# WiFi biztonság – A jó, a rossz, és a csúf

Buttyán Levente és Dóra László
Budapesti Műszaki és Gazdaságtudományi Egyetem
Híradástechnikai tanszék
CrySyS Adatbiztonság Laboratórium
{buttyan, doralaca}@crysys.hu

***Kivonat:*** *Jelen cikkben ismeretterjesztő jellegű áttekintést adunk a WiFi biztonsághoz kapcsolódó szabványokról, a WEP-ről és a 802.11i-ről.*

***Kulcsszavak:*** *WiFi, WLAN, WEP, 802.11, 802.11i, WPA, WPA2, RSN, 802.1X, EAP, RADIUS, TKIP, RC4, CCMP, AES, hitelesítés, hozzáférés-védelem, integritás-védelem, rejtjelezés*

## 1. Bevezetés

Napjainkban a vezeték nélküli hálózatok egyre nagyobb teret nyernek. A vezeték nélküli hálózatok nagy előnye az eszközök, s ezzel a felhasználók mobilitásának támogatása, hátrányuk viszont, hogy – a fizikai kapcsolatok hiánya és a rádiós csatorna jellege miatt – több potenciális támadásnak vannak kitéve, mint vezetékes társaik általában. Fontos tehát, hogy a vezeték nélküli hálózatok megfelelő védelmi mechanizmusokkal legyenek ellátva, melyek minden körülmények között (azaz rosszindulatú támadások esetén is) biztosítják a biztonságos működést.

Ebben a cikkben a 802.11 vezeték néküli LAN szabványhoz, közismertebb nevén a WiFi-hez kapcsolódó biztonsági problémákkal és megoldásokkal foglalkozunk. Cikkünk ismeretterjesztő jellegű, saját eredményeket nem tartalmaz, csupán tömör áttekintést ad a WiFi biztonsághoz kapcsolódó szabványokról és a mások által elért tudományos és gyakorlati eredményekről. Az érdeklődő olvasó a tématerület bővebb kifejtését találja [Edney+04]-ben.

A továbbiakban először a WEP működését és hibáit mutatjuk be. A WEP az első biztonsági architektúra, melyet 802.11 hálózatok számára javasoltak, ám hamar kiderült, hogy nem nyújt megfelelő védelmet. Utána a 802.11i szabványt ismertetjük, mely a WEP utódjának tekinthető. Áttekintjük a 802.11i-ben javasolt biztonsági architektúra elemeit: a hitelesítési és hozzáférés-védelmi mechanizmust, a kulcsmenedzsmentet, valamint a TKIP és az AES-CCMP protokollokat.

## 2. WEP

Az IEEE 802.11 vezeték nélküli LAN szabvány tervezői kezdettől fogva fontosnak tartották a biztonságot. Ezért már a 802.11 korai verziója [802.11] is tartalmazott biztonsági mechanizmusokat, melyek összességét WEP-nek (Wired Equivalent Privacy) nevezték el. Ahogy arra a név is utal, a WEP célja az, hogy a vezeték nélküli hálózatot *legalább* olyan biztonságossá tegye, mint egy – különösebb biztonsági kiegészítésekkel nem rendelkező – vezetékes hálózat. Ha például egy támadó egy vezetékes Ethernet hálózathoz szeretne csatlakozni, akkor hozzá kell férnie az Ethernet hub-hoz. Mivel azonban a hálózati eszközök általában fizikailag védve, zárt szobában találhatók, ezért a

támadó nehézségekbe ütközik. Ezzel szemben egy védelmi mechanizmusokat nélkülöző vezeték nélküli LAN-hoz való hozzáférés – a rádiós csatorna nyitottsága miatt – triviális feladat a támadó számára. A WEP ezt a triviális feladatot hivatott megnehezíteni. Fontos azonban megjegyezni, hogy a WEP tervezői nem törekedtek „tökéletes" biztonságra, mint ahogy a zárt szoba sem jelent tökéletes védelmet egy Ethernet hub számára.

A tervezők tehát nem tették túl magasra a lécet, ám a WEP még ezt a korlátozott célt sem érte el. Pár évvel a megjelenése után, a kriptográfusok és az IT biztonsági szakemberek súlyos biztonsági hibákat találtak a WEP-ben [Walker00, Borisov+01, Arbaugh+02], s nyílvánvalóvá vált, hogy a WEP nem nyújt megfelelő védelmet. A felfedezést tett követte, és hamarosan megjelentek az Interneten a WEP feltörését automatizáló programok. Válaszul, az IEEE új biztonsági architektúrát dolgozott ki, melyet a 802.11 szabvány *i* jelzésű kiegészítése tartalmaz [802.11i]. A 802.11i-t a következő fejezetben tárgyaljuk. Ebben fejezetben a WEP működését és hibáit tekintjük át. Ezt azért tartjuk szükségesnek, mert – bár a WEP-en már túlhaladt a kor – a legtöbb forgalomban levő hálózati eszköz még mindig támogatja a WEP-et. Azok a felhasználók, akik WEP-et használnak, jobb ha tisztában vannak annak korlátaival.

### *2.1 A WEP működése*

Vezeték néküli hálózatok esetében két alapvető biztonsági probléma merül fel. Egyrészt a rádiós csatorna jellege miatt a kommunikáció könnyen lehallgatható. Másrészt – s ez talán fontosabb – a hálózathoz való csatlakozás nem igényel fizikai hozzáférést a hálózati csatlakozóponthoz (Access Point, vagy röviden AP), ezért bárki megpróbálhatja a hálózat szolgáltatásait illegálisan igénybe venni. A WEP az első problémát az üzenetek rejtjelezésvel igyekszik megoldani, a második probléma megoldása érdekében pedig megköveteli a csatlakozni kívánó mobil eszköz (Station, vagy röviden STA) hitelesítését az AP felé.

