de xarxa: IPv4), 06 (longitud d'adreça MAC), 04 (longitud d'adreça IP).

- Els 2 bytes següents són el codi d'operació ARP: 00:01 (petició) o 00:02 (resposta).

- Els 6 bytes següents són l'adreça MAC d'origen, que ha de coincidir amb la que hi ha a la capçalera MAC 802.11 (no xifrada).

A continuació hi ha l'adreça IP d'origen, que no es pot deduir a partir de la mateixa trama, però sí del context observant altres trames. L'adreça MAC de destinació, o bé és zero en les peticions, o es troba desxifrada a la capçalera 802.11 en les respostes. Finalment hi ha l'adreça IP de destinació, que també es pot deduir del context.

Altres tipus de trames poden contenir paquets del protocol STP o, per defecte, s'assumeix que contenen paquets IPv4. En aquest últim cas es poden obtenir els primers 12 bytes de *keystream* a partir dels valors dels bytes desxifrats, fent certes suposicions que es compleixen en la gran majoria dels casos, com per exemple que el paquet no està fragmentat:

STP

STP és la sigla de *spanning tree protocol*.

- Els 8 primers bytes són la capçalera LLC/SNAP, igual que la de les trames amb paquets ARP, però canviant el tipus Ethernet a 08:00 (IPv4).

- En els 2 bytes següents hi ha la versió del protocol, la longitud de la capçalera i el camp de serveis diferenciats. En la gran majoria de paquets IPv4 aquests dos bytes són, en hexadecimal, 45:00.

- Els 2 bytes següents són la longitud del paquet IP, que es pot deduir per la longitud de la trama.

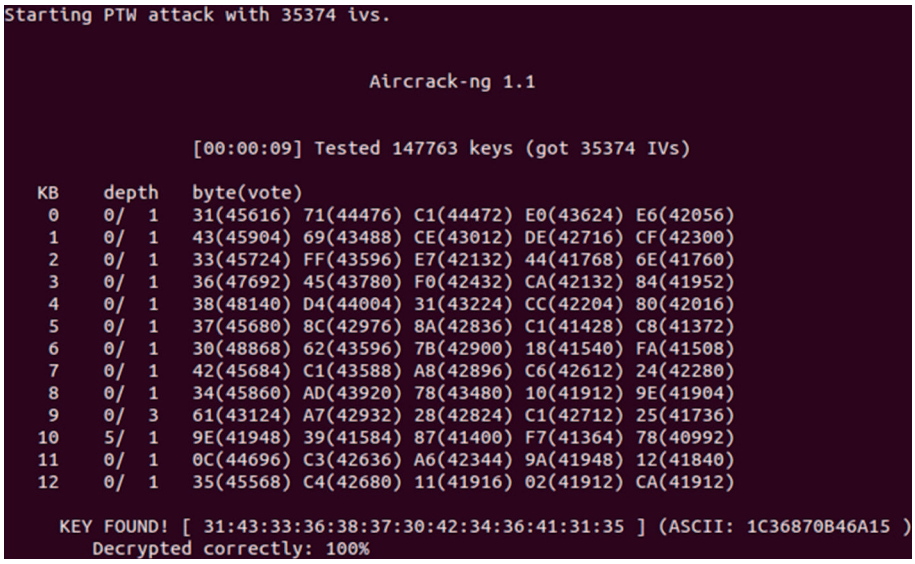
A continuació hi ha els 2 bytes de l'identificador de datagrama, que en general tindran un valor desconegut. A l'hora de trobar la clau WEP amb l'atac PTW, Aircrack-ng fa una cerca de tots els valors possibles d'aquests dos bytes si només disposa de paquets IPv4. En els 2 bytes següents, si el paquet no està fragmentat, hi haurà els valors 40:00 o 00:00, depenent de si està activat l'indicador DF (*Don't fragment*) o no. El primer d'aquests dos bytes és l'últim que s'utilitza per a obtenir el *keystream* que necessita l'atac PTW. Aircrack-ng assumeix que un 85% dels paquets tindran l'indicador activat i assigna el pes corresponent als vots que obtingui amb aquesta suposició. La resta de camps de la capçalera IPv4 poden tenir valors més indeterminats, però ja no s'utilitzen en l'atac PTW.

L'eina `aircrack-ng`

Aquesta és l'eina que a partir dels IV obtinguts implementa l'atac per a recuperar la clau WEP. Per defecte aplica l'atac PTW i els atacs KoreK en paral·lel, que inclouen l'atac FMS, però té opcions per a deshabilitar els atacs que no interessi fer. També pot executar automàticament un atac Klein quan a la clau hi ha molts bytes forts resistents a l'atac PTW.

La figura següent mostra un exemple del resultat obtingut amb una execució de l'eina `aircrack-ng`. En aquest exemple l'atac ha tardat 9 segons a trobar la clau correcta a partir de poc més de 35.000 IV. Mentre prova possibles valors de la clau, l'eina va mostrant per pantalla el nombre de vots ponderats que obtenen els valors candidats de cada byte de la clau.

Figura 12. Exemple d'execució de l'eina Aircrack-ng



5. Solucions a les vulnerabilitats WEP

Quan es va veure que el disseny inicial de la seguretat en l'estàndard IEEE 802.11 tenia deficiències importants, es van proposar diverses solucions per a intentar corregir aquests problemes.

Entre les propostes que es van fer podem destacar les següents:

- En l'article que descrivia l'atac FMS, de l'any 2001, els seus autors suggerien aplicar-lo a les trames en les quals l'IV comença per $K[0] = n$ i $K[1] = 255$ quan s'està buscant el valor $K[n]$ ($3 \leq n < 8$), perquè en aquests casos és més fàcil que es compleixi la hipòtesi de l'atac. Per tant, una primera solució era no enviar trames WEP xifrades amb aquests IV, anomenats **vectors d'inicialització febles**. La realitat és que aquesta mesura només incrementa molt lleugerament el nombre de trames que necessita l'atacant per a descobrir la clau, però tot i així la majoria de sistemes operatius la van incorporar al seu nucli, i a les targetes Wi-Fi que la implementaven en el seu maquinari els van donar l'etiqueta comercial **WEPplus**.

