Langattoman IEEE 802.11 -lähiverkon tietoturva

Sami Hallenberg

Helsinki 15.5.2012

Pro gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen tiedekunta	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Sami Hallenberg

Työn nimi — Arbetets titel — Title

Langattoman IEEE 802.11 -lähiverkon tietoturva

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu -tutkielma	15.5.2012	117 sivua

Tiivistelmä — Referat — Abstract

Tässä pro gradu -tutkielmassa käsitellään langattoman IEEE 802.11 -lähiverkon eli WLANin tietoturvaa. Siihen kuuluu laitteiden todentaminen, datan salaus, lähetysten muuttumattomuus eli eheys ja verkon saatavuus. WLANin kolme eri tietoturvaprotokollaa, WEP, WPA ja WPA2, täyttävät tietoturvan eri osa-alueille asetettuja tavoitteita hyvin vaihtelevasti. WEP on vanhin ja kaikilta osin puutteellinen. Epävirallinen ja väliaikaiseksi tarkoitettu WPA takaa jo luotettavan todennuksen, mutta on muilta osin puutteellinen. WPA2 on standardoitu ja tarjoaakin WLANissa luotettavan todennuksen, salauksen ja eheyden.

WPA2 ei kuitenkaan takaa verkon saatavuutta. Hyökkääjä voi palvelunestohyökkäyksillä estää verkon käyttöön oikeutettuja tahoja käyttämästä verkkoa. Vaarallisimmat palvelunestohyökkäykset WLANissa perustuvat joko radiohäirintään, yksittäisten lähetysten heikkoon todentamiseen tai suurella lähetysmäärällä tulvittamiseen. Palvelunestohyökkäyksillä voidaan helposti lamaannuttaa verkon toiminta, eikä mikään yksittäinen keino torju kaikkia hyökkäyksiä. Tässä tutkielmassa käydään läpi erilaisia hyökkäyksiä ja niiden torjuntakeinoja. Esitän myös uuden ajatuksen sekvenssinumeroihin perustuvasta tulvituksen torjunnasta. Lähetyksissä säännönmukaisesti eteneviä sekvenssinumeroita on aiemmin käytetty vain väärennetyn lähettäjästä kertovan MAC-osoitteen havaitsemiseen, mutta niitä voidaan käyttää myös tulvitushyökkäyksen torjumiseen.

ACM Computing Classification System (CCS):
C.2.1 [Network Architecture and Design]: Wireless communication,
C.2.0 [General]: Security and protection (e.g., firewalls),
K.6.5 [Security and Protection (D.4.6, K.4.2)]

Avainsanat — Nyckelord — Keywords
IEEE 802.11, palvelunestohyökkäys, DoS, WEP, WPA, WPA2

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

Internetin levitessä 90-luvulla yhä laajemmalle kasvoi myös tarve langattomille lähiverkkotekniikoille. Markkinoilla sai nopeasti valta-aseman IEEE:n (Institute of Electrical and Electronics Engineers) 1997 julkaisema 802.11-standardiin ja siihen myöhemmin tulleisiin lisäosiin perustuva lähiverkkotekniikka, joka tunnetaan yleisesti nimellä WLAN. Se ei vaadi kaapeleiden vetämistä ja tarjoaakin riippumattomuutta paikasta ja katkeamatonta verkkoyhteyttä kunhan pysytään kantaman sisällä.

Langattoman verkon avoimuus tekee siitä myös alttiin erilaisille hyökkäyksille ja hyvän tietoturvan saavuttaminen onkin WLANissa haastavaa. Ilmatiessä kulkevat lähetykset ovat kantaman sisällä kaikkien kuunneltavissa, toisin kuin kaapeliverkossa. Verkon laitteita ei myöskään voi paikallistaa yhtä tarkasti kuin kaapeliverkossa, jossa verkon rajoittuminen fyysisiin kaapeleihin antaa jo enemmän tietoa laitteiden sijainnista. Langattomassa verkossa tiedetään vain, että laitteet ovat jossain kantaman sisällä. Verkon avoimuus lisää tarvetta sekä luotettavalle laitteiden todennukselle että lähetysten salaukselle ja eheydelle. Eheyttä uhkaavat sekä häiriöaltis siirtotie että ulkopuoliset hyökkääjät.

WLANin tietoturva on kehittynyt useassa vaiheessa. Ensimmäisissä verkoissa oli vain puutteellinen WEP-protokolla, joka epäonnistui kaikilla tietoturvan osa-alueilla. Ennen virallisen tietoturvastandardin valmistumista otettiin ylimenokaudeksi käyttöön WPA-protokolla, joka korjasi WEPin puutteita. Sekin osoittautui kuitenkin tietoturvaltaan riittämättömäksi ja tarjosi vain luotettavan todennuksen. Vuonna 2004 julkaistu virallinen WPA2-tietoturvaprotokolla tarjosi lopulta luotettavan todennuksen ja salauksen ja varmisti lähetysten eheyden.

Edelleen WLAnissa ovat kuitenkin ongelmana saatavuutta uhkaavat palvelunestohyökkäykset, joihin edes WPA2 ei tuonut mitään ratkaisua. Palvelunestohyökkäykset estävät muita verkon käyttäjiä käyttämästä verkkoa, vaikka heillä olisi siihen oikeus. Palvelunestohyökkäyksiä esiintyy usein internetissäkin, mutta ne voivat olla vakava uhka myös lähiverkossa lamaannuttaen sen kokonaan. Radiohäirinnässä hyödynnetään langattoman verkon rajoittuneisuutta; vain yksi laite kerrallaan voi lähettää siirtotiellä muiden kuunnellessa. Useampi samanaikainen lähettäjä sotkee kaikki lähetykset lukukelvottomiksi. Yksittäiset lähetykset todennetaan WLANissa hyvin heikosti. Niiden lähettäjäksi on helppo väärentää mikä tahansa laite, ja väärennetyillä lähetyksillä hyökkääjä voi häiritä verkon toimintaa. Hyökkääjä voi

esimerkiksi tiputtaa asemia pois verkosta tai saada ne virheellisesti luulemaan, että siirtotie onkin varattu ja odottamaan sen vapautumista. WLANia uhkaa myös tulvitus, jossa verkko hukutetaan valtavaan lähetysten määrään. Verkkoon aiheutetaan keinotekoinen ruuhka, jolloin sen toiminta hidastuu tai lamaantuu kokonaan.

Tässä tutkielmassa esitellään WLANin toiminta, sen kolme tietoturvaprotokollaa, sekä kartoitetaan WLANissa esiintyviä palvelunestohyökkäyksiä. Mitä erilaisia palvelunestohyökkäyksiä tunnetaan? Miten hyökkäykset havaitaan ja miten niitä voidaan torjua?

Tutkielma rakentuu seuraavasti. Luvussa 2 esitellään 802.11-standardiin pohjautuva WLAN, sen rakenne ja toiminta sekä protokolla- että fyysisellä tasolla. Luvussa 3 käsitellään WLANin havaitsemista sekä WEPin tarjoamaa tietoturvaa. WEPin tietoturvan taso oli hyvin heikkoa ja sitä paremmin toimivia WPA- ja WPA2-tietoturvaprotokollia käsitellään luvussa 4. Internetissä esiintyviä palvelunestohyökkäyksiä kuvataan luvussa 5. WLANissa esiintyviin palvelunestohyökkäyksiin keskitytään luvussa 6. Niistä kartoitetaan tärkeimmät hyökkäystavat, niiden havaitseminen ja torjunta. Lopuksi luvussa 7 on yhteenveto tutkielman keskeisimmistä asioista.

2 IEEE 802.11 -verkko

WLAN (Wireless Local Area Network) on määritelty vuonna 1997 ilmestyneessä IEEE 802.11 -standardissa ja siihen myöhemmin tulleissa lukuisissa lisäosissa. Tässä luvussa esitellään WLAN-verkko, sen rakenne, toiminta ja ominaisuudet viimeisimmän aiemmat standardit ja lisäosat yhteen kokoavan vuonna 2007 ilmestyneen version [IEE07] pohjalta. Verkon tietoturvaa käsitellään kuitenkin vasta myöhemmin luvuissa 3, 4 ja 6. Hyvän kuvauksen WLANista saa myös Gastin aihetta perusteellisesti käsittelevästä kirjasta [GaS05].

2.1 Verkon elementit ja topologiat

WLAN koostuu kolmesta eri elementistä, joista ensimmäinen on *asema* (station, STA). Asema on mikä tahansa laite, joka noudattaa standardin mukaista fyysisen- ja siirtoyhteyskerroksen määrittelyä. Asema voi olla esimerkiksi kannettava tietokone tai älypuhelin. Ryhmä tällaisia toistensa kanssa kommunikoimaan kykeneviä asemia muodostaa *peruspalveluryhmän* (Basic Service Set, BSS).

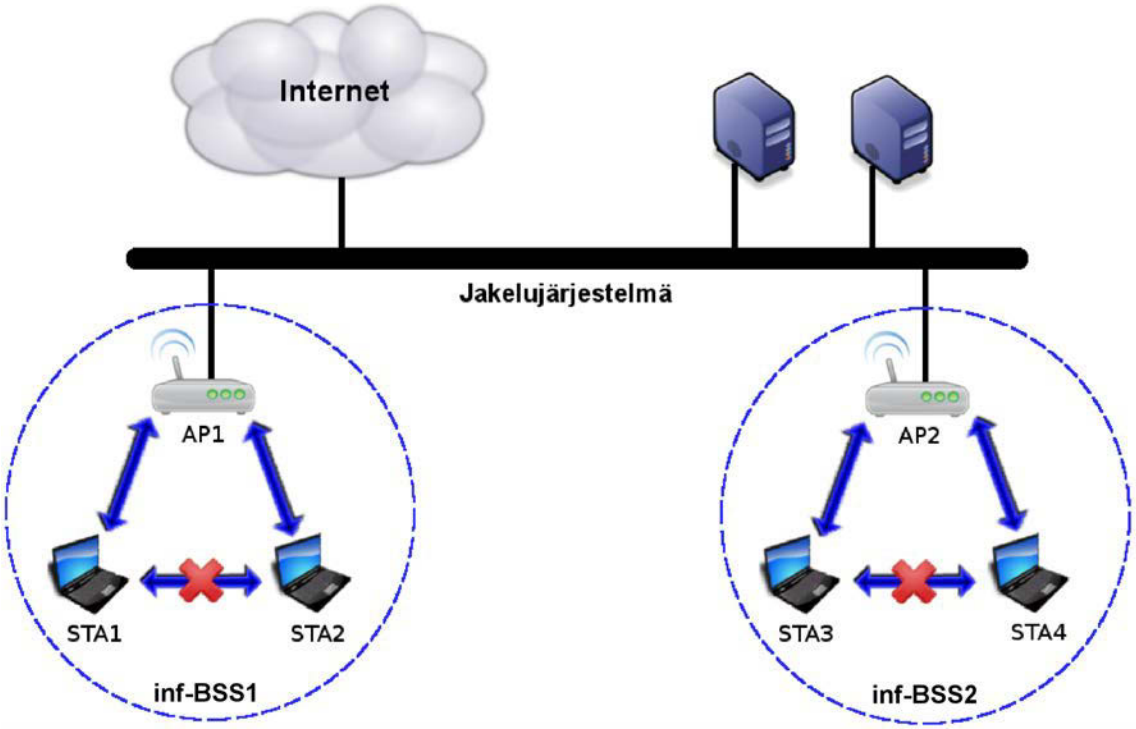
Toinen verkossa toimiva elementti on *tukiasema* (Acces Point, AP), joka on oi-

keastaan vain erikoistunut asema. Se toimii täysin samoin kuin asema, mutta tarjoaa lisäksi siihen liittyneille muille asemille välityspalvelua. Kaikki asemien väliset keskinäiset lähetykset kulkevat tukiaseman kautta eivätkä asemat kommunikoi suoraan keskenään, vaikka olisivatkin toistensa kantaman sisällä. Lisäksi kaikki asemien ja muun verkon väliset lähetykset kulkevat tukiaseman kautta. Tällaista yhdestä tukiasemasta ja yhdestä tai useammasta asemasta koostuvaa ryhmää kutsutaan *infrastruktuuri-BSS:ksi* (inf-BSS). Tässä tutkielmassa sekä asemasta että tukiasemasta käytetään yhteisnimitystä *laite*.

Jakelujärjestelmä (Distribution System, DS) on kolmas verkon elementti. Se yhdistää useita infrastruktuuri-BSS:iä toisiinsa sekä tarjoaa tukiasemille yhteyden internetiin. Tukiasemaan yhdistetty jakelujärjestelmä tarkoittaa usein kaapeloitua Ethernet-lähiverkkotekniikkaa.

WLANissa on kaksi vaihtoehtoa verkon topologiaksi: *itsenäinen BSS* (Independent BSS, IBSS) ja *laajennettu palveluryhmä* (Extended Service Set, ESS). IBSS koostuu pelkästään asemista ja kaikkien asemien on oltava toistensa kantaman sisällä, koska kaikki lähetykset menevät IBSS-verkossa suoraan lähettäjältä vastaanottajalle. Lähetykset eivät siis kulje erillisen tukiaseman kautta eivätkä asemat edes välitä lähetyksiä edelleen yhdeltä asemalta toiselle. IBSS-topologiaa sanotaan myös ad hoc -verkoksi, sillä se muodostetaan tyypillisesti vastaamaan yllättäen ja lyhyeksi aikaa syntyneeseen kommunikointitarpeeseen.

ESS-topologiassa on yksi tai useampi infrastruktuuri-BSS, jotka on yhdistetty jakelujärjestelmällä. Kuvassa 1 on esitetty ESS-verkon topologia. Kuvassa näkyy kaksi erillistä infrastruktuuri-BSS:ää, inf-BSS1 ja inf-BSS2, joissa kummassakin on yksi tukiasema ja kaksi asemaa. Tukiasemat AP1 ja AP2 yhdistävät nämä inf-BSS:t jakelujärjestelmän kautta samaksi loogiseksi verkoksi. Tällainen tilanne voisi olla esimerkiksi rakennus, jossa kauempana toisistaan olevat tukiasemat ovat Ethernet-verkon kautta yhteydessä toisiinsa. Jakelujärjestelmän kautta asemat voivat ottaa yhteyden internetiin sekä kommunikoida jakelujärjestelmään liitttyjen palvelimien kanssa. Toisin kuin IBSS:ssä asemat eivät ESS:ssä kommunikoi suoraan keskenään, vaikka olisivatkin kantaman sisällä, vaan kaikki lähetykset kulkevat aina tukiaseman kautta. Tässä tutkielmassa käsitellään jatkossa vain ESS-verkkoja ja jätetään IBSS-verkot tarkastelun ulkopuolelle.



Kuva 1: ESS-topologian mukainen verkko

2.2 Palvelut

IEEE 802.11 -standardissa ja siihen myöhemmin tulleissa lisäosissa on määritelty 13 erilaista palvelua, joita jakelujärjestelmän tai laitteiden on tunnettava ja tarjottava toisilleen. Taulukossa 1 on esitetty ESS:ssä tarvittavat palvelut. Palveluista on esitetty palvelun nimi, sen tarjoava elementti (L = laite tai DS = jakelujärjestelmä) sekä palvelun tarkoitus. Jos jakelujärjestelmä tarjoaa palvelun, se on usein toteutettu jakelujärjestelmän reunalla sijaitsevassa tukiasemassa. Palveluista tärkein on dataa sisältävän MSDUN:n (MAC Service Data Unit) välitys. Tätä tukee suurin osa muista palveluista ja kolme palvelua liittyy tietoturvaan.

Levittämisen tarjoaa tukiasema ja jakelujärjestelmä ja se on tärkein MSDU:n välitystä tukevista palveluista. Asema liittyy aina johonkin yhteen tukiasemaan ja tämän tiedon avulla jakelujärjestelmä osaa välittää samassa ESS:ssä, mutta eri inf-BSS:ssä olevien asemien väliset lähetykset oikeaan paikkaan. Esimerkiksi, kun kuvassa 1 oleva inf-BSS1:een kuuluva asema STA1 lähettää lähetyksen samaan ESS:ään, mutta

Palvelu	Tarjoaja	Tarkoitus
MSDU:n välitys	L	MSDU
Levittäminen	DS	MSDU
Liittyminen	DS	MSDU
Uudelleenliittyminen	DS	MSDU
Poistuminen	DS	MSDU
Yhdentyminen	DS	MSDU
Todentaminen	L / DS	tietoturva
Todennuksen purku	L / DS	tietoturva
Datan luottamuksellisuus	L	tietoturva

Taulukko 1: Elementtien tuntemat palvelut

eri inf-BSS:ään kuuluvalle asemalle STA4, se lähettää sen ensin oman omalle tukiasemalleen AP1:lle. AP1 ohjaa lähetyksen edelleen DS:lle, joka hoitaa sen välittämisen oikealle tukiasemalle, AP2:lle. Saatuaan DS:ltä lähetyksen AP2 osaa toimittaa sen lopulliselle vastaanottajalle STA4:lle. IEEE 802.11 -standardi ei ota kantaa siihen miten jakelujärjestelmä on toteutettu, mutta sen tulee tarjota lähetysten välittäminen oikealle kohdetukiasemalle lähettävän tukiaseman antamien tietojen perusteella. Levittämisen onnistumista tukee kolme muuta palvelua: liittyminen, uudelleenliittyminen ja poistuminen.

Jotta jakelujärjestelmä osaa välittää lähetykset oikeaan paikkaan ESS:n sisällä, sen levittämispalvelu tarvitsee tiedon siitä mille tukiasemalle lähetys on ohjattava. Tätä varten asemien on *liityttävä* johonkin tukiasemaan. Tämän asemien ja tukiasemien välisen liitoksen perusteella jakelujärjestelmä osaa välittää kehykset oikein.

Uudelleenliittyminen on tarpeellista, kun asema vaihtaa saman ESS:n sisällä toiseen tukiasemaan ja samalla myös toiseen inf-BSS:ään. Tällöin asemaan liittyvä tukiasema vaihtuu ja järjestelmä osaa ohjata inf-BSS:ää vaihtaneelle asemalle menevät lähetykset oikealle tukiasemalle. Jos asema siirtyy kokonaan toiseen ESS:ään, niin yhteydessä voi ilmetä katkoksia.

Kun asema *poistuu* kokonaan verkosta, esimerkiksi liikkumalla kantaman ulkopuolelle, akun loppuessa tai lopettaessa toimintansa, olisi sen ilmoitettava tästä tukiasemalle. Järjestelmälle ei aiheudu kuitenkaan ongelmia, vaikka ilmoitus jäisikin tekemättä.

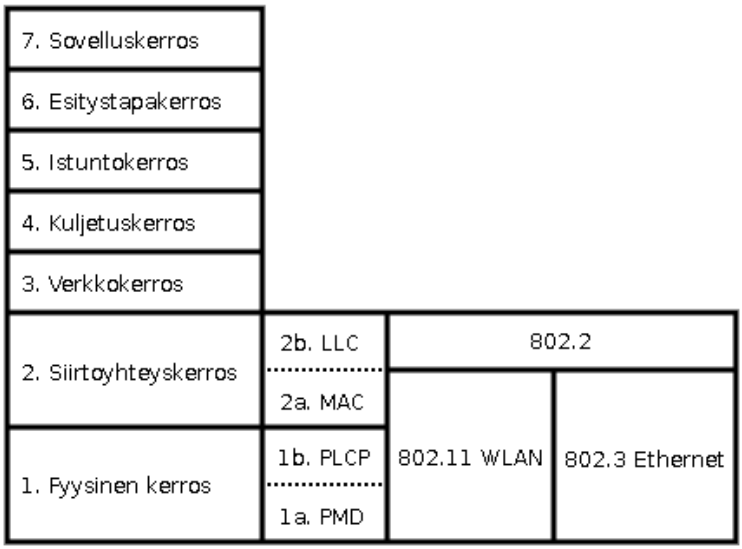
Jakelujärjestelmä tarjoaa tarvittaessa palveluna myös *yhdentymisen* muihin verkkoihin. Jos jakelujärjestelmän tukiasemalta saama lähetys onkin kuljetettava jonne-

kin muualle kuin toiselle samaan ESS:ään kuuluvalle tukiasemalle, niin yhdentymispalvelu huolehtii muualle lankaverkkoon ohjauksesta. Lähetykseen on ehkä tehtävä muutoksia osoitteeseen tai formaattiin ja jakelujärjestelmän yhdentymispalvelu huolehtii näistä.

Loput kolme palvelua, todentaminen, todennuksen purku ja datan luottamuksellisuus liittyvät tietoturvaan. Tietoturvan takaaminen on langattomassa lähiverkossa haastavampaa kuin kaapeliverkossa johtuen siirtotienä käytettävästä yhteisestä ilmatiestä. Kaapeliverkko tarjoaa lähtökohtaisesti paremman todennuksen, koska siinä kaikki lähetykset kulkevat rajattuja fyysisiä kaapeleita pitkin. On paljon helpompaa yhdistää lähetys sen lähettäjään kaapeliverkossa kuin langattomassa verkossa. Langattomassa verkossa lähetys voi tulla miltä tahansa kantaman sisällä olevalta laitteelta eikä lähettäjää voida paikallistaa yhtä selvästi kuin kaapeliverkossa. Kaapeliverkossa lähetys voidaan osoittaa halutuille vastaanottajille, mutta langattomassa verkossa radioaallot leviävät joka suuntaan ja lähetyksiä voidaan kantaman sallimissa rajoissa kuunnella millä tahansa sopivalla vastaanottimella, mikä on uhka lähetysten luottamuksellisuudelle. Näitä kolmea tietoturvaan liittyvää palvelua käsitellään luvuissa 3 ja 4.

2.3 IEEE 802.11 -verkko OSI-mallissa

OSI-malli on käsitteellinen hahmotustapa pakettivälitteisen tietoliikenteen kuvaamiseen. Siinä tiedonsiirtoprotokollat on jaettu kerroksiin, missä kukin kerros tarjoaa palveluja ylemmälle kerrokselle. Tiedonsiirrossa OSI-mallin kerros saa suoraan ylemmältä kerrokselta viestin. Kerros lisää viestiin omia kerroskohtaisia lisätietoja ja lähettää laajennetun viestinsä edelleen suoraan alemmalle kerrokselle. Kuvassa 2 on esitetty yleisen seitsemänkerroksisen OSI-mallin rakenne. Ylimpänä on sovelluskerros, jolla toimivat käyttäjälle näkyvät sovellusohjelmat. Tällä kerroksella muun muassa käynnistetään koko tiedonsiirtoprosessi. Sen alla ovat esitystapa- ja istuntokerros, jotka on usein sisällytetty sovelluskerrokseen eikä niitä edes esitetä viisikerroksisessa OSI-mallissa. Esitystapakerros tarjoaa sovelluskerrokselle riippumattomuutta sovellusten käyttämistä erilaisista datan esitystavoista ja merkistöistä. Istuntokerros hallinnoi yhdessä yhteydessä kulkevia istuntoja. Se käynnistää, hallinnoi ja lopettaa yhteyksiä paikallisen ja yhteyden toisessa päässä toimivan sovelluksen välillä. Luotettavan datan kuljetuksen ylemmille kerroksille tarjoaa kuljetuskerros. Se hallinnoi datan perillemenoa, kuittauksia ja mahdollisesti tarvittavaa datan uudelleenlähetystä. Verkkokerros huolehtii ylemmiltä kerroksilta saamiensa lähetysten



Kuva 2: WLANin sijoittuminen OSI-mallissa

välittämisestä verkossa olevien eri laitteiden välillä. Se osaa reitittää lähetyksen perille eri verkkojenkin välillä. Toiseksi alin kerros, siirtoyhteyskerros, huolehtii ja hallinnoi samassa verkossa olevien laitteiden välisestä liikenteestä. Se myös havaitsee ja mahdollisesti korjaa fyysisen kerroksen havaitsemia virheitä tiedonsiirrossa. Siirtoyhteyskerros jakaantuu vielä kahteen alikerrokseen, LLC-kerrokseen (Logical Link Control) ja MAC-kerrokseen (Media Access Control). IEEE 802.2 -standardissa on määritelty LLC-alikerroksen toiminta. Se tukee tiedonsiirrossa tapahtuneista virheistä toipumista pitämällä kirjaa, mitkä lähetykset onnistuivat, sekä lähettämällä uudelleen ne, joiden lähettämisessä oli ongelmia.

IEEE 802.11 -lähiverkkotekniikka määrittelee LLC:n alapuolella sijaitsevien MAC-alikerroksen sekä fyysisen kerroksen toiminnan. Myös kaapeloidussa lähiverkossa yleisesti käytössä oleva Ethernet-tekniikka eli IEEE 802.3 määrittelee samat kerrokset. WLANissa MAC-kerroksen tehtävä on kontrolloida ja hallinnoida langattoman verkon toimintaa ja kehysten lähettämistä yhteisesti jaetussa siirtotiessä eli ilmatiessä kulkevien radioaaltojen lähetystä. Fyysinen kerros jakaantuu vielä kahteen alikerrokseen, PLCP- (Physical Layer Convergence Procedure) sekä PMD-kerrokseen (Physical Medium Dependent). PLCP-kerros toimii välikerroksena, joka valmistelee MAC-kerrokselta saamansa kehykset lähetettäväksi radioaaltoina ilmatielle. Koska langattomassa lähiverkossa on useita erilaisia radioteknisiä tapoja koodata ja

välittää signaalia, vaaditaan jokaista PMD-kerroksen tuntemaa tapaa varten sopiva kehyksen valmistelu PLCP-kerroksella. PMD-kerros lähettää kaikki ylemmältä PLCP-kerrokselta tulevat kehykset antennien kautta ilmatielle sekä ohjaa muualta vastaanottamansa kehykset ylemmälle PLCP-kerrokselle.

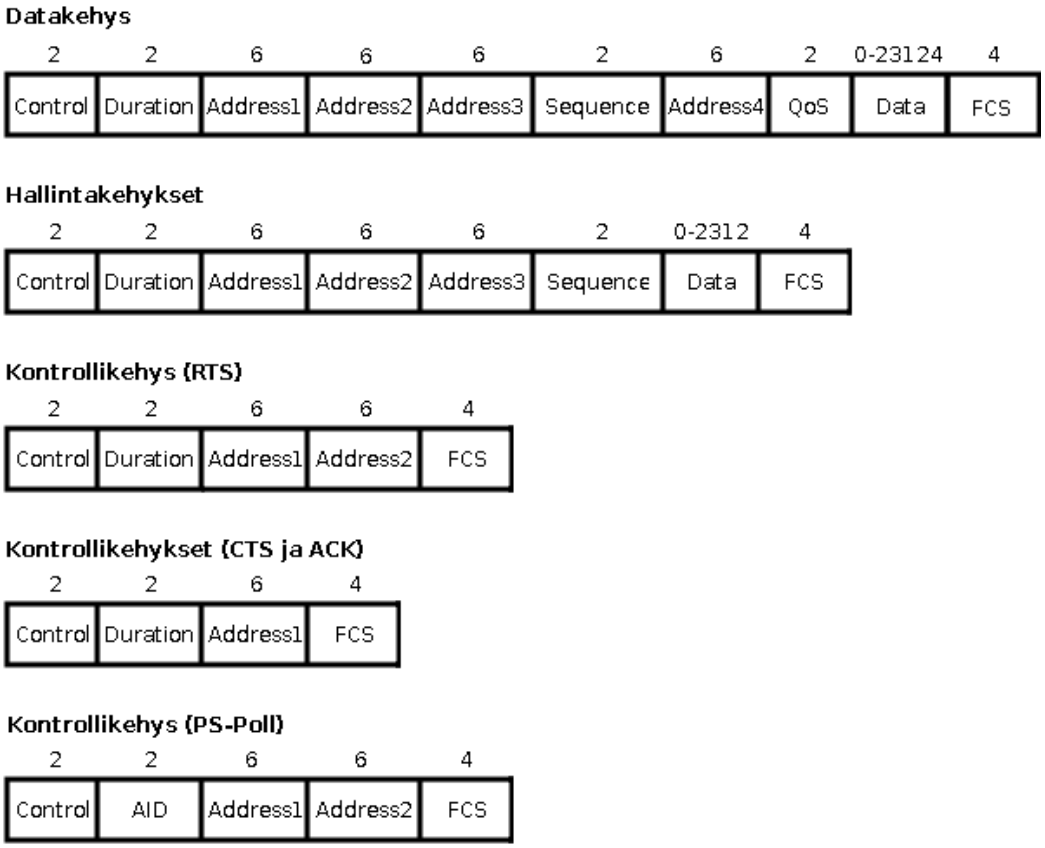
2.4 MAC-kerros

MAC-kerroksella on erittäin tärkeitä tehtäviä ja se onkin WLANin toiminnan ohjauksen kannalta kaikkein keskeisin osa. Se vastaa ylemmältä kerrokselta saamansa datan salaamisesta, toiselta asemalta saapuneiden pakettien salauksen purkamisesta, eheydentarkistuksesta sekä datan eteen päin lähettämisestä. Lisäksi MAC-kerroksella hallinnoidaan asemien todentamista ja liittämistä tukiaseman verkkoon sekä huolehditaan yhteisesti jaetun siirtotien koordinoidusta ja sujuvasta käytöstä, niin etteivät eri osapuolet häiritse toisiaan samanaikaisilla lähetyksillä. MAC-kerros toteuttaa aiemmin luvussa 2.2 mainitut palvelut.

2.4.1 Data-, hallinta- ja kontrollikehykset

MAC-kerroksella on kolmen tyyppisiä kehyksiä: data-, hallinta- ja kontrollikehyksiä. Kuvassa 3 on havainnollistettu, millaisista kentistä eri tyyppiset kehykset koostuvat. Jokainen kehys koostuu useammasta kentästä ja kentän yläpuolella on kyseisen kentän pituus tavuina. Data- ja hallintakehykset sisältävät aina samat kentät, mutta kontrollikehyksissä mukana olevat kentät vaihtelevat kehysten mukaan. Datakehyksen tehtävä on kuljettaa ylemmältä kerrokselta saatua dataa, joka sijoitetaan datakehyksen datakenttään. Hallintakehyksiä on useita erilaisia, mutta niiden kaikkien kenttärakenne on samanlainen. Hallintakehyksillä tukiasema myös mainostaa verkkoaan ja kertoo tarpeellisia tietoja, joita siihen liittyvä asema tarvitsee. Kontrollikehyksillä, joita on kehysrakenteeltaan kolmea eri tyyppiä, pidetään huolta yhteisen siirtotien hallitusta käytöstä ja sujuvasta tietoliikenteestä.

Jokaisen MAC-kehyksen ensimmäisenä on control-kenttä ja viimeisenä FCS-kenttä. control-kentässä välitetään koko kehyksen ohjaukseen ja käsittelyyn liittyviä tietoja, kuten esimerkiksi kehyksessä käytettävän protokollan versio, kehyksen tyyppi, tieto isomman datamäärän pilkkomisesta ja lähettämisestä useammassa datakehyksessä, käytössä oleva virranhallintatapa ja tietoa kehyksessä mahdollisesti käytetystä salauksesta. Lopussa olevaan FCS-kenttään (Frame Check Sequence) sijoitetaan kehyksen kaikista aiemmista kentistä laskettu tarkistussumma, 32-bittinen CRC-



Kuva 3: MAC-kerroksen kehysten kentät

tiiviste (Cyclic Redundancy Check). Sillä havaitaan tiedonsiirrossa tapahtuneita virheitä ja voidaan jopa korjata pienimpiä virheitä. PS-Poll-kehystä lukuuno ttamatta kaikissa MAC-kehyksissä tulee control-kentän jälkeen duration-kenttä, jonka arvo riippuu kehyksestä ja joka ilmaisee useammasta kehyksestä koostuvan viestienvaihdon tarvitseman ajan. Esimerkiksi datakehyksen jälkeen tulee aina vastaanottajalta kuittaus ja datakehyksessä oleva duration-kenttä kertoo, kuinka kauan datakehyksen jälkeen kuluu vielä aikaa mikrosekunneissa ennen kuin kuittaus on saapunut perille. Duration on myös oleellinen koordinoitaessa siirtotien käyttöä verkon laitteiden välillä. PS-Poll-kehyksessä on duration-kentän sijasta AID-kenttä (association identifier), eräänlainen liittymistunniste. Se on arvoltaan 1-2007 ja asema saa sen tukiasemalta liittyessään sen verkkoon. Ollessaan virransäästötilassa asema voi pyytää tukiasemalta oikean AID:n sisältävällä PS-Poll-kehyksellä datakehyksiä, jotka tukiasema on tallettanut aseman unitilan aikana. MAC-kehyksissä on usei-

ta address-kenttiä, joiden määrä vaihtelee kehystyypin ja tilanteen mukaan yhdestä neljään. Näissä kentissä on kehyksen lähettäjän ja vastaanottajan MAC-osoite. Joissain tapauksissa on tarpeellista käyttää vieläkin useampaa osoitetta ja mainita myös osoite, josta kehys juuri tuli ja mihin se on välittömästi seuraavaksi menossa. Data- ja hallintakehyksissä on sequence-kenttä, josta löytyy sirpalenumero, jota käytetään pilkottaessa isoa datamäärä useampaan datakehykseen sekä sekvenssinumero. Sekvenssinumero on laskuri, joka liikkuu välillä 0-4095 kasvaen yhdellä jokaista lähetettyä data- tai hallintakehystä kohden. Laskurin päästyä 4095:een se kääntyy nollaan. Jos iso datamäärä pilkotaan useaan kehykseen, niin jokaisessa kehyksessä on kuitenkin sama sekvenssinumero. Myös lähetettäessä sekvenssin sisältävää samaa jo aiemmin lähetettyä kehystä uudestaan, siinä on sama sekvenssinumero kuin aiemmassa lähetyksessä. Vain datakehyksessä voi olla QoS-kenttä (Quality of Service). Siinä välitetään erilaisia kehysten tärkeysjärjestykseen ja niiden luokitteluun liittyviä tietoja. Sekä data- että hallintakehyksessä on datakenttä. Datakehyksessä tähän upotetaan ylemmältä kerrokselta saatu eteenpäin lähetettävä data. Hallintakehyksissä datakenttään ei sijoiteta mitään ylemmiltä kerroksilta tulevaa dataa, vaan sen sisältö riippuu kulloinkin käytetystä hallintakehyksestä ja on erilaista vastaanottajan MAC-kerrokselle tarkoitettua informaatiota.

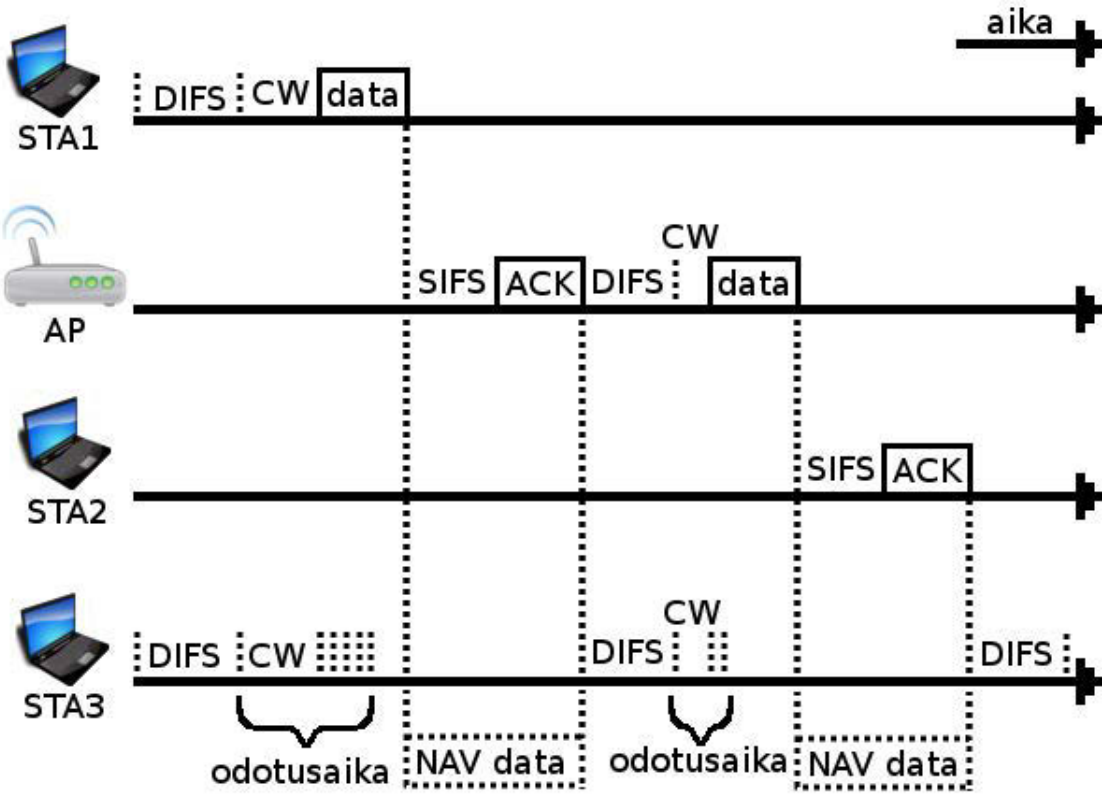
2.4.2 Yhteisesti jaetun siirtotien käyttäminen

IEEE 802.11 -standardi määrittelee MAC-kerrokselle kolme erilaista tapaa tunnustella onko yhteinen siirtotie vapaa muiden signaaleista, ennen kuin sitä voidaan käyttää. Ainoa kaikissa laitteissa pakollinen tapa on hajautettu koordinointifunktio (Distributed Coordination Function, DCF). Kaksi muuta, pistekoordinointifunktio (Point Coordination Function, PCF) ja hybridikoordinointifunktio (Hybrid Coordination Function, HCF), ovat valinnaisia ja hyödyntävät DCF:ää toteutuksessaan. HCF:ää voidaan käyttää vain, kun asemat hyödyntävät QoS:iä liikennöidessään. Valinnainen PCF perustuu kiertokyselyyn, missä tukiasema lähettää vuorollaan kullekin siihen liittyneelle asemalle pollausviestin. Jos tukiasemalla on kyseiselle asemalle dataa lähetettäväksi, se liittää sen pollausviestin mukaan. Saatuaan pollausviestin asema lähettää tukiasemalle vastauksen ja sen mukana dataa. PCF:ssä kukin asema saa siis säännöllisin väliajoin tukiasemalta oman toimintavuoron, jolloin voi lähettää ja vastaanottaa dataa.

Yleisin ja eniten käytetty tapa siirtotien seuraamiseen ja varaamiseen on DCF. WLANissa se perustuu CSMA/CA:han (Carrier Sense Multiple Access with Col-

lision Avoidance), joka pyrkii välttämään yhteentörmäyksiä. Ethernetissä käytetty CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ei mitenkään estä päällekäin tapahtuvia ja toisia sotkevia törmäyksiä. Se havaitsee ne kuitenkin helposti ja uudelleenlähetys on pienen tauon jälkeen nopeaa. WLAN on kuitenkin huomattavasti Ethernetiä hitaampi eikä radiolähetin lähettäessään signaalia pysty havaitsemaan toista samaan aikaan siirtotietä käyttävää signaalia. Siksi WLANissa pyritään ennemminkin estämään siirtotiellä tapahtuvia yhteentörmäyksiä sen sijaa, että niistä toivuttaisiin nopeasti.