A hitelesítést egy egyszerű kihívás-válasz alapú protokoll végzi, mely négy üzenet cseréjéből áll. Elsőként a STA jelzi, hogy szeretné hitelesíteni magát (authenticate request). Válaszul az AP generál egy véletlen számot, s azt kihívásként elküldi a STA-nak (authenticate challenge). A STA rejtjelezi a kihívást, s az eredményt visszaküldi az AP-nak (authenticate response). A STA a rejtjelezést egy olyan titkos kulccsal végzi, melyet csak a STA és az AP ismer. Ezért ha az AP sikeresen dekódolja a választ (azaz a dekódolás eredményeként visszakapja saját kihívását), akkor elhiszi, hogy a választ az adott STA állította elő, hiszen csak az ismeri a helyes válasz generálásához szükséges titkos kulcsot. Más szavakkal, a válasz sikeres dekódolása esetén az AP hitelesítette a STA-t, és ennek megfelelően dönthet arról, hogy a csatlakozást engedélyezi vagy sem. A döntésről az AP a protokoll negyedik üzenetében tájékoztatja a STA-t (authenticate success vagy failure).

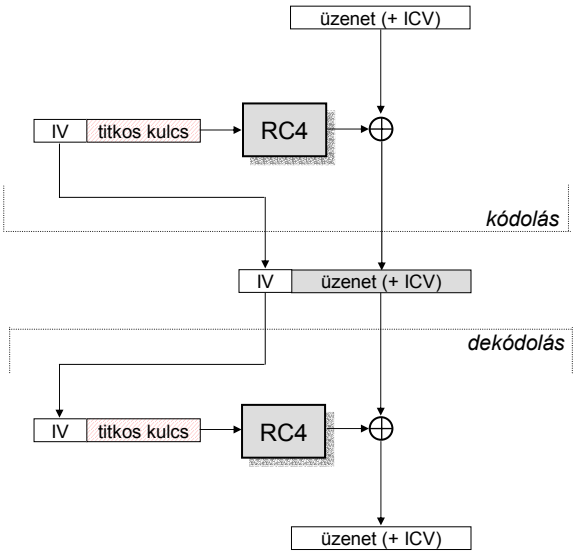
Miután a hitelesítés megtörtént, a STA és az AP üzeneteiket rejtjelezve kommunikálnak. A rejtjelezéshez ugyanazt a titkos kulcsot használják, mint a hitelesítéshez. A WEP rejtjelező algoritmusa az RC4 kulcsfolyam kódoló. A kulcsfolyam kódolók úgy működnek, hogy egy kis méretű, néhány bájtos titkos kulcsból egy hosszú álvéletlen bájtsorozatot állítanak elő, és ezen sorozat bájtjait XOR-olják az üzenet bájtjaihoz. Ez történik a WEP esetében is. Az $M$ üzenet küldője (a STA vagy az AP) a titkos kulccsal inizializálja az RC4 kódolót, majd az RC4 által előállított $K$ álvéletlen bájtsorozatot

XOR-olja az üzenethez. Az $M \oplus K$ rejtjelezett üzenet vevője ugyanazt teszi mint a küldő: a titkos kulccsal inicializálja az RC4 algoritmust, amely így ugyanazt a $K$ álvéletlen bájtsorozatot állítja elő, amit a rejtjelezéshez használt a küldő. Ezt a rejtjelezett üzenethez XOR-olva – az XOR művelet tulajdonságai miatt – a vevő az eredeti üzenetet kapja vissza: $(M \oplus K) \oplus K = M$.

A fent leírtak majdnem megfelelnek a valóságnak, van azonban még valami amit a WEP rejtejelezés kapcsán meg kell említeni. Könnyen látható, hogy ha a rejtjelezés a fentiek szerint működne, akkor minden üzenethez ugyanazt a $K$ álvéletlen bájtsorozatot XOR-olnánk, hiszen a kódolót minden üzenet elküldése előtt ugyanazzal a titkos kulccsal inicializáljuk. Ez több szempontból is hiba lenne. Tegyük fel például, hogy egy támadó lehallgat két rejtjelezett üzenetet, $M_1 \oplus K$-t és $M_2 \oplus K$-t. A két rejtjelezett üzenetet XOR-olva, a támadó a két nyílt üzenet XOR összegét kapja: $(M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2$. Ez olyan, mintha az egyik üzenetet a másik üzenettel, mint kulcsfolyammal rejtjeleztük volna. Ám ebben az esetben $M_1$ és $M_2$ nem álvéletlen bájtsorozatok. Valójában tehát $M_1 \oplus M_2$ egy nagyon gyenge rejtjelezés, és a támadó az üzenetek statisztikai tulajdonságait felhasználva könnyen meg tudja fejteni mindkét üzenetet. Az is elképzelhető, hogy a támadó esetleg (részlegesen) ismeri az egyik üzenet tartalmát, s annak segítségével a másik üzenet (részleges) tartalmához azonnal hozzájut.

Ezen problémák elkerülése érdekében, a WEP nem egyszerűen a titkos kulcsot használja a rejtjelezéshez, hanem azt kiegészíti egy IV-nek (Initialization Vector) nevezett értékkel, mely üzenetenként változik. A rejtjelezés folyamata tehát a következő: az IV-t és a titkos kulcsot összefűzzük, a kapott értékkel inicializáljuk az RC4 kódolót, mely előállítja az álvéletlen bájtsorozatot, amit az üzenethez XOR-olunk. A dekódolás folyamata ezzel analóg. Ebből következik, hogy a vevőnek szüksége van a kódolásnál használt IV-re. Ez a rejtjelezett üzenethez fűzve, nyíltan kerül átvitelre. Ez elvileg nem jelent problémát, mert az üzenet dekódolásához csupán az IV ismerete nem elegendő, ahhoz a titkos kulcsot is ismerni kell. A méreteket illetően megemlítjük – s ennek később még lesz jelentősége – hogy az IV 24 bites, a titkos kulcs pedig (általában) 104 bites[1]. A WEP rejtjelezés teljes folyamatát az 1. ábra szemlélteti.

[1] Különböző marketing anyagokban ezt gyakran úgy interpretálják, hogy a WEP „128 bites biztonságot” nyújt. Ez természetesen félrevezető (mint a marketing anyagok általában), hiszen a 128 bitből 24 nyíltan kerül átvitelre.