- Una altra solució proposada l'any 2001, anomenada **WEP2**, es basava en l'ús de vectors d'inicialització de 128 bits i claus arrel secretes també de 128 bits, és a dir, claus RC4 de 256 bits en total. Això de fet no impedeix fer els atacs dissenyats per al protocol WEP original, però sí que allarga el temps necessari per a completar-los amb èxit, tot i que el creixement no és exponencial sinó només lineal.

- Alguns fabricants van optar per una solució més efectiva, coneguda com **Dynamic WEP**. Com el nom indica, aquesta tècnica consisteix a anar canviant dinàmicament les claus WEP, cosa que complica considerablement els atacs respecte a les claus estàtiques. Les diferents implementacions, però, no eren interoperables entre si perquè cada fabricant seguia les seves pròpies especificacions.

A partir de la proposta de les claus dinàmiques van començar els treballs d'estandardització que donarien lloc a la publicació de l'especificació **IEEE 802.11i** l'any 2004. En l'edició de 2007, aquesta extensió va deixar de ser una especificació separada i es va incorporar com un capítol de l'estàndard base IEEE 802.11.

Mentre s'estava elaborant el text d'aquesta especificació, i atesa la urgència per a resoldre els problemes que presentava el protocol WEP, l'associació de fabricants Wi-Fi Alliance va desenvolupar una solució intermèdia anomenada **WPA**, amb la intenció que es pogués utilitzar amb el mateix maquinari de les targetes Wi-Fi existents, o introduint-hi només unes quantes modificacions al microprogramari (*firmware*). Aquesta solució estava basada en els esborranys que anava publicant el grup de treball IEEE 802.11i. Quan se'n va aprovar la versió oficial l'any 2004, l'estàndard IEEE 802.11i va ser incorporat a les especificacions de la Wi-Fi Alliance amb el nom de **WPA2**.

WPA

WPA és la sigla de *Wi-Fi protected access*.

5.1. WPA

L'estàndard WPA introdueix canvis fonamentals tant en el mètode d'autenticació de les estacions, com en l'algorisme de xifratge de les trames.

A diferència del protocol WEP, en què normalment hi ha una sola clau secreta compartida per l'AP i les estacions, WPA preveu l'ús de claus diferents en cada **associació segura**, és a dir en cada RSNA, i defineix els mecanismes per a establir aquestes claus dinàmicament.

RSNA

RSNA és la sigla de *Robust Security Network Association*.

L'ús d'una clau única compartida entre l'AP i les estacions, com preveu el protocol WEP, pot ser apropiada per a una xarxa sense fil domèstica, però és més problemàtica en una xarxa corporativa mitjana o gran. Quan hi ha desenes o centenars d'estacions amb la mateixa clau, si un atacant accedeix a la clau en una de les estacions, automàticament les comunicacions de totes les altres queden compromeses. A més, canviar la clau pot requerir actualitzacions manuals en cadascuna de les estacions, cosa que pot ser poc pràctica.

Per això, WPA preveu l'ús del mètode de control d'accés a la xarxa definit en un altre estàndard de la sèrie IEEE 802, concretament l'**IEEE 802.1X**. Aquest estàndard facilita l'intercanvi segur de claus de sessió entre dos nodes de la xarxa, amb una autenticació mútua prèvia. Que l'autenticació sigui mútua en WPA implica que l'estació s'autentica davant l'AP, però l'AP també s'autentica davant l'estació, perquè aquesta es pugui assegurar que no està parlant amb un AP falsificat.

L'estàndard IEEE 802.1X es basa al seu torn en el protocol EAP, que permet dur a terme una autenticació treballant al nivell d'enllaç, és a dir, sense necessitat de tenir assignada encara una adreça de xarxa (IP). EAP preveu l'ús de diversos mètodes d'autenticació i, com el nom indica, se n'hi poden afegir altres de definits en altres especificacions. Així, per mitjà de l'EAP es pot fer una autenticació basada, per exemple, en noms d'usuari i contrasenyes, en

EAP

EAP és la sigla d'*extensible authentication protocol*. Aquest protocol està definit a l'especificació RFC 3748.

claus públiques i certificats X.509, en dispositius físics com ara targetes amb xip, etc.

El que fa IEEE 802.1X és definir un format de trames anomenat EAPOL per a enviar els missatges del protocol EAP per una xarxa local. D'altra banda, en la terminologia IEEE 802.1X l'extrem de la comunicació EAP que sol·licita l'autenticació s'anomena **suplicant**, i l'altre extrem, el que la concedeix, s'anomena **autenticador**. L'autenticador pot concedir l'autenticació per si mateix, o bé pot comunicar-se amb un **servidor d'autenticació** que pren la decisió final. Típicament el servidor utilitzarà un protocol com ara RADIUS o Diameter per a dur a terme l'autenticació.

EAPOL

EAPOL és la sigla de *EAP over LANs*.

WPA també continua permetent l'ús d'una clau compartida o PSK, per simplicitat en el cas de xarxes petites com solen ser les xarxes domèstiques. Però en aquest cas la clau de xifratge no és directament la clau compartida més un VI, com en el protocol WEP, sinó que la clau compartida s'utilitza per a derivar les corresponents claus de sessió per a cada associació.

PSK

PSK és la sigla de *pre-shared key*.

En qualsevol cas, cada parell estació-AP utilitza les seves pròpies claus per a protegir les seves comunicacions. Així s'aconsegueix que una estació no pugui espiar les trames enviades entre l'AP i una altra estació del mateix BSS.