Siirtotien seuraamiseen ja sopivan lähetyshetken määrittämiseen käytetään sekä fyysistä siirtotien signaalitason havainnointia että virtuaalista siirtotien varauksesta kertovaa NAV-arvoa (Network Allocation Vector). Halutessaan lähettää data- tai hallintakehyksen eli kehyksen. jossa on datakenttä, laitteet odottavat kunnes siirtotie on vapaa ja varaavat sen sitten käyttöönsä kehyksen ja sen vastaanottajan siihen lähettämän kuittauksen ajaksi. Kuvassa 4 on havainnollistettu CSMA/CA:n toimintaa WLANissa. Verkossa on tukiasema (AP) ja kolme asemaa (STA1, STA2 ja STA3). Alussa sekä STA1 että STA3 haluaisivat siirtotien käyttöönsä. Kumpikin havaitsee siirtotien olevan vapaa, mutta ennen kuin siirtotietä voi käyttää, sen on oltava vapaa DIFSin (distributed interframe space) mittainen aika. Jos DIFSin jälkeen kumpikin asema lähettäisi siirtotielle heti oman signaalinsa, niin ne vain sotkisivat toisensa. Siksi DIFSin jälkeen onkin *kilpailuikkuna* (Contention Window, CW), jossa asemat kilpailevat pääsystä siirtotielle. Kumpikin asema arpoo luvun väliltä $[0, CW_{max} - 1]$, missä $CW_{max} = 2^i$ ja $i \in [3, 8]$. Kummankin aseman *odotusajaksi* muodostuu arvotun luvun osoittama määrä aikayksikköjä, joiden pituus riippuu verkon fyysisellä kerroksella käytetystä signaalinkäsittelytavasta. Aluksi i on kolme ja aina, kun lähetettäessä tulee yhteentörmäys, kasvatetaan i:tä yhdellä, kunnes se kasvaa kahdeksaan ja pysyy siinä niin kauan, kun yhteentörmäyksiä edelleen tulee. Yhteentörmäyksen jälkeen odotusaika arvotaan siis isommalta arvoalueelta, jolloin yhteentörmäysten todennäköisyys pienenee. Kun lähetys onnistuu eli lähettäjä saa vastaanottajalta kuittauksen perille tulleesta kehyksestä, asetetaan i aina heti takaisin pienimpään arvoonsa. Kuvassa STA1 arpoo itselleen lyhyemmän odotusajan kuin STA3. Niin kauan, kun asemat havaitsevat siirtotien olevan vapaa, niiden odotusaika pienenee kohti nollaa, jonka saavutettuaan asema saa lähettää. STA1:n odotusaika loppuu ensin ja se saa lähettää datakehyksensä siirtotielle. Kilpailuikkuna päättyy, kun siirtotie ei ole enää vapaa. Kun STA3 havaitsee siirtotien olevan varattu, sen odotusajan väheneminen keskeytyy, koska odotusaika pienenee vain kilpailuikkunassa. Tukiaseman saatua STA1:n lähettämän datakehyksen se odottaa SIFSin (short



Kuva 4: Siirtotien varaaminen CSMA/CA:lla

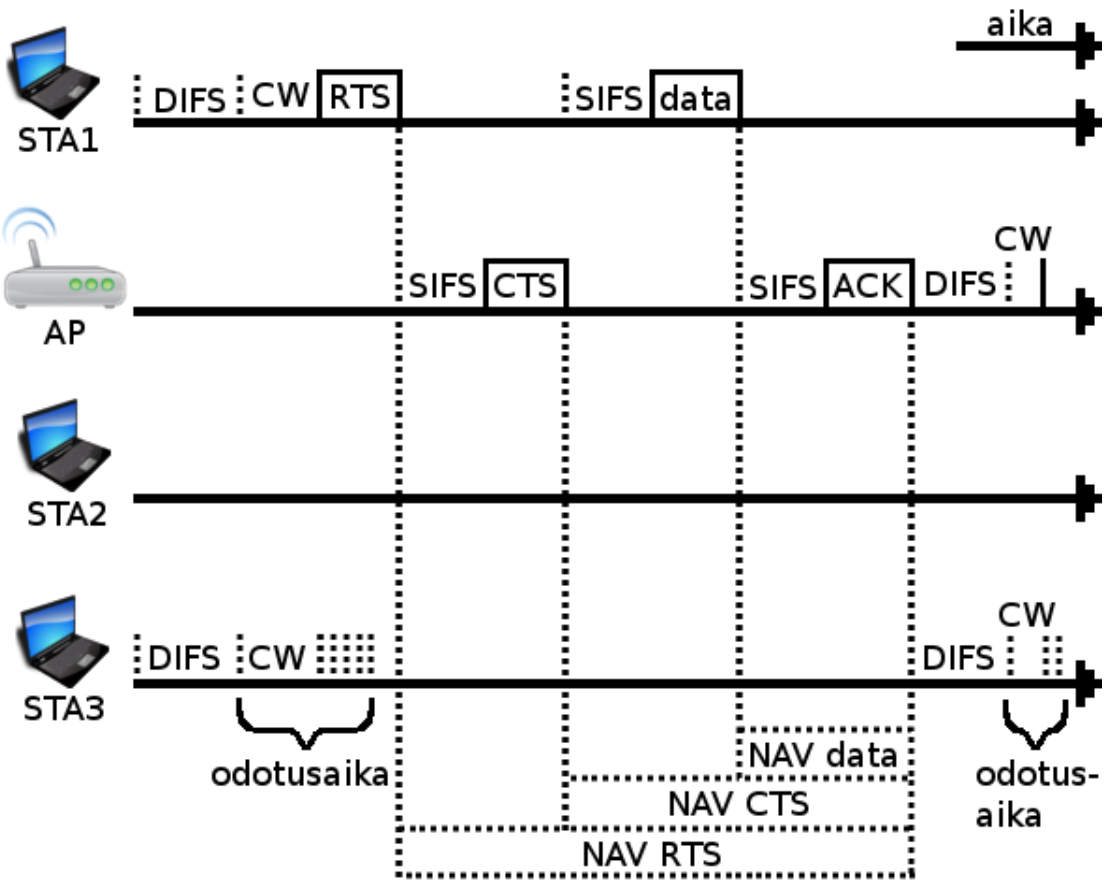
interframe space) mittaisen ajan ennen kuin lähettää STA1:lle kuittauksen (acknowledgment, ACK) virheettömästi perille tulleesta datakehyksestä. Kukaan muu asema ei pääse siirtotielle SIFSin aikana, sillä se on lyhyempi kuin DIFS ja aseman on aina ennen siirtotielle pääsyä havaittava siirtotien olevan vapaa vähintään DIFSin mittaisen ajan, sekä odotettava sen jälkeen vielä odotusaika. Vaikka siirtotie on SIFSin aikana vapaa, niin myöskään STA3:n odotusaika ei vähene kohti nollaa, sillä se pienenee vain kilpailuikkunassa. STA1:n saatua tukiaseman lähettämän kuittauksen on siirtotie taas vapaa. Nyt tukiasema haluaa myös päästä siirtotielle lähettämään STA1:ltä saamansa datakehyksen sen lopulliselle vastaanottajalle, STA2:lle. Jälleen kaikkien siirtotielle haluavien on odotettava ensin DIFS, jonka jälkeen siirrytään kilpailuikkunaan. Kilpailuikkunassa STA3:n jäljellä oleva odotusaika alkaa vähentyä. Kilpailuikkunassa tukiasema arpoo itselleen odotusajan ja saa pienemmän arvon, kuin STA3:n jäljellä oleva odotusaika. Tukiaseman odotusajan loputtua se saa siirtotien käyttöönsä ennen STA3:a, joka jää edelleen odottamaan omaa vuoroaan.

Kun tukiasema on lähettänyt datakehyksen STA2:lle ja saanut siihen kuittauksen, on siirtotie taas vapaa ja STA3 voi yrittää päästä siirtotielle DIFSin jälkeen.

Siirtotien tilan fyysisen seurannan lisäksi laitteet päivittävät NAVin arvoa lähes kaikissa MAC-kehyksissä olevan duration-kentän perusteella. Datakehyksessä se kertoo mikrosekunneissa kuittaukseen ja yhteen SIFSiin kuluvan ajan. Kaikki kantaman sisällä olevat laitteet asettavat NAViinsa kuulemastaan datakehyksestä löytyvän arvon, paitsi jos se on sama tai pienempi kuin NAVin nykyinen arvo. Niin kauan kuin NAVilla on jokin positiivinen arvo, laite ei saa lähettää siirtotielle mitään, koska joku muu on varannut sen käyttöönsä. Jos kumpi tahansa, siirtotien fyysinen seuranta tai NAV, kertoo siirtotien olevan käytössä, laite ei saa lähettää siirtotielle mitään. Kuvassa 4 sekä STA1:n että tukiaseman lähettämän datakehyksen kuultuaan STA3 päivitti NAViaan ja pidättäytyi sen osoittamaksi ajaksi siirtotielle pyrkimisestä.

Siirtotiellä voi sattua yhteentörmäys myös silloin, kun kaksi asemaa ovat tukiaseman, mutta eivät toistensa kantaman sisällä. Tällöin molemmat luulevat siirtotien olevan vapaa ja lähettävät tukiasemalle kehyksen. Tukiasema saa kehykset yhtä aikaa ja ne sotkevat toisensa. Tästä ongelmasta päästään, jos laitteen on aina ensin pyydettävä vastaanottajalta lupa kehyksen lähettämiseen. Lähetyslupaa pyydetään lyhyellä RTS-kontrollikehyksellä (Request To Send) ja vastaanottaja antaa luvan lyhyellä CTS-kontrollikehyksellä (Clear To Send). Kuvassa 5 on havainnollistettu kuvan 4 alkuosassa tapahtuvaa kommunikointia, kun käytetäänkin RTS/CTS:ää. Aluksi on normaalisti DIFS ja sen jälkeinen kilpailuikkuna odotusaikojen arpomisineen. Vuoron saanut STA1 lähettää nyt kuitenkin ensin lähetyslupaa pyytävän RTS:n tukiasemalle. Kaikki sen kuulevat muut laitteet asettavat NAViinsa RTS:ssä ilmoitetun kestoajan eivätkä sen aikana yritä itse siirtotielle. RTS:ssä ilmoitettu kesto kattaa CTS:n, datan, kuittauksen sekä kolme SIFSiä. Tukiasema antaa STA1:lle luvan lähettää datakehys lähettämällä sille CTS:n. CTS:ssä on myös kestoaika, joka on RTS:n kestoaika vähennettynä SIFSin ja CTS:n lähettämiseen kuluneella ajalla. Myös datakehyksessä on kestoaika aivan samoin kuin on CSMA/CA:ssakin. kuittauksen jälkeen NAVien osoittama aika on kulunut ja siirtotielle haluavat asemat odottavat DIFSin ja siirtyvät sen jälkeen taas kilpailuikkunaan. STA3:n vielä jäljellä ollut odotusaika alkaa vähentyä ja tukiasema arpoo itselleen odotusajan. Koska tukiaseman odotusaika on lyhyempi kuin STA3:n jäljellä oleva odotusaika, saa tukiasema ensimmäisenä siirtotien käyttöönsä ja STA3 jää edelleen odottamaan siirtotielle pääsyä.

RTS/CTS vähentää yhteentörmäyksiä samassa inf-BSS:ssä, mutta toistensa kanta-



Kuva 5: CSMA/CA:n laajennus RTS/CTS:llä

man ulkopuolella, olevien laitteiden välillä. Toki vielä RTS/CTS:ääkin käytettäessä voi tulla yhteentörmäyksiä, jos odotusajat loppuvat samaan aikaan. Jos tukiaseman eri puolilla olevat asemat lähettävät toisistaan tietämättä yhtä aikaa tukiasemalle pienen RTS:n, sen aiheuttamasta törmäyksestä toipuminen on kuitenkin paljon nopeampaa, kuin jos asemat olisivat lähettäneet ison datakehyksen. RTS/CTS lisää kuitenkin jonkin verran verkon liikennettä, joten sitä ei yleensä käytetä jatkuvasti. Siitä on hyötyä vain, jos verkossa on selvästi toistensa kantaman ulkopuolella olevia asemia tai datakehykset ovat hyvin isoja.

2.5 Fyysinen kerros

Oleellisin langattoman lähiverkon kaapeliverkosta erottava tekijä on viestien siirtotienä käytetty väylä. Kaapeliverkossa tieto kulkee johdoissa koodattuna sähköjän-

nitteen muutoksiin, mutta langattomassa verkossa viestit kulkevat ilmatiessä koodattuna radioaaltoihin. Erilainen ympäristö asettaa signaalien kuljettamiselle aivan uudenlaisia haasteita, joita lankaverkoissa toimivien järjestelmien ja protokollien toiminnassa ei tarvitse edes ottaa huomioon. Langattomassa verkossa signaalit leviävät kaapeliverkkoon verrattuna paljon hallitsemattomammin kaikkialle ympäristöön eivätkä etene vain ennalta asetettuja kaapeleiden osoittamia reittejä pitkin. Signaalit myös vaimenevat matkalla. Vaimenemiseen vaikuttaa etäisyyden lisäksi radioaallon taajuus ja sen kohtaamat fyysiset rakenteet ja jopa sääolosuhteet ulkona. Kaapeliverkoissakin on jossain määrin huomioitava signaalin vaimeneminen pitkillä matkoilla, mutta sen merkitys on langattomassa verkossa huomattavasti suuremmassa osassa. Lisäksi ilmatiessä liikkuvat muut radioaallot saattavat häiritä signaalia niin pahasti ettei vastaanottaja saa lähetyksestä selvää.

Radioaallot ovat sähkömagneettista säteilyä, elektromagneettista aaltoliikettä, joka kuljettaa energiaa. Langattomassa tiedonsiirrossa vastaanottaja havaitsee tämän lähettäjän liikkeelle sysäämän energian. Kaikella aaltoliikkeellä, kuten radioaalloillakin, on *taajuus*, joka kertoo sekunnissa tapahtuneiden värähdysten määrän. Radioaaltoihin katsotaan kuuluvaksi noin 3 Hz - 300 GHz taajuusalueella värähtelevä sähkömagneettinen säteily. 802.11-standardiin pohjautuvat lähiverkot toimivat kuitenkin vain tämän taajuusalueen pienellä osalla, muutaman gigahertsin taajuusalueella. Radioaaltojen vastaanottamiseen ja lähettämiseen tarvitaan sopiva antenni, joka on herkistynyt toimimaan tietyllä taajuusalueella. Radiotaajuuden voimakkuutta, sen sisältämän energian määrää, ilmaistaan värähtelyn *amplitudilla*. Koska ympäristössä on jatkuvasti taustakohinaa, on lähettäjän kasvatettava radiotaajuuden amplitudia tarpeeksi, jotta vastaanottaja huomaisi signaalin ja erottaisi sen taustakohinasta. Radioaalloilla on värähtelevänä liikkeenä kunakin ajanhetkenä *vaihe*, jonka poikkeavuutta vertailuarvosta voidaan mitata. Radioaallon vaihe kiertää 360 astetta aaltoliikkeen edetessä ajan myötä. Radiotaajuuksiin voidaan koodata tietoa erilaisilla joko amplitudin, taajuuden tai vaiheen muutoksiin pohjautuvilla menetelmillä, jolloin niiden avulla voidaan kuljettaa digitaalista informaatiota.

Informaatiota sisältävän radiotaajuussignaalin edetessä siinä tapahtuu signaalin häipymistä [Rap99]. Langattomissa lähiverkoissa tämä ilmenee radioaallon amplitudin tai vaiheen heilahteluina ajan funktiona. Tärkeimmät häipymistä WLANissa aiheuttavat syyt ovat vaimeneminen ja heijastuminen. Signaalin *vaimenemiseen* eli amplitudin pienentymiseen vaikuttaa etäisyyden lisäksi sen taajuus sekä lähettäjän ja vastaanottajan välissä olevat esteet. Radioaalto kulkee esteettömässä tilassa sitä pitemmälle mitä matalampi on sen taajuus. Toisaalta korkeampi taajuus voi kul-

jettaa samassa aikayksikössä enemmän informaatiota. Esteet vaimentavat radiotaajuussignaalia, sillä niillä on erilainen läpäisykyky. Signaalia vaimentaa tehokkaasti esimerkiksi sisätiloissa metalliovi ja ulkona rankkasade.

Erityisesti sisätiloissa radiotaajuussignaali saattaa *heijastua* erilaisista pinnoista. Vastaanottaja saa saman signaalin useita reittejä pitkin: suoraan, sekä rakenteiden kautta heijastuneena. Tällöin puhutaan *monitie-etenemisestä*. Heijastuneet radiotaajuussignaalit ovat vaimentuneet sekä eri vaiheessa kuin suoraan perille tullut. Jos eri reittejä tulleet radiotaajuussignaalit ovat ajallisesti liian lähellä toisiaan ei vastaanottaja pysty erottamaan niitä, varsinkaan suurilla tiedonsiirtonopeuksilla, ja ne häiritsevät toisiaan. Vastaanottajapäässä tapahtuu *interferenssiä*, missä samanaikaisesti tulleet radiotaajuussignaalit summautuvat ja riippuen niiden vaiheista ja amplitudeista heikentävät, vahvistavat tai vääristävät alkuperäistä signaalia. Haitallista interferenssiä ilmenee myös muiden samaan aikaan samalla radiotaajuudella lähettävien laitteiden signaalien sotkiessa lähetystä. Tämän johdosta radiotaajuuksien käyttö onkin luvanvaraista, jota eri maissa säätelevät viranomaiset, Suomessa Viestintävirasto. Poikkeuksena ovat kolme niin sanottua ISM-taajuusaluetta (Industrial, Scientific and Medical), joilla toimivien lähettimien käyttäminen on sallittua ilman viranomaisten lupaa. Vapaasta käytöstä huolimatta lähettimien tehoille on asetettu ylärajoja. WLAN voi toimia joko 2,4 GHz:n tai 5 GHz:n vapaalla ISM-taajuusalueella ja Suomessa tällaisen lähettimen teho saa olla korkeintaan 100 mW matalemmalla ja 200 mW korkeammalla taajuudella [Vie11].

Kaapeliverkossa, joka tarjoaa suhteellisen häiriöttömän siirtotien informaatiolle, riittää pelkkä bittien koodaaminen jännitteen muutoksina. Langattomassa verkossa käytettävä ilmatie on kuitenkin paljon alttiimpi häiriöille ja siksi siellä bittejä esittävä signaali saadaankin muokkaamalla kantoaaltoa. Tätä muokkausta sanotaan moduloinniksi. Tärkeimmät 802.11-standardin tuntemat modulointitekniikat perustuvat joko kantoaallon amplitudin, vaiheen tai taajuuden muokkaamiseen [Rap99]. Perinteisesti radioaallot lähetetään kapealla taajuuskaistalla, mutta tällöin samalla taajuudella toimivat muut lähettimet häiritsisivät pahasti toisiaan. Vapaalla ISM-alueella, missä toimii lukuisia eri laitteita, onkin siksi mielekästä käyttää moduloinnissa *hajaspektritekniikkaa* (Spread Spectrum, SS), missä siirrettävä signaali hajautetaan useammalle vierekkäiselle kapealle taajuudelle ja vastaanottaja muuttaa ne taas käänteisesti yhdeksi kapeaksi taajuudeksi [Rap99]. Samalla taajuusalueella toimiva kapealla taajuudella lähettävä lähetin ei häiritse hajaspektritekniikkaa käyttävää lähetystä isommasti, sillä ne menevät päällekkäin vain hyvin kapean taajuuden osalta. Hajaspektrin osalta tuon pienen taajuuden informaatio menetetään,

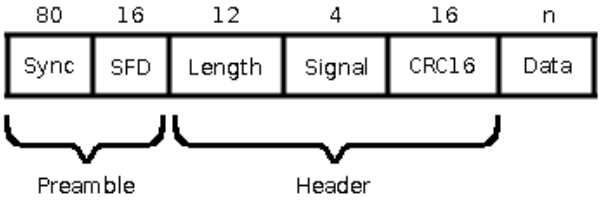
mutta pienuutensa vuoksi se ei ole iso ongelma. Kapealle taajuudelle suurella teholla lähetettyyn signaaliin aiheutuu hajaspektritekniikasta vain pientä häipymistä. Hajaspektritekniikka antaa lisäsuojaa myös heijastumista ja monitie-etenemistä vastaan. Toisaalta tiedonsiirtonopeus pienenee hajaspektritekniikkaa käytettäessä, sillä hajauttamatonta signaalia voitaisiin siirtää kapealla kaistalla enemmän.

2.5.1 Taajuushyppelyhajaspektritekniikka

Taajuushyppelyhajaspektritekniikassa (Frequecy Hopping Spread Spectrum, FHSS) 2,4 GHz:n taajuus on jaettu 79:ään 1 MHz:n levyiseen kanavaan. Lähetys tapahtuu lyhinä purskeina ja jokaisen purskeen jälkeen hypätään toiselle kanavalle. Kanavaa vaihdetaan 400 ms:n välein. Hyppelyjärjestys kanavalta toiselle on ennalta määrätty ja erilaisia mahdollisia hyppelykuvioita on useita. Lähettäjän ja vastaanottajan on tiedettävä mitä hyppelykuviota kulloinkin käytetään. Taajuushyppely vaatii onnistuakseen viestinnän osapuolilta tahdistusta ja koordinointia. Taajuushyppelyn vaatima tekniikka on kuitenkin suhteellisen halpaa eikä vaadi suurta virrankulutusta. Se on silti vanhentunutta tekniikkaa, eikä sitä käytetä enää nykyään, koska sillä päästään korkeintaan vain 2 Mbit/s datanopeuteen.

Digitaalisen signaalin bitit muutetaan analogiseksi signaaliksi hyvin pienillä taajuuksien muutoksilla kulloinkin käytössä olevassa kanavassa. Esimerkiksi alempi taajuus tarkoitta nollaa ja korkeampi taajuus ykköstä. Tällöin päästään 1 Mbit/s datanopeuteen. Nopeutta voidaan kasvattaa, jos otetaan enemmän eri taajuusarvoja käyttöön. Neljällä eri taajuudella voidaan ilmaista neljä eri symbolia, kaksi bittiä. Koska taajuudet ovat hyvin lähellä toisiaan, niiden määrää ei voida kasvattaa isommaksi, sillä vastaan tulee joko kanavan leveys tai vierekkäisiä symboleja kuvaavat taajuudet joutuvat liian lähekkäin, eikä niitä voida enää erottaa toisistaan. Tästä johtuen taajuushyppelyllä ei päästä 2 Mbit/s suurempaan datanopeuteen.

Ennen kuin MAC-kerrokselta tuleva kehys voidaan lähettää ilmatielle, lisää PLCP-kerros siihen tarvittavaa ohjaustietoa. Kuvassa 6 on esitetty PLCP-kehyksen rakentuminen kentistä, käytettäessä taajuushyppelyä. Kenttien koot on ilmoitettu bitteinä. Muut kentät ovat aina vakiomittaisia, mutta datakentän koko voi vaihdella suuresti, koska sen sisältö saadaan MAC-kerrokselta ja sitä käsitellään vielä ennen sen sijoittamista datakenttään. PLCP-kehys koostuu kolmesta osasta. Ensin on kahdesta kentästä koostuva preamble, sitten kolmen kentän mittainen header ja viimeisenä datakenttä. Sync- ja SFD (Start Frame Delimeter) -kenttiä käytetään tahdistamaan lähettäjän ja vastaanottajan radiolähettimet sekä osoittamaan lähetyksen alkukoh-



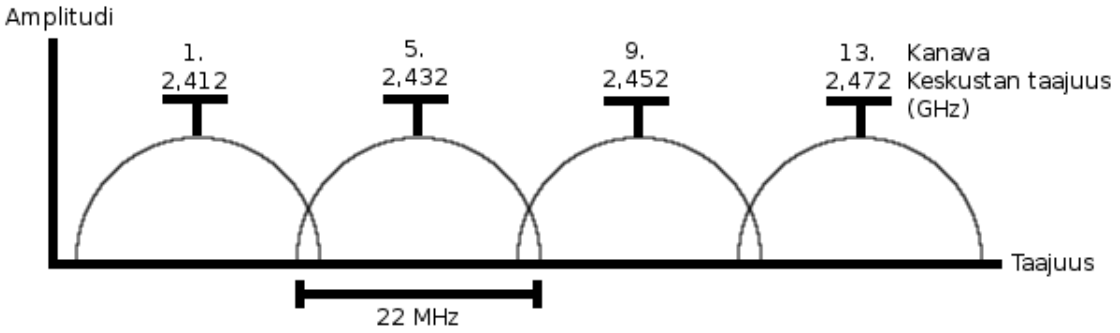
Kuva 6: PLCP-kehyksen kentät FHSS:ssä

taa. Length-kenttä kertoo kehyksen lopussa olevan datakentän pituuden ja signal-kentässä kerrotaan sen lähetysnopeus. CRC16 on length- ja signal-kentistä laskettu tarkistussumma.

2.5.2 Suorasekvenssihajaspektritekniikka

Suorasekvenssihajaspektritekniikassa (Direct Sequence Spread Spectrum, DSSS) 2,4 GHz:n taajuus on jaettu Suomessa ja suurimmassa osassa Eurooppaa 13:een kanavaan, joista kukin on leveydeltään 22 MHz. Käytettävien kanavien määrä vaihtelee maittain, esim. Japanissa on käytössä 14 ja USA:ssa 11 kanavaa. Suorasekvenssissä kapealla taajuudella lähetettävä signaali levitetäänkin matemaattisesti leveämmälle taajuusalueelle. Kapean radiotaajuuden energia levitetään niin, että keskellä kanavaa amplitudi on korkein ja pienenee reunoja kohti. Kuvassa 7 on havainnollistettu yksinkertaistaen neljän kanavan sijoittumista 2,4000 - 2,4835 GHz:n vapaalle ISM-taajuusalueelle. Vierekkäisten kanavien keskikohdat ovat vain 5 MHz:n päässä toisistaan ja näin lähellä olevat lähetykset häiritsevät toisiaan pahasti. Jos kantaman sisällä on useita tukiasemia, niiden on käytettävä omassa verkossaan tarpeeksi muista erottuvia kanavia. Muuten on vaarana yhteisen ilmatien ruuhkautuminen tai jopa lähetysten huomattava toistensa häiritseminen. Käytännössä 13:sta kanavasta voidaan käyttää vain kolmea kanavaa, esim. 1, 6 ja 11, ilman että ne menevät ollenkaan toistensa päälle. Neljää kanavaakin voidaan käyttää, kuten kuvassa 7 on esitetty eli kanavia 1, 5, 9 ja 13, jolloin ne menevät jo hiukan päällekkäin, mutta eivät vielä isommasti häiritse toisiaan.

Suorasekvenssissä käytetään Barkerin sarjaa lähetettävien bittien informaation levittämiseksi laajemmalle alueelle. Jokaisen lähetettävän bitin sijasta lähetetäänkin 11 bitin mittainen sarja (10110111000) tai sen käänteisarvo (01001000111). Jos lähetettävä bitti on nolla, lähetetään sarjan osoittama bittijono ja jos lähetettävä bitti on



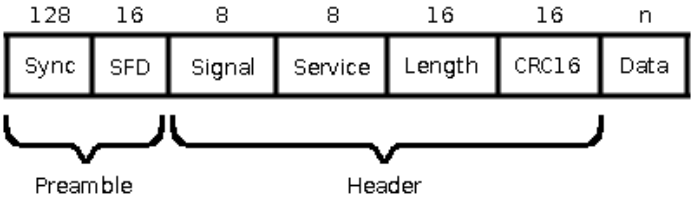
Kuva 7: Kanavien päällekkäisyys DSSS:ssä

yksi, lähetetään sarjan käänteisarvo. Lähetettävä bittimäärä kasvaa huomattavasti, mutta Barkerin sarjalla saadaan lisää vikasietoisuutta, sillä toisteisuuden vuoksi alkuperäinen bitti voidaan vielä tunnistaa, vaikka lähes puolet lähetetyistä biteistä muuttuisikin matkalla.

Toisin kuin taajuushyppelyssä, jossa käytetään taajuusmodulaatiota eli signaali koodataan kantoaaltoon taajuuden pienin muutoksin, suorasekvenssissä käytetään vaihemodulaatiota. Se perustuu kantoaallon vaiheen muutoksiin kapealla taajuusalueella. Differentiaalinen kaksivaihemodulaatio (Differential Binary Phase Shift Keying, DBPSK) on yksinkertaisin versio, missä nollabitti ei muuta signaalia mitenkään, mutta ykkösbitti kääntää signaalin vaihetta 180 astetta. Jos lisätään eroteltavissa olevien vaiheiden määrä neljään, saadaan differentiaalinen nelivaihemodulaatio (Differential Quadratic Phase Shift Keying, DQPSK). Tällöin voidaan koodata bittijono 00 normaalisti, bittijono 01 kääntämällä signaalin vaihetta 90 astetta, bittijono 10 kääntämällä signaalin vaihetta 180 astetta ja bittijono 11 kääntämällä signaalin vaihetta 270 astetta. Näin saadaan entistä nopeampaa tiedonsiirtoa, mutta toisaalta useamman vaiheen käyttö moduloinnissa lisää alttiutta monitie-etenemisestä johtuville häiriöille. Suorasekvenssin kehittyneemmässä versiossa, korkeanopeuksisessa suorasekvenssihajaspektritekniikassa (Hight Rate Direct Sequence Spread Spectrum, HR-DSSS) päästään vieläkin nopeampaan tiedonsiirtoon, kun bittien hajautukseen käytetään Barkerin sarjan sijasta komplementtikoodausta (Complementary Code Keying, CCK). CCK:ssa neljästä tai kahdeksasta bitistä muodostetaan matemaattisin muunnoksin kahdeksan bitin mittainen jono, joka moduloidaan signaaliin DQPSK:lla. Näin päästään jopa 11 Mbit/s datanopeuteen, joka on huima parannus

suorasekvenssin maksimissaan 2 Mbit/s datanopeuteen.

Kuvassa 8 on esitetty PLCP-kehyksen koostumus käytettäessä suorasekvenssiä. Ke-



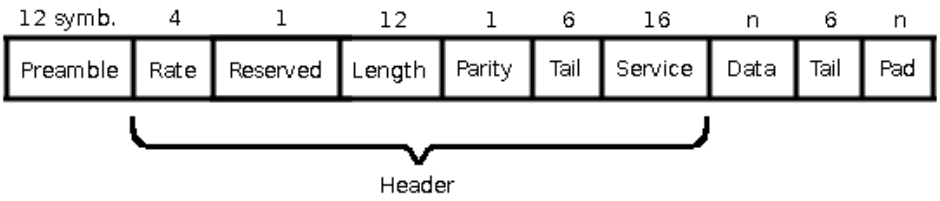
Kuva 8: PLCP-kehyksen kentät DSSS:ssä

hyksen rakenne on käytännössä lähes sama kuin FHSS:ä, kenttien paikat ja koot ovat vain hiukan muuttuneet. Service-kenttää, jota ei ollut FHSS:ssä, ei käytetä myöskään DSSS:ssä, mutta se on käytössä HR-DSSS:ä. Siinä on käytetyn hajautuksen ja nopeamman tiedonsiirron vaatimia lisätietoja. Toisin kuin FHSS:ä, datakenttää ei enää käsitellä mitenkään, vaan sen sisältönä on MAC-kerrokselta saatu MAC-kehys.

2.5.3 Monitaajuusmodulointi

Kolmas WLANissa käytetty hajaspektritekniikka on monitaajuusmodulointi (Orthogonal Frequency Division Multiplexing, OFDM). Siinä kaistanleveydeltään 20 MHz:n kanava jaetaan 52:een kapeampaan alikanavaan ja kullakin alikanavalla voidaan lähettää signaalia yhtä aikaa. Lähetykset voidaan tehdä kahdeksalla eri moodilla, joiden modulointitapa ja nopeus vaihtelevat. Kantoaallon modulointiin käytetään suorasekvenssitekniikasta tuttuja differentiaalista kaksi- ja nelivaihemodulaatiota tai aiemmista moduloinneista poikkeavia 16- ja 64-neliamplitudimodulaatioita (Quadrature Amplitude Modulation, QAM). OFDM:llä saavutetaan moduloinnista riippuen 6 - 54 Mbit/s datanopeus. OFDM:ää käytettiin aluksi ruuhkaisen 2,4 GHz:n taajuuden sijasta 5 GHz:n taajuusalueella, Suomessa tarkkaan ottaen taajuuksilla 5,150 - 5,350 GHz ja 5,470 - 5,725 GHz. 5 GHz:n ISM-taajuus on käytössä maailmanlaajuisesti, mutta käytettävät taajuudet vaihtelevat maittain. Alempaa 5 GHz:n taajuutta saa Suomessa käyttää vain sisätiloissa ja maksimissaan 200 mW:n lähetysteholla, korkeampaa taajuutta saa käyttää sekä sisällä että ulkona, mutta lähetysteho on rajattu 1 W:iin. Myöhemmin OFDM otettiin käyttöön myös 2,4 GHz:n taajuusalueella.

Kuvassa 9 on esitetty PLCP-kehyksen jakaantuminen kenttiin ja niiden pituudet bitteinä monitaajuusmoduloinnissa. Kehyksen rakenne eroaa jonkin verran FHSS:ä



Kuva 9: PLCP-kehyksen kentät OFDM:ssä

ja DSSS:ä käytetyistä PLCP-kehyksistä. Alussa on preamble-kenttä, joka koostuu 12 symbolista. Ensin lähetetään niin sanottu pieni symoboli kymmenen kertaa ja sen jälkeen isompi kaksi kertaa. Symbolien pituus riippuu käyteystä modulaatiosta. Preamblella synkronoidaan lähettäjän ja vastaanottajan ajastimet. Preamblen jälkeen tulevien kenttien pituudet on ilmoitettu bitteinä. Headeriin kuuluu kuusi kenttää. Rate-kenttä kertoo koko loppukehyksen lähetysnopeuden, reserved-kenttä on varattu myöhempää käyttöä varten ja length-kenttä kertoo datakenttään upotetun MAC-kehyksen pituuden. Parity-kentssä on yhden bitin mittainen niin sanottu parillinen pariteettibitti. Se on nolla, jos aiemmissa headerin kentissä on parillinen määrä ykkösbittejä, muuten ykkönen. Tail- ja service kentät ovat määrämuotoisia ja niiden tehtävä on osoittaa headerin loppuminen. Datakenttään sijoitetaan MAC-kehys ja sen jälkeen tulee kuudesta nollabitistä koostuva tail-kenttä ja lopuksi on täytteenä toimiva vaihtelevanmittainen pad-kenttä, jotta kehyksestä saadaan määrämittainen.

2.6 Standardin eri versiot ja lisäosat

Vuonna 1990 Institution for Elecrical and Electronic Engineers (IEEE) asetti työryhmän kehittämään standardia, joka määrittelisi Ethernetin ominaisuuksia vastaavan langattoman lähiverkon. Useiden välivaiheiden jälkeen vuonna 1997 julkaistiin IEEE 802.11 -standardi. Tämän jälkeen alkuperäiseen standardiin on ilmestynyt lukuisia lisäosia, jotka ovat vastanneet kehittyneen tekniikan tarjoamiin mahdollisuuksiin sekä kasvaneiden tietoturva- ja tiedonsiirtovaatimusten tarpeisiin. Seuraavassa on lyhyt katsaus alkuperäiseen standardiin, sen päivitykseen sekä näihin tulleisiin tärkeimpiin lisäosiin. Lisäosat tunnistaa vuosilukua edeltävästä pienestä kirjaimesta,

esimerkiksi 802.11a-1999.

2.6.1 802.11-1997: Alkuperäinen standardi

802.11-standardissa on määritelty käytettäväksi välitystapana joko infrapunaa tai radioaaltoja. Infrapunaa hyödyntäviä laitteita ei ole käytännössä juurikaan markkinoilla ja siksi se on jäänyt sivuosaan standardin myöhemmässä kehittämisessä. Standardi määrittelee laitteiden toiminnan radiotaajuudella 2,4000 - 2,4835 GHz, joka kuuluu vapaasti käytettävään ISM-taajuusalueeseen. Vapaasta käytöstä seuraa kuitenkin, että markkinoilla on runsaasti 2,4 GHz:n taajuudella toimivia laitteita. Ongelmaksi voivat muodostua kantaman sisällä samalla taajuudella toimivat laitteet, kuten esimerkiksi mikroaaltouunit tai Bluetooth-laitteet, jotka saattavat aiheuttaa häiriötä toistensa langattomaan signaaliin. Onneksi 2,4 GHz:n taajuusalueella toimivien laitteiden lähetystehoja on rajoitettu, jolloin niiden kantamakin on rajoitettua ulottuen optimiolosuhteissa sisätiloissa muutamiin kymmeniin metreihin ja ulkonakin vain kahteen- kolmeensataan metriin. Alkuperäisessä standardissa käytetään taajuushyppely- ja suorasekvenssihajaspektritekniikoita, joilla päästään 1 tai 2 Mbit/s datanopeuteen.

2.6.2 802.11a-1999: OFDM 5 GHz

2,4 GHz:n taajuusalueella toimivien laitteiden määrä lisääntyi ja helpottaakseen lähekkäin toimivien laitteiden toisilleen aiheuttamia häiriöitä julkaisi IEEE vuonna 1999 lisäosan alkuperäiseen standardiin, IEEE 802.11a:n. Se poikkesi aiemmasta 802.11-standardin määrittelemästä lähiverkosta olennaisilta osin eikä näin ollen ole sen kanssa yhteensopiva. 802.11a toimii korkeammalla 5,150-5,350 tai 5,470-5,725 GHz:n vapaalla ISM-taajuusalueella. Korkeammasta taajuudesta on sekä hyötyä että haittaa: kantama pienenee hiukan, mutta tiedonsiirtonopeus toisaalta kasvaa. Signaalin modulointitapakin vaihtui monitaajuusmodulointiin, jolla päästään huomattavasti aiempaa nopeampaan, jopa 54 Mbit/s datanopeuteen.

2.6.3 802.11b-1999: HR-DSSS 2,4 GHz

Samana vuonna 802.11a:n kanssa julkaistiin myös 802.11b-lisäosa. Se toimii 2,4 GHz:n taajuudella ja toisin kuin 802.11a se on täysin yhteensopiva alkuperäisen standardin kanssa. 802.11b käyttää edelleen suorasekvenssiä, mutta sen kehittyneempää muotoa niin sanottua korkeanopeuksista suorasekvenssiä, missä databitit voidaan

levittää signaaliin Barkerin sarjan sijasta komplementtikoodauksella. Näin saadaan nopeampaa tiedonsiirtoa ja päästään joko 5,5 tai 11 Mbit/s datanopeuteen riippuen käytetäänkö komplementtikoodauksessa 4 vai 8 databittiä. Jotta 802.11b olisi yhteensopiva aiemman standardin kanssa, se voi tarvittaessa käyttää myös Barkerin sarjaa ja liikennöidä 1:n tai 2:n Mbit/s datanopeudella. Taajuushyppelyä 802.11b ei kuitenkaan enää tunne. 802.11b:ssä on määritelty komplementtikoodauksen ja Barkerin sarjan lisäksi vielä pakettibinääripoimukoodaus (Packet Binary Convolutional Coding, PBCC), mutta sen toteutus on vapaaehtoista ja siksi puuttuukin useista laitteista kokonaan.

2.6.4 802.11g-2003: OFDM 2,4 GHz

Vaikka 802.11a tarjosikin nopeampaa tiedonsiirtoa, niin sen yhteensopimattomuus alkuperäisen standardin ja ennen kaikkea 802.11b:n kanssa jätti sen sivuosaan markkinoilla. Sen sijaan 802.11b:stä tuli suosituin ja sitä käyttävien laitteiden määrä kasvoi valtavasti. Ajan myötä kasvoi tarve nopeammalle, mutta 802.11b:n kanssa yhteensopiville laitteille. Ongelman ratkaisi vuonna 2003 IEEE:n julkaisema seuraava lisäosa, 802.11g, joka oli täysin yhteensopiva 802.11b:n kanssa. Se pohjautuu oikeastaan lähes täysin 802.11a:n käyttämään monitaajuusmodulointiin, mutta toisin kuin 802.11a toimii 2,4 GHz:n taajuudella. Tällöin päästään 54 Mbit/s datanopeuteen. Yhteensopivuuden takaamiseksi 802.11g toteuttaa myös kaikki 802.11b-lisäosassa mainitut pakolliset modulointitekniikat eli suorasekvenssin sekä korkeanopeuksisen suorasekvenssin. Maksimaaliseen 54 Mbit/s datanopeuteen päästään vain, kun verkossa on pelkästään 802.11g-laitteita. Jos verkossa on myös 802.11b-laitteita, hidastavat ne verkkoa, eikä 802.11g-laitteidenkaan välillä päästä enää huippunopeuksiin. Vaikka 802.11g toimiikin nopeammin, on sen ilmoitettava hitaammille 802.11b-laitteille niiden ymmärtämällä hitaammalla nopeudella, että siirtotie on varattuna.

2.6.5 802.11-2007: Standardin päivitys

Vuonna 2007 on ilmestynyt viimeisin alkuperäisen 802.11-1997-standardin päivitys, IEEE 802.11-2007 [IEE07]. Siinä on standardin ensimmäinen versio päivitettynä kaikilla sen jälkeen julkaistuilla lisäosilla eli IEEE 802.11a,b,d,e,g,h,i ja j. 802.11i paransi tietoturvaa ja sitä käsitellään lähemmin luvussa 4. Lisäosat d,e,h ja j ovat muita edellä mainittuja lisäosia pienempiä ja liittyvät muun muassa palvelun laatuun ja taajuuksien käyttöön.

2.6.6 802.11n-2009: MIMO-OFDM 2,4 ja 5 GHz

Viimeisin merkittävä lisäosa on vuonna 2009 ilmestynyt 802.11n [IEE09]. Se ei tuonut mitään uutta signaalin modulointitapaa eikä ottanut käyttöön uutta taajuusaluetta, kuten merkittävät lisäosat aiemmin, mutta paransi muilla keinoin huomattavasti verkon suoritustehoa. MAC-kerroksella tarkasteltaessa datan lähettämiseen ja siihen saatuun kuittaukseen menevään aikaan vaikuttaa moni tekijä. Tällaisia ovat esimerkiksi verkon ruuhkaisuus, siirtotien varaaminen, radioaaltojen häipyminen, erilaiset liikennöinnin ja ohjauksen vaatimat kehysten kentät, kuittaukset, uudelleenlähetykset ja kuljetettavan datakehyksen koko. Lukuisat tekijät siis heikentävät verkon suoritustehoa ja se jääkin yleensä huomattavasti datanopeutta pienemmäksi. 802.11n-lisäosa määrittelee MAC- ja fyysisen kerroksen parannukset, joilla pyritään vähintään 100 Mbit/s suoritustehoon. Suoritusteho jää silti luonnollisesti edelleen huomattavasti pienemmäksi 802.11n-verkon maksimaalisesta 600 Mbit/s datanopeudesta.

Aiemmat 802.11-verkon nopeuden parannukset tapahtuivat lähinnä kasvattamalla datanopeutta tai hyödyntämällä korkeampaa 5 MHz:n taajuusaluetta. Aina oli kuitenkin käytetty vain yhtä antennia liikennöimiseen eli niin sanottua yksiantennijärjestelmää (Single-Input Single-Output, SISO). 802.11n:ssä käytetään aikaisemmista lisäosista tuttua OFDM:ää, mutta hyödynnetään moniantennijärjestelmää (Multiple-Input Multiple-Output, MIMO), missä tiedonsiirto tapahtuu useamman antennin kautta. Oleellista ei ole pelkkä antennien määrän lisääminen, vaan että eri antenneja voidaan käyttää samanaikaisesti eri datavirtoihin (spatial stream). Lähetettäessä dataa se jaetaan useampaan osaan, datavirtaan, jotka lähetetään yhtä aikaa eri antennien kautta samalla taajuudella ja yhdistetään vastaanottajapäässä taas yhdeksi dataksi. Näin saadaan karkeasti ottaen moninkertaistettua siirretyn datan määrä aikayksikköä kohti. Eri antenneista lähetetyt datavirrat kulkevat hieman eri reittejä ja saapuvat vastaanottajan antenneihin eri aikoihin. Tällainen monitie-eteneminen oli aiemmin 802.11-verkossa ennemminkin haitta, jota vastaan tuli varautua, mutta 802.11n käyttää sitä hyödyksi. 802.11n-lisäosa vaatii, että sen mukaisesti toimivat laitteet pystyvät käyttämään lähettämiseen ja vastaanottamiseen vähintään kahta antennia ja käsittelemään vähintään kahta datavirtaa. Maksimissaan sekä liikennöintiin käytettäviä antenneja että datavirtoja sallitaan neljä. MIMO-tekniikan lisäksi toinen merkittävä, mutta vapaaehtoinen uusi parannus fyysisellä kerroksella on käytetyn kanavan leventäminen. Aiemmin OFDM:ssä oli käytetty 20 MHz:n kaistanleveyttä, mutta nyt kaksi vierekkäistä kanavaa voidaan

yhdistää isommaksi 40 MHz:n kanavaksi. Tämä lisää teoriassa datanopeuden kaksinkertaiseksi.