**1. ábra:** A WEP rejtjelezés folyamata

Az 1. ábra azt is mutatja, hogy a rejtjelezés előtt, a küldő egy integritás-védő ellenőrző összeggel (Integrity Check Value, vagy röviden ICV) egészíti ki a nyílt üzenetet, melynek célja a szándékos módosítások detektálásának lehetővé tétele a vevő számára. A WEP esetében az ICV nem más mint a nyílt üzenetre számolt CRC érték. Mivel azonban a CRC önmagában nem véd a szándékos módosítások ellen (hiszen egy támadó a módosított üzenethez új CRC értéket tud számolni), ezért a WEP a CRC értéket is rejtjelezi. A mögöttes gondolat az, hogy így a támadó nem tudja manipulálni az üzeneteket, hiszen a titkos kulcs hiányában nem tudja a módosított üzenethez tartozó rejtjelezett CRC értéket előállítani. Mint azt alább látni fogjuk, ez a gondolatmenet nem teljesen hibamentes.

Végezetül a WEP kulcsokról szólunk röviden. A szábvány lehetővé teszi, hogy minden STA-nak saját titkos kulcsa legyen, amit csak az AP-vel oszt meg. Ez azonban megnehezíti a kulcsmenedzsmentet az AP oldalán, mivel ekkor az AP-nek minden STA kulcsát ismernie és gondoznia kell. Ezért a legtöbb implementáció nem támogatja ezt a lehetőséget. A szabvány előír egy ún. default kulcsot is, amit az AP és a hálózathoz tartozó *minden* STA ismer. Eredetileg ezt a kulcsot azon üzenetek védelmére szánták, melyeket az AP többesszórással (broadcast) minden STA-nak el szeretne küldeni. A legtöbb WEP implementáció azonban csak ezt a megoldást támogatja. Így a gyakorlatban, egy adott hálózathoz tartozó eszközök egyetlen közös kulcsot használnak titkos kulcsként. Ezt a kulcsot manuálisan kell telepíteni a mobil eszközökben és az AP-ben. Nyílvánvaló, hogy ez a megoldás csak egy külső támadó ellen biztosítja a kommunikáció biztonságát; az eszközök (elvileg) dekódolni tudják egymás üzeneteit.

*2.2 A WEP hibái*

A WEP tulajdonképpen a rossz protokolltervezés mintapéldája. Az alábbi tömör összefoglalóból látható, hogy lényegében egyetlen kitűzött biztonsági célt sem valósít meg tökéletesen:

**Hitelesítés:** A WEP hitelesítési eljárásának több problémája is van. Elsőként mindjárt az, hogy a hitelesítés egyirányú, azaz a STA hitelesíti magát az AP felé, ám az AP nem hitelesíti magát a STA felé. Másodszor, a hitelesítés és a rejtjelezés ugyanazzal a titkos kulccsal történik. Ez azért nem kívánatos, mert így a támadó mind a hitelesítési, mind pedig a rejtjelezési eljárás potenciális gyengeségeit kihasználhatja egy, a titkos kulcs megfejtésére irányuló támadásban. Biztonságosabb lenne, ha minden funkcióhoz külön kulcs tartozna.

A harmadik probléma az, hogy a protokoll csak a hálózathoz történő csatlakozás pillanatában hitelesíti a STA-t. Miután a hitelesítés megtörtént és a STA csatlakozott a hálózathoz, bárki küldhet a STA nevében üzeneteket annak MAC címét használva. Úgy tűnhet, hogy ez annyira nem nagy gond, hiszen a támadó, a titkos kulcs ismeretének hiányában, úgysem tud helyes rejtjelezett üzenetet fabrikálni, amit az AP elfogad. Ám ahogy azt korábban említettük, a gyakorlatban az összes STA egy közös titkos kulcsot használ, s így a támadó megteheti azt, hogy egy $STA_1$ által küldött – és a támadó által lehallgatott – rejtejelezett üzenetet $STA_2$ nevében megismétel az AP felé; ezt az AP el fogja fogadni.

A negyedik probléma egy gyöngyszem a protokolltervezési hibák között. Emlékeztetünk arra, hogy a WEP rejtjelezési algoritmusa az RC4 folyamkódoló. Nemcsak az üzeneteket kódolják az RC4 segítségével, hanem a STA ezt használja a hitelesítés során is az AP által küldött kihívás rejtjelezésére. Így a támadó a hitelesítés során küldött üzenetek lehallgatásával könnyen hozzájut a $C$ kihíváshoz és az arra adott $R = C \oplus K$ válaszhoz, melyből $C \oplus R = K$ alapján azonnal megkapja az RC4 algoritmus által generált $K$ álvéletlen bájtsorozatot. A játéknak ezzel vége, hiszen $K$ segítségével a támadó bármikor, bármilyen kihívásra helyes választ tud generálni a STA nevében (s ezen az IV használata sem segít, mert az IV-t a rejtjelezett üzenet küldője, jelen esetben a támadó választja). Sőt, mivel a gyakorlatban minden, az adott hálózathoz tartozó eszköz ugyanazt a titkos kulcsot használja, a támadó ezek után bármelyik eszköz nevében csatlakozni tud a hálózathoz. Persze a csatlakozás önmagában még nem elegendő, a támadó használni is szeretné a hálózatot. Ehhez olyan üzeneteket kell fabrikálnia, amit az AP elfogad. A rejtjelezés követelménye miatt ez nem triviális feladat (hiszen magához a titkos kulcshoz még nem jutott hozzá a támadó), de a WEP hibáinak tárháza bőven tartogat még lehetőségeket.

**Integritás-védelem:** A WEP-ben az üzenetek integritásának védelmét az üzenetekhez csatolt ellenőrző összeg (ICV) hivatott biztosítani. Az ICV nem más, mint az üzenetre számolt CRC érték, mely az üzenettel együtt rejtjelezésre kerül. Formális jelöléseket használva, a rejtjelezett üzenet a következő módon írható fel: $(M \,||\, \mathrm{CRC}(M)) \oplus K$, ahol $M$ a nyílt üzenet, $K$ az RC4 által az IV-ből és a titkos kulcsból előállított álvéletlen bájtsorozat, CRC(.) jelöli a CRC függvényt, és || jelöli az összefűzés műveletét. Ismeretes, hogy a CRC lineáris művelet az XOR-ra nézve, azaz $\mathrm{CRC}(X \oplus Y) = \mathrm{CRC}(X) \oplus \mathrm{CRC}(Y)$. Ezt kihasználva, a támadó a rejtjelezett WEP üzenetek bármely bitjét