Aquest esquema, però, té un inconvenient: mentre que amb una clau compartida és fàcil enviar una trama xifrada simultàniament a més d'un node, com en el cas del trànsit de difusió (*broadcast*) o de difusió selectiva (*multicast*), amb claus independents seria necessari enviar tantes trames com destinataris, cadascuna xifrada amb la clau corresponent. Per a evitar aquesta ineficiència, en WPA es treballa amb dos tipus de claus:

- Les **claus entre parelles** són les que s'utilitzen per a les trames entre cada parell de nodes, és a dir entre l'AP i cada estació.

- Les **claus de grup** són conegudes per tots els membres del BSS i s'utilitzen per a les trames de difusió (*broadcast*) o de difusió selectiva (*multicast*). Es pot generar una clau de grup nova cada vegada que una estació abandona el BSS i es dissocia de l'AP, per a evitar que pugui continuar desxifrant el trànsit del grup.

5.1.1. Autenticació WPA i gestió de claus

WPA defineix dos modes d'autenticació:

- El **mode WPA-PSK**. És el que treballa amb una clau mestra predefinida, compartida entre l'AP i les estacions. Com hem comentat abans, sol usar-se només en xarxes amb poques estacions. La Wi-Fi Alliance també va donar a aquest mode el nom **WPA-Personal**.

- El **mode WPA-802.1X**. És el que utilitza el control d'accés basat en IEEE 802.1X més EAP, juntament amb un servidor d'autenticació. La Wi-Fi Alliance també va donar a aquest mode el nom **WPA-Enterprise**.

Tant si s'utilitza un mode com l'altre, una estació que vol entrar en un ESS ha de seguir aquests passos:

1) L'estació ha d'identificar l'ESS al qual vol accedir. Mitjançant les trames balisa descobreix la informació que necessita sobre l'ESS i l'AP que el gestiona, com ara el BSSID (és a dir, l'adreça MAC de l'AP), les velocitats de transmissió suportades, etc.

Entre els camps de la trama balisa, també anomenats *IE*, n'hi pot haver un de tipus RSN (*robust security network*). Si aquest IE és present, vol dir que l'AP suporta l'establiment d'associacions segures WPA. Els diferents subcamps de l'IE RSN indiquen els algorismes d'autenticació i de xifratge suportats.

IE

IE és la sigla d'*information element*.

2) Per compatibilitat amb els sistemes que implementen la màquina d'estats 802.11, l'estació primer ha de fer una autenticació de sistema obert, com hem vist a l'apartat 2, seguida d'una associació 802.11.

La trama de gestió que conté la sol·licitud d'associació inclou un IE de tipus RSN en què s'especifiquen l'algorisme d'autenticació i el de xifratge que l'estació està disposada a utilitzar, entre els anunciats per l'AP en les trames balisa. L'algorisme d'autenticació escollit determina si aquesta es farà en mode WPA-PSK o en mode WPA-802.1X.

Autenticació de clau compartida

Recordeu que l'autenticació de clau compartida és totalment insegura, i per això l'estàndard WPA no la recull.

3) L'estació i l'AP estableixen de manera segura una **clau mestra entre parelles** o PMK de 256 bits.

- Si es treballa en mode WPA-PSK, la clau mestra PMK és directament la clau compartida PSK prèviament configurada.

  Moltes vegades, per a facilitar la configuració de les estacions, la PSK no s'especifica directament sinó com una frase de pas (*passphrase*). En aquests casos els bits de la PSK són el resultat d'aplicar una funció de generació de claus, definida en l'estàndard PKCS#5 i basada en funcions resum (*hash*), a partir de la frase de pas i l'SSID.

- Si es treballa en mode WPA-802.1X, s'inicia el protocol EAP per a dur a terme l'autenticació. L'estació es posa d'acord amb l'autenticador, que pot ser l'AP o un servidor d'autenticació, sobre el mètode EAP que s'utilitzarà. Aquest mètode ha de garantir que un espia que observi la comunicació no pugui obtenir cap contrasenya o altra informació secreta que li permeti fer una autenticació fraudulenta.

PMK i MSK

PMK és la sigla de *pairwise master key*, i MSK és la sigla de *master session key*.

Llavors suplicant i autenticador s'intercanvien els missatges EAP necessaris fins a completar l'autenticació. Si el procés acaba amb èxit, el resultat és que l'estació i l'AP s'han autenticat mútuament de manera satisfactòria, i a més el mètode EAP utilitzat també ha de proporcionar un valor de 512 bits que s'utilitzarà com a **clau mestra de sessió** o MSK.

Finalment s'obté la PMK, que és igual als primers 256 bits de l'MSK.

4) L'autenticació es completa executant una **negociació en 4 passos** o *4-way handshake*. D'una banda, aquesta negociació permet verificar que tant l'AP com l'estació han obtingut correctament la clau mestra PMK, i d'aquesta manera comprovar que són els autèntics. I d'altra banda, com a resultat de la negociació s'obtenen també les claus necessàries per a protegir les trames WPA, tant entre parelles com de grup.

Els missatges de la *4-way handshake* s'envien en un tipus especial de trama, anomenat *EAPOL-Key*, definit en l'estàndard IEEE 802.1X. Durant la negociació s'obtenen una clau de xifratge i una clau d'autenticació de missatge, anomenades *KEK* i *KCK* respectivament, per ser utilitzades exclusivament en la negociació. Els missatges de la negociació s'envien xifrats amb RC4, excepte els dos primers perquè la clau KEK encara no està disponible, i autenticats amb HMAC-MD5, excepte el primer perquè la clau KCK tampoc està disponible. A més, un dels camps de les trames EAPOL-Key és un comptador per a detectar atacs de repetició.

KEK, KCK, PTK i GTK

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

L'intercanvi de missatges en la *4-way handshake* és el següent:

1) L'autenticador (AP) envia al suplicant (estació) un valor aleatori N_A.