Tärkein 802.11n:n esittelemä MAC-kerroksen uudistus on kehysten niputtaminen, jota on kahta tyyppiä. Niputetussa MSDU:ssa (Aggreagate MSDU, A-MSDU) MAC-kerroksella luotavan datakehyksen datakenttään voidaan upottaa useampikin LLC-kerrokselta saatu eteen päin välitettävä MSDU-kehys, jos ne eivät ole liian isoja. Normaalisti datakenttään sijoitettaisiin aina vain yksi MSDU-kehys. Toinen tyyppi on niputettu MPDU (Aggreagate MPDU, A-MPDU), missä useampia dataa sisältäviä MAC-kehyksiä, kokonaisia datakehyksiä, sijoitetaan fyysisellä kerroksella koostettavan PLCP-kehyksen datakenttään. Kumpikin niputustapa voidaan vielä yhdistää ja käyttää molempia yhtä aikaa. Niputusten yhteydessä käytetään toista MAC-kerroksen parannusta, lohkokuittausta (block acknowledge, BACK). Normaalisti 802.11-verkossa jokainen vastaanotettu lähetys kuitataan lähettäjälle yksitellen. Yhdellä lohkokuittauksella voidaan kuitata useita vastaanotettuja lähetyksiä kerralla. Kolmas merkittävä MAC-kerroksen uudistus on paluuviesti (reversed direction).

Normaalisti, kun laite haluaa lähettää dataa toiselle laitteelle, se joutuu aina ensin kuuntelemaan DIFSin mittaisen ajan onko yhteinen siirtotie vapaana ja kilpailemaan sen jälkeen kilpailuikkunassa siirtotielle pääsystä. Sallimalla datan vastaanottajan lähettää heti datan saatuaan paluuviestinä itse dataa takaisin päästään verkon toiminnassa vähemmällä siirtotien seuraamisella ja varaamisella. Kun asema saa siirtotien käyttöönsä lähettää se datan mukana vastaanottajalle tiedon, että se voi lähettää halutessaan saamaansa dataan liittyvän kuittauksen perään myös oman datakehyksen. Näin vastaanottajan ei tarvitse lähettää pelkkää kuittausta saamaansa viestiin ja yrittää myöhemmin varata siirtotietä itselleen hitaamman menettelyn kautta, vaan se voi lähettää heti kuittausten perässä paluuviestinä itsekin dataa.

802.11n on täysin yhteensopiva aiempien 802.11a/b/g-versioiden kanssa eli se tuntee myös kaikki näissä määritellyt MAC- ja fyysisen kerroksen kehysrakenteet ja modulointitavat. Toimiessaan kuitenkin tällaisten laitteiden kanssa muodostetussa sekaverkossa 802.11n-laitteet eivät yllä maksimaaliseen 600 Mbit/s datanopeuteensa, johon voidaan päästä vain, kun verkossa on ainoastaan 802.11n-laitteita, toimitaan 5 GHz:n taajuudella, käytetään neljää antennia ja datavirtaa ja kaikki lisäosassa vapaaehtoiseksikin määritellyt ominaisuudet ovat käytössä.

3 Tietoturva IEEE 802.11 -verkossa

Tietoturva on erittäin tärkeässä osassa kaikessa tietoliikenteessä, niin myös langattomassa lähiverkossa. Hyvä johdatus aiheeseen löytyy Stallingsin kirjasta [Sta10]. Tietoturva voidaan jakaa usealla tavalla osa-alueisiin, mutta tunnetuin lienee jako kolmeen pääkohtaan: tiedon luottamuksellisuuteen, eheyteen ja saatavuuteen. Näitä pääkohtia täydentävät vielä todennus ja kiistämättömyys.

Tiedon *luottamuksellisuus* (confidentiality) tarkoittaa sitä, ettei tieto paljastu ulkopuolisille. Luottamuksellista tietoa eivät saa käsiinsä tai eivät pysty tulkitsemaan muut kuin vain ne, joilla on oikeus siihen. Luottamuksellisuus varmistetaan salauksella sekä pääsynhallinnalla.

Tiedon *eheys* (integrity) estää tiedon huomaamattoman muuttumisen. Tahallisesta tai tahattomasta toiminnasta johtuva tiedon muuttuminen on kyettävä havaitsemaan. Eheys saavutetaan tiiviste- ja MAC-funktioilla (Message Authnetication Code) ja digitaalisilla allekirjoituksilla.

Saatavuus (availability) eli käytettävyys tarkoittaa, että tieto on siihen oikeutettujen osapuolten saatavilla aina, kun tietoa tarvitaan. Tiedon saantia ei ole estetty tai hidastettu tai muutenkaan vaikeutettu. Tätä on vaikea taata aukottomasti. Jossain määrin se onnistuu tiedonsiirtokapasiteettia lisäämällä, mutta vaatii usein paljon moniumutkaisempia toimenpiteitä.

Viestinnän osapuolten *todentamisella* (authentication) varmistetaan luotettavasti vastapuolen identiteetti. Todennus voi olla vain yksipuolistakin, mutta usein on tarpeen tehdä todennus molemminpuolisesti. Todennus perustuu usein johonkin molempien osapuolten tuntemaan salaiseen tietoon.

Kiistämättömyys (deniability) tarkoittaa tiedon tai tapahtuman jäljitettävyyttä, voidaan kiistämättömästi osoittaa tapahtunut. Tämä auttaa myös virheiden jäljittämisessä ja järjestelmään tunkeutuvien hyökkääjien havaitsemisessa. Tietotekniset järjestelmät eivät ole virheettömiä, mutta jäljitettävyys auttaa virheiden löytämisessä ja niiden alkuperän selvittämisessä. Kiistämättömyyttä saadaan muun muassa digitaalisilla allekirjoituksilla ja tapahtumalokeilla.

Tässä luvussa käsitellään tiedon luottamuksellisuuden ja eheyden sekä todennuksen ja kiistämättömyyden toteutumista alkuperäisen 802.11-standardin määrittelemässä lähiverkossa. Myöhemmin standardin lisäosa 802.11i paransi verkon tietoturvaa ja sen vaikutuksia tietoturvan osa-alueisiin käsitellään luvussa 4. Parannuksista huolimatta 802.11i oli edelleen haavoittuvainen verkon saatavuutta uhkaaville palvelu-

nestohyökkäyksille. Niitä käsitellään luvussa 6.

3.1 802.11-verkon havaitseminen

Langaton lähiverkko muodostuu sitä hallinnoivan tukiaseman ympärille. Verkkoon haluavan aseman on oltava tukiaseman kantaman sisällä, jotta sen signaali erottuisi vielä taustakohinasta ja kommunikointi onnistuisi. Suurentamalla lähetystehoa tai käyttämällä parempia antenneja kantamaa voidaan kasvattaa. Suomessa WLAN:in toimintataajuuksilla toimivien laitteiden lähetystehoa on kuitenkin rajoitettu. Lähetystehon antamissa rajoissa kantamaa voidaan kasvattaa myös käyttämällä parempia antenneja, vaihtamalla yleisesti laitteissa oleva ympärisäteilevä antenni suunta-antenniin. Tyypillinen ympärisäteilevä antennityyppi, piiska-antenni, säteilee tasaisesti joka suuntaan, mutta sen kantama ei ole kovin hyvä. Suunta-antennit, kuten parabolinen peiliantenni eli lautasantenni tai yagiantenni, säteilevät eri tavalla eri suuntiin. Tyypillisesti ne säteilevät yhteen kapeaan suuntaan hyvin ja selvästi heikommin muihin suuntiin. Tarkasti kohdistettuna suunta-antenneilla saadaan kantamaa kuitenkin kasvatettua huomattavasti.

Langattomia lähiverkkoja käytetään hyvin erilaisissa tilanteissa ja paikoissa, kuten esimerkiksi kodeissa, yrityksissä, lentokentillä ja nettikahviloissa. Osa verkoista on yksityisiä ja osa kaikille avoimia julkisia verkkoja, mistä seuraa hyvinkin erilaisia vaatimuksia verkon näkyvyydelle ja avoimuudelle. Jos verkkoa ei ole tarkoitettu täysin julkiseksi ja avoimeksi, sen kantamaa on syytä rajata kattamaan vain tarpeellinen alue. Tähän voidaan vaikuttaa säätämällä tukiaseman lähetystehoa ja sijoittamalla tukiasema sopivasti, sillä ympäristön rakenteet heikentävät myös kantamaa.

Langattoman lähiverkon tietoturvan kannalta verkon kantama on ensimmäinen askel. Jos hyökkääjä ei kuule mitään verkon lähetyksiä, ei verkkoa hyökkääjän näkökulmasta ole edes olemassa eikä sitä vastaan voi hyökätä langattomasti. WLANit toimivat usein kaapeliverkon reunalla, joten hyökkääjä voi tulla myös kaapeliverkon suunnasta, vaikka kantama olisikin liian lyhyt radioaalloilla tapahtuvaan hyökkäykseen. Myös verkossa toimivan laitteen joutuminen fyysisesti hyökkääjän haltuun mahdollistaa hyökkäyksen verkon sisältä käsin. Hyökkääjän kaappaama laite näyttää muille verkon laitteille luotettavalta todennetulta osapuolelta eikä ulkopuoliselta laitteelta.

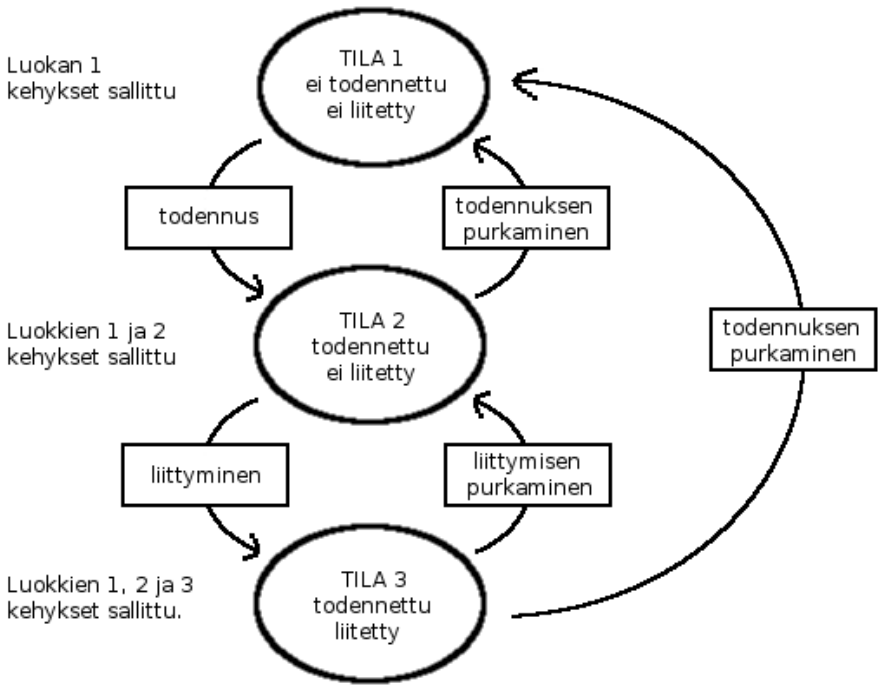
Tukiasemien ympäristöönsä säännöllisesti lähettämiä beaconeita kuuntelemalla voidaan tiheästi asutuilla urbaaneilla alueilla havaita helposti lukuisia langattomia

lähiverkkoja [IsM07], [Sec11]. Tällaista WLANien kartoittamista ajamalla autolla ympäriinsä ja käyttämällä sopivaa päätelaitetta, ohjelmistoa ja tarvittaessa tarpeeksi herkkää antennia kutsutaan *loiskäytöksi* (wardriving). Loiskäyttö tuli tunnetuksi, kun Peter Shipley raportoi hakkerikonferenssissa DEFCONissa vuonna 2001 tiettävästi ensimmäisen kerran systemaattisista ja automatisoiduista kokeiluistaan WLANien kartoittamiseksi [Ber04]. Edistyneimmillään kartoituksessa voidaan käyttää myös GPS-paikannusta, jolloin löydetyt WLANit yhdistetään fyysiseen paikkatietoon. Loiskäytössä ei kuitenkaan käytetä löydettyjä verkkoja. Siinä vain etsitään ja kartoitetaan ympäristöstä löytyviä WLANeja. Loiskäyttö ei ole ainakaan Suomessa laitonta ja sillä voidaankin esimerkiksi etsiä julkisia avoimia verkkoja, kartoittaa verkkojen levinneisyyttä ja niiden tietoturvan tasoa. Hyökkääjä saa toki selville samat tiedot ja voi käyttää niitä hyväksi suunnitellessaan hyökkäystä tai valitessaan sopivaa uhria. Vielä enemmän hyökkääjä saa tietoa verkosta kuuntelemalla beaconien lisäksi muitakin verkon laitteiden välillä lähetettyjä kehyksiä. Koska loiskäyttö ja liikenteen kuuntelu on helppoa on luotettavien todennuskeinojen käyttäminen erittäin tärkeää uuden aseman liittyessä verkkoon, ellei verkkoa ole tarkoitettu täysin avoimeksi ja julkiseksi. Myös liikenteen salauksesta on huolehdittava, jotta ulkopuoliset eivät pysty lukemaan avoimesti verkossa liikkuvaa dataliikennettä.

3.2 Todentaminen 802.11-verkossa

WLANissa jokainen laite pitää yllä kahta tilamuuttujaa jokaista muuta verkon laitetta kohti, jonka kanssa se kommunikoi. Toisin sanoen asemalla on yksi tilamuuttujapari, koska se kommunikoi vain tukiaseman kanssa ja tukiasemalla on tilamuuttujapari jokaista sen verkkoon kuuluvaa asemaa kohti. Tilamuuttujat ovat *todennus* ja *liittyminen*. Kuvassa 10 on esitetty näiden muuttujien muodostama kolmitilainen tilakaavio. Tilakaaviossa on oleellista, että nykyinen tila määrittelee, mitkä MAC-kehykset ovat sallittuja. Tilassa yksi oleva laite voi lähettää tai vastaanottaa kaikkein pienimmän valikoiman kehyksiä, mutta onnistuneen todennuksen ja liittymisen myötä tiloissa kaksi ja kolme sallittujen kehysten määrä kasvaa. Liittymisen ja todennuksen purkamiset vievät takaisin aiempiin tiloihin. Tärkeimmät eri tiloissa sallitut MAC-kehykset on esitelty taulukossa 2. Tilassa yksi on sallittu vain luokan yksi kehykset, tilassa kaksi myös luokan kaksi kehykset ja tilassa kolme kaikki kehykset ovat sallittuja.

WLANissa on useita todennustapoja, mutta alkuperäinen standardi määrittelee vain kaksi: avoimen ja jaettuun avaimeen perustuvan todennuksen. Näiden lisäksi valmis-



Kuva 10: Todennuksen ja liittymisen muodostama tilakaavio ja sallitut kehykset

tajat ovat kehitelleet omia laitekohtaisia todennustapoja, jotka myöhemmin levisivät käytännössä kaikkiin muihinkin laitteisiin. Tärkeimpiä näistä ovat suljettu verkko [Luc00] sekä MAC-osoitteisiin perustuva pääsynhallinta.

Avoin todennus ei ole oikeastaan todennus ollenkaan. Siinä tukiasema hyväksyy kaikki vastaanottamansa authentication requestit niiden lähettäjästä riippumatta. Avoimessa verkossa ei siis ole minkäänlaista kontrollia siinä, kuka voi yrittää verkkoon liittymistä. Tästä huolimatta toisen avointa todennusta käyttävään verkkoon liittyminen ilman verkon omistajan lupaa oli Suomessa aikaisemmin laitonta. Maaliskuussa 2011 voimaan astuneen lainmuutoksen myötä toisen avoimeen verkkoon liittyminen ja sen käyttäminen esimerkiksi internet-yhteyteen muuttui sallituksi.

Tukiasema lähettää ympäristöönsä lyhyin säännöllisin väliajoin verkkoansa mainostavia beaconeita. Niissä tukiasema kertoo hallinnoimastaan verkosta tarpeellisia tietoja, jotta asemat voivat liittyä sen verkkoon. Beaconista selviää muun muassa käytössä oleva salaustapa, tuetut yhteysnopeudet ja tukiaseman verkossa käytetty kanava. Jokaisella peruspalveluryhmällä on sitä hallinnoivan tukiaseman sille antama palveluryhmätunniste (service set identifier, SSID), joka verkkoon liittyvän aseman

Luokka	Tyyppi	Alityyppi
1	Kontrolli	RTS
1	Kontrolli	CTS
1	Kontrolli	ACK
1	Hallinta	Probe request/response
1	Hallinta	Beacon
1	Hallinta	Authentication request/response
1	Hallinta	Deauthentication
2	Hallinta	Association request/response
2	Hallinta	Reassociation request/response
2	hallinta	Disassociation
3	Kontrolli	PS-Poll
3	Data	

Taulukko 2: Tärkeimmät MAC-kehykset luokittain

on tiedettävä. Jos tukiasema ei lähetä beaconeita, voi asema lähettää probe requestin. Sen kuulevat tukiasemat vastaavat siihen probe responsella, jossa kertovat oleellisilta osin samat tiedot kuin beaconissakin.

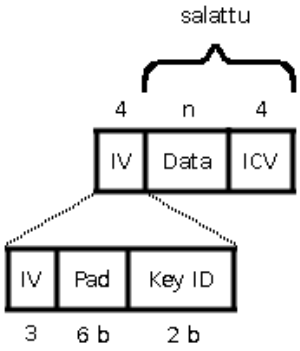
802.11-standardi ei tunne *suljettua verkkoa*, mutta useat tukiasemat tarjoavat sellaisen. Sen ilmestyminen markkinoilla oleviin laitteisiin oli yksi vastaus alkuperäisen 802.11-standardin määrittelemiin puutteellisiin tietoturvaominaisuuksiin. Siinä beaconeissa ei enää lähetetä SSID:tä. Sitä kuitenkin tarvitaan liityttäessä verkkoon, joten verkkoon pääsevät liittymän vain sellaiset asemat, jotka tietävät SSID:n jotain muuta kautta. Suljetulla verkolla voidaan kontrolloida, ketkä kaikki pääsevät verkkoon antamalla SSID vain halutuille tahoille, syöttämällä se esimerkiksi käsin laitteisiin. Satunnaiset verkkoja ympäristöstään skannaavat asemat eivät voi liittyä verkkoon, koska eivät tiedä sen oikeaa SSID:tä. Suljettu verkko ei kuitenkaan ole luotettava keino rajoittaa verkkoon pääsyä. Vaikka SSID-tunnusta ei enää lähetetäkään beaconissa, niin se kulkee salaamattomana muissa asemien ja tukiaseman välisissä lähetyksissä. Hyökkääjä voi selvittää oikean SSID:n esimerkiksi kuuntelemalla toisten asemien tukiasemalle lähettämiä authentication request- tai association request -kehyksiä. Niissä SSID on täysin selväkielisenä. Kuuntelemalla tarpeeksi kauan suljetun verkon liikennettä voi hyökkääjä selvittää oikean SSID:n, vaikka tukiasema ei sitä lähettäisikään beaconeissaan.

Verkkoon pääsyä voidaan rajoittaa myös *MAC-osoitteiden suodatuksella*. Jokaisella

verkkolaitteella on valmistajan sille antama MAC-osoite. IEEE jakaa niitä laitevalmistajille, joten jokaisella laitteella on yksilöllinen MAC-osoitteensa. Pääsynhallinta tapahtuu verkkoon haluavien laitteiden MAC-osoitteiden perusteella. Koska verkossa liikkuvissa kehyksissä näkyy sekä lähettäjän että vastaanottajan MAC-osoite, voi tukiasema sallia verkkoonsa pääsyn vain niille asemille, joiden MAC-osoite löytyy tukiasemassa määritellyltä sallittujen MAC-osoitteiden listalta. Ratkaisussa on kuitenkin kaksi huonoa puolta. Koska tukiaseman MAC-osoitteiden listaa on päivitettävä käsin, tulee siitä hyvin työlästä useita laitteita käsittävässä verkossa, jossa tapahtuu usein muutoksia. Toinen ongelma liittyy MAC-osoitteiden väärentämiseen. Koska MAC-osoitteet ovat kaikissa verkossa liikkuvissa kehyksissä aina täysin salaamattomina, voi hyökkääjä verkkoa kuuntelemalla saada selville jo verkossa olevien asemien MAC-osoitteet. Huolimatta laitteille valmistuksessa annetuista kiinteistä MAC-osoitteista, niitä voidaan kuitenkin muuttaa ohjelmallisesti. Näin hyökkääjä voi vaihtaa omaksi MAC-osoitteekseensa jonkin verkossa sallitun osoitteen, odottaa että kyseisen osoitteen omaava asema ei ole sillä hetkellä verkossa, vaihtaa oman MAC-osoitteensa sallittuun ja läpäistä tukiaseman MAC-osoitteeseen perustuvan pääsynhallintalistan. Suljettu verkko ja MAC-osoitteiden suodatus olivat aluksi valmistajakohtaisia ratkaisuja parantaa 802.11-verkon tietoturvaa, mutta ne ovat yleistyneet nykyään käytännössä kaikkiin laitteisiin. Näitä ei kuitenkaan ole suositeltavaa käyttää ainoina todennuskeinoina, vaan ainoastaan muiden keinojen ohessa antamassa lisäturvaa.

3.3 WEP-tietoturvaprotokolla

WEP (Wired Equivalent Privacy) oli mukana jo 1997 ilmestyneessä 802.11-standardissa. Se on siirtoyhteyskerroksen tietoturvaprotokolla, jonka oli tarkoitus tarjota nimensä mukaisesti langattomassa verkossa kaapeliverkon tasoista tietoturvaa. Sitä käytetään sekä jaettuun avaimeen perustuvassa todennuksessa että salaamaan todennettujen asemien ja tukiaseman väliset datakehykset. Tarkalleen ottaen sillä salataan vain osa datakehysten datakentästä ja kaikki muut kehykset ja datakehyksen osat liikkuvat verkossa täysin selväkielisenä. Kuvassa 11 on esitetty MAC-kerroksen datakehyksen datakentän sisältö käytettäessä WEP-salausta. Alussa ennen dataa on alustusvektorikenttä, joka jakaantuu vielä varsinaiseen kolmen tavun mittaiseen alustusvektoriin (Initialization Vector, IV), täytteeseen (Pad) sekä Key ID:hen. Jokainen salattu kehys saa uuden IV:n. Lisäksi salauksessa käytetään WEP-avainta, joita standardin mukaan voi olla päätelaitteissa määriteltynä korkeintaan neljä. Key

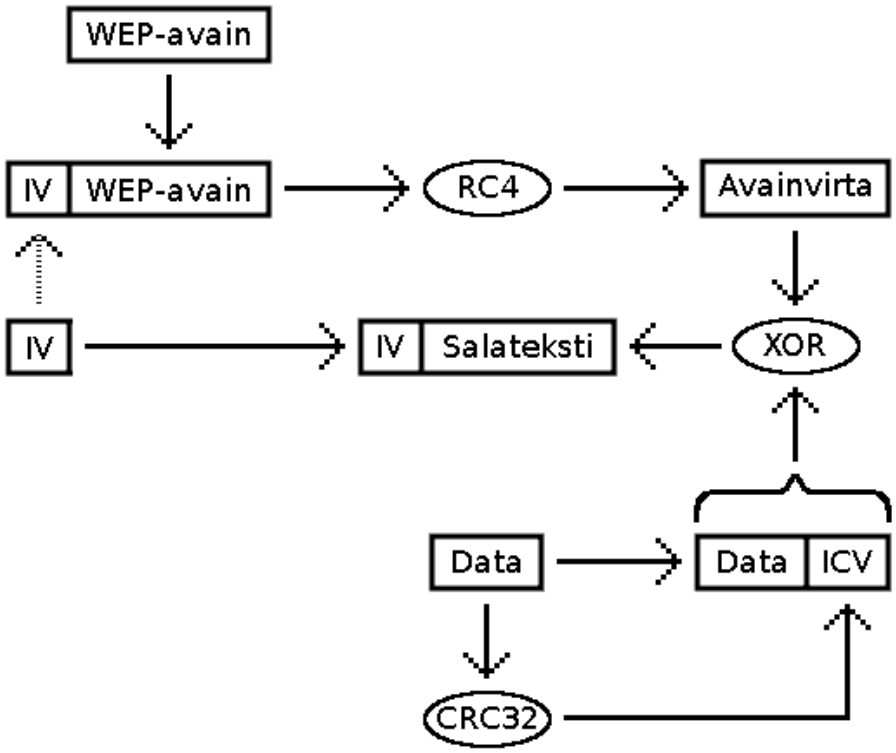


Kuva 11: Datakehyksen datakenttä WEPissä

ID:llä kerrotaan, mitä näistä neljästä avaimesta kulloinkin käytetään. Usein WEP-toteutuksissa käytetään kuitenkin vain yhtä ainoaa WEP-avainta. Alustusvektorikenttä kulkee aina selväkielisenä ja vain loppuosa datakentästä on salattu eli varsinainen data ja sen jälkeen tuleva datasta laskettu ICV-kenttään sijoitetu eheystarkistus. WEP perustuu jonosalaukseen ja 40 bitin mittaiseen salaiseen avaimeen, joka on syötetty verkon laitteisiin. Laajennetussa peruspalveluryhmässä kaikilla asemilla ja tukiasemilla on oltava sama yhteinen kaikkien tuntema salausavain.

Kuvassa 12 on esitetty datan salaaminen WEP-protokollassa. Alustusvektorikentästä löytyvä 24-bittinen IV ja salainen WEP-avain yhdistetään ja niiden muodostama merkkijono syötetään RC4-salausalgoritmille, jolta saadaan syötteenä avainvirta (keystream). Salattavasta datasta lasketaan eheyden tarkistussumma (Integrity Check Value, ICV) CRC32-algoritmilla ja laitetaan se datan perään. ICV auttaa havaitsemaan tiedonsiirrossa tapahtuneita virheitä. Datan ja ICV:n muodostamalle selvätekstille ja avainvirralle suoritetaan poissulkeva bittitason tai-operaatio (exclusive or, XOR). Näin on saatu salateksti lähetettäväksi. Salatekstin eteen liitetään vielä alustusvektorikenttä, joka sisälsi salauksessa käytetyn IV:n, jota ilman vastaanottaja ei voi purkaa salausta. Vaikka IV onkin lähetetyssä kehyksessä salaamattomana, niin ulkopuolinen tarkkailija ei voi purkaa salausta, koska siihen tarvitaan myös oikea WEP-avain.

WEP-avaimen pituus oli aluksi vain 40 bittiä, sillä USA:n vientirajoitukset eivät sallineet liian vahvojen kryptografisten salausmenetelmien maasta viemistä. Myöhemmin, kun näin lyhyeen avaimeen perustuva salaus osoittautui epäluotettavaksi, myös 104 bitin mittainen avain tuli mahdolliseksi ja tätäkin pitempiä avaimia saattaa olla yksittäisissä valmistajakohtaisissa laitteissa. WEPissä salaukseen käytetyn RC4-algoritmin kehitti Ron Rivest 1987. Se pysyi vuoteen 1994 asti Yhdysvalloissa



Kuva 12: Datan salaus WEPissä

kauppasalaisuutena, jolloin se julkaistiin nimettömänä internetissä.

WEP-salausprotokollan oli tarkoitus taata asemien luotettava todennus sekä verkossa liikkuvan datan salaus ja eheys. Mihinkään näistä tavoitteista ei päästä ja WEPiä ei pitäisikään enää käyttää tietoturvaratkaisuna missään 802.11-pohjaisessa lähiverkossa. WEP suojaa korkeintaan satunnaiselta avointa verkkoa etsivältä ohikulkijalta, mutta ei hiukankaan asiansa osaavalta hyökkääjältä. Sitä ei pitäisi käyttää kuin tietoturvavaatimuksiltaan hyvin matalan tason verkoissa tai taaksepäin yhteensopivuuden vuoksi ja tällöinkin mielellään vain väliaikaisesti. WEP on hyvä esimerkki siitä miten sinänsä turvallista RC4-salausalgoritmia käytetään sopimattomasti ja tuntematta 802.11-verkon toimintaympäristön sille asettamia vaatimuksia. Seuraavaksi esitellään lyhyesti WEPin keskeisimmät tietoturvaongelmat.

3.3.1 Avainvirran paljastuminen

WEP-protokollan eräs heikkous on siinä, että datan salaamiseen ei välttämättä tarvita WEP-avainta ollenkaan. Pelkän yhden avainvirran tunteminen riittää paketin salaamiseen. Kyseiseen avainvirtaan liittyvä alustusvektori on myös tunnettava ja

lähetettävä datan mukana, jotta vastaanottaja voi myös purkaa salauksen. Alustusvektorin selvittäminen ei kuitenkaan ole ongelma, sillä ne kulkevat kaikissa salatuissa lähetyksissä selväkielisenä. Yksittäisen avainvirran tunteminen riittää myös purkamaan salauksen kaikista paketeista, joiden salaukseen on käytetty samaa tunnettua avainvirtaa. Edelleen, jos hyökkääjä saa selville kaikki mahdolliset avainvirrat, hän voi purkaa salauksen mistä tahansa paketista. Avainvirran - yhden tai useamman - paljastuminen menettää kuitenkin merkityksensä, jos verkon yhteistä WEP-avainta vaihdetaan, mutta käytännössä sitä vaihdetaan harvoin. Avainvirran paljastumisongelma liittyy WEPissä käytettyyn symmetriseen jonosalaukseen. Salauksessa saadaan salateksti XOR-operaatiolla (merkitään $\oplus$):

$$C(P) = RC4(IV, k) \oplus P,$$

missä P on salattava selväteksti (data ja ICV) ja $RC4(IV, k)$ RC4-algoritmilla IV:stä ja WEP-avaimesta k tuotettu avainvirta. XOR-operaatio on kuitenkin käänteinen ja hyökkääjä tuntiessaan salatekstin ja selvätekstin voi selvittää avainvirran:

$$RC4(IV, k) = C(P) \oplus P.$$

Hyökkääjä voi selvittää avainvirran ainakin kahdella tavalla: yksinkertaisella passiivisella todennuksen kuuntelulla [BGM01] tai aktiivisella ja monimutkaisemmalla pilkkomishyökkäyksellä (chopchop) [Kor04a].

Avoimen todennuksen lisäksi toinen 802.11-standardissa määritelty todennustapa perustuu yhteiseen jaettuun avaimeen, WEP-avaimeen, sekä haaste-vaste-periaatteeseen. Siinä asema ja tukiasema vaihtavat keskenään useita authentication-kehyksiä. Aluksi asema ilmaisee halunsa käyttää yhteisen jaetun avaimen -todennusta. Tukiasema vastaa lähettämällä selväkielisen haastetekstin. Asema salaa sen molempien tuntemalla WEP-avaimella ja lähettää takaisin tukiasemalle. Tukiasema purkaa salauksen ja vertaa sitä alkuperäiseen lähettämäänsä haastetekstiin. Vain jos tekstit ovat samoja, on todennus onnistunut ja tukiasema lähettää todennuksen hyväksyvän authentication responsen. Jaetun avaimen -todennuksessa käytetään aina WEPiä ja yhteinen jaettu avain on sama WEP-avain, mitä käytetään myös datan salaamisessa.

Kuuntelemalla tukiaseman ja aseman välistä onnistuneen todennuksen viestienvaihtoa hyökkääjä saa selville sekä haasteena käytetyn selvätekstin että saman tekstin salattuna. Näiden avulla saadaan selville yksi kokonainen avainvirta. Lisäksi hyökkääjä tietää, mikä IV liittyy kyseiseen avainvirtaan, sillä se kulki selväkielisenä aseman tukiasemalle lähettämän salatekstin mukana. Avainvirran selvitettyään hyökkääjä voi todentaa itsensä verkkoon, vaikka ei tiedäkään yhteistä jaettua WEP-avainta.

Hyökkääjä toimii normaalin todennusprotokollan mukaisesti. Tukiaseman lähettäessä haastetekstin hyökkääjä muodostaa siitä eheyden tarkistussumman (ICV), käyttää aiemmin selvitettyä avainvirtaa salaamaan haastetekstin ja lähettää sen tukiasemalle yhdessä oikean IV:n kanssa. Tukiasema purkaa salatekstin ja löytää aiemmin lähettämänsä haastetekstin. Tukiasema päättelee tästä hyökkääjän omistavan oikean WEP-avaimen ja hyväksyy todennuksen. On suorastaan tragikoomista, että WEP-protokollan todennuksen heikko toteutus antaa suoraan hyökkääjälle eväät todennuksen huijaamiseen. Jos uusia asemia liittyy tukiasemaan harvakseltaan, voi hyökkääjä todennusprotokollan kehysten generoimiseksi lähettää todennuksen purkavan deauthentication-kehyksen verkkoon kuuluvalle asemalle, jonka jälkee asema todentaa itsensä uudestaan tukiasemalle. WEPin todennuksessa ilmenneisiin ongelmiin laitevalmistajat vastasivat tuomalla jo aiemmin mainitut suljettuun verkkoon ja MAC-osoitteiden suodatukseen perustuvat todennustavat laitteisiinsa.

Datakehyksen salaus voidaan purkaa ja selvittää sen salaukseen käytetty avainvirta myös nimimerkin KoreK internetin keskustelupalstalla vuonna 2004 esittämällä niin sanotulla *pilkkomishyökkäyksellä* [Kor04a]. Onnistuneen todennuksen kuuntelussa saadaan selville vain todennuksessa käytetty avainvirta, mutta pilkkomishyökkäyksellä voidaan selvittää minkä tahansa salatun datakehyksen salaamiseen käytetty avainvirta. Pilkkomishyökkäystä käytettäessä ei tarvitse ensin odottaa jonkun toisen aseman todentavan itseänsä verkkoon, vaan hyökkääjälle riittää yhden minkä tahansa salatun kehyksen kuuntelu. Kehyksen salatun osan viimeinen tavu tiputetaan pois eli kehystä typistetään ja muokataan jäljelle jäänyttä osaa sopivasti. Jos mitään muokkausta ei tehtäisi, niin ainakin CRC32-tarkistussumma olisi todennäköisesti aivan väärä. KoreK havaitsi, että poisjätetyn salatun tavun selväkielisestä muodosta voidaan johtaa typistetylle kehykselle tehtävät muutokset, jotta se olisi edelleen kelvollinen kehys. Koska hyökkääjä ei kuitenkaan tiedä, mikä selväkielinen tavu vastaa salatussa kehyksessä olevaa viimeistä salattua tavua, hän arvaa sen ja muokkaa typistetyn kehyksen ICV:n vastaamaan arvausta. Typistetty ja muokattu kehys lähetetään tukiasemalle. Pilkkomishyökkäyksessä tukiasemaa käytetään eräänlaisena oraakkelina, jonka vastauksen perusteella voidaan päätellä, oliko arvaus oikea vai väärä. Tukiasema saadessaan datakehyksen verkkoonsa todentamattomalta asemalta toimii eri lailla riippuen siitä, onko kehyksen eheydentarkistus kelvollinen vai ei. Jos ICV-kentässä oleva CRC32-eheydentarkistussumma on väärä, tukiasema hylkää kehyksen hiljaisesti. Tästä hyökkääjä päättelee arvauksensa olleen väärän. Sen jälkeen hyökkääjä arvaa tavulle seuraavan mahdollisen arvon, tekee sen mukaiset muutokset typistetyn kehyksen ICV-kenttään ja lähettää kehyksen

uudestaan tukiasemalle. Jos eheydentarkistuksessa ei ole vikaa, tukiasema lähettää todennuksen purkavan deauthentication-kehyksen. Tästä hyökkääjä tietää arvauksensa olleen oikean ja voi siirtyä selvittämään salatun osan toiseksi viimeistä tavua. Näin hyökkääjä etenee tavu tavulta ja saa lopulta purettua koko datakehyksen salauksen. Koska yhden tavun arvolla voi olla korkeintaan 256 eri vaihtoehtoa koko salauksen purkamiseen vaaditaan keskimäärin $n * 128$ pakettia, missä n on kaapatussa paketissa olevien salattujen tavujen lukumäärä. Salauksen purkamisen jälkeen sen salaukseen käytetyn avainvirran selvittäminen on hyvin helppoa, koska sekä salateksti että selväteksti ovat hyökkääjän tiedossa.

3.3.2 Eheydentarkistuksen heikkoudet

WEPissä datan todennukseen käytetään CRC32:lla datasta ennen salausta laskettua eheydentarkistusta. Se soveltuu hyvin satunnaisten liikennöinnissä tapahtuneiden bittivirheiden havaitsemiseen, mutta ei kuitenkaan riitä suojaamaan dataa tarkoituksellista ja suunnitelmallista muuttamista vastaan. Hyökkääjä voi muuttaa salattua dataa ja laskea sen aiheuttamat muutokset CRC32 tarkistussummaan, niin ettei kehyksen vastaanottaja voi tarkistussumman perusteella päätellä mitään muutosta tapahtuneenkaan [BGM01]. Tämä on erittäin kätevä keino hyökkääjälle, sillä hänen ei tarvitse tietää WEP-avainta tai edes avainvirtaa voidakseen muuttaa salatekstiä.

Kehyksen muokkausmahdollisuutta ilman sen paljastumista vastaanottajalle voidaan käyttää jopa viestin salauksen purkamiseen. Tämä perustuu huijaukseen, jossa tukiasema saadaan purkamaan salaus hyökkääjälle. Tämän onnistuminen vaatii kuitenkin, että tukiasema toimii myös reitittimenä internetin suuntaan ja että hyökkääjällä on siellä apulainen. Hyökkääjä kaappaa aseman lähettämän salatun datakehyksen, muuttaa sen vastaanottajaksi internetissä olevan apulaisensa IP-osoitteen ja lähettää tukiasemalle. Kehyksen saatuaan tukiasema purkaa sen salauksen ja lähettää hyökkääjän apulaiselle täysin selväkielisenä. Hyökkääjän on selvitettävä kaapatun kehyksen alkuperäinen IP-kohdeosoite, jotta siihen voidaan tehdä tarvittava muutos. Lisäksi on korjattava IP:n tarkistussumma, jotta IP-osoitteen muutos voidaan tehdä huomaamattomasti. Lopuksi on vielä korjattava WEP-salauksen CRC32-tarkistussumma muiden muokkausten piilottamiseksi [BGM01].

3.3.3 WEP-avaimen murtaminen

Koska WEP-salaus perustuu vain yhteen salaiseen avaimeen, sen paljastuminen antaa hyökkääjälle mahdollisuuden purkaa kaikkien samassa verkossa kulkevien datakehysten salauksen. Myös omien pakettien lähettäminen verkkoon onnistuu todennuksen jälkeen. WEP-protokollassa ei määritellä minkäänlaista avaimenhallintaa laitteiden kesken eli avaimen syöttämiseen tai vaihtamiseen ei oteta standardissa mitään kantaa. Käytännössä sama WEP-avain on syötettävä käsin jokaiseen verkon laitteeseen ja mitä isommasta verkosta on kyse sitä suuritöisempää tämä on. Tästä johtuen avaimia vaihdetaan harvoin. Avaimen paljastuminen takaa hyökkääjälle käytännössä pitkäksi aikaa rauhan kuunnella kaikkea verkossa kulkevaa liikennettä salauksesta huolimatta. Alkuperäinen vain 40 bitin mittainen WEP-avain oli murrettavissa jopa niin sanotulla väsytystekniikalla eli brute force -hyökkäyksellä. Siinä kokeillaan kaikki mahdolliset vaihtoehdot, mikä onnistuu vielä jossain määrin mielekkäässä ajassa avaimen ollessa lyhyt. Tämän johdosta WEP-avaimen pituutta kasvatettiinkin 104 bittiin, joka teki väsytystekniikasta käytännössä mahdottoman.