módosítani tudja (át tudja billenteni), bár magát az üzenetet nem látja. Jelöljük a támadó szándékolt módosításait $\Delta M$-mel. Ekkor a támadó az $((M \oplus \Delta M) \,||\, \mathrm{CRC}(M \oplus \Delta M)) \oplus K$ rejtjelezett üzenetet szeretné előállítani az eredetileg megfigyelt $(M \,||\, \mathrm{CRC}(M)) \oplus K$ rejtjelezett üzenetből. Ehhez egyszerűen $\mathrm{CRC}(\Delta M)$-et kell kiszámolnia, majd a $\Delta M \,||\, \mathrm{CRC}(\Delta M)$ értéket kell az eredeti rejtjelezett üzenethez XOR-olnia. A következő egyszerű levezetés mutatja, hogy ez miért vezet célra:

$$\begin{aligned}
&((M \,||\, \mathrm{CRC}(M)) \oplus K) \oplus (\Delta M \,||\, \mathrm{CRC}(\Delta M)) \;= \\
&((M \oplus \Delta M) \,||\, (\mathrm{CRC}(M) \oplus \mathrm{CRC}(\Delta M))) \oplus K = \\
&((M \oplus \Delta M) \,||\, \mathrm{CRC}(M \oplus \Delta M)) \oplus K
\end{aligned}$$

ahol az utolsó lépésben kihasználtuk a CRC linearitását. Mivel $\mathrm{CRC}(\Delta M)$ kiszámolásához nincs szükség a titkos kulcsra, ezért láthatóan a támadó könnyen tudja manipulálni a WEP üzeneteket, az integritás-védelem és a rejtjelezés ellenére.

Az üzenetfolyam integritásának védelme kapcsán szokás említeni az üzenetvisszajátszás detektálását, mint biztonsági követelményt. A WEP esetében ennek vizsgálatával egyszerű dolgunk van, mert a WEP-ben egyáltalán nincs semmilyen mechanizmus mely az üzenetek visszajátszásának detektálását lehetővé tenné. A tervezők nemes egyszerűséggel erről a biztonsági követelményről megfeledkeztek. A támadó tehát bármely eszköz korábban rögzített üzenetét vissza tudja játszani egy későbbi időpontban, s ezt a WEP nem detektálja. Nyílvánvaló, hogy ez miért gond, ha arra gondolunk, hogy a rögzített üzenet akár egy bejelentkezési folyamatból is származhat, s például egy felhasználói név/jelszó párt tartalmazhat.

**Titkosítás:** Mint azt korábban említettük, folyamkódolók használata esetén fontos, hogy minden üzenet más kulccsal legyen rejtjelezve. Ezt a WEP-ben az IV használata biztosítja; sajnos nem teljesen megfelelő módon. A probléma abból adódik, hogy az IV csak 24 bites, ami azt jelenti, hogy kb. 17 millió lehetséges IV van. Egy WiFi eszköz kb. 500 teljes hosszúságú keretet tud forgalmazni egy másodperc alatt, így a teljes IV teret kb. 7 óra leforgása alatt kimeríti. Azaz 7 óránként ismétlődnek az IV értékek, s ezzel az RC4 által előállított álvéletlen bájtsorozatok is. A problémát súlyosbítja, hogy a gyakorlatban minden eszköz ugyanazt a titkos kulcsot használja, potenciálisan különböző IV értékekkel, így ha egyszerre $n$ eszköz használja a hálózatot, akkor az IV ütközés várható ideje a 7 óra $n$-ed részére csökken. Egy másik súlyosbító tényező, hogy sok WEP implementáció az IV-t nem véletlen értékről indítja, hanem nulláról. Ezért beindítás után a különböző eszközök ugyanazt a nullától induló és egyesével növekvő IV sorozatot használják, legtöbbször ugyanazzal a közös titkos kulccsal. Azaz, a támadónak várakoznia sem kell, azonnal IV ütközésekhez jut.

A WEP teljes összeomlását az RC4 kódoló nem megfelelő használata okozza. Ismeretes, hogy léteznek ún. gyenge RC4 kulcsok, melyekre az a jellemző, hogy belőlük az RC4 algoritmus nem teljesen véletlen bájtsorozatot állít elő [Fluhrer+01]. Ha valaki meg tudja figyelni egy gyenge kulcsból előállított bájtsorozat első néhány bájtját, akkor abból következtetni tud a kulcsra. Ezért a szakemberek azt javasolják, hogy az RC4 által előállított bájtsorozat első 256 bájtját mindig dobjuk el, s csak az utána generált bájtokat használjuk a rejtjelezéshez. Ezzel a gyenge kulcsok problémáját meg lehetne oldani. Sajnos a WEP nem így működik. Ráadásul a változó IV érték miatt előbb-utóbb biztosan gyenge kulcsot kap a kódoló, s az IV nyílt átvitele miatt, erről a támadó is tudomást

szerezhet. Ezt kihasználva, néhány kriptográfus olyan támadó algoritmust konstruált a WEP ellen, melynek segítségével a teljes 104 bites titkos kulcs néhány millió üzenet lehallgatása után nagy valószínűséggel megfejthető. A WEP minden korábban leírt hibája eltörpül ezen eredmény mellett, ugyanis ezzel a támadással magához a titkos kulcshoz jut hozzá a támadó. Ráadásul a támadás könnyen automatizálható, és néhány „segítőkész" embernek köszönhetően, az Internetről letöltött támadó programok használatával amatőrök által is rutinszerűen végrehajtható.

## 3. 802.11i

A WEP hibáit felismerve, az IEEE új biztonsági megoldást dolgozott ki, melyet a 802.11i specifikáció tartalmaz [802.11i]. A WEP-től való megkülönböztetés érdekében, az új koncepciót RSN-nek (Robust Security Network) nevezték el. Az RSN-t körültekintőbben tervezték meg, mint a WEP-et. Új módszer került bevezetésre a hitelesítés és a hozzáférés-védelem biztosítására, mely a 802.1X szabvány által definiált modellre épül, az integritás-védelmet és a titkosítást pedig az AES (Advanced Encryption Standard) algoritmusra támaszkodva oldották meg.