2) El suplicant genera un altre valor aleatori N_S i calcula la **clau transitòria entre parelles** o PTK, de 512 bits. El càlcul es fa aplicant una funció unidireccional a la clau mestra PMK, els valors aleatoris N_A i N_S, i les adreces MAC d'autenticador i suplicant. Llavors s'obtenen les claus KCK i KEK prenent els primers 128 bits de la PTK i els 128 bits següents, respectivament.

Un cop obtingudes aquestes claus, el suplicant envia el valor N_S a l'autenticador.

3) L'autenticador fa els mateixos càlculs per a obtenir la clau PTK, i a partir d'aquesta, la KCK i la KEK. Llavors envia la **clau temporal de grup** o GTK al suplicant, xifrada amb la KEK.

4) El suplicant comprova que el missatge anterior és correcte. Si és així, l'autenticitat de l'autenticador (AP) haurà quedat confirmada. El suplicant envia

llavors a l'autenticador un missatge que no cal que contingui res en el seu camp de dades. Si l'autenticador veu que és correcte, l'autenticitat del suplicant (estació) també haurà quedat confirmada.

Com a resultat del procés anterior, l'AP i l'estació han acordat una clau PTK que utilitzaran entre si. D'aquesta PTK, prenent els bits 256-511, s'obté una **clau temporal** o TK, que serà la que es farà servir per a xifrar i autenticar les trames WPA.

TK

TK és la sigla de *temporal key*.

En la negociació, a més, l'AP envia a l'estació la clau de grup GTK. Aquesta clau de grup la determina unilateralment l'AP. La manera d'obtenir-la és una qüestió interna de l'AP però, per analogia amb la PTK, es pot obtenir per exemple aplicant una funció unidireccional a una clau mestra de grup o GMK més un valor aleatori N_G.

Un cop establerta la sessió, el protocol *4-way handshake* es pot tornar a iniciar en qualsevol moment per a renegociar la clau transitòria PTK; per exemple quan la sessió és llarga i ja fa cert temps que s'està utilitzant la mateixa clau. En aquest cas, l'enviament de la clau GTK és opcional si no ha canviat.

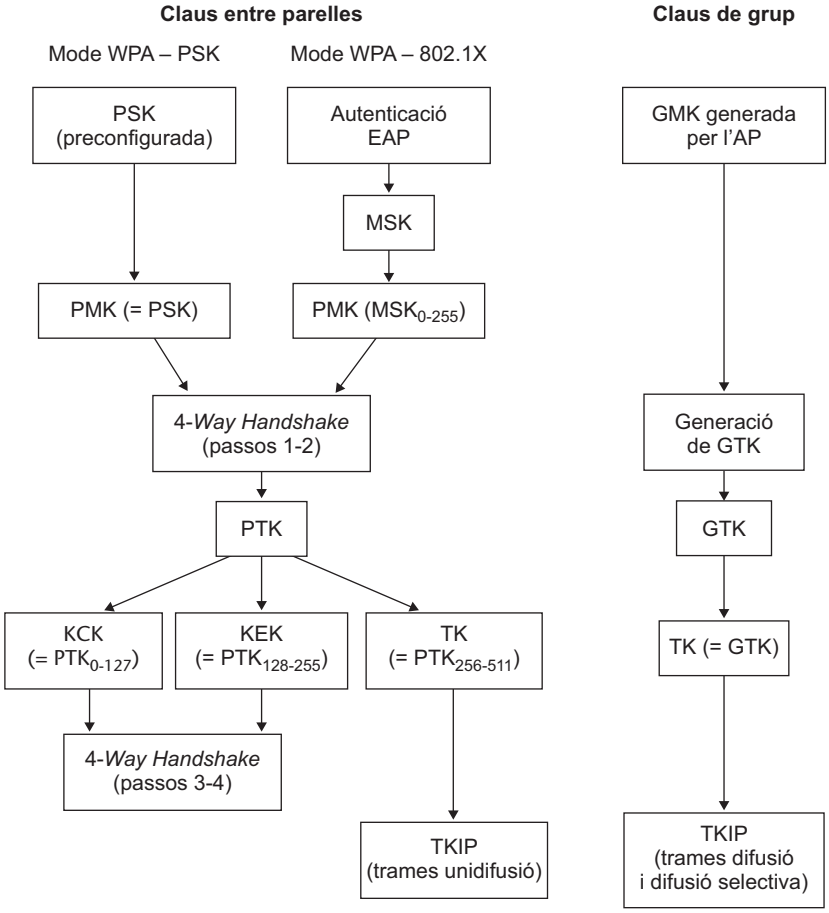
I en el moment en què canviï la GTK, per exemple perquè una estació ha sortit del BSS i ja no ha de continuar rebent trànsit de difusió, es fa un altre tipus de negociació anomenat *group key handshake* entre l'AP i cada estació. Aquesta negociació té dos passos perquè només cal enviar la nova clau GTK xifrada a l'estació, i que aquesta confirmi que l'ha rebut.

Com es pot comprovar, l'autenticació WPA introdueix fortes mesures de seguretat per a evitar qualsevol tipus d'atac: un mecanisme segur per a derivar una clau mestra PMK que és diferent per a cada sessió amb cada estació (excepte en el mode WPA-PSK), una clau transitòria PTK que es pot anar canviant periòdicament, i un protocol de generació de claus temporals en quatre passos afegit al mètode d'autenticació, amb l'ús de claus criptogràfiques independents de les de la comunicació normal, derivades de les adreces MAC i amb comptadors per a evitar atacs de repetició.

Una feblesa d'aquest esquema és l'ús dels algorismes RC4 i MD5 per al xifratge i l'autenticació de la negociació en quatre passos, que no són tan segurs com altres algorismes que s'han desenvolupat posteriorment. Però l'objectiu inicial de l'estàndard WPA era que es pogués utilitzar amb el maquinari disponible, i aquesta era una solució de compromís mentre no s'estenia la implementació del WPA2. D'altra banda, en el mode WPA-PSK una estació que capturi els valors N_A i N_S de la negociació d'una altra estació, que no s'envien xifrats, immediatament sabrà quina és la seva clau PTK i podrà desxifrar-ne el trànsit.