Vuonna 2001 Fluhrer, Mantin ja Shamir esittivät myöhemmin FMS-hyökkäyksenä tunnetun hyökkäyksen, jolla WEP-avain voidaan murtaa [FMS01]. Se perustuu siihen, että joidenkin IV:iden käyttäminen johtaa tietojen vuotamiseen WEP-avaimesta. Tällaisia niin sanottuja heikkoja IV:itä käytettäessä RC4-algoritmin tuottaman avainvirran ensimmäinen tavu antaa tietoa käytetystä WEP-avaimesta. Kun hyökkääjä saa käsiinsä useampia samalla heikolla IV:llä ja samalla WEP-avaimella muodostettuja avainvirtoja, kasvaa todennäköisyys päätellä oikein WEP-avaimen tietty tavu. Koska IV:itä on vain 2^{24} erilaista, niin jossain vaiheessa ne alkavat toistua. Hyökkääjä selvittää oikean WEP-avaimen tavu kerrallaan. Aluksi hyökkääjä selvittää WEP-avaimen ensimmäisen tavun kuuntelemalla verkon liikennettä niin kauan, että samaa IV:tä käytetään salaukseen useita kymmeniä kertoja. IV:thän kulkevat täysin selväkielisenä kaikissa paketeissa. Tänä aikana WEP-avain ei saa välissä vaihtua. Jokaisesta heikosta IV:stä tuotetusta avainvirrasta saadaan ehdotus oikeaksi WEP-avaimen ensimmäiseksi tavuksi ja mitä useampia samalla IV:llä salattuja paketteja kuunnellaan sitä varmemmin voidaan arvata tavun oikea arvo. Lopulta arvataan WEP-avaimen oikea tavu ja siirrytään selvittämään seuraavaa tavua WEP-avaimesta. WEP-avaimen tavut on selvitettävä järjestyksessä, sillä edeltävät tavut vaikuttavat sitä seuraavien tavujen arvoihin. Toisin sanoen, jos murrettu WEP-avain osoittautuukin vääräksi ja arvaukset menivät pieleen jo jossain alkupäässä, niin kaikki väärin arvatun tavun jälkeenkin olevat tavut ovat väärin. Hyökkääjä voi kuiten-

kin usein kuunnella verkon liikennettä täysin passiivisesti ja huomaamatta pitkiäkin aikoja ja riippuen verkon tietoliikenteen määrästä saa ennemmin tai myöhemmin selville verkossa käytetyn WEP-avaimen.

FMS-hyökkäys toteutettiin myös käytännön kokeissa ja havaittiin tarvittavan noin 5–6 miljoonaa kaapattua datakehystä, jotta WEP-avain voidaan päätellä oikein 50 %:n todennäköisyydellä [SIR02]. Määrä kertyi testiolosuhteissa muutamassa tunnissa. FMS-hyökkäyksen onnistuminen vaatii, että hyökkääjä tuntee avainvirran tai ainakin sen alkuosan. Tämä on kuitenkin selvitettävissä suhteellisen helposti. Kuten aiemmin esitettiin, avainvirran saamiseksi tarvitaan vain selväteksti ja vastaava salateksti. Selvätekstin alkuosa selviää esimerkiksi seuraamalla verkon ARP-pakettiliikennettä (Address Resolution Protocoll). ARP-paketit on helppo tunnistaa niiden koosta sekä vastaanottajasta, sillä ne lähetetään aina yleislähetyksenä (broadcast). WEP-salatuissa paketeissahan kaikki muu kuin data eli muun muassa osoitetiedot ovat täysin selväkielisenä. MAC-kerroksen LLC-kerrokselta saadun datan eli ARP-paketin otsakekentissä on tunnettuja toistuvia merkkejä. Hyökkääjä siis kuuntelee verkkoa ja pääteltyään salatun kehyksen kuuluvan ARP-protokollaan kaappaa sen ja suorittaa sille XOR-operaation tunnetun selvätekstisen alun kanssa ja saa avainvirran alkutavuja selville.

Laitevalmistajat reagoivat ilmenneisiin WEPin heikkouksiin muun muassa poistamalla heikot IV:t valikoimasta. Toisaalta jatkuvasti ilmeni lisää heikkoja IV:itä ja alkuperäiseen FMS-hyökkäykseen löytyi lukuisia parannuksia. Yksi tunnetuimmista on KoreK-hyökkäys, jonka vuonna 2004 nimimerkki KoreK esitti netin keskustelupalstalla [Kor04b]. Se perustuu FMS-hyökkäykseen, mutta siitä poiketen siinä käytetään avainvirran kahdesta ensimmäisestä tavusta saatavaa tietoa WEP-avaimen murtamiseen. KoreKissa on esitetty 16 uutta riippuvuussuhdetta avainvirran ja WEP-avaimen tavujen välillä jo aiemmin FMS-hyökkäyksessä havaitun riippuvuuden lisäksi. Nämä riippuvuudet nopeuttavat WEP-avaimen murtamista. KoreK on huomattavasti FMS:ää nopeampi ja sillä päästään 50 %:n todennäköisyydellä murtamaan WEP-avain, kun kehyksiä on kuunneltu ainoastaan 700000 [Tew07].

Vaikka WEP-salaus alkoikin menettää asemiaan 802.11-verkkojen tietoturvan tarjoajana siinä ilmenneiden vakavien puutteiden ja markkinoille tulleiden kehittyneempien ratkaisujen myötä, jatkui mielenkiinto sen murtamiseksi. Vuonna 2007 julkaistiin entistäkin tehokkaampi FMS-hyökkäystä muistuttava WEP-avaimen murtava hyökkäys, keksijöidensä mukaan nimetty PTW-hyökkäys [TWP07]. Se perustuu kahteen ideaan: Kleinin havaintoon, että murtamisessa voidaan käyttää kaikkia IV:itä

eikä vain heikkoja IV:itä [Kle08] sekä että WEP-avaimen tavuja ei tarvitse selvittää järjestyksessä. Vaikka voidaankin käyttää kaikkia IV:itä, niin kaikki eivät silti ole yhtä hyviä ja anna yhtä varmaa tietoa WEP-avaimen tavuista. Toisaalta kelvollisten kehysten määrä vähenee huomattavasti, kun kaikkia voidaan hyödyntää eikä tarvitse odottaa pelkkiä heikkoja alustusvektoreita sisältäviä paketteja. Lisäksi väärin arvatun WEP-avaimen tavun jälkeen ei tarvitse enää selvittää kaikkia sen jälkeisiä tavuja uudestaan. PTW-hyökkäyksellä saadaan oikea WEP-avain ratkaistua 50 %:n todennäköisyydellä vain noin 35000–40000 paketin kuuntelun jälkeen.

3.3.4 Sirpaloinnin hyväksikäyttäminen

Vuonna 2006 esiteltiin uusia hyökkäystapoja, joilla hyökkääjä saa purettua minkä tahansa viestin salauksen ja voi lähettää itse kehyksiä [BHL06]. Hyökkäykset perustuvat 802.11-protokollan käyttämään tapaan pirstoa (fragment) isommat paketit pienemmiksi. Hyökkääjän on tunnettava muutamia tavuja avainvirran alusta. Tämä onnistuu kuitenkin helposti, sillä salatun datan alkuosa koostuu LLC-kerroksen tunnetuista otsikkokentistä. Käytännössä hyökkääjä saa helposti selville kahdeksan tavua avainvirran alusta aiemmin kuvatulla salatekstin ja selvätekstin oletetun alun välisellä XOR-operaatiolla.

802.11-verkossa iso kehys voidaan pirstaloida korkeintaan 16 pienempään sirpaleeseen. Jokainen sirpale salataan erikseen, mutta samalla avainvirralla. Jos hyökkääjä saa siis selville kahdeksan tavua jostain avainvirran alusta, hän voi salata tällä kahdeksan tavun mittaisen selvätekstin. Hyökkääjä voi lähettää 16 sirpaletta, joista kussakin on kahdeksan tavun mittainen salateksti. Tästä vain neljä tavua voi olla dataa, sillä puolet eli neljä tavua menee jokaisessa datakehyksessä olevaan ICV-kentän tarkistussummaan. Dataa voidaan siis lähettää yhteensä 64 tavua.

Datan lähetysmahdollisuutta voidaan hyödyntää kokonaisen salatun kehyksen murtamisessa. Siihen vaaditaan kuitenkin, että hyökkääjällä on apulainen internetissä ja että sinne on yhteys langattomasta verkosta. Hyökkäys toimii seuraavasti. Hyökkääjä kuuntelee salatun kehyksen ja selvittää sen salauksessa käytetyn avainvirran alusta kahdeksan tavua. Tämän jälkeen hyökkääjä muodostaa kahdesta sirpaleesta koostuvan kehyksen. Ensimmäinen sirpale koostuu lyhyestä hyökkääjän apulaisen osoitteen sisältävästä IP-paketista ja toisena sirpaleena on kaapattu kehys. Molemmat sirpaleet salataan samalla kaapatulla avainvirran alkuosalla. Nämä lähetetään tukiasemalle, joka purkaa salauksen, yhdistää sirpaleet ja lähettää kehyksen selväkielisenä eteen päin internetiin hyökkääjän apulaiselle. Tukiasema ei mitenkään es-

tä sitä, että kokonainen kehys voidaan lähettää vain yhtenä sirpaleena osana vielä isompaa kehystä.

Jotta internetissä olevalle apulaiselle voidaan lähettää, on lähettäjän tiedettävä reitittimen MAC-osoite sekä lähettäjän IP-osoite, mutta nämä ovat suhteellisen helposti selvitettävissä. Tukiasema voi toimia itse reitittimenä tai MAC-osoitteen voidaan päätellä olevan jokin suosituimmista vastaanottajan MAC-osoitteista langattomassa verkossa kulkevissa kehyksissä. Oikeaa IP-osoitetta ei aina edes tarvita, mutta tarvittaessa sekin voidaan päätellä seuraamalla ja kuuntelemalla verkossa liikennöiviä kehyksiä.

Edellä esitetty pirstalointihyökkäys vaati muun muassa pääsyn internetiin ja siellä olevan apulaisen. Aina tämä ei ole mahdollista, mutta tällaisessakin tapauksessa pirstalointia voidaan käyttää yksittäisen avainvirran selvittämiseen. Hyökkäys perustuu tukiaseman tapaan toimia pakettien kanssa, joilla ei ole yhtä yksittäistä vastaanottajaa, vaan ovat yleislähetyksiä. Hyökkääjä toimii aluksi kuten edellä kerrotussa hyökkäyksessä ja salakuuntelee yhden salatun kehyksen. Tämän alusta selvitetään kahdeksan ensimmäistä tavua. Seuraavaksi hyökkääjä lähettää ison paketin yleislähetyksenä ja pilkottuna 16 pirstaleeseen. Kussakin sirpaleessa on kaapatulla avainvirralla salattua dataa neljän tavun verran. Tukiasemalla ei ole sirpaleet saatuaan mitään tarvetta lähettää eteen päin pieniä sirpaleita, vaan se purkaa niiden salauksen, yhdistää sirpaleet ja salaa koko paketin uudella avainvirralla ja lähettää yleislähetyksenä kaikille. Hyökkääjä kuulee sen myös, ja koska salattu data oli hänen lähettämäänsä, voi hän XOR-operaatiolla selvittää 68:n tavun mittaisen avainvirran. Seuraavaksi hyökkääjä toistaa aiemman toimenpiteensä, mutta voikin lähettää 16 sirpaletta, joissa kussakin on 64 tavua dataa ja neljän tavun mittainen ICV. Joka kierroksella hyökkääjä saa siis selville yhä isomman osan jostain avainvirrasta ja vain muutaman sekunnin ja 34 lähetetyn sirpaleen jälkeen hyökkääjä on saanut selville jonkin avainvirran kokonaan eli 1500 tavua. Selvitettyä avainvirtaa voidaan käyttää uusien avainvirtojen selvittämiseen. Usein kun tukiasemalle lähettää salatun paketin yleislähetyksenä, tukiasema purkaa salauksen ja salaa sen uudella alustusvektorilla ennen paketin lähettämistä kaikille verkkonsa asemille. Näin hyökkääjä saa selville yhden uuden avainvirran ja voi samalla lailla jatkamalla hankkia kokoelman kaikista tukiaseman käyttämistä avainvirroista.

4 802.11i-tietoturvastandardi (WPA2)

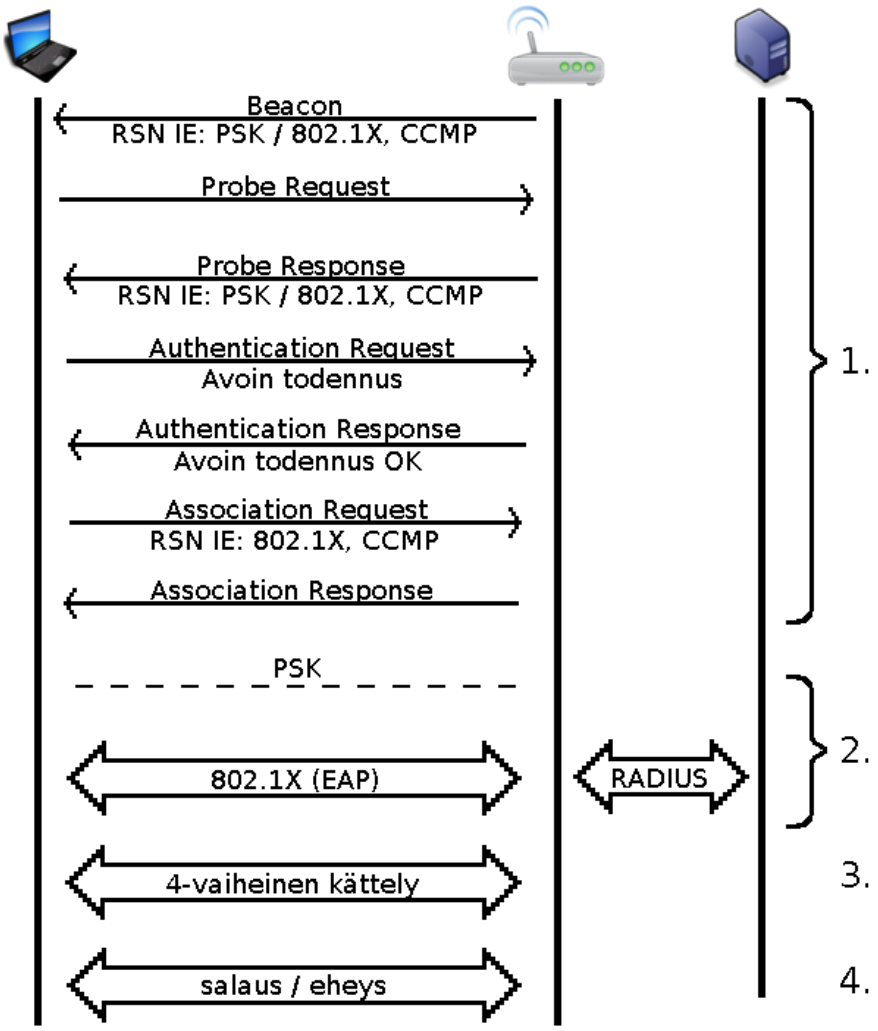
WEPissä ilmenneiden ongelmien myötä IEEE:ssä aloitettiin paremman tietoturvan takaavan standardin kehittäminen. Vuonna 2004 julkaistiin 802.11-standardin tietoturvaa käsittävä lisäosa 802.11i [IEE04a], joka myöhemmin liitettiin vuonna 2007 ilmestyneeseen 802.11-standardin viimeisimpään päivitykseen [IEE07]. Koska standardin valmistuminen kuitenkin kesti vuosia ja WEP oli osoittautunut epäluotettavaksi, alkoivat laitevalmistajat jo ennen 802.11i-standardin valmistumista ottaa siihen suunniteltuja ominaisuuksia käyttöön laitteissaan. Wi-Fi Alliance määritteli vuonna 2002 WPA-tietoturvaprotokollan (Wi-Fi Protected Access), johon sisällytettiin osa myöhemmin julkaistun 802.11i-standardin ominaisuuksista. Kun virallinen 802.11i-standardi ilmestyi, siitä käytettiin myös epävirallista WPA2-nimeä. WPA:n oli tarkoitus olla vain välivaihe ennen kuin virallinen 802.11i-tietoturvastandardi valmistuisi. WPA:n käyttöönoton piti olla mahdollista vanhemmissakin laitteissa vain ohjelmistoa päivittämällä. WPA2 käyttää muun muassa turvallisempaa salausta eikä sen käyttöönotosta selvitä enää pelkällä ohjelmistopäivityksellä, vaan se vaatii laitteiden uusimista.

WPA2:ssa tietoturvaa on parannettu neljällä osa-alueella: todennuksessa, avaintenhallinnassa, salauksessa sekä eheydentarkistuksessa. Näillä alueilla se korjaa WEPissä ilmenneitä vakavia tietoturvaongelmia. Todennustapoja on kaksi: pienissä verkoissa voidaan käyttää yhteiseen jaettuun avaimeen perustuvaa todennusta (Pre-Shared Key, PSK) ja isommissa sekä enemmän turvallisuutta vaativissa ympäristöissä käytetään 802.1X:ään [IEE04b] pohjautuvaa porttipohjaista pääsynhallintaa sekä EAP-protokollan (Extensible Authentication Protocol) [ABV04] tarjoamaa todennuskehystä eri todennustavoille. Tällöin varsinainen todennus voidaan myös ulkoistaa tukiaseaman ulkopuolelle, tyypillisesti RADIUS-palvelimelle (Remote Authentication Dial In User Service) [RWR00]. Avaintenhallintaa on parannettu usein tavoin. Käytössä on useampia avaimia ja niiden vaihtamiseen on automaattinen järjestelmä. WEPissä peruspalveluryhmän jokainen laite käytti samaa WEP-avainta, mutta WPA:ssa ja WPA2:ssa jokainen asema käyttää eri salausavainta. Lisäksi tukiaseman ja aseman välinen salausavain on eri kuin saman peruspalveluryhmän sisällä käytettävä monilähetysavain. Avaintenhallinassa WPA ja WPA2 eivät eroa toisistaan, paitsi että WPA:ssa käytetään eri avainta viestien salaamiseen ja eheydentarkistuksen muodostamiseen, mutta WPA2:ssa näitä ei ole erotettu. Todennuksen suhteen WPA ja WPA2 eivät eroa toisistaan, mutta viestien salauksen ja eheydentarkistuksen ne toteuttavat eri lailla. WPA:ssa näistä huolehtii TKIP-protokolla (Temporal Key In-

tegrity Protocol) ja WPA2:ssa CCMP-protokolla (Counter mode with CBC-MAC Protocol). TKIP on kuitenkin vapaaehtoinen myös WPA2-toteutuksissa takaamassa yhteensopivuutta vanhempien laitteiden kanssa. TKIP:ssä käytetään edelleen samaa RC4-salausalgoritmia viestien salaamiseen kuin WEPissäkin, mutta salausprotokollaa on kehitetty, eikä se ole enää yhtä pahasti alttiina WEPissä toimineille hyökkäyksille. TKIP:ssä viestien eheydestä vastaa Michael-algoritmi, joka soveltuu siihen paljon paremmin kuin WEPissä käytetty CRC32, mutta siinäkin on omat ongelmansa. CCMP:ssä on salauksessa hylätty kokonaan RC4:n käyttö ja korvattu se laskennallisesti huomattavasti vaativammalla ja entistäkin turvallisemmalla lohkosalaukseen perustuvalla AES-algoritmilla (Advanced Encryption Standard). CCMP:ssä myös eheydentarkistusta on parannettu entisestään.

Turvallisen lähiverkon (Robust Secure Network, RSN) muodostaminen koostuu neljästä vaiheesta ja sen muodostamista on havainnollistettu kuvassa 13. Vaiheessa yksi asema ja tukiasema muodostavat alustavan yhteyden ja neuvottelevat vaiheessa neljä käytettävästä todennus- ja salaustavasta. Aluksi tukiasema joko kuuluttaa itsestään ympäristöön beaconilla tai asema lähettää probe requestin, johon tukiasema vastaa probe responsella. Sekä beaconin että probe responsen mukana tukiasema lähettää RSN IE -tietueen (RSN Information Element), jossa se kertoo millaisia todennus- ja salaustapoja se tarjoaa verkossaan. Kuvan esimerkissä tukiasema tarjoaa todennustavaksi joko PSK:ta tai 802.1X:ää ja sekä täsmälähetysten että ryhmälähetysten salaukseen CCMP:tä. Tämän jälkeen on alustava todennus authentication-kehyksillä. Käytössä on pelkkä avoin todennus, jossa tukiasema hyväksyy todennuspyynnön ehdoitta. Vaihe yksi päättyy aseman liittämisellä tukiasemaan. Aseman on association requestissaan ilmoitettava, mitä tukiaseman aiemmin joko beaconissa tai probe responsessa tarjoamista todennus- ja salaustavoista se haluaa käyttää. Kuvan esimerkissä asema on valinnut kahdesta valinnaisesta todennustavasta 802.1X:n ja salaukseksi ainoan tukiaseman tarjoaman vaihtoehdon eli CCMP:n.

Neuvotellessaan vaiheessa yksi myöhemmin käytettävästä salaus- ja eheystavasta asema ja tukiasema pyrkivät valitsemaan turvallisimman mahdollisen tavan, jonka molemmat tuntevat. Turvallisinta on sallia tukiaseman verkossa käytettävän vain RSN:ää, mutta tarve yhteensopivuuteen vanhojen laitteiden kanssa voi vaatia myös turvattomamman WEPin käyttämistä. Tällöin voidaan käyttää väliaikaista turvallisuusverkkoa (Transient Security Network, TSN), jolloin tukiaseman verkossa voi olla sekä RSN:ää että WEPiä käyttäviä laitteita. TSN:n käyttäminen antaa kuitenkin hyökkääjälle mahdollisuuden huijata asemaa ja tukiasemaa valitsemaan WEPin, vaikka ne tuntisivatkin turvallisemman RSN:n [HeM05]. Koska vaiheessa yksi lähe-



Kuva 13: Turvallisen lähiverkon muodostamisen vaiheet

tettyjä kehyksiä ei ole vielä mitenkään salattu, voi hyökkääjä käyttää välimieshyökkäystä (man-in-the-middle attack) ja muuttaa tukiaseman lähettämiä beaconeita tai probe responseja, niin että tukiasema näyttäisi niissä tuntevan vain WEPin. Asema valitsee ainoan tarjotun vaihtoehdon ja salaus- ja eheystavaksi tulee WEP, vaikka molemmat osapuolet tuntisivat turvallisemmankin vaihtoehdon. Hyökkääjän on paljon helpompi hyökätä WEPillä salattua liikennettä vastaan aiemmin kuvatuin keinoin kuin huomattavasti turvallisempaa WPA:ta tai WPA2:ta vastaan, joita olisi käytetty ilman hyökkääjän häirintää. Ongelman helppona ratkaisuna on estää TSN ja sallia vain RSN, mutta tämä ei ole ehkä aina mahdollista yhteensopivuusongelmien vuoksi. Jos TSN:ää on käytettävä, voidaan tärkeimmät lähiverkon palvelut

rajata asemille, jotka käyttävät RSN:ää ja sallia WEPiä käyttäville asemille esimerkiksi vain yhteys internetiin.

RSN:n muodostamisen vaiheessa kaksi tapahtuu vasta varsinainen todennus. Vaiheen yksi neuvottelujen perusteella todennus tehdään joko käyttäen PSK:ta tai 802.1X:n porttipohjaista pääsynhallintaa yhdessä EAP-protokollan todennuskehyksen kanssa. Tällöin varsinainen todennus tapahtuu ulkoisella RADIUS-todennuspalvelimella tukiaseman toimiessa vain viestien välittäjänä aseman ja todennuspalvelimen välillä. Käytettäessä PSK:ta asema ja tukiasema eivät vaihda todennusvaiheessa mitään tietoja, mutta jos osapuolilla ei olekaan samaa avainta, tämä paljastuu vaiheen kolme 4-vaiheisessa kättelyssä. Todennusta käsitellään tarkemmin seuraavassa luvussa.

Onnistuneen todennuksen jälkeen RSN:n muodostamisen vaiheessa kolme asema ja tukiasema varmistavat, että kummallakin osapuolella on oikea avain. PSK:ssahan avaimet oli syötetty etukäteen laitteisiin ja 802.1X:ää käytettäessä todennus tapahtui ulkoisella palvelimella. 4-vaiheisessa kättelyssä tapahtuva tarkistus on yksi tärkeimmistä toimenpiteistä muodostettaessa RSN:ää. Siinä luodaan lisäksi tarvittavia avaimia sekä kahden osapuolen että monilähetysten ja ryhmälähetysten turvallista liikennöintiä varten. Vaihe kolme koostuu useista lähetyksistä ja niitä on käsitelty tarkemmin luvussa 4.2.

Lopuksi, kun osapuolet on luotettavasti todennettu ja on luotu tarvittavat avaimet viestien salaamiseksi ja eheyden takaamiseksi, voidaan niitä käyttämällä vaiheessa neljä salata kaikki datakehykset. Datan salaaminen ja eheydentarkistus tehdään WPA:ssa TKIP:llä, jota käsitellään luvussa 4.3 sekä WPA2:ssa CCMP:llä, jota käsitellään luvussa 4.4.

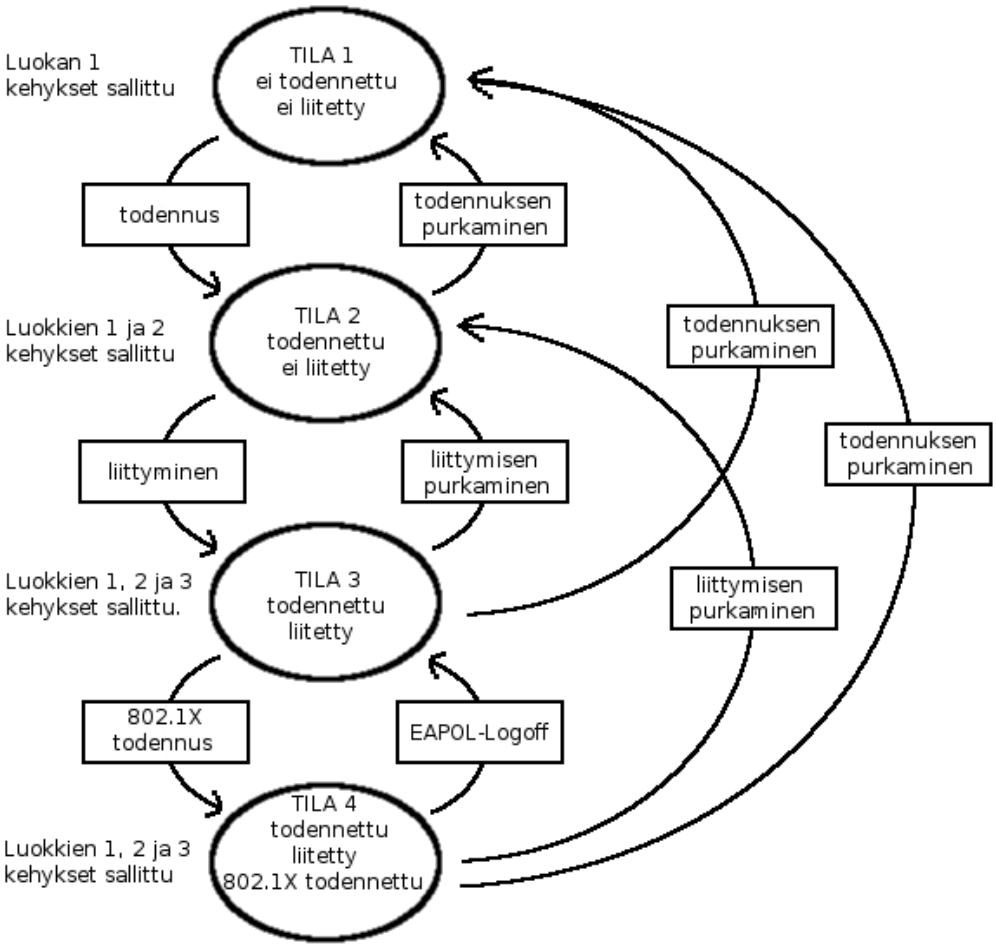
4.1 Todentaminen WPA:ssa ja WPA2:ssa

RSN:n muodostamisessa voidaan käyttää todennusvaiheessa kahta tapaa, joista PSK sopii pieniin verkkoihin, joissa laitteita on vähän. Tällöin sama avain on syötettävä ja tarvittaessa vaihdettava käsin kaikkiin verkon laitteisiin. Tältä osin PSK muistuttaa WEPissä käytössä ollutta todennusta. Toisaalta PSK:ta käytettäessä todennus ei tapahdu samanlaisella haaste-vaste-periaatteella kuin WEPissä, vaan asema ja tukiasema muodostavat syötetyn avaimen ja muiden tietojen perusteella uuden avaimen. Todennus tapahtuu varmistamalla, että molemmat ovat onnistuneet muodostamaan saman avaimen. Lisäksi käytössä on useampia avaimia eikä vain yhtä

ainutta, jota käytetään sekä todennuksessa että myöhemmin viestien salauksessa. 256-bittinen PSK-avain muodostetaan joko satunnaislukugeneraattorilla tai käyttäjä syöttää salasanan, josta yhdessä verkon SSID-tunnuksen kanssa muodostetaan 256-bittinen PSK-avain. PSK-avain toimii parittaisena pääavaimena (Pairwise Master Key, PMK) tukiaseman ja aseman välillä. Vaikka PSK-avain voi olla eri tukiaseman ja jokaisen siihen liittyneen aseman välillä, niin usein käytetään vain yhtä ja samaa avainta koko verkossa. Tästä seuraa, että saman verkon sisällä asema voi tietyissä tilanteissa purkaa salauksen muiden asemien lähettämistä kehyksistä. PSK:ta käytettäessä on myös valittava tarpeeksi vahva salasana, sillä se on heikoin lenkki todennuksessa ja vaikuttaa myös myöhemmin muodostettaviin salaus- ja eheysavaimiin. Suositeltavaa olisi käyttää vähintään 20 merkin mittaista salasanaa PSK:ta käytettäessä [IEE07].

PSK:ta parempi ja suositeltavampi toddennustapa RSN:ssä perustuu 802.1X:n mukaiseen porttipohjaiseen pääsynhallintaan, EAP-todennuskehyksen käyttämiseen todennusprosessin aikaisten viestien välittämisessä sekä ulkoisen RADIUS-palvelimen käyttöön. Tämä tapa sopii erityisesti isoihin verkkoihin, joissa laitteita on paljon ja avaimen asettaminen käsin laitteisiin muodostuisi epäkäytännölliseksi sekä tavoiteltaessa mahdollisimman hyvää tietoturvaa. Käytettäessä todennuksessa 802.1X:ää laajenee aiemmmin kuvassa 10 esitetty tilakaavio yhdellä tilalla kuvan 14 mukaiseksi.

802.1X on IEEE:n määrittelemä standardi lähiverkossa (sekä Ethernet että WLAN) toimivasta porttipohjaisesta pääsynhallinnasta. Siinä esitellään todennusprosessin kolme osapuolta: asiakas (supplicant), todentaja (authenticator) ja todennuspalvelin (Authentication Server, AS). Asiakas on asema, joka haluaa liittyä todentajan eli tukiaseman verkkoon. Sitä ennen asiakkaan on kuitenkin todennettava itsensä todentajalle luotettavan kolmannen osapuolen eli todennuspalvelimen avulla. Todennuspalvelin voi olla tukiaseman yhteydessäkin, mutta usein se on langattoman verkon ulkopuolella, kaapeliverkossa oleva erillinen palvelin, useimmiten RADIUS-palvelin. Todentaja muodostaa kaksi virtuaaliporttia: kontrolloidun ja kontrolloimattoman. Kaikki kontrolloidun portin kautta tulleet paketit ohjataan todennuspalvelimelle eikä niitä päästetä muualle todentajan verkkoon tai jakelujärjestelmään. Kontrolloimattomasta portista tulevat paketit ohjataan vapaasti eteenpäin niissä olevan kohdeosoitteen mukaisesti. Jokainen uusi verkkoon haluava asiakas liitetään aluksi todentajan kontrolloituun porttiin ja vasta onnistuneen todennuksen jälkeen se liitetään kontrolloimattomaan porttiin. Todennusta ei siis tee todentaja, vaan todennuspalvelin. Todentaja toimii niiden välissä vain liikenteenohjaajana ja odottaa



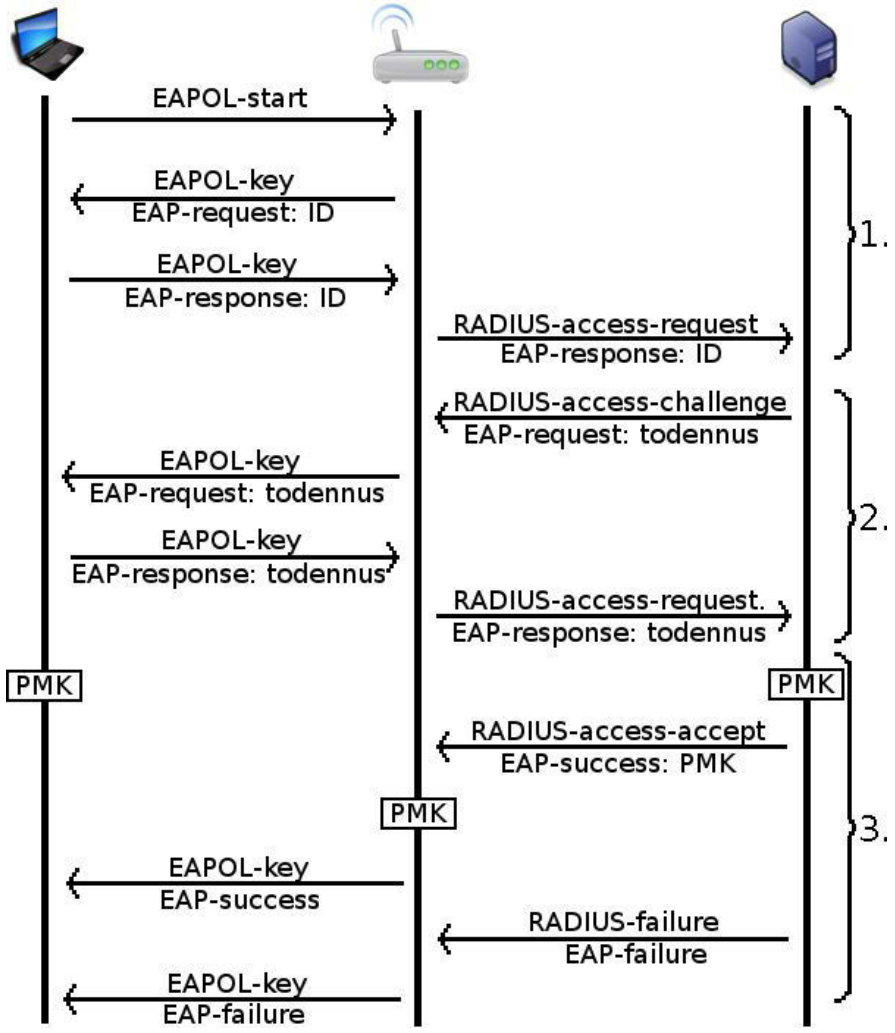
Kuva 14: Todennuksen, liittymisen ja kehysten väliset riippuvuudet käytettäessä 802.1X:ää

todennuspalvelimelta ilmoitusta todennuksen onnistumisesta tai epäonnistumisesta.

802.1X käyttää EAP-todennuskehystä, joka määrittelee tavan miten asema, tukiasema ja todennuspalvelin keskustelevat todennuksen aikana [ABV04]. Se toimii sekä Ethernetissä että WLANissa välittömästi siirtoyhteyskerroksen yläpuolella. EAPin tarkoitus on paketoida protokollapinon ylemmillä tasoilla toimivat todennusprotokollat yhteiseen kehykseen, joka on riippumaton alemman tason toteutuksesta. Näin ollen sillä voidaan kuljettaa paketteja WLANin ja Ethernetin välillä. EAP-kehykset ovat yksinkertaisia ja niitä on vain neljää eri tyyppiä: request, response, success ja failure. EAP-kehyksen otsakkeessa kerrotaan tyypin lisäksi tunniste ja kehyksen pituus. Otsakkeen jälkeen tulee vielä varsinainen data. Tunnistetta käytetään, yh-

distämään response oikeaan requestiin niissä olevan saman tunnisteen avulla. Koska EAP-kehykset ovat hyvin yksinkertaisia, voidaan lisäksi käyttää aseman ja tukiaseman välillä myös 802.1X:ssä määriteltyä EAPOL-kehystä (EAP over LAN). Se muistuttaa EAPia, mutta tuo muutaman uuden kehystyypin: EAPOLstartin, joka aloittaa EAP-viestinnän, EAPOL-logoffin, joka lopettaa EAP-viestinnän ja purkaa aiemman muodostetun todennuksen sekä EAPOL-keyin, jossa on datana todennuksessa tarvittavaa lisätietoa.

Kuvassa 15 on havainnollistettu jokaisen todennusprosessin perusviestienvaihtoa, johon eri todennusmetodit tuovat oman vivahteensa. Todentamisessa on kolme osa-



Kuva 15: Todennuksen vaiheet käytettäessä 802.1X:ää, EAPia ja RADIUSta

puolta: asema, tukiasema ja todennuspalvelin, tässä tapauksessa RADIUS-palvelin. Aseman ja tukiaseman välillä kulkee EAPOL-key-kehyksiä joihin on kapseloitu EAP-kehys. Tukiaseman ja todennuspalvelimen välillä EAP-kehykset on kapseloitu RADIUS-protokollan mukaisiin kehyksiin. Todennusprosessi koostuu kolmesta vaiheesta. Ensimmäinen vaihe alkaa joko aseman lähettämällä EAPOL-startilla, johon tukiasema vastaa EAP-requestilla, tai suoraan tukiaseman asemalle lähettämällä EAP-requestilla. Siinä tukiasema pyytää asemalta sen tunnistetietoja. Asema lähettää EAP-responsessa tunnistetietonsa, jotka tukiasema välittää edelleen todennuspalvelimelle kapseloituna RADIUS-access-requestiin. Vaiheessa kaksi tapahtuu varsinainen todennus, jossa todennuspalvelin lähettää asemalle todennusviestejä, haasteita, joihin asiakas vastaa. Välissä oleva tukiasema vain välittää viestejä aseman ja todennuspalvelimen välillä. Todennuspalvelimen ja aseman välinen kysymys-vastausvuoropuhelu voi tapahtua montakin kertaa, riippuen käytetystä todennusmetodista, joten vaihe kaksi voi toistua useitakin kertoja.

Lopulta vaiheessa kolme todennuspalvelin joko hyväksyy tai hylkää aseman todennuksen. Onnistuneen todennuksen jälkeen sekä asemalla että todennuspalvelimella on tarpeeksi tietoa, jotta ne voivat muodostaa PMK:n. Todennuspalvelin lähettää todennuksen onnistumisesta viestin tukiasemalle sekä kertoo samalla juuri muodostamansa PMK:n. Tukiasema lähettää vielä asemalle EAP-successilla kuittauksena todennuksen onnistumisesta. Jos todennuspalvelin hylkää todennuksen, lähettää tukiasema tiedon saatuaan tästäkin kuittauksen asemalle EAP-failurella. Tällöin RSN:n muodostaminen ja aseman yritys päästä tukiaseman verkkoon päättyy siihen.

Koska EAP ei vielä tee varsinaista todennusta, vaan määrittelee vain EAP-kehyksen muodon sekä tavan jolla todennuksen osapuolet keskustelevat, se tarvitsee vielä varsinaisen todennuksen tekijän. EAP tukee useita erilaisia todennusmetodeja, jotka suorittavat todennuksen. Varsinainen todennus voi pohjautua muun muassa salasanan, varmenteen, älykortin tai kännykän SIM-kortin avulla saatuun varmistukseen vastapuolen identiteetistä. Koska WLANeja käytetään hyvin erilaisissa ympäristöissä ja olosuhteissa, on varsinainen todennusmetodi jätetty määrittelemättä ja sallittu useita tapoja, jotta kuhunkin tilanteeseen voitaisiin valita sopivin. Seuraavaksi esitellään lyhyesti tunnetuimpia ja tärkeimpiä EAP-metodeja, joita on käsitelty perusteellisesti Hardjonon ja Dondetin kirjassa [HaD05].

EAP-MD5 on varhaisimpia EAP-metodeita [ABV04]. Se on suhteellisen turvaton, mutta helposti käyttöönotettavissa oleva EAP-metodi. Todennuksen toisessa vai-

heessa todennuspalvelin lähettää asemalle haasteena satunnaisen merkkijonon. Asiakas muodostaa tästä haasteesta sekä käyttäjän syöttämästä salasanasta MD5-tiivisteen ja lähettää sen todennuspalvelimelle. Todennuspalvelin joko muodostaa itse saman tiivisteen ja vertaa sitä aseman lähettämään tiivisteeseen tai todennuspalvelimelle on tallennettu jokaista sallittua asemaa kohti kutakin haastetta vastaavat oikeat MD5-tiivisteet. Salasanoja ei siis missään vaiheessa liikuteta verkon yli selväkielisinä. EAP-MD5:ttä ei kuitenkaan ole sellaisenaan enää käytössä, sillä sen haitat ovat ilmeisiä. Pahimpana puutteena todennus ei ole molemminpuolista. Ainoastaan asema todentaa itsensä todennuspalvelimelle, mutta todennuspalvelin ei todenna itseään mitenkään asemalle.

Netscape kehitti 90-luvun puolessa välissä SSL-protokollan (Secure Sockets Layer) nettiselaimiinsa luomaan turvallisen yhteyden selaimen ja palvelimen välille. IETF (Internet Engineering Task Force) käytti SSL:n 3.0-versiota pohjana kehittäessään TLS-protokollaa (Transport Layer Security) [DiR08]. TLS tarjoaa luotettavaa julkisiin avaimiin ja varmenteisiin perustuvaa todennusta, datan eheyttä sekä mahdollisuutta neuvotella myöhemmin käytettävästä salausavaimesta. EAP-TLS pohjautuu TLS:ään ja tarjoaa erittäin luotettavan todennuksen [SAH08]. Todennuksen vaiheessa kaksi TLS-kehykset kapseloidaan EAP-kehyksiin. Asema ja todennuspalvelin vaihtavat useita viestejä ja kumpikin osapuoli verifioi niissä olevan digitaalisen allekirjoituksen aitouden varmenteellaan. Vasta kun osapuolet ovat todentaneet itsensä, muodostetaan niiden välille turvallinen kanava, jossa voidaan vaihtaa avainten luomiseen tarvittavia kryptografisia tietoja. Vaikka EAP-TLS tarjoaakin luotettavaa todennusta, niin erityisesti asiakkaaltakin vaadittava sertifikaatti hankaloittaa sen käyttämistä.