Sajnos azonban az új RSN koncepcióra nem lehet egyik napról a másikra áttérni. Ennek az az oka, hogy a használatban levő WiFi eszközök az RC4 algoritmust támogató hardver elemekkel vannak felszerelve, és nem támogatják az RSN által előírt AES algoritmust. Ezen pusztán szoftver upgrade-del nem lehet segíteni, új hardverre van szükség, s ez lassítja az RSN elterjedésének folyamatát.

Ezt a problémát az IEEE is felismerte, és egy olyan opcionális protokollt is hozzáadott a 802.11i specifikációhoz, mely továbbra is az RC4 algrotimust használja, és így – szoftver upgrade után – futtatható a régi hardveren, de erősebb mint a WEP. Ezt a protokollt TKIP-nek (Temporal Key Integrity Protocol) nevezik.

A WiFi eszközöket gyártó cégek azonnal adaptálták a TKIP protokollt, hiszen annak segítségével a régi eszközökből álló WEP-es hálózatokat egy csapásra biztonságossá lehetett varázsolni. Meg sem várták amíg a 802.11i specifikáció a lassú szabványosítási folyamat során végleges állapotba kerül, azonnal kiadták a WPA (WiFi Protected Access) specifikációt [WPA], ami a TKIP-re épül. A WPA tehát egy gyártók által támogatott specifikáció, mely az RSN egy azonnal használható részhalmazát tartalmazza. A WPA-ban a hitelesítés, a hozzáférés-védelem, és a kulcsok menedzsmentje megegyezik az RSN-ben használt módszerekkel, a különbség csak az integritás-védelemre és a rejtjelezésre használt algoritmuskban mutatkozik.

A továbbiakban áttekintjük a 802.11i-ben definiált hitelesítési, hozzáférés-védelmi, és kulcsmenedzsment módszereket, melyek tehát megegyeznek az RSN-ben és a WPA-ban. Ezt követően röviden összefoglaljuk a TKIP (WPA) és az AES-CCMP (RSN) protokollok működését.

### *3.1 Hitelesítés és hozzáférés-védelem*

A 802.11i-ben a hitelesítés és hozzáférés-védelem modelljét a 802.1X szabványból kölcsönözték [802.1X]. Ezt a szabványt eredetileg vezetékes LAN-ok számára tervezték, de az elvek végülis vezeték nélküli WiFi hálózatokban is ugyanúgy alkalmazhatóak.

A 802.1X modell három résztvevőt különböztet meg a hitelesítés folyamatában: a hitelesítendő felet (supplicant), a hitelesítőt (authenticator), és a hitelesítő szervert (authentication server). A hitelesítendő fél szeretne a hálózat szolgáltatásaihoz hozzáférni, és ennek érdekében szeretné magát hitelesíteni, azaz kilétét bizonyítani. A hitelesítő kontrollálja a hálózathoz történő hozzáférést. A modellben ez úgy történik, hogy a hitelesítő egy ún. port állapotát vezérli. Alapállapotban a porton adatforgalom nincs engedélyezve, ám sikeres hitelesítés esetén a hitelesítő „bekapcsolja" a portot, ezzel engedélyezve a hitelesítendő fél adatforgalmát a porton keresztül. A hitelesítő szerver az engedélyező szerepet játsza. Tulajdonképpen a hitelesítendő fél hitelesítését nem a hitelesítő, hanem a hitelesítő szerver végzi[2], és ha a hitelesítés sikeres volt, engedélyezi, hogy a hitelesítő bekapcsolja a portot.

WiFi hálózatok esetében a hitelesítendő fél a mobil eszköz, mely szeretne a hálózathoz csatlakozni, a hitelesítő pedig az AP, mely a hálózathoz történő hozzáféfést kontrollálja. A hitelesítő szerver egy program, mely kisebb hálózatok esetében akár az AP-ben is futhat, nagyobb hálózatoknál azonban tipikusan egy külön erre a célra dedikált hoszton futó szerver alkalmazás. WiFi esetében a port nem egy fizikai csatlakozó, hanem egy logikai csatlakozási pont, amit az AP-ben futó szoftver valósít meg.

Vezetékes LAN esetében, a hitelesítendő fél egyszer hitelesíti magát, mikor fizikilag csatlakozik a hálózathoz. További védelmi lépésekre nincsen szükség (legalábbis hozzáférés-védelem tekintetében), hiszen a használatba vett portot más úgysem használhatja. Ahhoz ugyanis a hitelesítendő fél és a hitelesítő között létrejött fizikai kapcsolatot kellene megbontani (pl. a hálózati csatlakozót kihúzni egy Ethernet hub-ból). Ezt a hitelesítő hardvere detektálja és a portot azonnal letiltja. WiFi esetében más a helyzet, mert nincsen fizikai kapcsolat a STA és az AP között. Ezért a 802.11i azzal a követelménnyel egészíti ki a 802.1X modellt, hogy a hitelesítés során létre kell jöjjön egy titkos kulcs a STA és az AP között, melyet azok a további kommunikáció kriptográfiai védelmére használhatnak. Ezen kulcs hiányában egy támadó nem tudja a STA és az AP között kialakult logikai kapcsolatot csalárd módon saját céljainak megvalósítására felhasználni.

Maga a hitelesítés az EAP (Extensible Authentication Protocol) segítségével történik [EAP]. Az EAP egy igen egyszerű protokoll, aminek az az oka, hogy nem maga az EAP végzi a hitelesítést. Az EAP csak egy illesztő-protokoll, amit arra terveztek, hogy tetszőleges hitelesítő protokoll üzeneteit szállítani tudja (ezért „extensible"). Egy adott hitelesítő protokoll EAP-ba történő beágyazásának szabályait külön kell specifikálni. Több elterjedt hitelesítő protokollra létezik már ilyen specifikáció (pl. EAP-TLS, LEAP, PEAP, EAP-SIM).