A mode de resum, el diagrama següent mostra les relacions entre les diferents claus que formen l'anomenada *jerarquia de claus WPA*.

Figura 13. Jerarquia de claus WPA



Mètodes d'autenticació EAP usats en WPA-802.1X

Actualment hi ha desenes de mètodes EAP, entre els estandarditzats per l'IETF i els definits per diversos fabricants. Alguns dels usats més habitualment en el mode WPA-802.1X són els següents:

- **EAP-TLS**. En aquest mètode la comunicació amb el servidor d'autenticació, per exemple un servidor RADIUS, es protegeix mitjançant el protocol TLS amb autenticació mútua basada en certificats de servidor i de client.

- **EAP-TTLS** (*EAP-Tunneled TLS*). És una variant simplificada del mètode anterior en què no són necessaris els certificats de client, la qual cosa el fa molt més pràctic. S'utilitza el protocol TLS per a crear un canal segur o "túnel" només amb certificat de servidor. Llavors es fa l'autenticació del client amb un altre mètode, que pot ser per exemple basat en contrasenya, per mitjà d'aquest canal segur.

- **PEAP** (*Protected EAP*). És un mètode genèric per encapsular l'autenticació de client dins un altre mètode amb autenticació de servidor, per exemple basat en TLS.

A més dels passos per a fer l'autenticació, cadascun d'aquests mètodes ha de definir també com es genera el valor que es farà servir com a clau mestra de sessió (MSK).

Els mètodes com EAP-TTLS o PEAP estableixen l'autenticació del servidor, però llavors cal usar un altre mètode per a l'autenticació del client. Aquest altre mètode pot ser, per exemple:

Generació de l'MSK

En els mètodes EAP basats en TLS, l'MSK es genera normalment aplicant una funció unidireccional a les cadenes de bits aleatòries utilitzades en la fase de negociació TLS (*handshake protocol*) i el secret mestre que se n'obté.

- **EAP-MD5**. És un mètode de repte-resposta. La resposta és un *hash* MD5 d'una cadena formada per la contrasenya del client més el repte.

- **EAP-MSCHAPv2** (*EAP-Microsoft challenge handshake authentication protocol version 2*). Utilitza el protocol MSCHAPv2, definit en l'especificació RFC 2759.

- **EAP-GTC** (*EAP-generic token card*). També és un mètode de repte-resposta en el qual la resposta és generada per un dispositiu físic, com pot ser una targeta amb xip.

5.1.2. El xifratge TKIP

A més del mètode d'autenticació, l'altre canvi fonamental introduït en WPA respecte a WEP és l'algorisme de xifratge. O més exactament, la generació de les claus de xifratge, ja que l'algorisme pròpiament dit és el mateix: RC4. Això es va decidir, com ja hem vist, per a intentar aprofitar el maquinari de les targetes de xarxa que hi havia llavors.

L'esquema de xifratge que s'utilitza en l'estàndard WPA s'anomena *TKIP*. Les diferències principals que presenta respecte a l'esquema WEP són les següents:

TKIP

TKIP és la sigla de *temporal key integrity protocol*.

- La clau amb què s'encripten les dades de cada trama no s'obté a partir d'un vector d'inicialització variable i una part fixa, sinó que tots els bits de la clau RC4 es recalculen a cada trama.

- Les trames TKIP incorporen un codi MIC calculat a partir d'una clau secreta, com a prevenció contra els atacs de modificació o truncament com el *chopchop*. El codi MIC no substitueix sinó que complementa el camp ICV. D'altra banda, quan es produeix fragmentació aquest codi es calcula sobre la trama original abans de fragmentar, en comptes d'haver-hi un codi MIC per cada fragment.

MIC

MIC és la sigla de *message integrity code*, que és la nomenclatura amb què IEEE 802.11 es refereix al codi d'autenticació de missatge, per a evitar confusions amb *medium access control* (MAC).

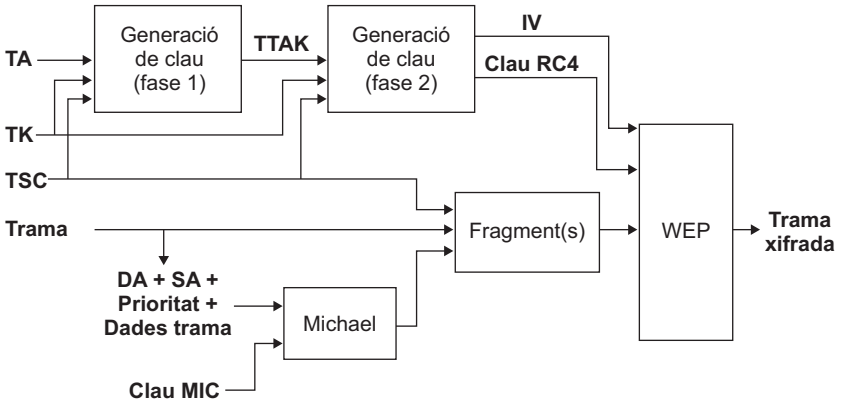
- Per a evitar atacs d'injecció, el codi MIC no es calcula només sobre les dades xifrades sinó que també s'hi afegeixen les adreces MAC de les estacions origen i destinació.

- Cada trama inclou un comptador de seqüència de 48 bits, anomenat *TSC*, com a mesura contra els atacs de repetició. Aquest comptador es reini-

cialitza a 1 cada vegada que es fa servir una clau temporal TK nova. El comptador N d'una trama enviada després que una altra amb comptador M ha de complir $N > M$ (si són trames consecutives, pot ser per exemple $N = M + 1$, però no necessàriament). Les trames rebudes que no segueixin aquesta regla són descartades.