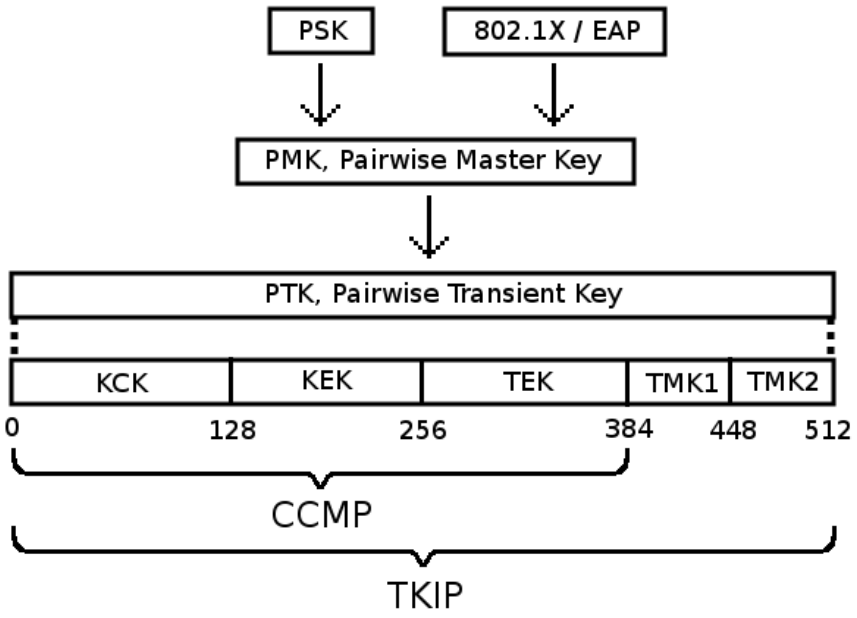
EAP-TTLS (Tunneled TLS) on EAP-TLS:n laajennus [FuB08]. Vaikka EAP-TLS onkin turvallinen, niin aseman identiteetti paljastui sekä selväkielisenä lähetetyn varmenteen johdosta että todennuksen alussa vaiheessa yksi (katso kuva 15 sivulla 47) aseman lähettäessä todentajalle tunnistetietonsa EAP-kehyksessä. EAP-TTLS:ssä aseman todentaminen todennuspalvelimelle sertifikaatin avulla on valinnainen, pelkkä todennuspalvelimen onnistunut todennus asemalle riittää niiden välisen turvallisen TLS-tunnelin muodostamiseen. Muodostettua tunnelia käytetään aseman todentamiseen todennuspalvelimelle. Todennustavan valinta on hyvin vapaata, voidaan käyttää esimerkiksi salasanaa tai EAP-MD5:ttä, joka sinänsä on turvaton todennustapa, mutta sen käyttäminen TLS-tunnelissa on turvallista. Koska vain todennuspalvelin tarvitsee varmenteen on EAP-TTLS:n hallinnointi helpompaa kuin EAP-TLS:n. Lisäksi EAP-TTLS:ässä aseman ei tarvitse lähettää tarkkoja tunniste-

tietojaan heti alussa. Pelkkä NAI-muotoinen (Network Access Identifier) [ABA05] käyttäjätunnus riittää. Vasta kun turvallinen TSL-tunneli on muodostettu, lähettää asema täydelliset tunnistetietonsa.

EAP tarjoaa siis yksinkertaisen todennusalustan, jonka päälle voidaan lisätä kuhunkin ympäristöön sopiva EAP-todennusmetodi. Tarkan todennusmetodin määrittelemättä jättäminen EAPissa mahdollistaa useisiin eri tarpeisiin parhaiten soveltuvan todennusmetodin valinnan. Useat todennusmetodit pohjautuvat TLS:ään ja kaikkein turvallisimman vaihtoehdon tarjoaa EAP-TLS. Sen käyttöönotto on kuitenkin usein turhan hankalaa molemminpuolisine varmennevaatimuksineen. EAP-TTLS tarjoaa joustavampaa ja suurempaa valikoimaa asiakkaan todentamiseen. Myös muitakin kuin edellä käsiteltyjä EAP-metodeja on runsaasti [DCA07].

4.2 Nelivaiheinen kättely

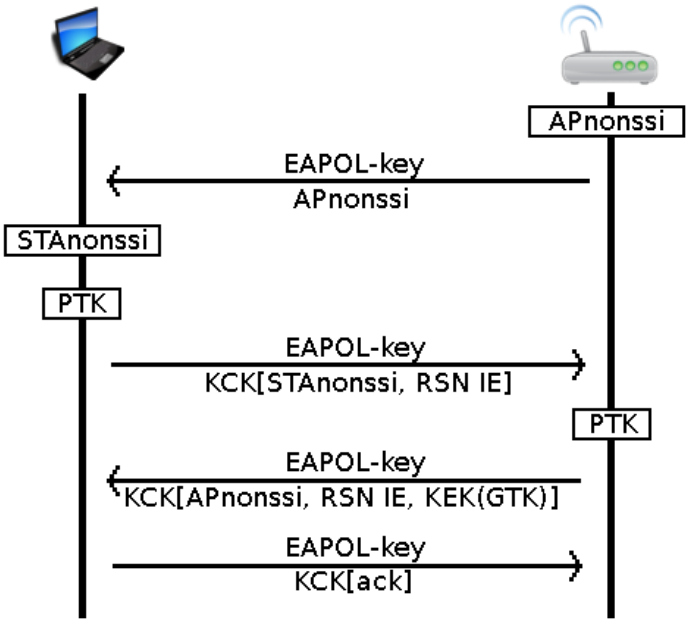
Nelivaiheinen kättely eli neljästä aseman ja tukiaseman välisestä lähetyksestä koostuva keskustelu on yksi tärkeimmistä vaiheista muodostettaessa RSN:ää. Tätä edeltävän todennusvaiheen eräs tarkoitus oli luoda parittainen pääavain, PMK. Onnistuneen todennusvaiheen jälkeen sekä asema että todennuspalvelin muodostivat vaihtamiensa tietojen perusteella PMK:n. Todennuspalvelin lähetti sen myös tukiasemalle, jotta asema ja tukiasema voivat nelivaiheisessa kättelyssä varmistaa sen, että kummallakin osapuolella on sama PMK ja muodostaa myöhemmin tarvittavia salaus- ja eheysavaimia. Kuvassa 16 on havainnollistettu RSN:ssä käytettäviä avaimia ja niiden hierarkiaa. Aluksi muodostetaan parittainen 256-bittinen pääavain (PMK). Jos todennus perustuu ennalta jaettuun avaimeen (PSK), muodostetaan PMK siitä ja verkon SSID-tunnuksesta. Jos todennus tapahtuu 802.1X- ja EAP-protokollan avulla, muodostavat asema ja todennuspalvelin onnistuneen todennuksen jälkeen PMK:n. Tällöin PMK:lle annetaan myös jokin elinaika, jonka kuluttua se muodostetaan uudestaan. PMK:ta ei koskaan käytetä salaukseen tai eheyden tarkistamiseen, vaan siitä muodostetaan nelivaiheisessa kättelyssä tilapäinen pariavain (Pairwise Transient Key, PTK). Sillekin on määrätty elinaika, joka ei voi olla PMK:n elinaikaa pitempi. PTK:n elinajan loputtua se on muodostettava uudestaan aseman ja tukiaseman välisessä nelivaiheisessa kättelyssä. PTK koostuu viidestä osasta, joista osaa tarvitaan vain väliaikaisesti nelivaiheisessa kättelyssä ja osaa myöhemmin varsinaisen datan salauksessa ja eheydentarkistuksessa. 128-bittistä KCK:ta (EAPOL-Key Confirmation Key) ja KEKiä (EAPOL-Key Encryption Key) käytetään vain nelivaiheisessa kättelyssä. MIC-eheydentarkistuksessa käytetään KCK:ta ja KEKiä käyte-



Kuva 16: Pariavainten hierarkia ja koostumus RSN:ssä

tään joidenkin EAPOL-key-kehysten salaamiseen. Väliaikaista salausavainta, TEKiä (Temporal Encryption Key), käytetään nelivaiheisen kättelyn jälkeen datan salaamiseen. CCMP:ssä PTK on vain 384 bittiä ja koostuu KCK:sta, KEKistä ja TEKistä. Käytettäessä TKIP:tä PTK on 512 bittiä ja koostuu lisäksi kahdesta 64-bittisestä väliaikaisesta MIC-eheydentarkistuksessa käytettävästä avaimesta (Temporal MIC Key, TMK), joita käytetään nelivaiheisen kättelyn jälkeen datan eheyden varmistamiseen. TMK-avaimia on kaksi, koska kumpaankin suuntaan on omansa: TMK1:tä käytetään tukiaseman lähettäessä datakehyksen ja TMK2:ta aseman lähettäessä datakehyksen.

Nelivaiheisen kättelyn vaiheet on esitetty kuvassa 17. Aluksi tukiasema muodostaa satunnaisen luvun, niin sanotun nonssin (APnonssi), jonka se lähettää asemalle EAPOL-key-kehyksessä. Tätä kehystä ei ole mitenkään salattu. Saatuaan kehyksen asema tarkistaa aluksi, että siinä oleva EAPOL-key-kehykseen kuuluva avaimen toistolaskuri (key replay counter) näyttää oikein. Laskuri nollattiin PMK:n muodostamisen jälkeen. Laskurin arvoa kasvatetaan jokaisessa kehyksessä yhdellä ja virheellinen laskurin arvo johtaa kehyksen hylkäämiseen. Saatuaan kelvollisen kehyksen asema muodostaa vuorostaan nonssin (STAnonssi), jonka jälkeen asemalla on kaik-



Kuva 17: Nelivaiheinen kättely

ki tarvittava tieto PTK:n muodostamiseksi. Siihen tarvitaan aiemmin muodostettu PMK, sekä aseman että tukiaseman MAC-osoitteet, APnonssi ja STAnonssi. Tämän jälkeen asema lähettää tukiasemalle EAPOL-key-kehyksessä muodostamansa STAnonssin sekä täsmälleen saman RSN IE:n minkä lähetti aiemmin tukiasemalle association requestissa RSN:n muodostamisen vaiheessa yksi (katso kuva 13 sivulla 43). Kehystä ei salata mitenkään, mutta ennen lähetystä sille lasketaan MIC-eheydentarkistus käyttäen KCK:ta. Kehyksen saatuaan tukiasema tarkistaa aluksi toistolaskurin arvon. Sen ollessa oikea tukiasema voi muodostaa saman PTK:n kuin asema hetkeä aiemmin. PTK:ssa olevan KCK:n avulla tukiasema voi muodostaa MIC-arvon, jonka on täsmättävä kehyksessä olleeseen MIC-arvoon, jolloin varmistutaan ettei pakettia ole muutettu matkan varrella. Nyt tukiasema tietää asemankin tietävän oikean PTK:n. Lopuksi tukiasema tarkistaa, että kehyksessä ollut RSN IE on oikea. Toinen tukiaseman nelivaiheisessa kättelyssä asemalle lähettämä kehys sisältää saman RSN IE:n kuin tukiaseman asemalle aiemmin RSN:n muodostamisen vaiheessa yksi lähettämässä kehyksessä (katso kuva 13 sivulla 43). Lisäksi kehyksessä lähetetään APnonssi uudestaan sekä ryhmälähetysten salaamiseen tarvittava väliaikainen ryhmäavain (Group Temporal Key, GTK), joka salataan KEKillä. Tukiasema muodostaa GTK:n 256-bittisestä pääryhmäavaimesta (Group Master Key,

GMK), tukiaseman MAC-osoitteesta sekä muodostamastaan satunnaisesta luvusta. GTK jakaantuu kahteen osaan ryhmäsalausavaimeen (Group Encryption Key, GEK) ja ryhmäeheysavaimeen (Group Integrity Key, GIK). TKIP:ssä GTK koostuu molemmista osista, mutta CCMP:ssä vain GEKistä. Lopuksi muodostetaan koko EAPOL-kehyksestä MIC-eheydentarkistus käyttäen KCK:ta. Viimeisenä nelivaiheisessa kättelyssä asema lähettää vielä kuittauksen tukiasemalle. Tämän jälkeen kumpikin osapuoli käyttää keskinäisessä viestinnässään TKIP:ssä juuri alustettuja salaus- ja eheydentarkistusavaimia, TEKiä ja TMK:ta ja CCMP:ssä pelkästään TEKiä.

Ennen todennusvaihetta asema liitettiin tukiaseman virtuaaliseen kontrolloituun porttiin, jonka kautta liikenne sallitiin vain todennuspalvelimelle, ei muualle verkkoon. Vasta nyt, kun todennus on suoritettu ja tarvittavat liikennöinnissä käytettävät avaimet ovat kummankin osapuolen tiedossa ja molemmat ovat varmistuneet, että kummallakin osapuolella on samat oikeat avaimet, liittää tukiasema aseman kontrolloimattomaan porttiin ja päästää sen liikennöimään muualle verkkoon.

Nelivaiheisen kättelyn jälkeen voidaan vielä suorittaa ryhmäavaimen kättely. GTK:ta tarvitaan, kun aseman lähettämä paketti on osoitettu useammalle saman verkon asemalle. Ryhmäavaimen kättelyssä tukiasema muodostaa uuden GTK:n ja lähettää sen asemille. Kättely tehdään aina, kun asema poistuu verkosta tai todennetaan tukiasemalle.

PSK on 802.1X:ään ja EAPiin perustuvaa todennusta turvattomampi. Oletetaan, että tukiasema käyttää vain PSK-todennusta ja sen verkkoon on liittynyt aiemmin asema A. Myöhemmin asema B liittyy samaan verkkoon suorittamalla normaalin nelivaiheisen kättelyn. Koska asemilla on sama PSK:sta johdettu PMK, niin kuunneltuaan nelivaiheisen kättelyn kaksi ensimmäistä viestiä, joita siis ei ole salattu mitenkään, asema A saa selville sekä APnonssin että STAnonssin ja voi muodostaa saman PTK:n kuin mitä asema B ja tukiasema muodostavat nelivaiheisessa kättelyssä. Tämän jälkeen asema A voi purkaa kaikki tukiaseman ja aseman B väliset salatut kehykset. Tarvittaessa erityisen turvallista verkkoa ei ole suositeltavaa käyttää PSK:ta [IEE07].

GTK:lla salatut viestit eivät anna riittävää suojaa MAC-osoitteen väärentämistä vastaan. Ahmad esitti vuonna 2010 Hole196-nimeä kantavan hyökkäyksen, joka osoittaa GTK:n heikkoudet [Ahm10]. Hyökkäys on saanut nimensä 802.11-standardin [IEE07] sivulla 196 olevasta GTK:n määrittelystä. GTK:ta käytetään vain tukiaseman lähettämien ryhmälähetysten (yleis- tai monilähetys) salaamiseen. Jokainen

asema saa nelivaiheisessa käyttelyssä tukiaseman muodostaman GTK:n, jolla voi purkaa sillä salatut kehykset. Verkkoon onnistuneesti todennettu hyökkääjä voi kuitenkin käyttää GTK:ta kehyksen salaamiseen ja lähettää muille asemille väärennettyjä ryhmälähetyksiä, jotka näyttävät tulevan tukiasemalta. Hyökkääjä voi esimerkiksi lähettää väärennetyn ARP-requestin, jossa kyselyn lisäksi väärentää oman MAC-osoitteensa olevan verkon yhdyskäytävän osoite. Tämän jälkeen uhrin lähiverkon ulkopuolelle lähettämät paketit kulkevatkin tukiaseman kautta suoraan hyökkääjälle.

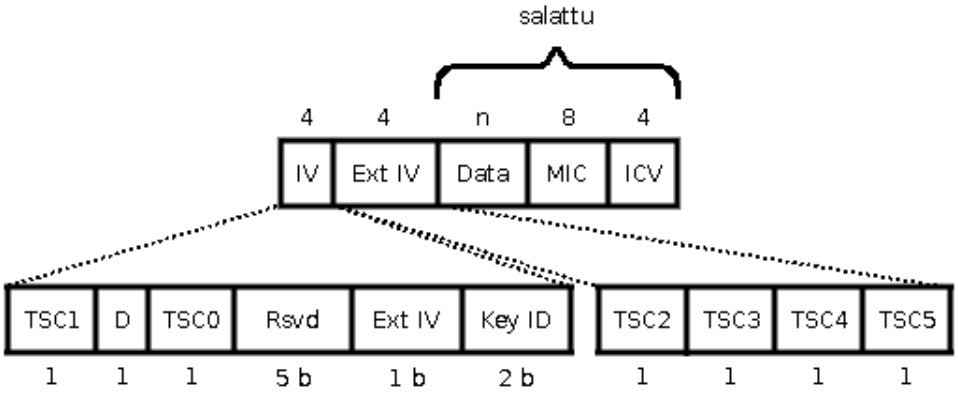
Hole196:tta voidaan jossain määrin torjua muutamin keinoin [Ahm10]. Asema voi tarkkailla ARP-taulunsa tietojen muuttumista ja erityisesti yhdyskäytävän osoitteen muuttumista. Näin mahdollinen hyökkäys voitaisiin ainakin havaita. Toinen keino olisi estää asemien välisten kehysten välittäminen verkossa eli tukiasema ei lähettäisi eteen päin kehystä, jonka lähettäjä ja vastaanottaja kuuluisivat samaan tukiaseman ylläpitämään peruspalveluryhmään. Kolmas torjuntakeino olisi hylätä kokonaan GTK:n käyttäminen. Tällöin taaksepäin yhteensopivuuden vuoksi tukiasema joutuisi kuitenkin yhden yhteisen GTK:n sijasta muodostamaan jokaiselle asemalle oman yksilöllisen GTK:n. Ryhmälähetyksen sijasta tukiasema lähettäisi kehyksen erikseen jokaiselle vastaanottajalle sen tuntemalla yksilöllisellä GTK:lla salattuna. Tämä luonnollisesti hidastaisi verkon suoritustehoa. Myös niin sanotuilla tunkeilijan havaitsemisjärjestelmillä (Intrusion Detectection System, IDS) voidaan yrittää havaita hyökkäys. Vaikka Hole196 onkin merkittävä uhka, niin on huomattava, että se ei paljasta hyökkääjälle mitään salausavaimia ja hyökkäyksen voi tehdä vain verkkoon jo aiemmin onnistuneesti todennettu asema, mikä asettaa luotettavalle todennukselle entistäkin enemmän painoarvoa.

4.3 TKIP

TKIP:tä käytetään vain WPA:ssa ja se koostuu joukosta algoritmeja, jotka pyrkivät korjaamaan WEPissä ilmenneitä ongelmia. Se on kuitenkin tarkoitettu vain väliaikaiseksi ratkaisuksi ja suositeltavaa on käyttää WPA2:ta, jossa TKIP:n sijasta käytetään CCMP:tä. WPA:n on tarkoitus toimia vanhoissakin WEPin sisältävissä laitteissa ja sen käyttöönoton on onnistuttava pelkällä ohjelmistopäivityksellä. Siksi WPA ei voi käyttää mitään kovin monimutkaisia ja raskaita algoritmeja korjatessaan WEPissä ilmenneitä ongelmia. TKIP tuo kuitenkin useita parannuksia: alustusvektori on pitempi ja sitä käytetään järjestelmällisemmin, RC4-salausalgoritmille syötettävä merkkijono muodostetaan paremmin ja eheydentarkistukseen käytetään

kehittyneempää Michael-algoritmia.

TKIP pohjautuu WEP-salausprotokollaan, mutta parantaa sitä korjaten siinä ilmenneitä puutteita. Kuvassa 18 on esitetty MAC-kerroksen datakehyksen datakentän rakenne WPA:ssa. Verrattaessa rakennetta WEPiin (kuva 11 sivulla 32) ovat



Kuva 18: Datakehyksen datakenttä WPA:ssa

selvimpinä eroina laajennettu alustusvektorikenttä (extended IV, Ext IV) ennen salatekstiä sekä MIC-eheydentarkistus (Message Integrity Check) datan ja ICV:n välissä. WEPissä salauksessa käytetty alustusvektori oli vain 24-bittinen, mutta WPA:ssa sen pituus on kasvatettu 48 bittiin. WPA:ssa alustusvektori on sarjanumero, joka muodostuu kuudesta tavun mittaisesta TSC-kentästä (TKIP Sequence Counter). Sarjanumeron on kasvettava aina edellisestä lähetetystä datakehyksestä. Jos vastaanottajan saamassa kehyksessä on sama tai pienempi sarjanumero kuin samalta lähettäjältä viimeksi tulleessa kehyksessä, se hylätään. Tällä vaikeutetaan toistohyökkäyksiä, missä hyökkääjä lähettää aiemmin kaappaamansa kehyksen uudestaan verkkoon. TSC-kentistä TSC5 on kaikkein tärkein ja TSC0 vähiten merkitsevä. Nelitavuisessa IV-kentässä olevan D:n arvo muodostetaan TSC1:stä ja sitä käytetään salauksessa. Sen tarkoitus on estää ongelmallisten heikkojen alustusvektorien muodostuminen. Ne mahdollistivat WEPissä tietojen vuotamisen käytetystä WEP-avaimesta. IV-kentän viimeisessä tavussa on viiden bitin mittainen Rsvd (Reserved), joka on varattu myöhempää käyttöä varten, yhden bitin mittainen Ext IV, joka kertoo onko käytössä WEP vai TKIP sekä jo WEPistäkin tuttu Key ID. Ext IV -kentässä on laajennetun alustusvektorin neljä tärkeintä tavua (TSC2-TSC5). Ennen salausta datasta lasketaan kaksi eheydentarkistusta. ICV:hen tulee datasta ja MICistä CRC32:lla laskettu tarkistussumma aivan kuten WEPissäkin. Se ei kuitenkaan auttanut havaitsemaan muuta kuin satunnaisia bittivirheitä tietoliikenteessä.

sä. WPA:ssa on lisäksi MIC-kenttä, johon tulee Michael-algoritmilla datasta laskettu tarkistussumma. Sen tarkoitus on havaita datan tarkoituksellinen muuttaminen. Datan lisäksi myös MIC ja ICV salataan.

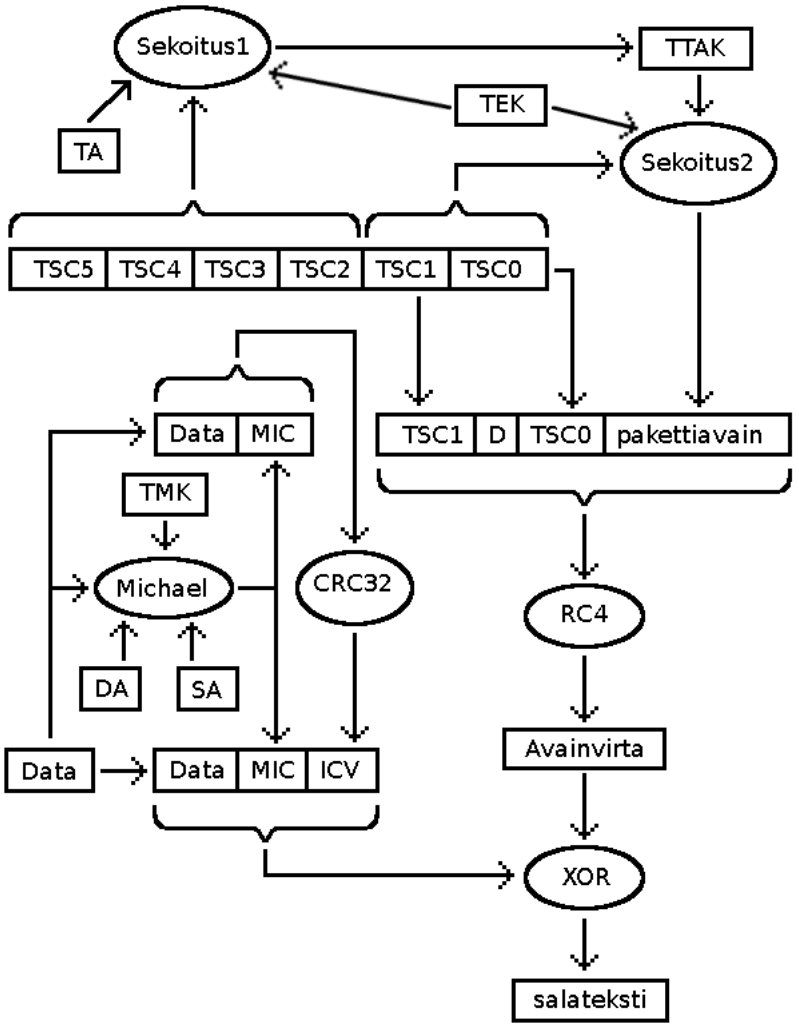
Kelvollisia eheydentarkistusalgoritmeja on olemassa useitakin, mutta ne vaativat liikaa laskentatehoa, jotta niitä voitaisiin käyttää vanhoissa WEP-pohjaisissa laitteissa pelkällä ohjelmistopäivityksellä ilman laitteen uusimista. Tähän tarpeeseen kehitettiin Michael-algoritmi [Fer02], jonka laskeminen on tarpeeksi nopeaa ja joka tarjoaa CRC32:ta paremman suojan kehysten tietoista muuttamista vastaan. Käytettäessä CRC32:ta hyökkääjä pystyi muun muassa muuttamaan salatekstiä ja laskemaan toimenpiteensä aiheuttamat muutokset CRC32-tarkistussummaan purkamatta edes salausta. Michael takaa paremman suojan kehysten eheydelle, mutta yhteensopivuusvaatimus vanhojen laitteiden kanssa rajoitti myös Michaelin kehittämistä samoin kuin valittua salaustapaakin. Tiedonsiirrossa tulleen bittivirheen havaitsee ennen MICiä ICV, joka korjaa pienet virheet. Vain kehykset, joissa ei ICV:ssä ei ole virhettä tai ne on saatu korjattua, näkyvät MICille asti. Normaalissa liikenteessä MICille asti tulee erittäin vähän viallisia kehyksiä. Jos esimerkiksi sekunnissa saapuu 100 datakehystä, jotka ovat sotkeentuneet matkalla niin pahasti ettei CRC32 enää pysty korjaamaan niissä olevia virheitä, niin korkeintaan joka $2^{32}/100$ sekunti paketti läpäisee virheellisesti CRC32:n, vaikka sen pitäisi jäädä siinä kiinni. Tällöin MICille asti tulee eheyden rikkova kehys harvemmin kuin kerran vuodessa [Fer02]. Hyökkääjä voi yrittää lähettää valtavan määrän kehyksiä kokeillen eri MIC-arvoja toivoen jonkin niistä olevan oikea. Kehyksen oikean ICV:n arvon hyökkääjä voi laskea CRC32:lla, sillä se ei perustu mihinkään salaiseen tietoon. MIC taasen lasketaan Michaelilla, joka käyttää muun muassa salaista TMK:ta avaimena. Koska MIC on hyvin harvoin väärä normaalissa tietoliikenteessä, voidaan virheilmoituksen erittäin todennäköisesti olettaa johtuvan aktiivisesta hyökkäyksestä, varsinkin jos virheellinen MIC esiintyy toistuvasti.

Tämän johdosta TKIP:ssä on määritelty vastatoimenpiteitä havaittaessa väärä MIC-arvo [Fer02] [IEE07]. Kun tukiasema havaitsee ensimmäisen virheellisen MICin tai saa asemalta viestin, että se on havainnut virheellisen MICin, tukiasema käynnistää 60 sekunnin mittaisen ajastimen. Jos seuraava MIC-virhe havaitaan tai saadaan tästä viesti asemalta ennen ajastimen loputtua, puretaan kaikkien asemien todennukset ja käytössä olleet parittaiset pääavaimet poistetaan. Jos väliaikainen ryhmäavain oli käytössä, myös se poistetaan. Tämän jälkeen tukiasema odottaa 60 sekuntia ennen kuin luo uuden väliaikaisen ryhmäavaimen ja sallii asemien todentaa itsensä 802.1X:ää ja EAP-metodeja käyttäen. Jos käytössä on PSK-todennus, niin 60 se-

kunnin odottamisen jälkeen siirrytään suoraan nelivaiheiseen kättelyyn. Jos asema havaitsee virheellisen MICin, se käynnistää oman 60 sekunnin mittaisen ajastimen ja lähettää tukiasemalle tiedon löytyneestä MIC-virheestä. Jos edellisestä virheellisestä MICistä on kulunut alle 60 sekuntia, asema lähettää tukiasemalle tiedon virheellisestä MICistä, poistaa käytössä olleen parittaisen pääavaimen ja väliaikaisen ryhmäavaimen ja purkaa todennuksensa. Tämän jälkeen asema odottaa 60 sekuntia ennen kuin yrittää todentaan itsensä uudestaan tukiasemaan.

Näillä vastatoimenpiteillä estetään hyökkääjää saamasta mitään tietoa kokeilemastaan MIC-arvosta sekä vaikeutetaan käytettyjen salaus- ja eheysavaimien paljastumista. Vastatoimenpiteiden lisäksi MIC on vielä salattava aivan kuin datakin, eikä sitä saa lähettää selvätekstinä. Jos hyökkääjä saisi käsiinsä selväteksin ja siitä lasketun MICin, hän voisi selvittää MIC-avaimen [Fer02]. Tämä johtuu MICissä käytetyn Michael-algoritmin käänteisyydestä, joka on eheydentarkistusalgoritmeille harvinainen piirre. Hyökkääjä suorittaa selvittämälleen selvätekstille ja siitä muodostetulle MIC-arvolle käänteisesti samat toimenpiteet kuin MIC-arvoa laskettaessa tehtiin [Woo04]. MICin salaaminen vaikeuttaa tätä, vaikka hyökkääjä saisikin selville käytetyn selvätekstin. MIC on vain väliaikainen WPA:han kehitetty ratkaisu ja turvallinen, kun se salataan ja käytetään vastatoimenpiteitä havaittaessa virheellinen MIC-arvo.

TKIP:ssä datan salaus tapahtuu samalla RC4-algoritmilla kuin WEPissäkin, mutta sitä käytetään paljon tarkoituksenmukaisemmin. Kuvassa 19 on havainnollistettu salausta TKIP:ssä. Aivan kuin WEPissäkin RC4 tuottaa avainvirran, jolle suoritetään XOR-operaatio datan ja siitä laskettujen eheydentarkistusten kanssa. Ennen RC4-algoritmin käyttöä sille syötettävää merkkijonoa sekoitetaan huomattavasti WEPiä monimutkaisemmin. WEPissähän vain liitettiin yhteen 24-bittinen alustusvektori ja WEP-avain. Sekoitus koostuu kahdesta vaiheesta. Ensimmäisessä vaiheessa syötteenä toimii lähettäjän MAC-osoite (Transmission Address, TA), TSC-sarjanumeron neljä merkitsevintä tavua sekä TEK. Vaihe yksi tuottaa 80-bittisen TTAK:n (TKIP-mixed Transmit Address and Key). Sekoituksen toisessa vaiheessa syötteenä ovat TTAK, TEK ja TSC-sarjanumeron kaksi vähiten merkitsevää tavua. Näistä saadaan 104-bittinen pakettiavain, joka vastaa WEP-avainta. Ennen sitä on oltava 24-bittinen alustusvektori, joka muodostetaan TSC1:stä, TSC0:sta ja niiden väliin sijoitetusta TSC1:stä kahdella yksinkertaisella bittioperaatiolla saadusta tavusta D. Sen tarkoituksena on poistaa WEPissä olleita heikkoja alustusvektoreita. TTAK:ta ei tarvitse vaihtaa jokaiseen pakettiin, mutta samaa arvoa voidaan käyttää korkeintaan 2^{16} paketissa. TTAK on muodostettava uudestaan myös silloin, kun TEKin



Kuva 19: Datan salaus TKIP:ssä

elinaika tulee täyteen ja vaihdetaan. Aivan kuin WEPissä WPA:ssa on edelleen käytössä ICV-kenttään laskettu CRC32-tarkistussumma. Sitä ennen datasta, kohteen (Destination Address, DA) ja lähettäjän (Source Address, SA) MAC-osoitteista ja TMK:sta muodostetaan Michael-algoritmilla eheydentarkistussumma ja sijoitetaan datan perään MIC-kenttään. Datasta ja MIC:stä lasketaan CRC32:lla vielä toinen tarkistussumma, joka sijoitetaan edellisten jälkeen ICV-kenttään. Lopuksi avainvirrasta, datasta ja sen perässä olevista eheydentarkistussummista muodostetaan XOR-operaatiolla salateksti lähetettäväksi vastaanottajalle.

Vaikka WPA onkin huomattavasti turvallisempi kuin WEP, sen käytössä on oltava huolellinen. Sekä TEKin että sekoitusvaiheessa muodostettava pakettiavain on

molemmat pidettävä ehdottomasti salaisina. Yhdenkin pakettiavaimen paljastuminen hyökkääjälle paljastaa myös MICin laskennassa käytetyn TMK-avaimen, jolloin hyökkääjä voi muuttaa kaapattua pakettia ja laskea siihen uuden oikean MIC-arvon. Tätä kuitenkin vaikeuttaa vielä TSC-sarjanumerointi. Hyökkääjän on estettävä vastaanottajaa saamasta kaapattua pakettia sekä estettävä vastaanottajaa saamasta tämän jälkeenkään hänelle samalta lähettäjältä lähetettyjä paketteja. Jos tämä ei onnistu, niin hyökkääjän kaappaamassa paketissa, jota on muutettu ja laskettu uusi MIC, on väärä sarjanumero ja vastaanottaja hylkää sen [Fer02]. Vielä vakavampaa on, jos hyökkääjä saa haltuunsa kaksi tai useampia pakettiavaimia, joiden muodostamiseen on sekoittamisen vaiheessa yksi käytetty samoja 32:ta merkitsevintä TSC:n bittiä (TSC2-TSC5). Tällöin hyökkääjä voi selvittää TEKin [MRH04]. Nämä ongelmat ovat kuitenkin aika teoreettisia, ja niiltä vältytään, kun salaukseen käytettävä pakettiavain vaihtuu joka kerta ja kun huolehditaan PTK:n tarpeeksi tiheästä vaihtamisesta.

Vuonna 2009 Beck ja Tews esittivät, että aiemmin WEPissä käytettyä pilkkomishyökkäystä voitiin käyttää myös WPA:ssa [BeT09]. Hyökkäystä kuitenkin vaikeuttavat ja hidastavat väärään MIC-arvoon liittyvät vastatoimenpiteet sekä TSC:n muodostama sarjanumero. Onnistuneessa hyökkäyksessä saadaan purettua yksittäisen paketin salaus, selville sen salaamiseen käytetty avainvirta sekä MICin muodostamisessa käytetty MIC-avain. Lisäksi hyökkääjä voi vielä lähettää verkkoon muutaman kaapatusta paketista väärennetyn paketin. Aluksi hyökkääjä kuuntelee verkon liikennettä, kunnes havaitsee tukiaseman lähettämän ARP-lähetyksen [Plu82]. ARPit ovat hyökkääjän kannalta hyviä, sillä ne on salauksesta huolimatta helppo tunnistaa muun liikenteen joukosta vakiokokonsa ja yleislähetysosoitteen ansiosta. Lisäksi niiden sisällöstä suurin osa on vakiomuotoista tai helposti arvattavissa. ARPissa ainoastaan lähettäjän ja vastaanottajan IP-osoitteiden viimeiset vähiten merkitsevät tavut ovat hyökkääjälle tuntemattomia. Osoitteiden muut tavut ovat helposti selvitettävissä tai arvattavissa. Salatun datan jälkeen ARP-paketissa on salattuna kaksi eheydentarkistusta: kahdeksantavuinen MIC ja neljätavuinen ICV. Koko paketissa on siis hyökkääjälle yhteensä 14 tuntematonta tavua, jotka on selvitettävä. Beck-Tews-hyökkäyksessä selvitetään pilkkomishyökkäyksellä MIC ja ICV. Typistetään salattua paketin osaa yhdellä tavulla, arvataan mikä poistetun tavun salaamaton muoto olisi ja tehdään arvauksen aiheuttama muutos ICV:hen. Tällainen typistetty paketti lähetetään asemalle. Jos ICV on väärä, asema hylkää paketin hiljaisesti. Jos paketin ICV on oikea, mutta MIC on väärä, niin asema tulkitsee sen mahdolliseksi hyökkäykseksi ja vastatoimenpiteenä käynnistää 60 sekunnin mittaisen ajastimen

ja lähettää tukiasemalle tiedon virheellisestä MICistä. Tästä aseman tukiasemalle lähettämästä varoituksesta hyökkääjä tietää arvauksensa olleen oikean. Ajastin kuitenkin hidastaa hyökkäystä, sillä arvauksia ei voi tehdä liian usein. Jokaisen oikeaan osuneen arvauksen jälkeen on siis odotettava minuutti ennen kuin voidaan siirtyä seuraavan tavun arvaukseen. 11:ssä minuutissa hyökkääjä on saanut selville sekä ICV:n että MICin. Kahta vielä tuntematonta tavua eli lähde- ja kohdeosoitteiden viimeisiä tavua ei tarvitse enää selvittää pilkkomishyökkäyksellä. Ne saadaan nopeammin selville yksinkertaisesti kokeilemalla kaikki vaihtoehdot. Arvatut tavut sisältävään dataan lisätään jo selvitetty MIC-arvo ja lasketaan ICV ja verrattaan sitä jo selvitettyyn oikeaan ICV-arvoon.

Hyökkäyksen onnistuminen vaatii vielä TSC:n huomioimista. TSC:tä kasvatetaan aina, kun laite vastaanottaa kelvollisen kehyksen, missä sekä ICV että MIC ovat oikein. Hyökkääjän salakuuntelemassa ARPissa oli jokin TSC-arvo ja asema ei enää hyväksy kehyksiä, joissa on sama tai pienempi TSC-arvo. Asema hylkää hiljaisesti kelvottoman TSC:n sisältävän kehyksen. Koska TSC tarkistetaan ennen salauksen purkamista, ei pilkkomishyökkäyksen käyttäminen sellaisenaan onnistu. Jos hyökkääjän lähettämän arvauksen jälkeen asema ei lähetä tukiasemalle viestiä virheellisestä MICistä, se voi johtua joko siitä, että arvaus oli väärä, jolloin ICV oli myös väärä, tai siitä että arvaus oli oikea, mutta paketissa oli liian pieni TSC-arvo. Ongelma voidaan kuitenkin kiertää, jos verkossa on käytössä QoS ja sen kanavat. Kahdeksalla kanavalla on kullakin oma TSC-laskurinsa. Usein kaikki kanavat eivät ole yhtä runsaassa käytössä ja osa voi olla jopa täysin käyttämättömiä, jolloin niiden TSC-laskurin arvo on hyvin pieni. Hyökkääjä käyttää arvaustensa lähettämiseen jotain sellaista kanavaa, jolla TSC-arvo on pienempi kuin kanavalla, jolla salakuunneltu kehys lähetettiin.

Lopulta hyökkääjä on saanut purettua salauksen ARPista. Selvätekstin ja MICin perusteella voidaan helposti laskea myös Michael-algoritmin käyttämä salainen MIC-avain. Riippuen salakuunnellun ARPin tyypistä (pyyntö vai vastaus) saadaan selville joko TMK, jota tukiasema käyttää muodostaessaan eheydentarkistussummaa asemalle lähetettävään pakettiin, tai GIK, jota käytetään lähetettäessä kehystä ryhmälle asemia. Myös ARPin salaukseen käytetty avainvirta saadaan selville XOR-operaatiolla selvä- ja salatekstistä. Myöhemmin, kun salakuunnellaan seuraava ARP-kehys, sen murtaminen onnistuu nopeammin, jos siinä käytetään samaa MIC-avainta kuin jo aiemmin puretussa paketissa. Tällöin riittää, kun puretaan datan salaus tuntemattomien tavujen osalta. Sen jälkeen MIC voidaan laskea, koska tiedetään MIC-avain ja lopuksi ICV:kin voidaan laskea. Pientä salakuunneltua ARP-kehystä

voidaan myös väärentää ja muokata ja lähettää jokaisella sellaisella QoSin kanavalla, jolla on pienempi TSC-arvo kuin salakuunnellussa kehyksessä. Näin voidaan esimerkiksi sotkea ARP-liikennettä verkossa.

Periaatteessa Beck-Tews hyökkäystä voidaan käyttää isompienkin pakettien salauksen purkamiseen, mutta mitä enemmän salattuja tavuja pitää purkaa sitä kauemmin se vie aikaa vastatoimenpiteiden vuoksi. Lisäksi ARPit ovat pienen kokonsa vuoksi hyviä, sillä niitä ei lähetettäessä pirstaloida enää pienempiin osiin. ICV lasketaan erikseen jokaiselle pirstaloidulle osalle ja tämä hankaloittaisi hyökkäystä.

Mielestäni Beck-Tews-hyökkäyksestä saisi vielä hiukan nopeamman, kun selvittäisi pilkkomishyökkäyksellä ensin MICin kahdeksan tavua ja sen jälkeen datan kaksi tuntematonta tavua. Näiden jälkeen ICV voitaisiin laskea aivan normaalisti, sillä sen laskemiseen ei tarvita mitään salaista tietoa. Tällöin hyökkäykseen kuluisi aikaa vain reilu kymmenen minuuttia.