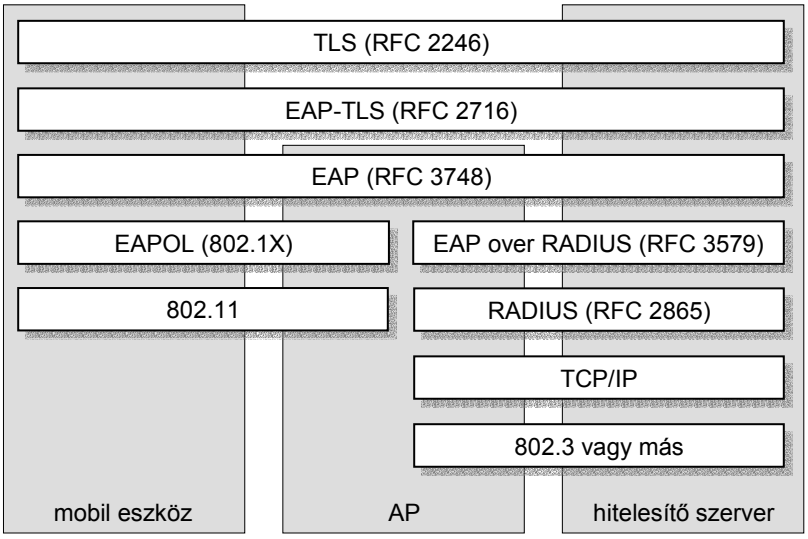
Négy fajta EAP üzenet létezik: request, response, success, és failure. Az EAP request és response üzenetek szállítják a beágyazott hitelesítő protokoll üzeneteit. Az EAP success és failure speciális üzenetek, melyek segítségével a hitelesítés eredményét lehet jelzni a hitelesítendő fél felé.

Ahogy azt fentebb említettük, a 802.1X modellben, a hitelesítendő fél hitelesítését a hitelesítő szerver végzi. WiFi esetében ez azt jelenti, hogy az EAP protokollt (és az abba

[2] Ebből a szempontból az elnevezések kicsit szerencsétlenek.

beágyazott tényleges hitelesítő protokollt, például a TLS-t) lényegében a mobil eszköz és a hitelesítő szerver futtatják. Az AP csak továbbítja az EAP üzeneteket a mobil eszköz és a hitelesítő szerver között, de nem érti azok tartalmát. Az AP csak az EAP success és failure üzeneteket érti meg, ezeket figyeli, és ha success üzenetet lát akkor engedélyezi a mobil eszköz csatlakozását a hálózathoz.

Az EAP üzeneteket a mobil eszköz és az AP között a 802.1X-ben definiált EAPOL (EAP over LAN) protokoll szállítja. Az AP és a hitelesítő szerver között a WPA a RADIUS protokoll [RADIUS] használatát írja elő. A RADIUS-t az RSN opcióként ajánlja, de más alkalmas protokoll használatát is lehetővé teszi a specifikáció. Végülis tetszőleges protokoll használható, amely az EAP üzenetek szállítására alkalmas. Elterjedtsége miatt azonban várhatóan a legtöbb hálózat RADIUS-t használ majd. Az így kialakuló protokoll architektúrát a 2. ábra szemlélteti.

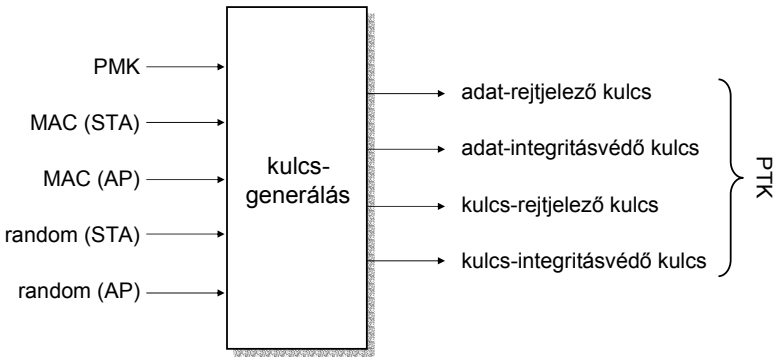



**2. ábra:** Hitelesítési protokoll-architektúra a 802.11i-ben TLS használata esetén

Ahogy azt korábban említettük, a hitelesítés eredményeként nemcsak a hálózathoz való hozzáférést engedélyezi a hitelesítő szerver, hanem egy titkos kulcs is létrejön, mely a mobil eszköz és az AP további kommunikációját hivatott védeni. Mivel a hitelesítés a mobil eszköz és a szerver között folyik, ezért a protokoll futása után ezt a kulcsot csak a mobil eszköz és a hitelesítő szerver birtokolja, és azt még el kell juttatni az AP-hez. A RADIUS protokoll biztosít erre használható mechanismust az MS-MPPE-Recv-Key RADIUS üzenet-attribútum formájában, mely kifejezetten kulcs-szállítás céljára lett specifikálva. A kulcs rejtjelezett formában kerül átvitelre, ahol a rejtjelezés egy a hitelesítő szerver és az AP között korábban létrehozott (tipikusan manuálisan telepített) kulcs segítségével történik.

*3.2 Kulcsmenedzsment*

A hitelesítés során, a mobil eszköz és az AP között létrehozott titkos kulcsot páronkénti mesterkulcsnak (pairwise master key, vagy röviden PMK) nevezik. Azért „páronkénti”, mert csak az adott mobil eszköz és az AP ismeri (na meg a hitelesítő szerver, de az megbízható entitásnak tekinthető), s azért „mester”, mert ezt a kulcsot nem használják közvetlenül rejtjelezésre, hanem további kulcsokat generálnak belőle. Egészen pontosan a PMK-ból mind a mobil eszköz, mind pedig az AP négy további kulcsot generál: egy adat-rejtjelező kulcsot, egy adat-integritás-védő kulcsot, egy kulcs-rejtjelező kulcsot, és egy kulcs-integritás-védő kulcsot. Ezeket együttesen páronkénti ideiglenes kulcsnak (Pairwise Transient Key, vagy röviden PTK) nevezik. Megjegyezzük, hogy az AES-CCMP protokoll az adatok rejtjelezéséhez és az adatok integritás-védelméhez ugyanazt a kulcsot használja, ezért AES-CCMP használata esetén csak három kulcs generálódik a PMK-ból. A PTK előállításához a PMK-n kívül felhasználják még a két fél (mobil eszköz és AP) MAC címét, és két véletlenszámot, melyet a felek generálnak. Ezt a 3. ábra szemlélteti.