Algunes trames poden tenir assignada una prioritat, i pot passar que trames de prioritats diferents siguin rebudes en ordre diferent al d'enviament. Per tant, emissor i receptor han de mantenir un comptador TSC independent per cada prioritat utilitzada (n'hi pot haver com a màxim 8 de diferents).

Figura 14. Generació de trames xifrades TKIP



TSC

TSC és la sigla de *TKIP sequence counter*.

El procés que se segueix en l'algorisme TKIP per a generar una trama xifrada inclou els passos següents:

1) Es genera el codi MIC aplicant un algorisme anomenat *Michael* a les entrades següents:

- La informació següent de la trama: l'adreça MAC de destinació (DA), l'adreça MAC d'origen (SA), la prioritat i el camp de dades.
- La clau MIC de 64 bits. Per a evitar atacs de repetició en sentit contrari s'utilitzen dues claus MIC diferents per a les trames de l'AP a l'estació i per a les trames de l'estació a l'AP. La primera s'obté dels bits 128-191 de la clau TK, i la segona dels bits 192-255 de la mateixa clau.

L'algorisme Michael és una funció resum senzilla amb operacions simples que es pot calcular molt ràpidament. No és, però, una funció resum segura perquè no té la propietat de la unidireccionalitat.

2) En cas que sigui necessari, s'aplica la fragmentació a la trama més el codi MIC. A cada fragment se li assigna un comptador TSC diferent, sempre respectant l'ordre creixent.

3) S'aplica una funció criptogràfica, anomenada **fase 1**, a les entrades següents:

- La clau temporal TK obtinguda en la *4-way handshake*. Com a clau per al xifratge s'utilitzen els primers 128 bits (0-127) de la clau TK.

- L'adreça MAC de l'estació transmissora, TA.

- El comptador TSC. Per a la fase 1 s'utilitzen els 24 bits de més pes del comptador.

El resultat de la fase 1 és un valor TTAK (*TKIP-mixed transmit address and key*) de 80 bits. Aquest valor serà el mateix per a totes les trames que tinguin els mateixos 24 bits de més pes del comptador TSC, i per tant no caldrà recalcular-lo cada vegada.

4) A continuació s'aplica una altra funció criptogràfica, anomenada **fase 2**, a les entrades següents:

- El resultat TTAK de la fase 1.

- La mateixa clau de xifratge que en la fase 1 (els bits 0-127 de la clau TK).

- El comptador TSC. Per a la fase 2 s'utilitzen els 24 bits de menys pes del comptador.

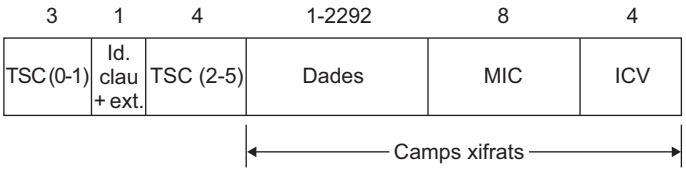
El resultat de la fase 2 és una clau de xifratge RC4 de 128 bits amb 24 bits d'IV i 104 bits de clau arrel.

La propietat principal del xifratge TKIP és que els 104 bits de clau arrel són diferents per a cada trama, amb la qual cosa els atacs estadístics contra el xifratge WEP no són aplicables.

L'IV es construeix de manera que els bytes primer i tercer es copien dels 16 bits de menys pes del TSC, i el segon byte es deriva del primer amb la precaució que el resultat no sigui un VI feble, és a dir, que no tingui el segon byte igual a 255.

5) Finalment la trama, o cada fragment de trama si és el cas, se xifra igual que en el protocol WEP.

Figura 15. Dades d'una trama TKIP



L'estructura de la trama xifrada TKIP que es genera és lleugerament diferent de la de les trames WEP normals.

- El primer camp conté l'IV, igual que en les trames WEP, en aquest cas obtingut a partir dels dos bytes de menys pes del TSC.

- El camp següent té activat un bit d'extensió per a indicar que a continuació hi ha un camp addicional.

- El camp addicional d'extensió s'aprofita per a incloure-hi la resta de bytes del TSC (2-5).

- A continuació de les dades de la trama, i abans del camp ICV, s'insereixen els 64 bits del codi MIC.