Beck-Tews-hyökkäys on kuitenkin helppo torjua [BeT09]. Koska sen suorittaminen vaatii useita minuutteja pienellä ARP-kehykselläkin, niin tarpeeksi tiheä PTK:n uusiminen estää hyökkääjää purkamasta kokonaisen kehyksen salausta. Toinen hyökkäystä rajoittava keino on muuttaa aseman toimintatapaa sen havaitessa väärän MIC-arvon. Asema voisi olla lähettämättä varoitusta havaitessaan väärän MIC-arvon ja käynnistää pelkästään oman 60 sekunnin mittaisen ajastimensa. Vasta jos asema saisi toisen väärän MICin sisältävän kehyksen ennen kuin ajastin on kulunut, käynnistettäisiin normaalit vastatoimenpiteet. Näin hyökkääjä ei voisi käyttää aseman lähettämiä varoituksia merkkinä arvauksensa oikeellisuudesta. Myös QoSin kanavat ovat aivan oleellisia, jos hyökkääjä haluaa jotenkin hyödyntää purkamansa ARPin salausta. Ilman niitä salauksen purkamisen jälkeen salakuunneltua kehystä tai siitä muokattua kehystä ei kannata enää lähettää uhrille, sillä liian pienen TSC:n vuoksi niitä ei huomioida ollenkaan.

QoSin kanavien käyttäminen ei kuitenkaan ole välttämätöntä Teck-Bews-hyökkäyksessä. Ohigashi ja Morii käyttivät kanavien sijasta välimieshyökkäystä. He osoittivat myös, että ensimmäisen onnistuneesti murretun ARPin salauksen jälkeen seuraavat ARPit voidaan murtaa minuutissa [OhM09]. Murtaminen onnistuu kuitenkin vain 37 %:n todennäköisyydellä. Hyökkääjä asettuu tukiaseman ja aseman väliin, niin että kaikki tukiaseman ja aseman väliset kehykset kulkevat sen kautta. Joko tukiasema ja asema ovat niin kaukana toisistaan, ettei kantama riitä niiden välillä, tai hyökkääjä käyttää radiohäirintää ja suunnattua antennia voidakseen kontrolloida aseman ja tukiaseman välistä liikennettä haluamallaan tavalla. Koska hyökkääjä

voi kontrolloida, mitkä tukiaseman lähettämät kehykset menevät asemalle asti, se voi myös kontrolloida TSC:n kasvamista. Ensin hyökkääjä tekee normaalisti aiemmin mainitun Beck-Tews-hyökkäyksen jollekin ARP-vastaukselle, jonka tukiasema on lähettänyt asemalle, jolle menevää liikennettä hyökkääjä kontrolloi. Hyökkäys paljastaa käytetyn MIC-avaimen, tässä tapauksessa toisen TMK-avaimista sekä aseman IP-osoitteen. Siten myöhemmin hyökkääjän kaappaamissa samalle asemalle lähetetyissä ARP-vastauksissa on vain yksi tuntematon tavu lähettäjän IP-osoitteessa. Lisäksi ICV on tietysti tuntematon. Purkaakseen seuraavan ARP-vastauksen salauksen hyökkääjä kaappaa kehyksen eikä päästä sitä asemalle asti. Jos koko ICV selvitettäisiin pilkkomishyökkäyksellä, niin vastatoimenpiteiden vuoksi siihen menisi reilu kolme minuuttia. Ajan säästämiseksi siitä selvitetäänkin ainoastaan viimeinen tavu. Lopuksi hyökkääjä käy läpi kaikki mahdolliset ARPissa olevan tuntemattoman tavun mahdolliset arvot. Koska TMK tunnetaan, voidaan jokaiselle arvauksen sisältävälle datalle laskea MIC. Seuraavaksi datalle ja sitä seuraavalle MICille lasketaan ICV ja sen viimeistä tavua verrataan pilkkomishyökkäyksellä saatuun oikeaan arvoon. Toki useampikin eri kokeiltu arvo puuttuvalle tavulle voi tuottaa ICV:n viimeiseksi tavuksi oikean arvon. Ainoastaan, kun vain yksi vaihtoehto kaikista mahdollisista tavun arvoista antaa oikean ICV:n viimeisen tavun, tiedetään arvauksen olevan oikea. Siksi Ohigashin ja Moriin Beck-Tews-hyökkäykseen pohjautuvalla menetelmällä saadaankin reilussa minuutissa purettua salaus vain 37 %:ssa kaapatuista ARP-vastauksista.

Hyökkäystä on kuitenkin hyvin vaikea havaita, sillä se ei välttämättä aiheuta pitkiä katkoja liikenteeseen ja käytettäessä vielä suunnattua antennia kehykset menevät vain hyökkääjän haluamalle osapuolelle. Pisimmän katkon liikenteeseen aiheuttaa alussa tehtävä Beck-Tews-hyökkäys, jonka aikana on estettävä kaikkien tukiaseman asemalle lähettämien datakehysten perille pääsy. Tätäkin katkoa voidaan pienentää seuraamalla liikennettä ja päästämällä joidenkin krittisten sovellusten kehykset läpi. Tällöin toki aseman ylläpitämä TSC-arvo kasvaa liian isoksi ja Beck-Tews-hyökkäys joudutaan aloittamaan uudestaan alusta. Myös jos hyökkääjä haluaa salauksen purkamisen jälkeen lähettää itse väärennetyn ARPin asemalle, on koko ARPin kaappaamisen ja väärennetyn paketin lähettämisen välisen ajan estettävä asemaa saamasta yhtään tukiasemalta tulevaa datakehystä, jotta aseman TSC-arvo ei kasva liian isoksi.

ARPille tehdystä Beck-Tews-hyökkäyksestä saadun avainvirran hyödyntämistä rajoittaa sen lyhyys, vain 40 tavun mittainen pituus. Halvorsen ja Haugen onnistuivat tekemään Beck-Tews-hyökkäyksen DHCP-kehykselle (Dynamic Host Confi-

guration Protocol) [Dro97] ja saamaan selville jopa 596 tavun mittaisen avainvirran [HaH09]. ARP sopi hyvin purettavaksi, sillä iso osa sen rakenteesta on tunnettua. Samoin DHCP-protokollan kuittausviestit ovat suurimmaksi osaksi täysin tunnettuja ja lisäksi niissä on runaasti täytetavuja. Hyökkääjälle tuntemattomia ovat ainoastaan ICV, MIC, IP-osoitteet sekä liikennöintitunnus (Traffic identifier, TID), jolla asema ja DHCP-palvelin liittävät lähettämänsä pyynnöt- ja vastaukset toisiinsa. Hyökkäys kestää selvästi kauemmin kuin ARPille tehtävä hyökkäys, mutta sillä saatua pitempää avainvirtaa voidaan käyttää muiden hyökkäysten tekemiseen. Hyökkääjä voi hankkia lisätietoa ja häiritä verkon toimintaa muodostamalla verkon kontrolli-informaatiota välittäviä paketteja ja salaamalla ne kelvollisesti selvittämällään avainvirralla. Tällaisia paketteja ovat muun muassa TCP:n SYN- ja ACK-paketit sekä DNS-, DHCP-, ICMP- ja ARP-protokollien mukaiset paketit. DHCP-kuittausviestin murtaminen kesti testiympäristössä noin 20-25 minuuttia, mutta jos IP-osoitteiden selvittämiseksi on ensin suoritettava Beck-Tews-hyökkäys ARP-paketille, niin aikaa menee kokonaisuudessaan lähemmäs 40 minuuttia [HaH09].

Toinen tapa muodostaa pitempiä avainvirtoja onnistuu käyttämällä hyväksi ARP- tai IPv4-pakettien sisällön ennustettavuutta sekä pakettien pirstaloimista [Bec10]. Aiemmin käsiteltiin jo ARP-paketin sisältöä ja ettei siinä ole hyökkääjälle, kuin maksimissaan kaksi tuntematonta tavua. IP-pakettien alkuosakin on varsin hyvin tunnettua tai hyökkääjälle helposti arvattavissa. LLC-kerroksen otsikko on 8 tavun mittainen ja sama jokaiselle IPv4-paketille. Sen jälkeen tulevassa IP-kehyksen otsikon alussa on muun muassa käytetyn version ja otsikon pituuden ilmaisevia kenttiä, joiden muoto on tiedossa. Hyökkääjä voi suhteellisen helposti arvata jokaisen IPv4-paketin alusta 12 tavua. Tästä hyökkääjä saa selville 12 tavun mittaisen avainvirran, mutta sen lyhyys rajoittaa käyttöä huomattavasti. Siitä voidaan kuitenkin sirpaloinnin avulla saada pitempi. Hankkimalla ensin 16 tällaista 12 tavun avainvirtaa voidaan niistä muodostaa isomman paketin sirpaleita. Jokaiselle sirpaleelle lasketaan oma ICV, mutta MIC lasketaan koko paketille ennen sirpalointia. 15 sirpaleessa on 8 tavua dataa ja 4 tavun ICV ja viimeisessä sirpaleessa on vain 8 tavun MIC ennen ICV:tä. Dataa voidaan lähettää 16 sirpaleessa yhteensä maksimissaan 120 tavua. Jos käytetään ARP-paketteja IP-pakettien sijasta, niin esimerkiksi Beck-Tews-hyökkäyksellä puretusta yhdestä ARP-paketista saadulla 40 tavun avainvirralla voidaan 16 sirpaleessa lähettää yhteensä jopa 568 tavua dataa. Hyökkääjä voi kasvattaa lyhyttä tuntemaansa 12 tavun avainvirtaa lähettämällä asemalle TCP-protokollassa yhteyden kahden koneen välille avaavaa TCP-SYN-pakettia,

jossa lähettäjäksi on väärennetty tukiasema. Paketin koko on 48 tavua koostuen 8 tavun LLC:stä sekä 20 tavun IP- ja TCP-otsikoista. Lisäksi koko paketille lasketaan 8 tavun MIC. Koska jokaisen sirpaleen loppuun tulee 4 tavun ICV, niin yhdestä 12 tavun avainvirrasta voidaan käyttää 8 tavua datan kuljettamiseen. TCP-SYN-paketin muodostamiseen tarvitaan siten 7 sirpaletta. Sirpaleet on lähetettävä QoSin eri kanavilla, jotta TSC-arvo ei olisi liian pieni ja sirpaleita ei hylättäisi sen takia. Aseman saatua TCP-SYN-paketin sirpaleet se kokoaa ne ja lähettää tukiasemalle TCP-SYN/ACK-kuittauspaketin. Koska tukiasema ei kuitenkaan ole lähettänyt aiempaa TCP-SYN-pakettia, johon se juuri sai kuittauksen, se lähettää TCP-yhteyden peruuttavan TCP-RST-paketin. Tästä paketista hyökkääjä saa itselleen 60 tavun mittaisen avainvirran, koska hän tietää tai voi helposti päätellä paketin selväkielisen sisällön johtuen itse ensimmäisenä lähettämästään TCP-SYN-paketista ja siihen määrittelemistään tiedoista. Jatkossa hyökkääjä voi käyttää näin saamiansa 60 tavun avainvirtoja muodostaessaan vieläkin isomman paketin sirpaleita. Mahdollisten alkutoimenpiteiden, kuten tukiaseman IP-osoitteen selvittämisen, jälkeen hyökkääjä saa edellä kuvatulla tavalla muodostettua 60 tavun avainvirran käytännössä muutamassa sekunnissa. Huomattakoon, että avainvirtaa voidaan käyttää vain asemalle lähetettävien pakettien salaukseen, ei siis tukiasemalle, koska paketeissa on oltava myös oikea MIC-arvo. Beck-Tews-hyökkäyksellä voitiin selvittää vain asemalle lähetettyjen pakettien eheydentarkistuksen laskennassa Michael-algoritmin käyttämä TMK.

Sirpalointia voidaan hyödyntää toisellakin tavalla, joka perustuu eheydentarkistusta laskevan Michael-algoritmin heikkouksiin. Michael-algoritmi tuottaa 8-tavuisen tarkistussumman, jonka laskennassa tarvitaan muun muassa salaista väliaikaista MIC-avainta. Michael-algoritmi pitää laskennan aikana yllä kahta 32 tavun mittaista tilamuuttujaa. Jos nämä tilamuuttujat saavat jossain laskennan vaiheessa samat arvot kuin mitä ne olivat aivan alussa, niin tarkistussumma tavallaan nollautuu ja sen laskenta alkaa taas alusta. Tällöin lopulliseen tarkistussummaan vaikuttavat vain tilamuuttujien nollautumisen jälkeen käsitellyt selvätekstin tavut. Käytännössä tilamuuttujien palautuminen alkutilaansa on erittäin harvinaista, mutta Beck esitti miten sellainen voidaan tehdä tarkoituksellisesti [Bec10]. Voidaan muodostaa edeltävästä selvätekstistä kaksi neljän tavun mittaista merkkijonoa, niin sanottua taikasanaa, jotka selvätekstin jälkeen sijoitettuna, palauttavat Michael-algoritmin tilamuuttujat alkutilaansa. Näitä kahta taikasanaa voidaan käyttää yhdessä sirpaloinnin kanssa hyväksi asemalle menevän paketin salauksen purkamiseksi. Purkaminen vaati lisäksi, että lähiverkosta on pääsy internetiin ja että hyökkääjällä on siellä apu-

lainen. Ensin hyökkääjä salakuuntelee minkä tahansa tukiaseman jollekin asemalle lähettämän paketin, joka ei kuitenkaan ole liian pitkä. Hyökkääjä muodostaa kaksi sirpaletta, joista ensimmäinen koostuu ICMP echo request -paketin alkuosasta, jonka lähettäjäksi on väärennetty hyökkääjän internetissä sijaitseva apulainen. Koska hyökkääjä tuntee ensimmäisen sirpaleen selvätekstin kokonaisuudessaan, hän voi selvittää tarvittavat taikasanat, joilla Michael-algoritmin tilamuuttujat nollautuvat. Nämä taikasanat sijoitetaan vielä ensimmäisen sirpaleen loppuun, mutta kuitenkin ennen ICV:tä. Koska ICMP-kehysten viimeinen kenttä voi sisältää valinnaista dataa, sijoitetaan toiseen sirpaleeseen aiemmin salakuunneltu paketti, jonka salaus siis halutaan purkaa. Tämä kahden sirpaleen paketti lähetetään asemalle, joka vastaa saamaansa pakettiin ICMP echo response -paketilla ja sijoittaa sen datakenttään ICMP echo requestin datakentässä olleen datan. Asema lähettää paketin tukiasemalle, joka purkaa paketin salauksen ja lähettää edelleen internetiin hyökkääjän apulaiselle. Näin hyökkääjä saa purettua alkuperäisen salakuunnellun paketin. Lisäksi hyökkääjä saa pitemmän, lähettämänsä sirpaleista koostuvan paketin mittaisen, avainvirran tietoonsa myöhempää käyttöä varten. Hyökkäys on erittäin nopea ja vaatii korkeintaan joitakin kymmeniä sekunteja, mutta se toki edellyttää tiettyä valmistelua ja pääsyä lähiverkosta internettiin, QoSin kanavien käyttöä sekä ettei ICMP-viestejä ole otettu pois käytöstä.

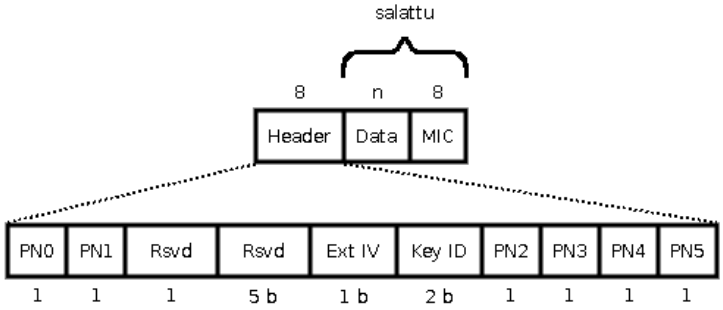
WEPissä ja siihen pohjautuvassa TKIP:tä käyttävässä WPA:ssa on eräänä perusongelmana valittu salaustapa: jonosalaus. Se perustuu selvätekstin ja avainvirran väliseen XOR-operaatioon, joka tuottaa salatekstin. Tämä on kuitenkin haavoittuvainen, sillä hyökkääjän saadessa selville mitkä tahansa kaksi merkkijonoa avainvirran, selvätekstin ja salatekstin muodostamasta joukosta hän voi selvittää kolmannen merkkijonon XOR-operaatiolla. Parempi ratkaisu on käyttää jonosalauksen sijasta lohkosalausta, johon pohjautuvaa AES-salausalgoritmia käytetäänkin WPA2:n CCMP:ssä.

4.4 CCMP

TKIP oli tarkoitettu vain väliaikaiseksi ylimenovaiheen ratkaisuksi kehysten salaamisessa ja eheyden tarkistamisessa. Se korjasi WEPissä ilmenneitä vakavia puutteita, mutta on vapaaehtoinen. 802.11i-standardi määrittelee pakolliseksi salauksesta ja eheydestä huolehtivaksi protokollaksi ainoastaan CCMP:n [IEE07]. TKIP oli muodostettu WEPin päälle korjaten sen puutteita, mutta CCMP on toteutettu kokonaan puhtaalta pöydältä. CCMP perustuu CCM:ään, yleiseen malliin, jota voi-

daan käyttää valitun lohkosalausalgoritmin kanssa [WHF03]. CCMP:ssä käytetään salaukseen AES-algoritmia (Advanced Encryption Standard, AES) [NIS01]. CCM koostuu kahdesta osasta: salauksesta vastaavasta laskurimoodista (counter mode, CTR) sekä MIC-tarkistussumman muodostavasta Michael-algoritmia kehittyneemmästä eheydentarkistuksesta (Cipher Block Chaining Message Authentication Code, CBC-MAC).

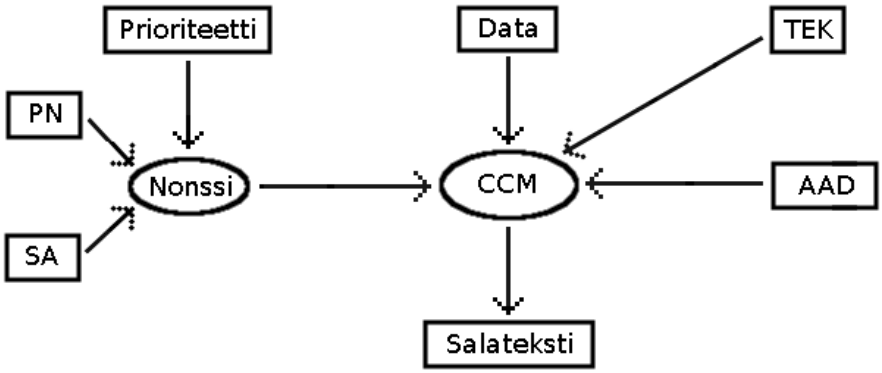
CCMP:n sisältävää WPA2:ta käytettäessä MAC-kerroksen datakehyksen datakenttä on kuvassa 20 esitetyn kaltainen. Kenttien pituudet on ilmoitettu tavuina kol-



Kuva 20: Datakehyksen datakenttä WPA2:ssa

mea poikkeusta lukuun ottamatta, joiden pituus on ilmoitettu bitteinä. Datakenttä koostuu salaamattomasta Headerista sekä varsinaisesta datasta ja sitä seuraavasta MICistä, jotka on salattu. Toisin kuin WEPissä ja TKIP:ssä ICV:tä ei enää käytetä ollenkaan eheydentarkistukseen. Pelkkä kahdeksan tavun mittainen MIC riittää, koska se muodostetaan aiempaa turvallisemmin. Kahdeksantavuinen Header koostuu kuuteen osaan jaetusta pakettinumerosta (Packet Number, PN), myöhempään käyttöön varatuista osista (Rsvd) sekä Ext IV- ja Key ID -biteistä. Ext IV ilmaisee, että käytössä on CCMP ja Key ID kertoo mikä mahdollisesta neljästä eri salausavaimesta on käytössä, joskin yleensä käytetään vain yhtä. PN:ää käytetään kehyksen salaamisessa ja sitä kasvatetaan yhdellä jokaisessa salatussa kehyksessä. Salaamisessa käytettävää TEKiä on vaihdettava tarpeeksi usein ettei PN ala toistua ja ettei kehyksen salauksessa käytetä koskaan yhtä aikaa samaa TEKiä ja PN:ää kuin on käytetty jo aiemmin.

Datan salaus tapahtuu WPA2:ssa selvästi eri lailla ja yksinkertaisemmin kuin WEPissä tai WPA:ssa ja tätä on havainnollistettu kuvassa 21. Salauksen ja eheyden muodostava CCM saa syötteenä nonssin, salattavan datan, TEKin ja AADin (Ad-



Kuva 21: Datan salaus CCMP:ssä

ditional Authentication Data, AAD), joka muodostetaan MAC-kehyksen otsikosta saatavista tiedoista. Nonssi muodostetaan lähettäjän MAC-osoitteesta (SA), PN:stä, sekä prioriteetista, jonka arvo riippuu siitä onko QoS käytössä vai ei. Jos QoS puuttuu, koostuu prioriteetti pelkistä nollabiteistä. Jos QoS on käytössä, sijoitetaan prioriteetin arvoksi QoS-kentän neljä ensimmäistä bittiä eli niin sanottu liikennöintitunnus (traffic identifier, TID). Data saadaan protokollapinon ylemmältä, LLC-kerrokselta ja TEK muodostettiin nelivaiheisessa kättelyssä. AAD muodostetaan MAC-kehyksessä ennen datakenttää olevista kentistä, pois lukien kuitenkin Duration (katso kuva 3 sivulla 9). Osa mukaan otettujen MAC-kehyksen kenttien osista korvataan kuitenkin AAD:ssä nollabitillä ja jätetään tavallaan huomiotta. Näin tehdään, jos kentästä puuttuu jotain osia tai jos sen arvo saattaisi muuttua. MAC-kehyksestä suoraan AAD:hen otettujen bittien osalta CCMP varmistaa niidenkin eheyden datan lisäksi. AES:ää salausalgoritmina käyttävä CCM tuottaa salatekstin, joka koostuu datasta ja sitä seuraavasta 8-tavuisesta MICistä [IEE07].

CCMP:tä käyttävä WPA2 suunniteltiin ilman historian painolastia ja taaksepäin yhteensopivuutta ja se tarjoaakin luotettavaa datakehysten salausta ja varmistaa tehokkaasti niiden eheyden, eikä sitä vastaan tunneta mitään raakaan laskentaan perustuvaa väsytyshyökkäystä oleellisesti tehokkaampaa keinoa purkaa salausta [HeM05]. Eheydentarkistussumman muodostava CBC-MAC on huomattavasti TKIP:ssä käytettyä Michael-algoritmia luotettavampi. Michaelin heikkous on sen käänteisyydessä. Selvitettyään selvätekstin ja siitä muodostetun MICin hyökkääjä sai helposti laskettua Michael-algoritmin MICin muodostamisessa käyttämän salaisen TMK:n ja pystyi sen jälkeen laskemaan kelvollisen MICin mihin tahansa itse lähettämään-

sä pakettiin [Fer02]. Koska CCMP:ssä MIC on muodostettu turvallisemmin, ei siinä tarvita myöskään mitään TKIP:ssä olevia MICiä suojaavia vastatoimenpiteitä, kuten ajastimia ja väliaikaisten avainten uudelleenneuvottelua liian usein havaitun virheellisen MICin jälkeen. Toisin kuin TKIP:ssä CCMP:ssä eheydentarkistus kattaa datan lisäksi myös ison osan MAC-kehyksen otsikkokentistä. Toistohyökkäyksiä CCMP:ssä estää PN, jonka vastaanottaja tarkistaa, ja jonka on oltava aina isompi kuin edellisessä samalta lähettäjältä tulleessa kehyksessä.

Vaikka CCMP tarjoaakin luotettavaa salausta ja pakettien eheyttä, on WPA2:ssa muutama jo aiemmin mainittu riskitekijä. Käytettäessä todennuksessa PSK:ta on sen oltava tarpeeksi pitkä ja monimutkainen täyttäen kaikki yleiset salasanoillekin asetetut vaatimukset. Lisäksi, jos lähiverkkoon kuuluu useampi asema ja käytetään PSK:ta, voi verkkoon aiemmin liittynyt asema A kuuntelemalla myöhemmin verkkoon liittyvän aseman B tukiaseman kanssa käymää nelivaiheista kättelyä selvittää tukiaseman ja aseman B keskinäisessä liikennöinnissä käyttämän PTK:n [IEE07]. Näistä ongelmista päästään, kun käytetään todennukessa 802.1X:ää ja EAPia. Kolmas riskitekijä liittyy heikkoon GTK:n määrittelyyn standardissa, niin sanottuun Hole196-ongelmaan, joka mahdollistaa välimieshyökkäyksen [Ahm10]. Hole196:ssa lähiverkkoon kelvollisesti todennettu hyökkääjä voi syöttää toiselle samassa lähiverkossa olevalle asemalle vääriä osoitetietoja ARP-protokollan avulla ja saada aseman lähettämään muualle tarkoitetut kehykset hyökkääjälle.

Keskityttäessä vuosien saatossa 802.11-lähiverkon tietoturvan parantamiseen on huomio kohdistunut ennen kaikkea tiedon luottamuksellisuuden ja eheyden takaamiseen sekä laitteiden luotettavaan todennukseen. Saatavuuden varmistaminen on jäänyt tietoturvaa parannettaessa kuitenkin vähemmälle huomiolle ja sitä uhkaavat erityisesti palvelunestohyökkäykset.

5 Palvelunestohyökkäykset

Alun perin palvelunesto-käsite liitettiin lähinnä käyttöjärjestelmiin [Gli84] ja vasta myöhemmin se yhdistettiin verkkoympäristöihin [Nee94]. Tässä luvussa käsitellään lyhyesti OSI-mallin (katso kuva 2 sivulla 7) verkko-, kuljetus- ja sovelluskerroksella vaikuttavia DoS-hyökkäyksiä. Tällaiset internetin kautta tulevat DoS-hyökkäykset joko lamaannuttavat uhrin, kuluttavat uhrin resursseja niin, etteivät muut oikeutetut käyttäjät voi käyttää uhrin verkossa tarjoamia palveluja tai häiritsevät verkon infrastruktuurin toimintaa niin, etteivät uhrille osoitetut paketit pääse edes peril-

le asti. OSI-mallin kahdessa alimmassa kerroksessa toimivassa WLANissa esiintyviä DoS-hyökkäyksiä käsitellään seuraavassa luvussa.

Internet koostuu valtavasta määrästä toisiinsa liitettyjä päätelaitteita sekä niiden välisestä sujuvasta tietoliikenteestä huolehtivasta verkon infrastruktuurista ja lukuisista liikennöintiprotokollista. Hyökkääjä voi tarkoituksellisesti häiritä verkossa olevan julkisen palvelun saatavuutta muilta palveluun oikeutetuilta käyttäjiltä DoS-hyökkäyksellä, joita on kahta päätyyppiä [MVS01]. *Loogisessa hyökkäyksessä* suhteellisen pienellä määrällä tarkoituksellisesti muokattuja paketteja käytetään hyväksi jotain tietoturva-aukkoa ohjelmistossa tai protokollan toiminnassa ja saadaan järjestelmä hidastumaan tai jopa kaatumaan kokonaan. *Tulvituksessa* käytetään hyväksi internetin erinomaista tehokkuutta pakettien välittämisessä ja hukutetaan kohde valtavaan määrään turhia paketteja, joiden käsittely kuluttaa kohteen resurssit loppuun. Resursseja voivat olla esimerkiksi CPU-aika, muisti tai tietoliikennekapasiteetti.

5.1 Loogiset hyökkäykset

Loogiset DoS-hyökkäykset perustuvat sovellusohjelmissa, käyttöjärjestelmissä ja protokollissa olevien haavoittuvuuksien, ohjelmointi- ja konfigurointivirheiden sekä tietoturva-aukkojen hyödyntämiseen tavalla, joka johtaa järjestelmän hidastumiseen, kaatumiseen tai uudelleenkäynnistymiseen. Seuraavassa joitakin tunnetuimpia vuosien varrella havaittuja loogisia DoS-hyökkäyksiä.

90-luvulla oli useampia IP-protokollassa huonosti toteutettuun sirpaleiden käsittelyyn liittyvää hyökkäystä. IP-paketin suurin sallittu koko on 65535 tavua ja tätä suurempia IP-paketteja ei hyväksytä, mutta Ping of Death -hyökkäyksessä lähetetään tätä isompi paketti pieninä sirpaleina. Vastaanottajan yhdistäessä sirpaleet kokonaiseksi paketiksi siitä tulee liian iso, josta seuraa puskurin ylivuoto ja käyttöjärjestelmän kaatuminen tai uudelleenkäynnistyminen [Ken96]. Hyökkäys tehdään ICMP:n ping- eli echo request -paketilla, jossa on liian iso datakenttä. Hyökkääjä lähettää muokattuja sirpaleita, joista kasattaessa tulee yksi iso, ylisuuri ping-paketti. Koska ongelma ei kuitenkaan liity niinkään ping-paketteihin, vaan sirpaleiden kasaamiseen, saadaan sama kaatuminen aikaiseksi myös muillakin IP-kehyksiä käyttävillä protokollilla. Hyökkäys on erittäin helppo toteuttaa, mutta onneksi sen korjaus oli myös helppoa ja käyttöjärjestelmiin tehtiinkin nopeasti tarvittavat korjaukset estämään puskurin ylivuoto sirpaleita yhdistettäessä.

1997 havaittiin Teardrop-hyökkäys, joka perustui myös IP-protokollan tapaan yhdistää sirpaleista kokonainen paketti. Teardropissa hyökkääjä hyödyntää IP-rotokollan toteutukseen jäänyttä virhettä ja lähettää muutaman muokatun sirpaleen, joita yhdistäessä käyttöjärjestelmä kaatuu. Sirpaleet on muodostettu oikein, mutta niiden otsikkokentässä on väärennetyt arvot pituudesta ja sirpaleiden paikasta niistä koottavassa paketissa. Kun tällaisista sirpaleista kasataan ehjää pakettia, niin virheellisten tarkistusten vuoksi sirpaleiden dataa kirjoitetaankin niille varatun puskurin ulkopuolelle, väärään paikkaan muistiin, jolloin käyttöjärjestelmä yleensä kaatuu [HPW02]. Ensiapuna hyökkäyksen torjunnassa voitiin palomuurilla estää sirpaloitujen pakettien liikenne ja koko ongelmakin oli pian korjattu pienellä käyttöjärjestelmän tietoturvapäivityksellä IP-pakettien sirpaloinnista huolehtivaan koodiin.

Sen sijaan että looginen DoS-hyökkäys kohdistetaan yksittäiseen palveluntarjoajaan se voidaan kohdistaa itse verkon kykyyn välittää paketit oikeaan kohteeseen. Tällöin hyödynnetään puutteellista todennusta ja päästään syöttämään virheellistä tietoa reititysprotokollille tai nimipalvelinjärjestelmälle (Domain Name System, DNS), jolloin paketit ohjautuvatkin tarkoitetun kohteen sijasta esimerkiksi hyökkääjän hallinnoimaan osoitteeseen.

Reitittimet vaihtavat keskenään reititystietoja, jotta ne osaisivat ohjata saamansa paketit oikeaan suuntaan. Jos hyökkääjä pääsee muokkaamaan näitä paketteja tai syöttämään reitittimelle tekaistuja ohjaustietoja, niin reititin ei enää välitä paketteja oikeaan kohteeseen. Reititysprotokollia on useita erilaisia ja niiden herkkyydessä sietää väärin toimivia reitittimiä tai tahallaan syötettyjä valheellisia ohjaustietoja on eroja. Hyökkäyksen torjumiseksi reitittimien keskenään vaihtamat ohjaustiedot tulisi todentaa ja varmistaa niiden eheys [PaH02].

Hyökkääjä voi myös estää kohdetta saamasta sille osoitettuja paketteja estämällä lähettäjiä saamasta selville kohteen IP-osoitetta. Hyökkääjä syöttää virheellisiä osoitetietoja kohteen verkosta vastaavalle nimipalvelimelle [Che06]. Tällöin nimipalvelin antaa väärän osoitteen vastauksena toiselta nimipalvelimelta tulleeseen kyselyyn, jossa halutaan tietää kohteen nimeä vastaava IP-osoite. Väärennetty osoite voi johtaa hyökkääjän hallinnoimalle palvelimelle tai olla osoite, jota ei ole edes olemassa.

Loogisista DoS-hyökkäyksistä voi olla hankalaa selvittää mikä ne aiheutti, mutta sen selvittyä ongelma on usein suhteellisen helppo korjata. Ohjelmointivirheen korjaus, protokollan toiminnan pieni muutos tai usein pelkkä virheen päivittävä tietoturvapäivitys poistaa ongelman. Hyökkäyksiltä vältytään usein pitkälle kokonaan huoleh-

timalla, että käyttöjärjestelmästä ja sovelluksista on käytössä viimeisimmät versiot ja että tietoturvapäivitykset ovat ajan tasalla.

5.2 Tulvitushyökkäykset

Toisin kuin looginen DoS-hyökkäys tulvitushyökkäys perustuu valtavaan lähetettyjen pakettien määrään, valitun kohteen hukuttamiseen suuren pakettitulvan alle. Paketteja ei ole muokattu mitenkään älykkäästi, vaan ne ovat hyvin toistensa kaltaisia eivätkä juurikaan eroa normaalista tietoliikenteestä ja muista uhrin saamista paketeista. Tulvitushyökkäykset näyttävätkin usein vain tavalliselta, runsaan liikenteen aiheuttamalta ruuhkalta. Siksi niiden torjuminen on loogisia hyökkäyksiä vaikeampaa ja 90-luvun lopulta lähtien onkin tehty lukuisia isoja julkisuutta saaneita hyökkäyksiä.

Ensimmäinen suurta huomiota saanut hyökkäys tapahtui elokuussa 1999, jolloin Minnesotan yliopisto joutui laajamittaisen hyökkäyksen kohteeksi ja sen verkko oli kaksi päivää käytännössä suljettuna. Seuraavan vuoden helmikuussa useat tunnetut nettisivustot, kuten Yahoo, eBay, CNN ja Amazon joutuivat massiivisen DoS-hyökkäyksen kohteeksi. Verkkosivustot joutuivat pahimmillaan ottamaan vastaan yli yhden gigabitin verkkoliikennettä. Sitä määrää niiden ei oltu suunniteltu kestävän ja ne ruuhkautuivatkin täysin. Hyökkäys oli niin raju ja laajamittainen, että jopa koko internetin tasolla liikenne hidastui selvästi ja pahimpina hyökkäysten hetkinä odotusajat olivat reilusti yli 20 prosenttia hitaampia kuin normaalisti [Gar00]. DoS-hyökkäyksistä ei ole päästy näidenkään jälkeen ja niitä voidaan käyttää myös poliittisiinkin tarkoituksiin, kuten Viron joutuessa vuonna 2007 muutaman viikon kestäneen hyökkäyksen kohteeksi sen siirrettyä Tallinnassa sijaitsevaa Pronssisoturipatsasta toiseen paikkaan [Tra07].

Tulvitukseen perustuvia DoS-hyökkäyksiä voidaan kohdistaa loogisten hyökkäysten tavoin myös internetin infrastruktuuria kohtaan. DNS:ää vastaan hyökätään usein ja vuonna 2002 sen kaikki 13 juurinimipalvelinta joutuivat erittäin kovan DoS-hyökkäyksen kohteeksi [VSS02]. Juurinimipalvelimia ei onnistuttu kaatamaan täydellisesti, mutta osan toiminta hidastui niin, että nimipalvelupyyntöihin vastaaminen hidastui selvästi. Hyökkäyksessä oli poikkeuksellista, että se oli kohdistettu kaikkiin juurinimipalvelimiin. Nimipalvelimet ovat pienemmässä määrin lähes jatkuvasti DoS-hyökkäysten kohteena.

Tulvituksen tarkoituksena on kasvattaa kohteelle tulevien pakettien määrä niin suu-

reksi, että se joutuu käyttämään ylettömästi resursseja, kuten CPU-aikaa ja muistia, niiden käsittelemiseen. Muut kohteen palveluja käyttävät asiakkaat joutuvat kilpailemaan samoista resursseista ja valtavasta hyökkäysliikenteen määrästä johtuen ne eivät joko saa palvelua ollenkaan tai sen saaminen hidastuu. Myös kohteen tiedonsiirtokapasiteetti voi kulua turhaan pakettiliikenteeseen, jolloin kaikkien palveluihin oikeutettujen asiakkaiden paketit eivät pääse ollenkaan perille asti.

Yksikin kone voi tuottaa runsaasti pakettiliikennettä, joka voi ainakin ruuhkauttaa tai jopa tukkia kokonaan hitaan nettiyhteyden päässä olevan palveluntarjoajan. Hyökkääjän tuottama pakettitulva olisi helppo suodattaa pois sopivassa matkan varrella olevassa reitittimessä lähettäjän IP-osoitteen perusteella ellei sen väärentäminen olisi erittäin helppoa. Peittääkseen jälkiään vielä enemmän hyökkääjä voi myös käyttää kolmatta osapuolta, eräänlaista välikättä hyökkäyksessään. Tällainen niin sanottu *heijastushyökkäys* perustuu siihen, että hyökkääjän välikädelle lähettämä paketti tuottaa vastauksen, jonka välikäsi lähettää uhrille. Välikäsi on itsekin eräänlainen uhri, toissijainen kohde, jota käytetään hyväksi. Heijastushyökkäys voidaan toteuttaa esimerkiksi ICMP:n echo-paketeilla. Hyökkääjä lähettää välikädelle ICMP echo request -paketin (ping), jossa on väärennetty lähettäjän osoitteeksi hyökkäyksen kohteen, uhrin, IP-osoite. Välikäsi vastaa saamaansa pakettiin ICMP:n echo response -paketilla ja lähettää sen uhrille.

Hyökkääjä ei välttämättä saa tuotettua tarpeeksi paketteja, että ne aiheuttaisivat uhrille isompaa haittaa, mutta heijastusta voidaan käyttää hyökkäyksen voimistamiseen, niin että yksi hyökkääjän lähettämä paketti tuottaa uhrille monta pakettia. Tällainen on esimerkiksi Smurffi-hyökkäys, jossa hyökkääjä lähettää echo request -pakettinsa välikätenä käytettävän verkon yleislähetysosoitteeseen, jolloin se välittyy reitittimen ohjaamana kaikille verkon koneille. Kun jokainen kone lähettää vastauksensa uhrille, on yksi hyökkääjän lähettämä paketti tuottanut uhrille monikertaisen määrän paketteja. Ongelmaa voidaan torjua muun muassa rajoittamalla reitittimiä välittämästä eteen päin verkon ulkopuolelta tulevia yleislähetysosoitteeseen osoitettuja paketteja sekä jättämällä vastaamatta echo-paketteihin välikätenä toimivissa koneissa [Cer00].

Hyökkääjä voi kasvattaa DoS-hyökkäyksensä tehoa entisestään käyttämällä useampaa tietokonetta. Mitä useampi taho tuottaa liikennettä sitä suuremmaksi saadaan kasvatettua liikenne uhrin päässä. Tällaista useiden tietokoneiden koordinoidusti tekemää tulvitushyökkäyksestä kutsutaan *hajautetuksi palvelunestohyökkäykseksi* (Distributed Denial of Service, DDoS). Käytännössä kaikki tulvitushyökkäykset ovat

nykyään DDoS-hyökkäyksiä. Hyökkäyksessä mukana olevia tietokoneita käytetään usein DDoSsissa niiden omistajan tietämättä. Hyökkääjä on kaapannut tietokoneet ja voi ohjata niiden toimintaa. Tällaisia kaapattuja tietokoneita kutsutaan *boteiksi* (tai zombeiksi tai agenteiksi) ja ne muodostavat *bottiverkon*. DDoS-hyökkäys muistuttaa erittäin paljon normaalia verkon liikennettä ja uhrin on erittäin vaikea erottaa sitä normaalista ruuhkasta ja botit voivat käyttää jopa oikeita IP-osoitteitaan niiden väärentämisen sijasta. Jos bottiverkon koneet sijaitsevat vielä hyvinkin hajautetusti ympäri internetiä, DDoS-hyökkäyksen kohteeksi joutuneen uhrin on lähes mahdotonta tunnistaa niitä hyökkääjiksi ja jäljittää niitä.

Bottiverkko koostuu siis hyökkääjän kontrolloimasta bottiverkosta, joka voi koostua jopa kymmenistä tuhansista kaapatuista koneista. Viattomasta tietokoneesta tehdään hyökkääjän kontrolloima botti asentamalla siihen sopiva haittaohjelma. Se onnistuu hyödyntämällä jotain tietoturva-aukkoa tai haavoittuvuutta. Haittaohjelman asennuttua se ilmoittaa itsestään hyökkääjälle ja pyrkii pysyttelemään piilossa odottaen hyökkääjän lähettämiä toimintaohjeita. Alussa haavoittuvien tietokoneiden etsiminen ja bottiohjelmien asennus tehtiin manuaalisesti, mutta myöhemmin tämäkin automatisoitui. Haittaohjelman saastutettua tietokoneen ja liitetyä sen bottiverkoon se etsii uuden uhrin, johon tartuttaa itsensä. Näin bottiverkko kasvaa ja laajenee automaattisesti. Ensimmäisissä bottiverkoissa kommunikointi osapuolten välillä tapahtui sellaisten porttien kautta, joita eivät normaalisti käyttäneet mitkään muut ohjelmat. Vaikka itse viestit oli joissakin bottiverkoissa jopa salattu, niin tunnettujen porttien käyttäminen teki bottiverkosta haavoittuvaisen paljastumiselle [HoW01].

Myös runsas bottiverkon sisäinen liikenne oli paljastumiselle altistava tekijä. Tätä mahdollisuutta pienensivät hyökkääjän ja bottien välissä olevat *käsittelijät*, jotka olivat myös kaapattuja päätelaitteita. Hyökkääjä hallinoi käsittelijöitä, jotka välittävät bottien ja hyökkääjän välisiä viestejä [HoW01]. Käsittelijät pyrittiin sijoittamaan verkkopalvelimille tai reitittimiin, jotka pystyivät käsittelemään suuriakin liikennemääriä [SpL04]. Näin hyökkääjältä käsittelijöille kulkeva liikennemäärä pysyi kohtuullisissa määrissä ja käsittelijöiden lukuisille boteille lähettämä liikenne hukkui niiden isäntäkoneen muutenkin suureen tietoliikenteen määrään.