**3. ábra:** A PTK generálása a PMK-ból, a felek MAC címéből, és a véletlenszámokból

A véletlenszámokat az ún. *négy utas kézfogás* (four way handshake) protokollt használva juttatják el egymáshoz a felek. Ennek a protokollnak további fontos feladata az, hogy segítségével a felek közvetlenül meggyőződjenek arról, hogy a másik fél ismeri a PMK-t. A négy utas kézfogás protokoll üzeneteit az EAPOL protokoll Key típusú üzeneteiben juttatják el egymáshoz a felek. Az üzenetek tartalma és a protokoll működése vázlatosan a következő:

1. Első lépésként az AP elküldi az általa generált véletlenszámot a mobil eszköznek. Mikor a mobil eszköz ezt megkapja, rendelkezésére áll minden információ a PTK előállításához. A mobil eszköz tehát kiszámolja az ideiglenes kulcsokat.

2. A mobil eszköz is elküldi az általa generált véletlenszámot az AP-nek. Ez az üzenet kriptográfiai integritás-ellenőrző összeggel (Message Integrity Code, vagy röviden MIC) van ellátva, amit a mobil eszköz a frissen kiszámolt kulcs-integritás-védő kulcs segítségével állít elő. Az üzenet vétele után az AP-nek is rendelkezésére áll minden információ a PTK kiszámításához. Kiszámolja az ideiglenes kulcsokat, majd a kulcs-integritás-védő kulcs segítségével ellenőrzi a MIC-et. Ha az ellenőrzés sikeres, akkor elhiszi, hogy a mobil eszköz ismeri a PMK-t.
3. Az AP is küld egy MIC-et tartalmazó üzenetet a mobil eszköznek, melyben tájékoztatja a mobil eszközt arról, hogy a kulcsokat sikeresen telepítette, és készen áll a további adatforgalom rejtjelezésre. Ez az üzenet tartalmaz továbbá egy kezdeti sorozatszámot. A későbbiekben ettől az értéktől kezdik majd sorszámozni a felek az egymásnak küldött adatcsomagokat, és a sorszámozás segítségével detektálják a visszajátszásos támadásokat. Az üzenet vétele után a mobil eszköz a kulcs-integritás-védő kulccsal ellenőrzi a MIC-et, és ha az ellenőrzés sikeres, akkor elhiszi, hogy az AP ismeri a PMK-t.
4. Végül a mobil eszköz nyugtázza az AP előző üzenetét, mely egyben azt is jelenti, hogy a mobil eszköz is készen áll a további adatforgalom rejtjelezésére.

A továbbiakban a mobil eszköz és az AP az adat-integritás-védő és az adat-rejtjelező kulccsal védik egymásnak küldött üzeneteiket. Szükség van azonban még olyan kulcsokra is, melyek segítségével az AP többesszórással küldhet üzeneteket biztonságosan minden mobil eszköz számára. Értelemszerűen, ezeket a kulcsokat az összes mobil eszköznek és az AP-nek is ismernie kell, ezért ezeket együttesen ideiglenes csoportkulcsnak (Group Transient Key, vagy röviden GTK) nevezik. A GTK egy rejtjelező és egy integritás-védő kulcsot tartalmaz. AES-CCMP esetén a két kulcs ugyanaz, ezért csak egy kulcsból áll a GTK. A GTK-t az AP generálja, és a négy utas kézfogás során létrehozott kulcs-rejtjelező kulcsok segítségével titkosítva juttatja el minden mobil eszközhöz külön-külön.

*3.3 TKIP és AES-CCMP*

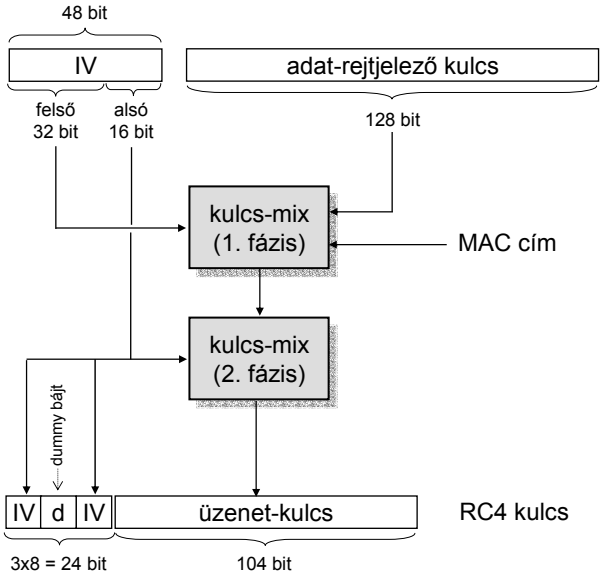
A TKIP (Temporal Key Integrity Protocol) és az AES-CCMP (AES CTR Mode and CBC MAC) a fent leírt kulcsmenedzsment megoldásra támaszkodó protokollok, melyek az üzenetek integritás-védelmével és rejtjelezésével foglalkoznak. Mint azt korábban említettük, a TKIP egy olyan köztes megoldás, amely a régi, WEP-es hardveren is működik, de a WEP-nél jóval magasabb szintű biztonságot nyújt. Az AES-CCMP új hardvert igényel, de cserébe egyszerűbb, tisztább megoldást nyújt, mint a TKIP.

A TKIP a WEP hibáit a következő módon igyekszik javítani:

**Integritás-védelem:** A TKIP egy új integritás-védelmi mechanismussal egészíti ki a WEP-es megoldást (utóbbi általában hardverben van implementálva, úgyhogy benne hagyták a TKIP-ben is). Az új mechanzmust Michael-nek hívják. A Michael SDU szinten működik (azaz a felsőbb protokollszintről a MAC szintre érkező adatokon, fragmentálás előtt végzi az integritás-védő ellenőrző összeg számítását), ami lehetővé teszi a hálózati kártya meghajtó programjában (device driver) történő megvalósítást. Ez azért fontos, mert így a Michael bevezetése egyszerű szoftver upgrade-del megoldható.