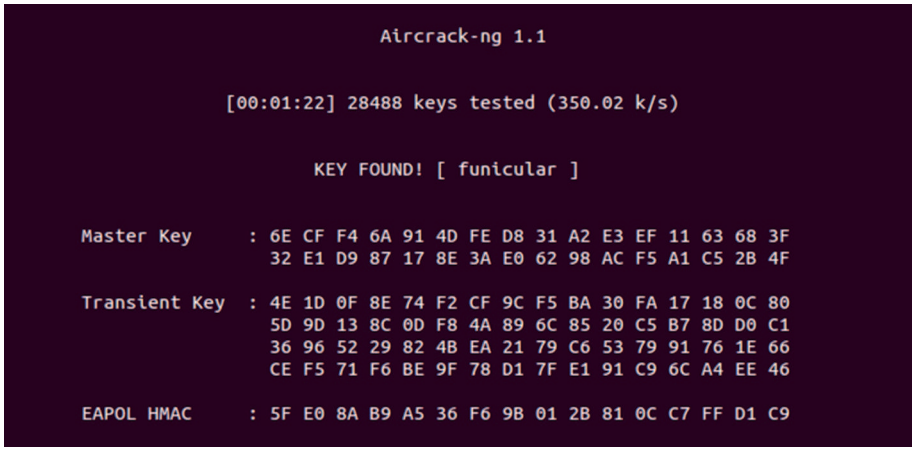
5.1.3. Vulnerabilitats i contramesures

La vulnerabilitat principal del sistema WPA es troba en l'ús del mode d'autenticació de clau compartida, WPA-PSK. Encara que es treballi amb una clau mestra PMK de 256 bits, si aquesta clau prové exclusivament d'una paraula més o menys fàcil de recordar, l'espai de claus possibles queda molt reduït i fa viable un atac per força bruta. Per exemple, l'eina `aircrack-ng` permet fer un atac de diccionari sobre els paquets de la negociació *4-way handshake* entre una estació i l'AP. A diferència dels atacs WEP, que com més trames tinguin disponibles més ràpidament poden trobar la clau, l'atac de diccionari WPA només necessita les trames d'una sola negociació. De fet, amb dues de les quatre trames n'hi ha prou. Aquestes trames es poden obtenir amb `airodump-ng`, esperant de manera passiva que alguna estació estableixi una associació amb l'AP o bé provocant de manera activa l'associació amb el mode desautenticació de l'eina `aireplay-ng`. L'atac consisteix a anar provant, per a cada paraula del diccionari, si les claus que se'n deriven quadren amb el contingut xifrat i autenticat de les trames capturades.

La conclusió és que si s'utilitza el mode WPA-PSK cal escollir una clau que no sigui cap paraula de diccionari en cap idioma ni una combinació trivial de paraules (per exemple, una paraula escrita del revés). S'aconsella utilitzar frases llargues, de 20 caràcters com a mínim, que no continguin paraules de diccionari.

La figura següent mostra l'exemple d'una execució de l'eina `aircrack-ng` per a descobrir una clau WPA-PSK amb l'atac de diccionari. En menys d'un minut i mig, i després de provar poc més de 28.000 paraules de diccionari, en aquest exemple l'eina ha trobat que la clau és la paraula *funicular*.

Figura 16. Exemple d'execució d'un atac de diccionari WPA



Pel que fa al protocol TKIP, com hem vist abans, el seu disseny inclou un seguit de proteccions criptogràfiques per a evitar els atacs d'injecció, de repetició, de modificació, de truncament, etc. Però a més l'especificació inclou també una mesura en el funcionament del protocol per a intentar contrarestar els atacs contra el codi MIC. L'objectiu és evitar atacs d'assaig i error, de l'estil de l'atac *chopchop* sobre l'ICV. Si un atacant aconseguís trencar el codi MIC podria injectar trames correctes.

Per a evitar-ho, el protocol TKIP està dissenyat per a alentir la velocitat a la qual es poden fer els intents d'atac. Concretament, l'especificació estableix que les trames amb els camps CRC, ICV o TSC incorrectes han de ser ignorades. Però si aquests camps són correctes i el codi MIC és erroni, això ha de ser considerat com un possible atac i senyalitzat com a tal en els registres (*logs*) de seguretat. L'estació que detecti l'error ha d'enviar a l'AP un tipus especial de trama EAPOL-Key, anomenat *Michael MIC failure report*. I si es detecten dues trames errònies en menys de 60 segons, s'ha de deshabilitar la recepció de trames TKIP durant un minut. Quan es restableixi, les claus haurien de ser renegociades.

Això implica que un atacant no podrà fer més de dos intents per minut. Tot i així s'han proposat atacs, com l'anomenat *atac Beck-Tews*, que en certes condicions teòricament permetrien a un atacant desxifrar els últims 12 bytes d'una trama (MIC i ICV) en poc més de 12 minuts. Si és una trama ARP amb contingut conegut, es pot obtenir fàcilment la clau MIC ja que l'algorisme Michael no està dissenyat per a ser unidireccional. I en 4 o 5 minuts més es podria obtenir prou *keystream* per a poder injectar determinats tipus de trames.

5.2. WPA2

L'especificació WPA2 incorpora tota la funcionalitat de l'estàndard IEEE 802.11i. Els canvis que introdueix respecte a WPA són de dos tipus:

- D'una banda defineix mecanismes com la preautenticació i l'emmagatzemament de claus mestres (*PMK caching*) que fan més ràpida i eficient la reautenticació d'una estació mòbil quan surt d'un BSS i entra en un BSS adjacent del mateix ESS (*roaming*).

- D'altra banda introdueix un algorisme de xifratge nou, anomenat **CCMP**, que no està basat en l'RC4 sinó en la xifra AES-128. Aquest mètode de xifratge és molt més segur, ja que actualment no se'n coneixen vulnerabilitats significatives, i tot i que no és tan senzill d'implementar com l'RC4, és força més eficient que la majoria d'altres xifres de bloc existents.

Els sistemes WPA2 han de suportar obligatòriament el xifratge CCMP. L'ús del xifratge TKIP és opcional, per compatibilitat amb els sistemes WPA. D'altra banda, quan es treballa amb CCMP els algorismes criptogràfics utilitzats en la *4-way handshake* són AES (segons l'estàndard RFC 3394) per al xifratge i HMAC-SHA1 per a l'autenticació de missatge, en comptes de RC4 i HMAC-MD5, respectivament.

CCMP

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

El xifratge CCMP consisteix a aplicar el mode CCM definit a l'especificació RFC 3610 a la xifra de bloc AES amb clau de 128 bits. El mode CCM proporciona alhora autenticació de missatge i confidencialitat, tot amb la mateixa clau. La clau CCMP és de 128 bits i s'obté, com en TKIP, dels bits 0-127 de la clau temporal TK.

Bits de la clau TK

Els primers 128 bits de la clau TK són els únics que es necessiten en el protocol CCMP perquè s'utilitzen tant per a xifrar com per a autenticar. Quan es treballa amb CCMP, doncs, no cal generar 512 bits de PTK en els passos 2 i 3 de la *4-way handshake* sinó que n'hi ha prou amb 384.

- El codi d'autenticació MAC es genera fent un xifratge AES-128 en mode CBC amb la tècnica coneguda com a *CBC-MAC*. El vector d'inicialització, segons l'especificació CCM, té 1 byte d'indicadors, 2 bytes que indiquen la longitud de les dades, i els altres 13 bytes han de ser únics per a cada trama. Per a aconseguir-ho, aquests 13 bytes es construeixen de la manera següent:

– 1 byte codifica la prioritat.