Käsittelijöissäkin oli vielä riskinsä. Koska ne tunsivat suuren joukon botteja, niin käsittelijän paljastuttua paljastuivat kaikki sen tuntemat botitkin. Vuonna 2000 havaittiin ensimmäiset bottiverkot, jotka olivat hylänneet käsittelijöiden käytön viestien välittämisessä. Sen sijaan ne käyttivät IRC-protokollaa (Internet Relay Chat).

IRC-kanavia käytettiin käsittelijöiden sijasta yhdeydenpitoon bottien ja hyökkääjän välillä. Enää ei ollut tarvetta käyttää liikennöintiin mitään epäilyttävää porttia, vaan voitiin käyttää suosittua ja huomiota herättämätöntä IRC-protokollan porttia, jolloin bottiverkon komentoliikennettä oli vaikeampi jäljittää. IRC-palvelimilla on normaalistikin runsaasti liikennettä ja hyökkääjän aiheuttama liikennemäärä peittyy helposti muun liikenteen joukkoon. Myöskään käsittelijöiden ylläpitämää listaa sen hallinnoimista boteista ei enää tarvita, sillä hyökkääjä voi selvittää botit kirjautumalla oikealle kanavalle [HoW01].

IRC-kanavien käyttämisessä bottiverkkojen sisäisessä kommunikoinnissa on kuitenkin edelleen ongelmia. Kanavat ovat edelleen hyökkääjän kannalta arimpia bottiverkon pisteitä, sillä kanavan paljastuessa myös kaikki sitä käyttävät botit paljastuvat. Viime vuosina havaitut bottiverkot ovatkin luopuneet IRC-kanavien käytöstä ja siirtyneet hyödyntämään vertaisverkkoja [WSZ10].

Huomattakoon, että vaikka DDoS-hyökkäyksiä tehdäänkin pääsääntöisesti kaapatuista koneista muodostetuilla bottiverkoilla, niin hyökkäyksiä voidaan tehdä myös ilman koneiden luvatonta kaappausta. Niin sanotussa vapaaehtoisessa bottiverkossa koneen omistaja antaa suostumuksensa koneensa käyttämiseen DDoS-hyökkäyksessä. Esimerkiksi Anonymous-ryhmässä, joka hyökkäsi muun muassa luottokorttiyhtiöitä vastaan vuonna 2010 niiden katkaistessa Wikileaksin rahaliikenteen, jokainen hyökkäykseen osallistuva antaa koneensa vapaaehtoisesti käytettäväksi DDoS-hyökkäyksessä [Eco10].

DDoS-hyökkäyksiä on monenlaisia. Aiemmin oli jo puhetta ICMP-paketeilla tehtävästä tulvituksesta ja sitä voidaan tehdä myös bottiverkoilla. Tällöin välikätenä toimivat toissijaiset uhrit sijoittuvat bottien ja uhrien väliin [Pax01]. Toinen suosittu ja jo kauan tunnettu DDoS-hyökkäys perustuu TCP-protokollan kolmivaiheisen kättelyn väärinkäyttämiseen. Siinä hyökkääjä lähettää uhrille TCP SYN -paketin, jossa on lähettäjän IP-osoitteena osoite, jota ei ole käytössä tai joka ei vastaa. Uhri tallettaa muodostettavana olevaan yhteyteen liittyviä tilatietoja puskuriin ja lähettää TCP ACK -paketin. Tähän ei kuitenkaan tule koskaan kuittausta ja uhri jää puoliavoimeen tilaan. TCP-protokollassa on ajastin, joka lopulta purkaa tällaiset puoliavoimet yhteydet, mutta kuitenkin lukuisten bottiverkon lähettämien TCP SYN -pakettien takia varattu puskuritila täyttyy nopeasti eivätkä muut asiakkaat saa enää yhteyttä uhriin [SKK97].

Edellä esitetyt DDoS-hyökkäykset liittyivät OSI-mallin verkkokerroksen (ICMP) tai kuljetuskerroksen (TCP SYN) protokolliin, mutta myös sovelluskerroksella voidaan

toteuttaa DDoS-hyökkäyksiä. Hyökkääjä voi esimerkiksi kuormittaa WWW-sivuilla käytettyä http-protokollaa runsaalla sivupyyntöjen määrällä. Tämä kuluttaa uhrin tietoliikennekapasiteettia, eikä se voi enää vastata muiden asiakkaiden sivupyyntöihin. Hyökkäystä voi joissakin tapauksissa tehostaa liittämällä sivupyyntöihin ison tiedoston lataamisen. Hyökkäyksestä voi tehdä luonnollisemman ja enemmän normaalia verkkoliikennettä muistuttavan pyytämällä palvelimelta myös WWW-sivuston alasivuja eikä pelkästään etusivua. Palvelimen ruuhkauttaminen sivupyynnöillä ei kuitenkaan onnistu, jos lähettäjän IP-osoite on väärennetty. Ennen sivupyyntöjen lähettämistä on muodostettava onnistunut TCP-yhteys kolmivaiheisella kättelyllä eikä se onnistu, jos molemmat osapuolet eivät osallistu siihen. Näin ollen hyökkäys on toteutettava bottiverkon avulla, koska botin IP-osoitetta ei tarvitse väärentää. Yhtä tai muutamaa hyökkäävää konetta käytettäessä niiden muodostama liikenne erottuisi selvänä piikkinä ja ne olisi helppo suodattaa oikean IP-osoitteen perusteella pois. Bottiverkkoa käytettäessä hyökkäys näyttää mahdollisimman aidolta ja normaalilta liikenteeltä ja mistään yksittäisestä IP-osoitteesta ei tule huomiota herättävän paljon sivupyyntöjä [PLR07].

DoS-hyökkäykset ovat olleet tunnettuja jo 90-luvulta lähtien ja ne ovat olleet siitä lähtien runsaan tutkimuksen ja mielenkiinnon kohteena. Loogiset hyökkäykset on usein suhteellisen helppo torjua esimerkiksi muuttamalla protokollan toimintaa tai korjaamalla ohjelmointivirhe, mutta DDoS-hyökkäyksiin ei ole löydetty mitään yhtä ainoaa jokaiseen tilanteeseen ja ympäristöön toimivaa ratkaisua. DDoS-hyökkäyksiä vastaan taistelemisen voisi jakaa kolmeen osaan: hyökkäysten estämiseen, niiden havaitsemiseen sekä hyökkäyksen torjumiseen [PLR07].

Hyökkäyksiä voidaan yrittää estää paikallisten verkkojen rajareitittimissä seuraamalla pakettien IP-osoitteita ja suodattamalla paketit, jotka tulevat suunnasta, josta niiden ei pitäisi lähettäjäksi merkityn IP-osoitteen perusteella tulla [FeS00]. Samaa IP-osoitteeeseen perustuvaa suodatusideaa voidaan laajentaa internetin paikallisista verkoista itsenäisten järjestelmien (Autonomous System, AS) välillä oleviin rajareitittimiin [PaL01]. Tällaiset väärennettyyn IP-osoitteen havaitsemiseen perustuvat suodatuskeinot ovat kuitenkin varsin heikkoja bottiverkoilla tehtäviä hyökkäyksiä vastaan, sillä niissä lähettäjän IP-osoitetta ei tarvitse väärentää.

DDoS-hyökkäysten havaitseminen ei ole ihan helppoa, sillä ne muistuttavat erehdyttävästi normaalia liikennettä, on vaikea erottaa onko kasvanut liikenne normaalia vai hyökkäyksestä aiheutuvaa. DDoS-hyökkäysten havaitsemiseen on kaksi tapaa. Joko hyökkäys havaitaan kullekin DDoS-hyökkäykselle tyypillisistä erityispiirteistä

tai mallinnetaan verkon toimintaa normaaliaikana, jolloin hyökkäyksen aiheuttamat muutokset voidaan havaita.

Eräs tapa jonkin tietyn hyökkäyksen havaitsemiseksi on seurata verkkoliikennettä jollain mittarilla ja etsiä hyökkäyksen aiheuttamaa muutosta, esimerkiksi TCP-protokollassa SYN-pakettien suhdetta FIN- ja RST-paketteihin [WZS02]. Näin voidaan havaita SYN-paketeilla tapahtuvan hyökkäyksen aiheuttama epätavallisen suuri SYN-pakettien suhde muihin paketteihin. Hyökkääjä voi kuitenkin kiertää tämän lähettämällä SYN-paketin lisäksi sopivasti FIN- tai RST-paketteja, jotta liikenne vaikuttaisi normaalilta.

Toisessa, poikkeavuuksien havainnointiin perustuvassa havaitsemistavassa on ensimmäisenä vaiheena tilastollisesti normaalin liikenteen mallintaminen ja sopivien tätä kuvaavien parametrien valitseminen. On esitetty lukuisia tapoja valita sopivia parametreja sekä mitata normaalin liikenneprofiiliin ja mitatun liikenteen välistä eroa. Vaikeutena on saada normaalin liikenteen profiili niin laajaksi, että se kattaa kaikentyyppiset normaalioloissa esiintyvät liikennevirrat, mutta tunnistaa hyökkäykset. Ongelmaa voidaan kiertää kuvaamalla liikennettä yhä hienojakoisemmin ja useammalla parametrilla, mutta tämä johtaa laskennallisesti ja ajallisesti yhä suurempaan resurssien kulutukseen ja alkaa jo itsessään muodostua järjestelmälle taakaksi.

DDoS-hyökkäyksiä vastaan taisteleminen on sitä tehokkaampaa mitä lähempänä hyökkääjää se voidaan tehdä. Hyökkäystä on toisaalta vaikea tai lähes mahdotonta erottaa normaalista liikenteestä lähellä hyökkääjää. Lähellä uhria hyökkäys on helppo havaita, mutta vaikea torjua. Voidaan yrittää esimerkiksi sopivaa liikenteen suodattamista tai jopa liikennöintikapasiteetin lisäämistä ottamalla käyttöön uusia palvelimia. Hyökkäyksen torjuminen puolessa välissä uhria ja hyökkääjää eli niiden välisissä reitittimissä tapahtuva liikenteen suodattaminen vaatii enemmän osapuolten välistä kommunikointia. Suodatus tapahtuu joko uhrin ilmoituksen perusteella tai pohjautua verkon suorittamaan liikenteen älykkääseen havainnointiin [PLR07].

Miksi tulvitukseen perustuvat DoS-hyökkäykset ovat sitten niin vaikeasti torjuttava ja edelleenkin kiusallinen ongelma? Ongelman juuret johtavat siihen miten internet syntyi, sen syntyvaiheisiin ja silloin asetettuihin tavoitteisiin. Jos internet keksittäisiin nyt, sille asetettaisiin varmasti osin erilaisia tavoitteita.

Vuonna 1969 syntynyt ARPANET-verkko oli alunperin Yhdysvaltojen puolustusvoimien rahoittama hanke, jonka yksi tavoitteista oli kehittää verkko, joka pysyisi toimintakykyisenä, vaikka osia verkosta menetettäisiinkin. Verkon oli kuljetettava viestit perille vaikka hitaamminkin, jos osapuolten välillä oli edes jokin fyysisesti

toimiva reitti. Ratkaisuksi kehitettiin pakettikytkentäinen verkko, jossa toisin kuin piirikytkentäisessä verkossa, datasta muodostettiin pieniä paketteja ja kukin kulki verkon läpi itsenäisesti vastaanottajalle.

Merkittävimmät internetiin vaikuttavat päätökset tehtiin, kun päätettiin kehittää tehokas tekniikka yhdistämään pakettikytkentäinen ARPANET-verkko muihin verkkoihin, ennen kaikkea pakettiradioverkkoon [Cla88]. Yhdistettävälle verkolle asetettuja tavoitteita oli useita, mutta nykyisten DDoS-hyökkäysten kannalta oleellisimpia oli kolme: 1) internetin on toimittava, vaikka joitakin osia siitä lakkaisi toimimasta 2) internetin on tuettava erilaisia palvelutarpeita ja 3) internetillä ei ole keskitettyä hallintoa.

ARPANETin pakettikytkentäisyys otettiin internetin pohjaksi. Pakettien luotettava kuljettaminen verkossa vaatii erilaisten hallintotietojen, kuten esimerkiksi lähetettyjen pakettien määrä ja kuittaukset sekä vuonhallinta, ylläpitämistä. Verkosta saadaan vikasietoisempi, kun hallintotiedot keskitetään yhteyden päissä oleviin laitteisiin ja niiden välissä olevan verkon toiminta pidetään mahdollisimman yksinkertaisena. Päätelaitteissa oleva TCP-protokolla vastaa luotettavasta yhteydestä ja IP-protokollan avulla verkon reitittimet vain välittävät saamansa paketit perille niissä olevan IP-osoitteen perusteella. Protokollien erottaminen oli myös vastaus erilaisten sovellusten vaihteleviin verkolle asettamiin vaatimuksiin. Internet pystyi palvelemaan sekä luotettavaa pakettien välitystä tarvitsevia palveluja että palveluja, joille nopeus on tärkeämpää kuin jokaisen paketin perille pääsy. Luotettavuutta saatiin TCP-protokollalla ja nopeutta jättämällä pois sen tarjoama luotettava pakettien välitys eli käyttämällä UDP-protokollaa. Verkon kaikki äly on siis yhteyden päissä ja välissä olevat reitittimet ovat hyvin yksinkertaisia, jotta ne voisivat tehokkaasti välittää paketteja. Ennen kaikkea reitittimiltä ei edellytetä minkäänlaista IP-osoitteen alkuperän tai oikeellisuuden tarkistamista. Tämän seurauksena IP-osoite voidaan väärentää eikä vastaanottaja voi luottaa lähettäjän IP-osoitteen oikeellisuuteen. Pakettien reitittämiseen ei myöskään liity mitään jäljitettävyyttä, joten paketin käyttämä reitti ei ole jälkikäteen selvitetävissä. Lisäksi verkon reitittimet eivät tiedä, haluaako vastaanottaja edes saada sille osoitettua pakettia. Internet onkin erittäin tehokas kuljettamaan paketteja perille, halusi vastaanottaja niitä tai ei.

Yksi internetille asetettu tavoite oli myös sen hallinnon hajautuneisuus. Tämä onkin toteutunut hyvin, verkon liikenteestä huolehtivat reitittimet kuuluvat eri organisaatioiden hallintaan. Tästä on toisaalta seurannut se, että verkkoa ei hallitse kukaan

kokonaisuutena. Laajalta alueelta tulevat DDoS-hyökkäykset kulkevat useamman jotain verkon osaa hallinnoivan ylläpitäjän kautta. Tulvitushyökkäyksiä vastaan taisteleminen edellyttäisi ylläpitäjiltä yhteistyötä, mutta se ei aina ole helppoa useiden toimijoiden, niiden noudattaman erilaisen lainsäädännön ja byrokratian vuoksi.

Internetin alkuaikoina verkko oli vielä hyvin pieni ja käyttäjiä suhteellisen vähän eikä ollut tarvetta juuri salasanaa monimutkaisemmille tietoturvaratkaisuille. Suunnittelijat eivät arvanneet kuinka keskeiseen osaan internet vielä tulisi maailmassa ja että sitä käytettäisiin lukuisilla tärkeillä yhteiskunnan osa-alueilla kuten esimerkiksi liike-elämässä, energiantuotannossa ja finanssimaailmassa. Internetin syntyhetkillä sen tietoturvalle asetettiin huomattavasti pienempiä vaatimuksia kuin mitä tietoturvalta nykyään vaaditaan kaikilla niillä elämänalueilla missä internetiä käytetään [Lip02].

6 Palvelunestohyökkäykset IEEE 802.11 -verkossa

Langaton verkko on kaapeliverkkoa alttiimpi palvelunestohyökkäyksille. Ensinnäkin fyysiset kaapelit eivät rajoita lähetysten etenemistä. Radioaallot leviävät joka suuntaan ympäristöönsä ja jokainen kantaman sisällä oleva voi kuunnella tai häiritä muiden lähettämiä viestejä. Toisaalta kaapeliverkossa hyökkääjä voi olla vaikka toisella puolella maapalloa, mutta WLANissa hyökkääjän on oltava suhteellisen lähellä, jossain kantaman sisällä. Toiseksi langattoman verkon tiedonsiirtokapasiteetti on huomattavasti kaapeloitua lähiverkkoa ja internetiä pienempi, WLAN kärsii helpommin ruuhkasta. Kolmas WLANia DoS-hyökkäyksille altistava tekijä on todennuksen ja salauksen puute MAC-kerroksen kehyksistä. Ainoastaan datakehysten data on salattu. Hallinta- ja kontrollikehyksiä ei ole salattu eikä todennettu mitenkään.

802.11-lähiverkon kehityksessä keskityttiin vuosien varrella lähinnä vain datan salaukseen, luotettavaan osapuolten väliseen todennukseen ja kehysten eheyteen. WLANissa oli aluksi vain WEP huolehtimassa tietoturvasta. Sen osoittauduttua nopeasti puutteelliseksi tuli sitä korvaamaan väliaikaisesti WPA, jossa oli käytössä luotettavampi 802.1X:ään perustuva todennustapa sekä parannettu salausalgoritmin käyttötapa. Myös WEPissä ollut eheydentarkistus vaihdettiin WPA:ssa parempaan, sillä se havaitsi vain satunnaiset bittivirheet eikä tarkoituksellista viestien muuttamista. WPA:ssakin alkoi ilmetä ongelmia ja sen korvasi WPA2, joka oikein käytettynä takaa luotettavan osapuolten välisen todennuksen, datan salauksen sekä viestien muuttumisen havaitsemisen. Johtui muuttuminen sitten häiriöisestä siirtotiestä tai

hyökkääjän tahallisesta lähetyksen muuttamisesta. WPA2 on kuitenkin edelleen altis verkon saatavuutta uhkaaville DoS-hyökkäyksille. Tässä luvussa tarkastellaan erilaisia WLANissa edelleenkin ilmeneviä DoS-hyökkäyksiä sekä keinoja niiden havaitsemiseen ja torjumiseen.

6.1 Radiohäirintä

Radiohäirinnällä (jamming) tarkoitetaan laitteiden yhteisesti jakaman siirtotien käyttämisen tai sille pääsyn fyysistä häiritsemistä. Koska siirtotiellä liikennöidään radiotaajuuksilla, voi kantaman sisällä vain yksi laite kerrallaan lähettää muiden kuunnellessa. Siirtotiellä tapahtuvien yhteentörmäysten välttämiseksi WLANissa käytetään yleensä hajautettua koordinointifunktiota vuoron saamiseksi siirtotien käyttöön. Joko vähennetään yhteentörmäyksiä CSMA/CA:lla seuraamalla siirtotien vapautta tai estetään yhteentörmäykset lähes kokonaan tehostamalla CSMA/CA:ta RTS/CTS:llä, jolloin siirtotien käyttöön on saatava vastaanottajan lupa.

Erilaiset radiohäirintää käyttävät DoS-hyökkäykset voidaan jakaa kahteen pääluokkaan sen perusteella onko hyökkääjän oltava tietoinen verkossa käytettävästä siirtotien vuoronvarausprotokollasta. *Tyhmää radiohäirintää* voidaan suorittaa, ilman että hyökkääjän tarvitsee olla tietoinen käytetystä protokollasta. *Älykkäässä radiohäirinnässä* hyökkääjän on tiedettävä onko verkossa käytössä pelkkä CSMA/CA vai onko sitä tehostettu vielä RTS/CTS:llä. Lisäksi hyökkääjän on älykkäässä häirinnässä seurattava liikennettä ja ajoitettava hyökkäyksensä tiettyihin protokollan vaiheisiin.

6.1.1 Tyhmä radiohäirintä

Tyhmää radiohäirintää käyttäessään hyökkääjän ei tarvitse olla mitenkään tietoinen verkossa siirtotien seurantaan tai varaamiseen käytetystä protokollasta. Tyhmällä radiohäirinnällä aiheutetaan joko muiden lähettämien signaalien sotkeentuminen lukukelvottomaksi lähettämällä samanaikaisesti omaa häirintäsignaalia tai estetään muita saamasta siirtotietä ollenkaan käyttöönsä sen ollessa koko ajan varattuna. Hyökkääjän lähettämä signaali voi olla joko oikeita kelvollisia paketteja tai pelkkiä satunnaisia bittejä [XTZ05]. Erilaisia tyhmiä radiohäirinnällä tehtäviä palvelunestohyökkäyksiä voidaan kuvata kolmikolla (E, t_h, t_t), missä E (wattia) on hyökkääjän radiolähettimen häirintäsignaalin energia, t_h (sekuntia) häirintäsignaalin kesto ja t_t (sekuntia) tauon pituus kahden peräkkäisen häirintäsignaalin välillä.

Signaalin energian E on oltava tarpeeksi suuri, jotta häirintä tuottaa täydellisen palvelunestohyökkäyksen, jolloin yksikään lähetys ei pääse virheettömänä perille. Verkon täysin lamaannuttavan E:n minimiarvo on tapauskohtainen rippuen verkosta ja olosuhteista.

Kaikkein yksinkertaisin radiohäirintä on *jatkuva* verkon toimintataajuudella lähetettävä signaali, jota kuvaa kolmikko $(E, \infty, 0)$. Jos siirtotie on hyökkäyksen käynnistyessä jonkin laitteen käytössä, sotkee hyökkäys lähettäjän signaalin täysin eivätkä vastaanottajat saa siitä mitään selvää virheenkorjauksesta huolimatta. Jos siirtotie on vapaana hyökkäyksen alkaessa, eivät muut laitteet yritä päästä siirtotielle havaitessaan sen olevan koko ajan varattuna [AST04]. Hyökkääjä ei siis noudata protokollan mukaista toimintatapaa, vaan tunkeutuu siirtotielle, vaikka siellä olisikin jo joku toinen eikä noudata yhtäjaksoiselle lähettämiselle asetettuja maksimiaikoja antaen välillä muillekin mahdollisuuden päästä siirtotielle, vaan käyttää siirtotietä itse koko ajan.

Jatkuvan häirintäsignaalin sijasta hyökkääjä voi tehdä *pulssimaista* radiohäirintää lähettämällä lyhyehkön signaalin säännöllisin väliajoin, jossa tyypillisesti t_h on paljon pienempi kuin t_t. Pulssimainen radiohäirintä lamaannuttaa verkon täysin, jos t_t on tarpeeksi lyhyt, esimerkiksi alle DIFSin mittainen (katso kuva 4 sivulla 12 ja kuva 5 sivulla 14). Tällöin hyvinkin lyhytkestoinen häirintäsignaali saa muut verkon laitteet luulemaan verkon olevan koko ajan varattuna [AST04]. Verkon liikenteen voi estää myös vain osittain lyhentämällä t_h:ta ja ennen kaikkea pidentämällä t_t:tä, jolloin häirintä ei pidä siirtotietä jatkuvasti varattuna ja jotkin lyhyimmät lähetykset pääsevät häirinnästä huolimatta onnistuneesti vastaanottajalle. Pulssimaista radiohäirintää voidaan tehdä myös satunnaisemmin, jolloin t_h:n ja t_t:n arvoja voidaan vaihdella jatkuvasti [XTZ05]. Tällöin hyökkäys ei näytä täysin säännölliseltä ja sitä on vaikeampaa havaita. Hyökkääjä voi myös tehdä pulssimaista häirintää, joka reagoi siirtotien vapauteen. Hyökkääjä lähettää häirintäsignaalia havaittuaan siirtotien olevan vapaana DIFSin mittaisen ajan [AST04]. Tällöin häirintäsignaali joko sotkee jonkun toisen laitteen lähettämän datakehyksen, (katso kuva 4 sivulla 12) tai RTS:n (katso kuva 5 sivulla RTS/CTS) tai estää muita laitteita lyhentämästä omaa odotusaikaansa kilpailuikkunassa. Käytännössä häirinnästä tulee pulssimaista ja hyökkääjä lähettää häirintäsignaalia aina DIFSin mittaisen tauon jälkeen.

Edellä kuvatut verkon liikenteen kokonaan estävät tyhmän radiohäirinnän muodot ovat kaikki aktiivisia perustuen siihen, että riippumatta verkon liikenteestä ne pitävät käytännössä koko ajan siirtotien itsellään lähettämällä aktiivisesti signaalia.

Tyhmä radiohäirintä voi olla myös *reagoivaa*. Siinä hyökkääjä kuuntelee siirtotietä, ja vasta kun havaitsee siirtotiellä jonkun muun lähettämää signaalia, aloittaa häirinnän. Hyökkääjä lähettää lyhyen hetken häirintäsignaalia ja sotkee toisen lähetyksen, jonka jälkeen hiljentyy kuuntelemaan siirtotietä. Hyökkääjä ei siis yritäkään reagoivassa radiohäirinnässä estää muiden pääsyä siirtotielle pitämällä sitä itse koko ajan omassa käytössä, vaan sotkemaan aina muiden signaalin heti niiden lähettäessä jotain [XTZ05].

Hyökkääjän kannalta optimaalinen radiohäirinnällä tehtävä palvelunestohyökkäys estäisi verkon liikenteen täydellisesti, hyökkäystä ei olisi helppoa havaita ja torjua, eikä lähettimen energiankulutus muodostuisi ongelmaksi. Näitä kaikkia on mahdotonta saavuttaa yhtä aikaa tyhmällä radiohäirinnällä ja erilaiset häirintätavat eroavatkin toisistaan sen suhteen miten ne yltävät näihin tavoitteisiin.

Yleisesti ottaen tyhmä radiohäirintä on tehokasta, mutta sitä rasittaa sen vaatima suuri kokonaisenergiankulutus. Vaikka lähetysteho saattaisi sinällään olla kohtuullinen, niin häirintäsignaalia muodostetaan lähes jatkuvasti. Jatkuva häirintä kuluttaa kaikkein eniten energiaa ja pulssimainen jonkin verran vähemmän. Pulssimaisessa radiohäirinnässä energiankulutusta voidaan vähentää taukoja pidentämällä, mutta tällöin häirinnän vaikuttavuus voi kärsiä ja osa verkkoon lähetetyistä lähetyksistä voi päästä läpi häirinnästä huolimatta [AST04]. Reagoiva häirintä ei säästä niin paljoa energiaa, kuin voisi luulla, sillä siinä lähetin kuluttaa energiaa myös kuunnellessaan aktiivisesti siirtotietä havaitakseen signaalin. Reagoivan häirinnän etu muihin tyhmiin radiohäirintöihin onkin lähinnä sen vaikeampi havaittavuus [XTZ05]. Energiankulutus ei välttämättä ole hyökkääjälle ongelma, mutta lähetintä ei voi aina kytkeä verkkovirtaan tai muuhun vakaaseen energianlähteeseen, vaan on toimittava mobiilin laitteen akkujen varassa. Tällöin runsaasti energiaa kuluttavaa tyhmää radiohäirintää voidaan tehdä vain varsin rajallinen aika.

Hyökkäyksen havaitsemisessa vaikeutena voi olla sen erottaminen verkon muusta epätavallisesta käyttäytymisestä, kuten ruuhkasta, akun loppumisesta tai jonkin verkkolaitteen rikkoutumisesta. Tyhmä radiohäirintä voidaan kuitenkin tunnistaa ja erottaa luotettavasti muusta verkon toiminnasta seuraamalla samanaikaisia muutoksia pakettien lähetyssuhteessa (Packet Delivery Rate, PDR) ja siirtotiellä havaitun signaalin voimakkuudessa [XTZ05].

PDR tarkoittaa onnistuneesti perille lähetettyjen ja virheenkorjauksen jälkeen lukukelpoisten pakettien suhdetta niihin paketteihin, jotka eivät menneet virheettä perille, mutta jotka kuitenkin tunnistettiin saapuneiksi paketeiksi. PDR voidaan

mitata joko lähettäjän päässä vertaamalla lähetettyjen pakettien suhdetta niihin saatuihin kuittauksiin tai vastaanottajan päässä fyysisellä kerroksella vastaanotettujen pakettien suhdetta CRC-virheenkorjauksen jälkeen lukukelpoisiin paketteihin. Mitä suurempi PDR sitä useampi paketti menee virheettä perille. Jos yksikään paketti ei pääse perille, on PDR nolla. PDR pienenee huomattavasti ja laskee jopa nollaan häirinnän aikana, mutta se reagoi samalla lailla aseman liikkuessa kantaman ulkopuolelle tai virran loppuessa. Se ei siis yksinään riitä erottelemaan luotettavasti tyhmää radiohäirintää kaikista muista verkossa kohdattavista tilanteista [XTZ05].

Toinen yritys tyhmän radiohäirinnän havaitsemiseksi on seurata sen aiheuttamia muutoksia siirtotiellä havaittujen signaalien voimakkuuksiin. Siirtotiellä havaitusta radiosignaalista tehdään useita mittauksia, joiden perusteella lasketaan joko signaalin keskimääräinen voimakkuus tai signaalien kokonaisenergia. Ideana on selvittää näiden arvot normaalioloissa, jolloin ei vielä ole mitään radiohäirintää, ja saada vertailuarvot, joita sitten verrataan oletetun radiohäirinnän aikana mitattuihin arvoihin. Signaalin voimakkuuden vertailu ei kuitenkaan auta erottamaan luotettavasti tyhmää radiohäirintää kaikista verkossa ilmenevistä tilanteista. Sekä signaalin voimakkuuden keskiarvo että kokonaisenergia eivät pysty erottamaan jatkuvaa radiohäirintää esimerkiksi pahasta ruuhkasta. Lisäksi reagoiva häirintä muistuttaa erittäin paljon verkon normaalia ruuhkatonta liikennettä eikä näitäkään voida erottaa toisistaan selvästi [XTZ05].

Tarkastelemalla kuitenkin yhtä aikaa sekä signaalin voimakkuutta että PDR:ää voidaan erottaa tyhmä radiohäirintä muista verkon tietoliikenteen poikkeustilanteista. Kun PDR pienenee eli entistä vähemmän paketteja pääsee virheettä perille, niin normaalisti tähän on jokin luonnollinen syy, kuten aseman liikkuminen tai jonkin esteen tuleminen asemien väliin. Tällöin myös signaali heikkenee. Jos PDR laskee nollaan, tarkoittaa se normaalioloissa esimerkiksi aseman liikkumista kantaman ulkopuolelle, laitevikaa tai virran loppumista. Näissä tapauksissa myös signaalin taso tippuu dramaattisesti. Normaalisti siis PDR:n laskuun tai nollaan saakka laskemiseen liittyy aina myös signaalin selkeä lasku. Tyhmän radiohäirinnän aikana PDR alenee huomattavasti, mutta signaali ei kuitenkaan heikkene, vaan pysyy luonnottoman korkeana johtuen hyökkääjän siirtotielle syöttämästä energiasta [XTZ05].

Kun hyökkäys on havaittu, sitä voidaan yrittää torjua muutamilla keinoilla. Lähetystehoja voidaan kasvattaa, jolloin hyökkääjän häirintäsignaalista huolimatta vastaanottaja saa lähetykset lukukelpoisena. Tätä rajoittaa ISM-taajuusalueen radiolähettimille yleensä säädetty maksimiteho. Hyökkääjällä saattaa myös olla käytössä

laittoman tehokas lähetin. Teoreettisena ratkaisuna erityisesti pitkiä taukoja sisältävää pulssimaista häirintää vastaan saattaisi auttaa pienempien pakettien käyttäminen, jotka pääsisivät häirintäsignaalien välissä olevien taukojen aikana onnistuneesti perille. Yksinkertaisinta tyhmää radiohäirintää on kuitenkin torjua joko siirtymällä häiritsijästä kauemmaksi, käyttää suunnattua antennia tai paikallistaa hyökkääjän lähetin. WLANissa kantamat ovat kuitenkin suhteellisen pieniä ellei käytetä suunnattua antennia ja hyökkääjän on oltava varsin lähellä häirinnän onnistumiseksi. Voimakas häirintäsignaali voidaan paikallistaa ja eliminoida lähetin. Myös verkossa käytettyä kanavaa voidaan vaihtaa, mutta hyökkääjä voi seurata helposti perässä ja tämä auttaakin vain hetkeksi.

Huomattakoon, että vaikka WLANissa käytetäänkin hajaspektritekniikoita kantoaallon modulointiin ne eivät auta estämään tyhmää radiohäirintää. Alun perin hajaspektritekniikat kehiteltiin sotilasteollisuudessa 1900-luvun puolivälissä torjuntakeinoksi vihollisen radiohäirintää vastaan käyttämällä laajempaa taajuusaluetta ja joko piilottamalla signaali kokonaan (DSSS) tai vaihtamalla käytettyä osataajuutta jatkuvasti (FHSS) [Sch82]. WLAN toimii tietyillä tunnetuilla taajuuksilla ja WLANissa tukiasema kertoo sekä beaconissa että probe responsessa ympäristöön tarpeellisia tietoja hallinnoimastaan verkosta, jotta asemat voivat liikennöidä siinä. Myös hyökkääjä kuulee nämä samat tiedot ja saa selville muun muassa verkon käyttämän kanavan ja osataajuuksien välisen hyppelyjärjestyksen. Hajaspektritekniikoiden käyttö WLANissa liittyy ennemminkin niiden häiriösietoisuuteen ruuhkaisella ISM-taajuusalueella, kuin niiden antamaan suojaan tietoista radiohäirintää vastaan.

6.1.2 Älykäs radiohäirintä

Tyhmän radiohäirinnän ongelmia ovat sen vaatima suuri energiankulutus sekä hyökkäyksen suhteellisen helppo havaitseminen. Häirintää saatetaan myös tehdä turhaan, vaikka verkossa ei olisikaan mitään liikennettä eikä yksikään laite halua saada siirtotietä haltuunsa. *Älykkäällä radiohäirinnällä* pyritään saavuttamaan tyhmän radiohäirinnän edut, mutta välttämään sen ongelmia. Sen tarkoitus on lamauttaa verkon liikenne yhtä tehokkaasti, mutta kuluttaa vähemmän energiaa ja olla huomaamattomampi. Lisäksi hyökkäys voidaan kohdistaa tarkemmin vain tiettyyn laitteeseen, kun taas tyhmä radiohäirintä vaikuttaa aina jokaiseen kantaman sisällä olevaan laitteeseen valitsematta uhriaan. Tyhmä radiohäirintä toimii vain fyysisellä kerroksella ja perustuu häirintäsignaalin lähettämiseen tai siirtotien pitämiseen varattuna koko ajan. Älykkäässä radiohäirinnässä on edelleen tarkoituksena sotkea siirtotiellä

havaittu lähetys, mutta tehdä se vähemmän energiaa kuluttavasti ja kohdistaa häirintä vain tiettyihin siirtotiellä havaittuihin kehyksiin. Älykkäässä radiohäirinnässä on oltava tietoinen siirtotien varaukseen ja käyttöön liittyvistä CSMA/CA- ja RTS/CTS-protokollista ja se perustuu fyysisen kerroksen yläpuolella olevan MAC-kerroksen toiminnan tuntemiseen.

Älykkäässä radiohäirinnässä voidaan sotkea verkossa liikkuvia kehyksiä, mutta häirintä ajoitetaan tiettyihin data- tai kontrollikehyksiin. Hyökkääjä voi kuunnella verkon liikennettä ja sotkea omalla signaalillaan datakehyksen. Jos käytössä on pelkästään CSMA/CA, niin pienestä datakehyksestä voi onnistua vain loppuosan sotkeminen, koska hyökkääjän on vaihdettava tilansa siirtotien kuuntelemisesta sille lähettämiseen. Loppuosankin sotkeminen yleensä riittää, jotta vastaanottaja ei voi lukea kehystä kelvollisesti. Jos käytössä on RTS/CTS, niin hyökkääjä voi kuunnella verkon liikennettä ja ajoittaa häirintänsä heti CTS:n ja sitä seuraavan SIFS:n jälkeen [AST04]. Datakehys ei siis mene virheettä perille saakka, eikä vastaanottaja voi lähettää siihen lähettäjän odottamaa kuittausta. Muutamien uudelleenlähetysten jälkeen lähettäjä luopuu yrityksistä kokonaan, koska hyökkääjä sotkee aina datakehyksen lukukelvottomaksi.

Datakehysten sijasta hyökkääjä voi sotkea kuittaukset. Hyökkääjä seuraa verkon liikennettä ja sotkee häirintäsignaalillaan datakehyksen jälkeen tulevan ACK:n. Datakehyksen lähettäjä jää odottamaan vastaanottajan kuittausta, jota ei koskaan tule. Ensimmäinen datakehys menee virheettä perille vastaanottajalle, mutta kuittauksen puuttumisen vuoksi lähettäjä ei saa tätä koskaan tietoonsa. Tästä seuraa joitakin turhia uudelleenlähetyksiä, mutta lähettäjän näkökulmasta ne eivät mene koskaan perille, koska niihin lähetetyt kuittaukset sotketaan, ja se luopuukin lopulta kokonaan lähettämisestä luullen yhteyden olevan poikki [AST04].

Jos verkossa on käytössä myös RTS/CTS pelkän CSMA/CA:n lisäksi, niin se tarjoaa edellä mainittujen häirintäkeinojen lisäksi hyökkääjälle vielä yhden keinon vastaavanlaiseen palvelunestohyökkäykseen. Hyökkääjä seuraa verkon liikennettä ja havaittuaan RTS:n odottaa SIFSin mittaisen ajan ja lähettää lyhyen häirintäsignaalin. Häirintä sotkee CTS:n ja RTS:n lähettäjä ei näin ollen saa lupaa siirtotien käyttämiseen [AST04]. Periaatteessa hyökkääjän voisi ajatella tekevän älykästä radiohäirintää sotkemalla RTS:n, mutta koska RTS on erittäin pieni kehys, niin käytännössä hyökkääjä tuskin ehtii ensin havaita paketin olevan RTS:n, ja sen jälkeen vielä ajoissa lähettää oman RTS:n sotkevan häirintäsignaalinsa [AcT05].

6.2 Hallinta- ja kontrollikehysten väärentäminen

Yksi pahimmista puutteista 802.11-pohjaisessa langattomassa lähiverkossa on luotettavan kehyskohtaisen todennuksen puuttuminen. Kun asema haluaa liittyä tukiaseman verkkoon, niin sen todentamiseen ja myöhemmin viestien salauksessa tarvittavien avainten luomiseen on panostettu erittäin paljon, kuten aiemmin luvussa 4 on esitetty. Kun asema on todennettu ja se on liittynyt tukiaseman hallinnoimaan verkkoon, niin tukiaseman ja asemien väliset paketit todennetaan kuitenkin ainoastaan erittäin helposti väärennettävissä olevan MAC-osoitteen perusteella. Hyökkääjä voi helposti naamioitua toiseksi asemaksi, väärinkäyttää verkon muille asemille tarjoamia palveluja ja häiritä verkon liikennettä. Datakehyksissä data on salattu, mutta hallinta- ja kontrollikehyksiä ei salata mitenkään eikä mitään kehyksiä todenneta epäluotettavaa lähettäjästä kertovaa MAC-osoitetta lukuun ottamatta. Seuraavaksi esitellään kolme erilaista palvelunestohyökkäystä, jotka perustuvat MAC-osoitteen puutteelliseen todentamiseen, sekä niiden havaitsemis- ja torjuntakeinoja.

6.2.1 Virtuaalihäirintä

Virtuaalihäirinnän tavoite on osittain sama kuin tyhmässä radiohäirinnässä: estää uhria edes pyrkimästä siirtotielle. Tyhmässä häirinnässä uhri havaitsi siirtotien olevan fyysisesti varattuna hyökkääjän lähettäessä siirtotielle omaa signaaliansa. Virtuaalihäirintä ei perustu siirtotien fyysiseen käyttämiseen, vaan sen virtuaaliseen varaamiseen omaan käyttöön NAVin avulla. Seuraavaksi kuvattu hyökkäys perustuu aiemmin luvussa 2.4 käsiteltyyn CSMA/CA:lla tai sen laajennuksella RTS/CTS:llä tapahtuvaan siirtotien varaamiseen.

Virtuaalihäirinnässä hyökkääjä lähettää verkkoon RTS-, CTS-, data- tai kuittauskehyksen, jonka durationissa on tarpeettoman suuri kestoaika. Muut verkon laitteet päivittävät NAVinsa vastaamaan havaitsemaansa durationia ja odottavat sen osoittaman ajan ennen kuin yrittävät siirtotielle pääsyä [BeS03]. Näin hyökkääjä saa pidettyä yhdellä kehyksellä muut asemat suhteellisen kauan poissa siirtotieltä. Helpointa on käyttää RTS:ää tai CTS:ää, sillä ne ovat kaikkein pienimpiä paketteja. RTS:n käyttäminen on hyvä myös siksi, että sen kuultuaan tukiasema lähettää siihen protokollan mukaisesti vastauksena CTS:n, jolloin kaikki sen kuulevat asemat asettavat NAViinsa hyökkääjän haluaman pitkän odotusajan. Data- ja kuittauskehyksessä iso kesto on mahdollinen vain, kun on sirpaloitu iso datamäärä useampaan kehykseen ja varataan siirtotietä loppujenkin sirpaleiden lähettämiseen. Normaalisti

kesto on datakehyksessä vain SIFSin ja sitä seuraavan kuittauksen lähettämiseen kuluvan ajan mittainen. Kuittauksessa duration on yleensä nolla. Koska sirpalointia ei käytännössä juurikaan käytetä WLANissa, voidaan se ottaa kokonaan pois käytöstä. Näin päästään eroon datakehyksiin ja kuittauksiin liittyvästä NAVin huijaamisesta.