Az üzenet-visszajátszás detektálása érdekében a TKIP az IV-t használja sorozatszámként. Ennek megfelelően, a TKIP előírja, hogy az IV értékét minden üzenetben eggyel növelni kell (a WEP-ben ez nem volt kötelező). A vevő nyílvántartja az utolsó néhány vett IV értéket. Ha egy frissen érkezett üzenet IV-je kisebb, mint a legkisebb nyílvántartott IV, akkor a vevő eldobja az üzenetet, míg ha az üzenet IV-je hagyobb, mint a legnagyobb nyílvántartott IV, akkor a vevő megtartja az üzenetet és módosítja a nyílvántartását. Ha egy vett üzenet IV-je a legkisebb és a legnagyobb nyílvántartott IV közé esik, akkor a vevő ellenőrzi, hogy az adott IV szerepel-e a nyílvántartásában. Ha igen, akkor eldobja az üzenetet, ha nem, akkor megtartja azt és módosítja a nyílvántartását.

**Titkosítás:** Emlékeztetünk arra, hogy a WEP titkosítás legfőbb hibáját az IV kis mérete és a gyenge RC4 kulcsok használata jelentette. A TKIP-ben ezért az IV méretét 24 bitről megnövelték 48 bitre. Ez egyszerű megoldásnak látszik, ám a nehézséget az okozza, hogy a WEP-et támogató hardverek adott hosszúságú (128 bites) értékkel inicializálják az RC4 algoritmust, s így a megnövelt IV-t, a rejtjelező kulccsal együtt, valamilyen módon „bele kell gyömöszölni” ebbe az adott hosszúságba. A gyenge kulcsok problémáját a TKIP úgy oldja meg, hogy minden üzenet rejtjelezését más kulccsal végzi. Így a támadó nem tud a sikeres támadáshoz szükséges számú, azonos (potenciálisan gyenge) kulccsal kódolt üzenetet megfigyelni. Az üzenetkulcsokat a TKIP a négy utas kézfogás során generált adat-rejtjelező kulcsból állítja elő. A TKIP IV mechanizmusát és az üzenet-kulcsok előállítását a 4. ábra szemlélteti.




**4. ábra:** Az RC4 kulcs előállítása a TKIP-ben

Az AES-CCMP tervezőinek bizonyos értelemben könnyebb dolguk volt, mint a TKIP tervezőinek, hiszen nem volt megkötés arra vonatkozóan, hogy a protokollnak milyen hardveren kell futnia. A tervezők ezért egyszerűen megszabadultak az RC4 algoritmustól,

s helyette az AES blokkrejtjelezőre építették fel a protokollt. Definiáltak egy új AES használati módot, mely a régóta ismert CTR (Counter) mód és a CBC-MAC (Cipher Block Chaining – Message Authentication Code) kombinációja. Ebből származik a CCMP rövidítés. CCMP módban, az üzenet küldője először kiszámolja az üzenet CBC-MAC értékét, ezt az üzenethez csatolja, majd az üzenetet CTR módban rejtjelezi. A CBC-MAC számítás kiterjed az üzenet fejlécére is, a rejtjelezés azonban csak az üzenet hasznos tartalmára és a CBC-MAC értékre vonatkozik. A CCMP mód tehát egyszerre biztosítja a teljes üzenet (beleértve a fejlécet is) integritásának védelmét és az üzenet tartalmának titkosságát. A visszajátszás ellen az üzenetek sorszámozásával védekezik a protokoll. A sorszám a CBC-MAC számításhoz szükséges inicializáló blokkban van elhelyezve.

## 4. Összegzés

Cikkünkben ismeretterjesztő jellegű áttekintést adtunk a WiFi biztonsághoz kapcsolódó szabványokról, a WEP-ről és a 802.11i-ről. Bemutattuk a WEP működését és hibáit. Ismertettük a 802.11i-ben definiált hitelesítési és hozzáférés-védelmi mechanizmust, kulcshierarchiát, valamint a TKIP és az AES-CCMP protokollokat. Láttuk, hogy a TKIP protokoll célja, hogy lehetővé tegye a 802.11i-ben definiált új biztonsági architektúra alkalmazását a régi eszközökön, melyek hardvere csak a WEP-et támogatja. Ez a követelmény nagy mértékben megkötötte a tervezők kezét, ezért működő, de nem túl szép megoldás született. Ezzel szemben az AES-CCMP protokoll biztonságos és elegáns, viszont új hardvert igényel.

Manapság a boltban vásárolt új WiFi eszközök már támogatják 802.11i-ben definiált biztonsági architektúrát. Új eszközök vásárlása esetén tehát érdemes az RSN-t használni, amit a gyártók WPA2-nek is neveznek. Ha csak régebbi eszközök állnak rendelkezésre, akkor a WPA használatát tanácsoljuk.

## 5. Köszönetnyílvánítás

Jelen anyag elkészítését a Mobil Innovációs Központ (www.mik.bme.hu) és a 2 027 04 számú NKFP projekt (Adaptív médiafolyam szolgáltatási architektúra a legújabb mobil távközlési rendszerek céljaira) támogatta.

## 6. Bibliográfia

[Arbaugh+02] W. Arbaugh, N. Shankar, J. Wan, K. Zhang. Your 802.11 network has no clothes. *IEEE Wireless Communications Magazine,* 9(6):44-51, 2002.

[Borisov+01] N. Borisov, I. Goldberg, D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *Proceedings of the 7th ACM Conference on Mobile Computing and Networking*, 2001.

[EAP] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748. 2004.

[Edney+04] J. Edney, W. Arbaugh. *Real 802.11 Security: WiFi Protected Access and 802.11i.* Addison-Wesley, 2004.

[Fluhrer+01] S. Fluhrer, I. Mantin, A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Proceedings of the 8th Workshop on Selected Areas in Cryptography. 2001.

[RADIUS] B. Aboba, P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support for Extensible Authentication Protocol (EAP), RFC 3579, 2003.

[Walker00] J. Walker. Unsafe at any key size: An analysis of the WEP encapsulation. *IEEE 802.11-00/362*, 2000.

[WPA] Wi-Fi Alliance. Wi-Fi Protected Access. http://www.wi-fi.org/white_papers/whitepaper-042903-wpa/ (elérhetőség ellenőrizve 2006. április 19-én)

[802.1X] IEEE Std 802.1X-2001. IEEE Standard: Port-based Network Access Control, 2001.

[802.11] IEEE Std 802.11. IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[802.11i] IEEE Std 802.11i. IEEE Standard Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.