– Els 6 bytes següents són l'adreça MAC (*medium access control*) de l'estació transmissora.

– Els últims 6 bytes són un **número de paquet** o PN (*packet number*) de 48 bits que s'incrementa a cada trama.

  Després d'aquest vector d'inicialització, els 2 blocs següents que es xifren contenen una combinació dels camps invariants de la capçalera MAC (*medium access control*) de la trama, que són bàsicament tots excepte el camp Durada/ID.

  A continuació d'aquests 2 blocs es xifren les dades de la trama, completades al final amb bytes iguals a 0 si la seva longitud no és múltipla de 16. Del

resultat de xifrar l'últim bloc es prenen els primers 64 bits, i aquest serà el codi MAC que autenticarà la trama.

- Les dades xifrades s'obtenen aplicant un xifratge en el mode comptador (*CTR mode*) segons la terminologia CCM. Per a cada bloc de dades que es vol xifrar, es construeix un bloc auxiliar que té 1 byte d'indicadors, 2 bytes que contenen un comptador igual a 1 per al primer bloc i que s'incrementa a cadascun dels blocs següents, i els altres bytes són iguals als 13 bytes únics que s'utilitzen per a generar el codi MAC.

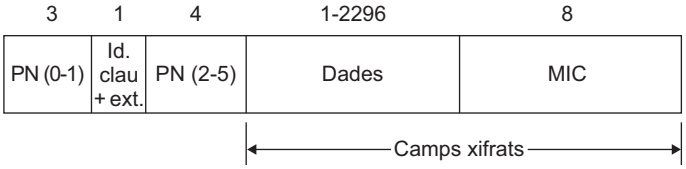
  Llavors se xifra aquest bloc auxiliar amb AES-128, i el resultat se suma bit a bit (amb l'operació XOR) amb el bloc corresponent que es vol xifrar, com si fos una xifra de flux. Si la longitud de l'últim bloc és menor de 16 bytes, només s'utilitzen els bytes del bloc auxiliar xifrat que calguin.

Terminologia CCM

En el context de l'especificació CCM, MAC significa *message authentication code*, i vector d'inicialització té el sentit que se li dóna normalment en les xifres de bloc, és a dir, el bloc aleatori que s'utilitza com si fos l'anterior al primer bloc que es vol xifrar.

El codi MAC obtingut en el primer pas també se xifra sumant-lo amb un altre bloc auxiliar xifrat, en el qual el comptador és igual a 0. El resultat és el codi MIC, segons la terminologia WPA.

Figura 17. Dades d'una trama CCMP



Després d'aplicar l'autenticació de missatge i el xifratge ja es pot generar la trama xifrada CCMP. La seva estructura és semblant a la de les trames TKIP, substituint els bytes del comptador TSC pels del número de paquet PN, i amb la diferència principal que en CCMP no s'inclou el camp ICV. Aquest camp, que en TKIP serveix per a reforçar el codi MIC basat en l'algorisme Michael, no es considera necessari en CCMP atesa la fortalesa de l'autenticació CBC-MAC amb la xifra AES.

Els únics atacs pràctics que es coneixen sobre el sistema WPA2 són els mateixos que afecten el sistema WPA quan s'utilitza el mode PSK. És a dir, WPA2-PSK és exactament igual de vulnerable a atacs de diccionari que WPA-PSK, ja que aquests atacs actuen sobre la negociació *4-way handshake* i són independents de l'algorisme de xifratge de les trames.

Resum

En aquest mòdul didàctic hem estudiat els diferents mecanismes que existeixen per a la protecció de la informació en les xarxes WLAN.

En particular hem vist com el protocol WEP, per mitjà del criptosistema en flux RC4, permet xifrar la informació que viatja per la xarxa per a dificultar-ne la intercepció per part d'un atacant. Tot i la millora que suposa el protocol WEP respecte a enviar la informació en clar, també hem pogut veure que aquest protocol presenta un seguit de debilitats que fan que sigui atacable de diferents maneres. A més, existeixen un seguit d'eines específiques, com l'Aireplay-ng o l'Aircrack-ng, que permeten explotar les vulnerabilitats del protocol WEP.

Finalment, hem estudiat que el protocol WPA protegeix les xarxes WLAN de manera molt més efectiva. En concret, hem vist els diferents modes de funcionament del WPA i com gestiona de manera més segura tant la fase d'autenticació dels usuaris com el sistema de xifrat de la informació.

Glossari

AP Access Point. Estació específica que permet la interconnexió amb altres xarxes, amb fil o sense.

EAP (Extensible authentication Protocol) Protocol que permet dur a terme una autenticació treballant al nivell d'enllaç, és a dir, sense necessitat de tenir assignada encara una adreça de xarxa (IP).

IEEE 802.11 Estàndard definit per l'Institute of Electrical and Electronics Engineers per a les comunicacions en xarxes locals sense fils, també conegut com a Wi-Fi.

PSK Pre-Shared Key. Clau compartida.

RC4 Algorisme criptogràfic de xifrat en flux dissenyat per Ronald Rivest (d'aquí l'acrònim Ron's code 4) que s'utilitza per a xifrar les trames WEP.

service set identifier *m* Identificador de format lliure de fins a 32 bytes que fa referència a una estació sense fils.

SSID *m* Vegeu ***service set identifier***.

TKIP Temporal Key Integrity protocol. Esquema de xifratge que s'utilitza en l'estàndard WPA basat en l'algorisme de xifrat en flux RC4.

WEP *m* Vegeu ***wired equivalent privacy***.

wired equivalent privacy *m* Sistema de protecció que incorpora l'estàndard IEEE 802.11 per a tecnologia LAN sense fils.

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

Bibliografia

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

IEEE Computer Society (1999). *Std. 802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Nova 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 Berlin.