Väärennettyä RTS:ää ja CTS:ää vastaan taisteleminen on vaikeampaa. Jollekin toiselle lähetetyn RTS:n havaitseva laite voisi suhtautua siinä ilmoitettuun durationiin aluksi ehdollisesti [BeS03]. RTS:n jälkeen pitäisi protokollan mukaisesti toimittaessa havaita tietyn ajan päästä myös datakehys (ja sitä ennen CTS). Jos datakehystä ei havaita, on lähettäjä joko liikkunut havainnoitsijan kantaman ulkopuolelle tai RTS oli väärennetty. Koska lähettäjän siirtyminen juuri RTS:n ja datan lähettämisen välissä on erittäin epätodennäköistä, voi RTS:n havainnut laite kummassakin tapauksessa unohtaa RTS:ssä olleen durationin, nollata NAVinsa ja yrittää halutessaan päästä itse siirtotielle. Itse asiassa standardi jopa sallii tällaisen ilmoitetun durationin sivuuttamisen, jos tarpeeksi nopeasti RTS:n jälkeen ei kuulla datakehystä [IEE07].

Jos havaitaan yksittäinen, jollekin toiselle laitteelle osoitettu CTS, niin se on joko vastaus havaitsijan kantaman ulkopuolelta lähetettyyn RTS:ään tai se on väärennös [BeS03]. Jos väärennös on osoitettu jollekin verkon laitteelle, niin vain kyseinen laite tietää varmuudella sen olevan väärennetty, koska se ei ole vastaus sen aiemmin lähettämään RTS:ään. Jos väärennetty CTS on osoitettu laitteelle, jota ei ole olemassa tai on havaitsijan kantaman ulkopuolella, havaitsija ei voi erottaa sitä mitenkään oikeasta vastauksesta aiemmin lähetettyyn RTS:ään.

WLANissa käytetään kuitenkin matalampaa kynnysarvoa signaalin havaitsemisessa eli siirtotien varatuksi tunnistamisessa kuin kehysten vastaanottamisessa. Jos asema kuulee esimerkiksi tukiaseman jollekin toiselle etäiselle asemalle lähettämän CTS:n, niin vaikka havaitsija ei kuulisikaan selvästi toisen aseman tukiasemalle lähettämää datakehystä, niin se erottaa hyvin todennäköisesti siirtotien kohonneen signaalitason. Jos signaalitaso pysyy matalana, niin CTS oli todennäköisesti väärennetty. Näin ollen voidaan aika turvallisesti jättää myös yksittäiset CTS:t kokonaan huomiotta [BeS03].

Virtuaalihäirintä ei vielä yksinään tuota välttämättä täydellistä liikenteen kokonaan katkaisevaa palvelunestohyökkäystä. Jos hyökkääjä noudattaa muilta osin vuoronvarausprotokollaa, niin se kilpailee aina lähetyksen jälkeen kilpailuikkunassa siirtotielle pääsystä muiden sinne haluavien laitteiden kanssa. Hyökkääjä voi kuitenkin tehostaa virtuaalihäirintäänsä *etuilemalla* kilpailtaessa siirtotielle pääsystä. Sen sijaan,

että DIFSin jälkeen valittaisiin kilpailuikkunassa protokollan mukainen odotusaika väliltä $[0, CW_{max} - 1]$, hyökkääjä pitääkin odotusajan nollassa. Näin hyökkääjä pääsee siirtotielle heti eikä sen tarvitse kilpailla siitä muiden osapuolten kanssa. Toki joku toinen protokollan mukaisesti toimiva laite on voinut arpoa itselleen odotusajaksi nollan, jolloin molempien lähettäessä samaan aikaan tulee yhteentörmäys. Tällöin hyökkääjä ei kasvata omaa CW_{max}:n arvoa, kuten protokollan mukaisesti toimiessa pitäisi, vaan pitää sen edelleen nollana. Hyökkääjä saa siis siirtotien käyttöönsä aina ennen muita heti DIFSin jälkeen. Siirtotien saatuaan hyökkääjä lähettää CTS:n maksimaalisella durationilla varustettuna, jolloin muiden laitteiden tulee pidättäytyä siirtotieltä NAVin osoittaman ajan, noin 32 millisekuntia. Näin hyökkääjä näyttää toimivan protokollan mukaisesti, mutta pääsee silti jatkuvasti ainoana siirtotielle ja estää verkon muun liikenteen täydellisesti. Ulkopuolelta katsoen yksi asema pääsee toki jatkuvasti siirtotielle, mutta voihan olla, että se on vain sattunut arpomaan kilpailuikkunassa itselleen aina pienemmän odotusajan kuin muut laitteet. Mitä enemmän verkossa on muita siirtotielle haluavia laitteita, niin sitä epätodennäköisempää on, että vain yksi asema saa siirtotien haltuunsa koko ajan. Ovela hyökkääjä voisi pienentää epäilysten heräämistä päästämällä silloin tällöin jonkun muunkin siirtotielle asettamalla $CW_{max} - 1$:n arvoksi jonkin pienen luvun. Tällöin palvelunestohyökkäys ei olisi kuitenkaan täydellistä, mutta sopiva tasapainoilu paljastumisen ja liikenteen rajoittamisen välillä on joissakin tapauksissa paras vaihtoehto, jos hyökkääjä haluaa häiritä pitkän aikaa huomaamattomasti. Hyökkääjä voi myös vaihdella lähettäjäksi merkittyä MAC-osoitetta, jotta siirtotie ei näyttäisi olevan jatkuvasti vain yhden aseman hallussa.

Edellä kuvattu röyhkeä etuileminen kilpailuikkunassa on erikoistapaus niin sanotusta huonosti käyttäytyvästä asemasta [KyV03]. Sillä tarkoitetaan tukiaseman verkkoon liittynyttä asemaa, joka yrittää päästä siirtotielle useammin, kuin sen protokollan mukaan pitäisi. Huonosti käyttäytyvä asema eroaa edellä kuvatusta etuilemisesta heti DIFSin jälkeen siinä, että siirtotielle pyritään, vain kun itsellä on jotain lähetettävää. Sen pääasiallisena tarkoituksena ei siis ole estää toisia pysymään kokonaan poissa siirtotieltä, vaan saada se omaan käyttöön tarvittaessa mahdollisimman nopeasti.

Pelkkä etuileminen kilpailuikkunassa siirtotielle pääsyssä, siis ilman NAViin vaikuttamista, ei yksinään aiheuta mitään haittaa ellei hyökkääjä käytä saamaansa siirtotietä jotenkin. Jos hyökkääjä lähettää vain satunnaisia kehyksiä tai bittejä, niin hyökkäys ei käytännössä eroa tyhmästä radiohäirinnästä. Hyökkääjä vain odottaa aina DIFSin mittaisen ajan ennen kuin ottaa siirtotien haltuunsa. Jos hyökkääjä on

kuitenkin ensin onnistunut todentamaan itsensä onnistuneesti verkkoon tai verkossa on käytössä pelkkä avoin todennus, niin tilanne on hiukan toinen. Tällöin hyökkääjä kaapattuaan kilpailuikkunassa siirtotien aina ensimmäisenä itselleen voi lähettää siinä protokollan mukaisesti dataa. Energiankulutuksen kannalta tällainen toiminta ei kuitenkaan eroa mainittavasti tyhmästä radiohäirinnästä, mutta sitä on erittäin vaikea havaita.

Eräs ratkaisu hyökkääjän epäreilusti manipuloimaan odotusaikaan on siirtää odotusajasta päättäminen asemalta tukiasemalle [KyV03]. Sen sijaan, että asema arpoo kilpailuikkunassa käyttämänsä odotusajan väliltä $[0, CW_{max} - 1]$, tukiasema arpoo odotusajan, jota aseman on käytettävä seuraavassa kilpailuikkunassa, ja lähettää sen ACK:n mukana asemalle. Jos käytössä on CTS/RTS, niin ACK:n sijasta tukiasema voi lähettää odotusajan CTS:n mukana. Ensimmäiseen lähetykseensä asema voi päättää odotusajan itse, mutta kaikissa myöhemmissä lähetyksissä tukiasema kertoo sen. Tukiasema myös valvoo antamiensa odotusaikojen noudattamista. Jos asema poikkeaa saamastaan odotusarvosta liikaa pienempään suuntaan yrittäen päästä siirtotielle nopeammin kuin pitäisi, voi tukiasema rangaista sitä antamalla seuraavalla kerralla normaalia pidemmän odotusajan. Näin huonosti käyttäytyvää asemaa rangaistaan, jotta se ei pitkässä juoksussa pääse siirtotielle useammin kuin mitä sen normaalisti kuuluisi. Jos asema on jatkuvasti välittämättä sille annetuista odotusajoista, niin sitä vastaan voidaan ruveta vastatoimenpiteisiin. Jos käytössä on RTS/CTS, huonosti käyttäytyvää asemaa voidaan estää enää liikennöimästä verkossa jättämällä lähettämättä CTS:iä sen RTS:iin. Jos käytössä on pelkkä CSMA/CA, niin MAC-kerros voi kertoa protokollapinon ylemmille kerroksille huonosti käyttäytyvästä asemasta, jolloin ne voivat ryhtyä tarvittaviin toimenpiteisiin tai ilmoittaa järjestelmän käyttäjälle tilanteesta.

Huomattakoon, että täysin vastoin standardia jotkin laitteet eivät kunnioita CTS:ssä tai ACK:ssa olevaa durationia, jos siirtotie pysyykin niiden jälkeen vapaana. Jo muutaman millisekunnin odottelun jälkeen laitteet pyrkivät siirtotielle, vaikka niiden kuuluisi ensin odottaa durationin perusteella asetetun NAVin kulumista loppuun [BeS03]. Näin ollen virtuaalihyökkäyksen tehokkuus voi käytännössä olla hyvin vaihtelevaa.

6.2.2 Todennuksen ja liittymisen purkaminen

Ennen kuin asema voi lähettää tai vastaanottaa dataa tukiaseman hallinoimassa verkossa, sen on ensin sekä todennettava itsensä että liityttävä tukiaseman verkkoon.

Todennukseen ja liittymiseen liittyvät tilakaaviot on esitelty jo aiemmin (katso kuva 10 sivulla 29 ja kuva 14 sivulla 46). Asema voi olla todennettuna useampaankin tukiasemaan, mutta liittyneenä vain yhteen kerrallaan. Asema voi nopeammin vaihtaa samaan laajennettuun palveluryhmään kuuluvien tukiasemien välillä tekemällä todennuksen jo etukäteen ja liittymisen vasta siirtyessään uuteen tukiasemaan.

Hyökkääjä voi väärentää MAC-osoitteensa ja purkaa yhdellä kehyksellä jonkun verkkoon liittyneen aseman todennuksen tai liittymisen [BeS03]. Tällöin kyseinen asema siirtyy tilakoneessa tilaan yksi tai kaksi ja sen on tehtävä todennus ja liittyminen. Protokollan mukaisesti toimittaessa liittymisen purku (disassociation) ja todennuksen purku (deauthentication) ovat pelkkiä ilmoituksia eivätkä pyyntöjä ja ne on aina hyväksyttävä heti. Tehokkainta on purkaa todennus, jolloin asema siirtyy suoraan alkutilaan ja josta palaaminen on hitaampaa kuin pelkän liittymisen purkamisen jälkeen. Hyökkäys siis estää asemaa liikennöimästä verkossa purkamalla aina uudestaan ja uudestaan todennuksen ja liittymisen ja on erittäin tehokas palvelunestohyökkäys lamauttaen kokonaan aseman dataliikenteen. Hyökkäys toimii kumpaankin suuntaan eli hyökkääjä voi naamioitua joko asemaksi tai tukiasemaksi ja lähettää väärennetyn dissasociationin tai deauthenticationin. Hyökkäyksen voi kohdistaa kerrallaan vain yhteen asemaan tai tukiasemaksi naamioituneena lähettää yleislähetyksenä koko verkkoon ja purkaa kaikkien asemien todennuksen tai liittymisen [AIK06a]. Normaalisti aseman todennusta tai liittymistä ei jatkuvasti vuorotellen muodosteta ja pureta, joten tällainen edestakainen aaltoliike tilakoneen tilojen välillä saattaa olla merkki palvelunestohyökkäyksestä.

Yksinkertainen torjuntatapa on pitää jokaista deauthenticationia ja disassociationia aluksi ehdollisena. Purkukehyksen jälkeen ei heti purettaisikaan todennusta tai liittymistä, vaan seurattaisiin muutamia sekunteja tuleeko purkukehyksen lähettäjältä vielä muita kehyksiä. Jos ei tule, purkukehys tulkitaan aidoksi, mutta jos tulee, niin purkukehys oli väärennetty ja se voidaan unohtaa [BeS03]. Protokollan mukaisesti toimiva laite ei koskaan lähettäisi purkukehystä ellei se olisi aikeissa lopettaa liikennöintiä sen jälkeen. Hetken odottelu torjuu hyökkäyksen tehokkaasti, mutta voi aiheuttaa hetkellisesti ongelmia pakettien reitityksessä. Voi kestää jonkin aikaa ennen kuin jakelujärjestelmä, tyypillisesti Ethernet-verkko, saa tiedon aseman siirtymisestä toisen tukiaseman verkkoon. Ennen tiedon saapumista ohjaa jakelujärjestelmä asemalle tulevat lähetykset väärälle tukiasemalle. Hyökkääjä voi myös lähettää tukiasemalle, jolta uhri poistui, kehyksiä uhrin MAC-osoitteella, jolloin tukiasema tulkitsee uhrin aiemmin lähettämän purkukehyksen olleen väärennetty [BeS03].

Eräs tapa torjua väärennettyjä disassociationeja perustuu kahdesta alkuluvusta muodostetun tulon tekijöihinjaon vaikeuteen [NTN08]. Asema muodostaa kaksi isoa alkulukua X_{STA}:n ja Y_{STA}:n, laskee niiden tulon $N_{STA} = X_{STA}Y_{STA}$ ja lähettää N_{STA}:n tukiasemalle association requestissa. Tukiasema muodostaa vastaavasti jokaista siihen liittyvää asemaa kohti tulon $N_{AP} = X_{AP}Y_{AP}$ kahdesta muodostamastaan alkuluvusta ja lähettää N_{AP}:n asemalle association responsessa hyväksyessään aseman liittymisen. Kun asema haluaa myöhemmin purkaa liittymisensä, se lähettää disassociationin mukana joko X_{STA}:n tai Y_{STA}:n. Tukiasema voi helposti tarkistaa onko N_{STA} jaollinen aseman lähettämällä luvulla. Jos N_{STA} on jaollinen saadulla luvulla, niin disassociation oli aito. Muussa tapauksessa kehys oli väärennetty ja voidaan jättää huomiotta. Tukiasema toimii täysin vastaavasti halutessaan purkaa jonkin aseman liittymisen ja lähettää disassociationin mukana oikean X_{AP}:n tai Y_{AP}:n. Muodostamiensa asemakohtaisten alkulukutulojen lisäksi tukiasema on muodostanut vielä yhden koko verkolle yhteisen alkulukutulon $B_{AP} = T_{AP}U_{AP}$. Sen tukiasema on myös lähettänyt jokaisen hyväksyvän association responsen mukana. Tukiasema hyödyntää jokaisen verkkonsa aseman tuntemaa B_{AP}:tä halutessaan purkaa kaikkien asemien liittymisen kerralla. Yleislähetyksenä lähetettyyn disassociationiin tukiasema liittää mukaan joko T_{AP}:n tai U_{AP}:n, jolloin asemat voivat varmistua kehyksen tulleen tukiasemalta.

Hyökkääjä on toki voinut salakuunnella aiemmin liittymisen yhteydessä lähetetyn N_{STA}:n, N_{AP}:n ja B_{AP}:n, mutta niistä on erittäin vaikea selvittää alkulukutekijöitä, jos alkuluvut ovat tarpeeksi isoja. Alkulukutulot ovat kuitenkin kertakäyttöisiä. Kun toinen tulon muodostava alkuluku on lähetetty disassociationin mukana, niin kyseistä N_{STA}:ta, N_{AP}:tä tai B_{AP}:tä ei voida enää käyttää. Tukiaseman purkaessa yhden aseman liittymisen tai aseman purkaessa liittymisensä on uusittava sekä N_{STA} että N_{AP}. Tukiaseman purkaessa kaikkien asemien liittymisen on uusittava kaikki kolme alkulukutuloa.

Isojen alkulukujen muodostaminen on kuitenkin hidasta ja hyökkääjä voi käyttää alkulukuihin perustuvaa väärennetyn disassociationin torjuntakeinoa palvelunestohyökkäyksen tekemiseen. Lähettämällä runsaasti association requesteja hyökkääjä voi kuluttaa huomattavasti tukiaseman resursseja ja hidastaa sen toimintaa [NTN08]. Ainoastaan WEPissä tällainen liittymispyyntöjen tulvitus ei onnistu, koska siinä liittymistä edeltävä todennus ei ole avoin todennus, kuten WPA:ssa tai WPA2:ssa, vaan perustuu haaste-vaste-periaatteeseen ja ennalta jaetun avaimen tuntemiseen. WEPissä on kuitenkin runsaasti muita tietoturvaongelmia, kuten aiemmin luvussa 3.3 laajasti käsiteltiin, eikä sen käyttö ole siksi kuitenkaan suositel-

tavaa. Alkulukuihin perustuva disassociationin torjuminen ei siis ole lopullinen ratkaisu kehysten todentamiseen WPA:ssa ja WPA2:ssa. Sen käyttäminen estää kyllä väärennetyn liittymisen purkamisen, mutta itse torjuntakeino altistaa toiselle palvelunestohyökkäykselle, association requestien tulvitukselle.

Myös käänteistä ARP-kyselyä (Reversed ARP, RARP) on esitetty keinoksi väärennetyn MAC-osoitteen paljastamiseksi [Car03]. Siinä lähetetään yleislähetyksenä käänteinen ARP-kysely, jossa kysytään annettua MAC-osoitetta vastaavaa IP-osoitetta. Annetun MAC-osoitteen haltija vastaa ja kertoo oman IP-osoitteensa. Jos esimerkiksi jostain MAC-osoitteesta havaitaan saapuvan jatkuvasti useita purkukehyksiä, niin ne voivat olla väärennettyjä. Kyseiseen MAC-osoitteeseen liittyvää IP-osoitetta kysytään RARPilla. Jos osoite oli väärennetty, niin saadaan vastaukseksi kaksi eri IP-osoitetta, kun sekä oikea MAC-osoitteen haltija että hyökkääjä vastaavat siihen. Jos MAC-osoitetta ei ole väärennetty, saadaan vastauksena vain yksi IP-osoite oikealta asemalta. RARP ei kuitenkaan kerro aina luotettavasti väärennetystä MAC-osoitteesta [AIK06a]. Hyökkääjä voi väärentää MAC-osoitteen lisäksi myös IP-osoitteensa, jolloin RARP-kyselyyn saadaan vain yksi IP-osoite. Jollain asemalla voi olla myös kaksi eri IP-osoitetta yhdistettynä samaan MAC-osoitteeseen, jolloin se kertoo molemmat ja päätellään virheellisesti MAC-osoitteen olevan väärennetty.

Lupaavin väärennetyn purkukehyksen havaitsemiskeino perustuu sekvenssinumeroihin. Seuraamalla data- ja hallintakehyksissä olevien sekvenssinumeroiden säännönmukaisuutta voidaan tunnistaa väärennetyn MAC-osoitteen sisältävät kehykset [Wri03]. Jos esimerkiki aseman tukiasemalle viimeksi lähettämässä kehyksessä oli sekvenssinumero 500 ja hyökkääjä lähettää aseman MAC-osoitteella purkukehyksen, niin siinä tuskin on tukiaseman odottamaa sekvenssinumeroa 501, joka numeroiden säännönmukaisesti kasvaessa siinä pitäisi olla, vaan jokin aivan muu. Poikkeava sekvenssinumero tai useamman sarja herättää helposti huomioita tukiaseman saadessa samasta MAC-osoitteesta kehyksiä, esimerkiksi sekvenssinumeroilla 77, 78 ja 850.

Menetelmän käyttäminen vaatii kuitenkin saapuvien kehysten ja niissä olevien sekvenssinumeroiden seurantaa. Liian pienistä muutoksista peräkkäisten pakettien sekvenssinumeroissa ei kuitenkaan pitäisi heti tulkita pakettien olevan väärennettyjä. Sekvenssinumerot saattavat normaalistikin poiketa odotetusta sarjasta. Käytetyn kanavan vaihtaminen saattaa aiheuttaa sekvenssinumeroihin selvän hyppäyksen ja jotkin asemat saattavat toimia standardista poikkeavalla tavalla ja hypätä sekvenssinumeroissaan hetkellisesti käyttämään samoja mitä tukiasema käyttää ja palata sitten aiempaan omaan sarjaansa [Wri03].

Fanglu ja Tzi-cker [FaT05] tutkivat kokeellisesti sekvenssinumeroiden säännönmukaisuutta ja havaitsivat niiden noudattavan pääsääntöisesti standardia, mutta jossain määrin, jopa ilman hyökkäystäkin, peräkkäin havaittujen kehysten sekvenssinumerojen ero ei kuitenkaan ollut aina tasan yksi. Jos kehys joudutaan lähettämään uudestaan, niin sen sekvenssinumero ei kasva. Riippuen havaitsijasta voi olla, ettei se huomaa kaikkia kehyksiä ja tästä saattaa seurata myös kahden luvun hyppäystä eteen päin sekvenssinumeroissa. Tukiasemat saattavat myös priorisoida beacontai probe response -kehysten lähettämisen ennen muita, vaikka sekvenssinumeron mukaan ne tulisivatkin liian aikaisin. Näin ollen tukiaseman lähettämän beaconin tai probe resposen sekvenssinumero voi olla isompi kuin sitä seuraavan kehyksen sekvenssinumero.

Jos aseman lähettämän kehyksen sekvenssinumero on sama tai korkeintaan kaksi numeroa isompi kuin edellisen kehyksen sekvenssinumero, niin se tulkitaan vielä normaaliksi. Muussa tapauksessa asema asetetaan tarkkailutilaan ja sille lähetetään ARP-kysely. Jos jatkossa asemalta tulevien kehysten sekvenssinumerot kasvavat normaalisti suhteessa edelliseen poikkeavaan, tarkkailutilan aiheuttaneeseen sekvenssinumeroon, niin tarkkailutila puretaan ja hyökkäystä ei tulkita tapahtuneen. Jos sekvenssinumerot ovat tarkkailutilassa kuitenkin poikkeuksen aiheuttaneen sekvenssinumeron ja sitä edeltäneen välistä, niin poikkeuksen aiheuttanut kehys oli väärennetty [FaT05].

Sekvenssinumeroiden seuraamisella voidaan käytännössä vain havaita hyökkäys, ei estää sitä [FaT05]. Havainnointia olisi hyvä tehdä liikennöivässä laitteessa, asemassa tai tukiasemassa, muokkaamalla ajuria kaappaamaan sekvenssinumero analysoitavaksi. Tämä ei kuitenkaan onnistu helposti, sillä laiteohjelmistot (firmware) eivät välitä kaikkia saapuneita kehyksiä, kuten esimerkiksi hallintakehyksiä, ollenkaan ajurille. Toinen keino on seurata tietoliikennettä ulkopuolisella laitteella, joka niin sanotussa monitor mode -tilassa kaappaa kaiken liikenteen. Tällöin voidaan analysoida kaikkia kaapattuja kehyksiä ja havaita sekvenssinumeroitakin, mutta havaitun käynnissä olevan hyökkäyksen torjuminen on teknisesti hankalaa. Pitäisi esimerkiksi pystyä ilmoittamaan ajoissa tukiasemalle, että sen juuri saama purkukehys olikin väärennetty.

Sekvenssinumeroiden käyttöön väärennetyn deauthnetication tai disassociationin torjunnassa liittyy myös sekvenssinumeron väärentämisen mahdollisuus. Kehysten sekvenssinumeroita käsittelee laiteohjelmisto ja siihen on vaikea päästä käsiksi, mutta se saattaa onnistua joissain tapauksissa [BeS03]. Tällöin hyökkääjälle tarjoutuu

keino kiertää sekvenssinumeroihin perustuva kehysten todennus. Hyökkääjä kuuntelee aseman lähettämiä kehyksiä ja saa selville sekä uhrin MAC-osoitteen että käytetyt sekvenssinumerot. Hyökkääjä asettaa purkukehykseensä sopivan sekvenssinumeron ja väärentää lähettäjäksi aseman MAC-osoitteen. Tukiaseman näkökulmasta purkukehys on aivan oikea ja purkaa aseman todennuksen tai liittymisen.

Ratkaisuna on luopua purkukehyksissä säännönmukaisesti yhdellä kasvavasta sekvenssinumerosta ja korvata se näennäisesti satunnaisella sekvenssinumerolla [AIK06b]. WPA:ssa ja WPA2:ssa todennusprosessin aikana asema ja tukiasema muodostavat yhteisen tilapäisen pariavaimen PTK:n, kuten aiemmin luvussa 4.2 esitettiin. Näennäissatunnainen sekvenssinumero muodostetaan PTK:sta, edellisestä käytetystä sekvenssinumerosta ja sekä tukiaseman että aseman MAC-osoitteista. Menetelmää voidaan käyttää myös tukiaseman purkaessa kaikkien verkkonsa asemien todennuksen tai liittymisen. Tällöin näennäissatunnainen sekvenssinumero muodostetaan PTK:n sijasta kaikkien verkon asemien tuntemasta väliaikaisesta ryhmäavaimesta GTK:sta, edellisestä käytetystä sekvenssinumerosta sekä tukiaseman MAC-osoitteesta. Näennäissatunnaislukujen käyttäminen vaatii laiteohjelmistojen päivittämistä, mutta on erittäin tehokas torjuntatapa väärennettyjä purkukehyksiä vastaan. Hyökkääjä ei voi enää pelkkiä sekvenssinumeroita seuraamalla päätellä helposti oikeaa sekvenssinumeroa purkukehyksiinsä.

Näennäissatunnaisten sekvenssinumeroiden käyttäminen ei kuitenkaan onnistu ennen nelivaiheista kättelyä, jossa vasta muodostetaan PTK ja GTK. Näin ollen hyökkääjä voi aseman vasta muodostaessa turvallista lähiverkkoa tehdä palvelunestohyökkäyksen vaiheessa yksi tai kaksi (katso kuva 13 sivulla 43) avoimen todennuksen jälkeen deauthenticationilla ja onnistuneen liittymisen jälkeen myös disassociationilla. Ongelman voisi mielestäni ratkaista yhdistämällä näennäissatunnaisten sekvenssinumeroiden käyttämisen aiemmin mainittuun ehdolliseen purkukehyksiin suhtautumiseen. Jos tukiasema saa deauthenticationin tai disassociationin ennen PTK:n ja GTK:n muodostamista, niin se odottaa muutaman sekunnin. Jos odotusaikana ei saavu yhtään kehystä samasta MAC-osoitteesta, niin purkukehys oli aito. Jos odotusaikana tulee lisää kehyksiä, niin purkukehys oli väärennetty ja voidaan hylätä. PTK:n ja GTK:n muodostamisen jälkeen käytetään näennäissatunnaisia sekvenssinumeroita väärennösten tunnistamiseen. Aiemmin mainittua odotusaikaan liittyvää jakelujärjestelmän reititysongelmaa aseman vaihtaessa toiseen tukiasemaan samassa laajennetussa palveluryhmässä ei nyt kuitenkaan voi syntyä. Koska asema on vasta muodostamassa yhteyttään tukiasemaan, ei jakelujärjestelmä tiedä asemasta vielä mitään eikä näin ollen voi lähettää asemalle osoitettuja kehyksiä väärälle tukiasemal-

le. Jos asema on jo liikennöinyt tukiasemansa kautta ja jakelujärjestelmäkin tietää sen sijainnin, niin aseman lähettämä purkukehys vanhalle tukiasemalle hyväksytään välittömästi kunhan näennäissatunnainen sekvenssinumero on oikea.

6.2.3 Virransäästötila

Koska langatonta verkkoa käyttävät laitteet toimivat usein vain akkujen varassa ja energian riittävyys on elintärkeää, tarjoaa 802.11-verkko asemille mahdollisuutta siirtyä pelkällä ilmoituksella virransäästötilaan (Power Save, PS). PS:ssä asemat eivät voi lähettää eivätkä vastaanottaa kehyksiä ja tukiasema puskuroi asemalle sen PS:n aikana tulleet kehykset. Asema herää ajoittain hetkeksi kuuntelemaan tukiaseman säännöllisesti ympäristöönsä lähettämiä beaconeita. Jokaisessa beaconissa on datakentässä TIM-elementti. TIMissä on muun muassa virtuaalinen 2008 bitin kokoinen bittikartta, jossa jokainen bitti vastaa yhtä tukiaseman antamaa AID:tä. Tukiasema antaa jokaiselle asemalle oman yksilöllisen AID:n sen liittyessä tukiaseman verkkoon. Jos asemalle on puskuroituja kehyksiä, niin aseman AID:tä vastaavan bitin arvo kertoo sen TIMissä. Aseman huomattua, että sille on puskuroituja kehyksiä, se pyytää niitä tukiasemalta PS-Poll-kehyksellä. Sen saatuaan tukiasema lähettää asemalle sen virransäästötilan aikana sille puskuroidut kehykset ja poistaa ne puskuristaan. Tämän jälkeen asema voi palata virransäästötilaan. Milloin tahansa asema voi ilmoittaa myös lopettavansa virransäästötilan ja palata normaaliin aktiiviseen toimintatilaansa.

Koska PS-Poll-kehystä ei muiden kontrollikehysten tavoin todenneta eikä salata mitenkään, voi hyökkääjä väärentää sen. Lähettämällä aseman MAC-osoitteella ja AID:llä varustetun PS-Pollin ennen asemaa tukiasema luulee sitä oikeaksi ja lähettää asemalle puskuroidun datan ja tyhjentää puskurinsa. Koska data on salattua, hyökkääjä ei hyödy siitä, mutta voi näin helposti aiheuttaa palvelunestohyökkäyksen virransäästötilassa olevalle asemalle estäen sitä saamasta sille lähetettyjä tukiaseman tallettamia kehyksiä [BeS03].

Voidakseen lähettää uskottavan väärennetyn PS-Pollin hyökkääjän on selvitettävä uhrinsa MAC-osoite sekä AID. MAC-osoite selviää helposti lähes mistä tahansa uhrin lähettämästä kehyksestä, mutta MAC-osoitteeseen liittyvän oikean AID:n selvittäminen on hiukan hankalampaa. AID esiintyy vain kolmessa kehyksessä: association responsessa, reassociationissa ja PS-Pollissa. Salakuuntelemalla jonkun näistä hyökkääjä saa selville kyseisen aseman AID:n. Kun asema vaihtaa liittymisensä toiseen samaan laajennettuun palveluryhmään kuuluvaan tukiasemaan, lähettää asema

reassociationin vanhalle tukiasemalle. Näiden kehysten kuuntelemisen lisäksi hyökkääjä voi onnistua päättelemään MAC-osoitteen ja AID:n välisen yhteyden seuraamalla aseman lähettämiä kehyksiä ja havainnoimalla hetkeä, jolloin asema ilmoittaa haluavansa siirtyä virransäästötilaan sekä tukiaseman lähettämissä beaconeissa ilmoitettuja AID:itä, joille on puskuroituja kehyksiä. Jos esimerkiksi vain yksi asema siirtyy virransäästötilaan, niin sen MAC-osoite voidaan yhdistää ainoaan AID:hen, jolle on tullut puskuroituja kehyksiä. Käytännössä on kuitenkin kaikkein helpointa vain odottaa hetki aseman siirryttyä PS:ään ja napata AID aseman lähettämistä PS-Polleista. Tällöin palvelunestohyökkäystä ei kuitenkaan voida aloittaa aivan heti aseman mentyä virransäästötilaan.

Väärennetyllä PS-Poll-kehyksellä toteutettava palvelunestohyökkäys voidaan estää salaamalla AID-kenttä PS-Poll-kehyksissä [QAM07]. Salaus mudostetaan käyttämällä standardissa salausavaimienkin muodostamisessa käytettyä näennäissatunnaisfunktiota (Pseudo Random Function, PSR) [IEE07], AID:tä sekä aseman ja tukiaseman MAC-osoitteita. PSR:n käytön vuoksi menetelmä on mahdollinen vain WPA:ssa tai WPA2:ssa, mutta ei WEPissä, koska PSR:ää ei käytetä siinä ollenkaan. AID:n salaus sekä estää että paljastaa mahdollisen hyökkäysyrityksen.

PS-Pollin lisäksi on esitetty myös kaksi muuta virransäästötilaan liittyvää palvelunestohyökkäystä, mutta ne ovat varsin teoreettisia [BeS03]. Jos asema herätessään ja kuunnellessaan tukiaseman lähettämää beaconia kuuleekin hyökkääjän väärentämän beaconin, minkä perusteella asemalle ei näyttäisikään olevan puskuroituja kehyksiä, asema vaipuu takaisin unitilaan. Jos joka kerta herätessään asema kuuleekin väärennetyn beaconin, se ei saa koskaan tietää sille tulleista ja tukiaseman puskuroimista kehyksistä. Lopulta tukiasema pidettyään asemalle tulleita kehyksiä puskurissaan maksimiajan poistaa ne. Toinen hyökkäys liittyy aseman ja tukiaseman välisen synkronoinnin häiritsemiseen. Aseman on tiedettävä, milloin tukiasema lähettää beaconeitaan, jotta se osaa herätä oikeaan aikaan kuuntelemaan niitä pieneksi hetkeksi. Tukiasema lähettää jokaisessa beaconissa aikaleiman, jonka perusteella asemat tahdistavat ja korjaavat oman kellonsa samaan aikaan. Jos hyökkääjä väärentää beaconin ja asettaa sen aikaleiman sopivasti, eivät asema ja tukiasema tomi enää synkronoidusti ja asema herää vääriin aikoihin odottelemaan beaconia. Virransäästötilan ulkopuolella tästä ei ole niinkään haittaa, koska asemat kuuntelevat siirtotietä jatkuvasti, paitsi kun itse lähettävät jotain, ja kuulevat näin ollen jokaisen beaconin.

Huomattakoon, että mitään kolmea edellä esitettyä virransäästötilaan kohdistuvaa

palvelunestohyökkäystä ei ole testattu dokumentoidusti, ei myöskään AID:n salaukseen perustuvaa väärennetyn PS-Pollin estämistä. Virransäästötilaan perustuvia palvelunestohyökkäyksiä ja niiden torjuntakeinoja tulisikin selvittää myös kokeellisesti ennen kuin niiden luomaa uhkaa ja käyttökelpoisuutta voidaan arvioida luotettavasti.

6.3 Tulvitus väärennetyillä hallintakehyksillä

Tulvitukseen perustuvat palvelunestohyökkäykset eivät ole mikään uusi ilmiö. Jo ennen langattomien lähiverkkojen yleistymistä oli internetissä esiintynyt useita vakavia tulvitukseen perustuvia palvelunestohyökkäyksiä, kuten luvussa 5.2 mainittiin. WLANissa tulvitushyökkäys kohdistuu aina tukiasemaan ja perustuu yhteyden muodostuksen eri osapuolilta vaatimaan epätasaiseen resurssien kulutukseen. Hyökkääjä voi lähettää helposti valtavan määrän sopivia hallintakehyksiä väärennetyllä MAC-osoitteella, mutta tukiasema joutuu vastaanottamastaan kehyksestä riippuen tallentamaan erilaisia yhteyden muodostuksessa tarvittavia tilatietoja puskuriinsa. Muistin lisäksi tukiaseman resursseja kuluu myös sen lähettäessä protokollan mukaisen vastauksen saamaansa kehykseen. Koska vastaanottajaa ei kuitenkaan ole olemassa, ei tukiasema saa kehykseensä kuittausta ja joutuu tekemään useita uudelleenlähetyksiä. Puskurin täyttyminen ja vastausten käsittely kuluttavat tukiaseman resursseja, eikä se voi enää palvella tehokkaasti muita verkon käyttäjiä.

6.3.1 Tulvitushyökkäykset eri kehyksillä

Tulvitushyökkäystä voidaan tehdä probe-. authentication tai association requestilla [FBV04]. Lähettämällä kehyksiä runsaasti voi hyökkääjä kuluttaa tukiaseman resursseja hidastaen sen toimintaa tai jopa kaataa sen. On kuitenkin yksilöllistä ja laitekohtaista, kuinka helposti ja kuinka toimintakyvyttömäksi hyökkääjä voi tukiaseman saada tulvituksella [BFV08]. Tulvitushyökkäykset kohdistuvat suoranaisesti vain tukiasemaan, mutta koska tukiasema on koko verkon toiminnan kannalta aivan olennaisessa asemassa, vaikuttaa sen hidastuminen tai kaatuminen koko langattoman lähiverkon toimintaan.

Probe requestilla tehtävässä tulvituksessa (Probe Request Flood, PRF) tukiasema luulee valtavan määrän asemia tiedustelevan siltä verkon tietoja, jotta ne voisivat mahdollisesti liittyä verkkoon. Tukiasema vastaa jokaiselle probe responsella, mutta koska se ei saavuta ketään, ei tukiasema saa kuittausta ja lähettää kehyksen uudes-

taan useita kertoja [FBV04].

Väärennetyillä MAC-osoitteilla tehty authentication requestien tulvitus (authentication request flood, AutRF) kuluttaa myös tukiaseman resursseja. Jos käytössä on WEP, tukiasema varaa muistia haastetekstin muodostamiseksi. WPA:ssa ja WPA2:ssa on prosessin alussa käytössä kevyempi avoin todennus (katso kuva 13), mutta siinäkin on varattava muistia uuden todennettavan aseman tietoja varten [FBV04]. Koska todennuspyynnössä oli keksitty MAC-osoite, ei tukiaseman lähettämä authentication response -kehys saavuta ketään eikä tukiasema saa kuittausta sen perillemenosta, jolloin se lähetetään useita kertoja uudestaan.

Tukiaseman saadessa association requestin se tarkistaa puskuristaan onko liittymistä pyytävä asema todennettu aiemmin [LiY07]. Koska kehys tuli keksitystä MAC-osoitteesta, ei aseman tietoja löydy ja kehys olisi parasta hylätä kokonaan. Usein tukiasemat kuitenkin vastaavat siihen joko disassociationilla tai deauthenitcationilla, johon eivät kuitenkaan saa koskaan kuittausta [FBV04]. Hyökkääjän tulvittaessa tukiasemaa association requesteilla (association request flood, AssRF) kuluttavat ne tukiaseman resursseja.

Tehdessään PRF:ää, AutRF:ää tai AssRF:ää hyökkääjä toimii kuitenkin siirtotien käytössä täysin protokollan mukaisesti eli odottaa omaa vuoroansa ja käyttää siirtotietä vain sallitun ajan. Tukiaseman näkökulmasta hyökkäys näyttää vain yhtäkkiseltä yhteydenottojen runsaalta kasvulta. Vaikka hyökkääjä väärentääkin MAC-osoitteensa, niin toisin kuin aiemmin käsitellyissä hallinta- ja kontrollikehysten väärentämisissä tulvituksessa MAC-osoite voi olla mikä tahansa. Sen ei tarvitse kuulua jollekin jo tukiaseman verkossa olevalle asemalle, koska tulvituksessa ei ole tarkoitus naamioitua miksikään tietyksi asemaksi.

Edellä mainittujen kolmen kehyksen lisäksi tulvitus saattaa onnistua myös kahdella muullakin kehyksellä. Halutessaan vaihtaa samassa laajennetussa palveluryhmässä toiseen tukiasemaan lähettää asema uudelle tukiasemalle reassociation requestin. Jos tukiasema saa reassociation requestin, asemalta jota ei ole todennettu, lähettää tukiasema vastauksena deauthenticationin. Jos siihen jäädään odottamaan vielä kuittausta, avaa se hyökkääjälle mahdollisuuden reassociation requesteilla tehtävään tulvitukseen. Toinen mahdollinen tulvitettava kehys on virransäästöön liittyvä aseman lähettämä PS-Poll. Sen saatuaan tukiasema lähettää datakehyksen ja jää odottamaan siihen kuittausta, jota ei koskaan tule. Jälleen tukiasema tekee lukuisia uudelleenlähetyksiä enenn kuin luopuu yrityksistä. Reassociation requestin ja PS-Pollin käyttökelpoisuus tulvitushyökkäyksen aikaansaamisessa vaatisi kuitenkin

tarkempaa kokeellista tutkimista.

6.3.2 Tulvituksen torjuntakeinoja

Tulvitushyökkäystä on haastavaa torjua millään kehysten todentamiseen liittyvällä menetelmällä, sillä todentaminen perustuu usein johonkin yhteiseen salaiseen tietoon aseman ja tukiaseman välillä ja tällainen syntyy vasta onnistuneen 802.1X:ään perustuvan todennuksen ja nelivaiheisen kättelyn jälkeen (WPA:ssa ja WPA2:ssa). PRF, AutRF ja AssRF tapahtuvat kuitenkin jo ennen kuin mitään yhteistä todentamiseen käytettävää tietoa on muodostettu. Kehysten todentamisen sijasta tulvitusta vastaan on esitetty useita tehtävien (puzzle) ratkaisemiseen perustuvia torjuntakeinoja, joissa authnetication- tai association requestin lähettäjän on ensin ratkaistava jokin resursseja kuluttava tehtävä. Seuraavaksi esitellään lyhyesti muita torjuntakeinoja ja niiden jälkeen tarkastellaan lähemmin tehtäviin perustuvia tulvitushyökkäysten torjuntakeinoja. Lopuksi esittelen uuden idean sekvenssinumeroiden käyttämisestä tulvitushyökkäysten havaitsemisessa.

MAC-osoitteiden suodatuksessa vain ennalta määritellyistä MAC-osoitteista tulevat kehykset hyväksytään ja muut hylätään kokonaan reagoimatta niihin mitenkään [LiY07]. Kuten luvussa 3.2 mainittiin, MAC-osoitteisiin perustuva suodatus ei ole luotettava tapa todentaa asemaa. Sitä voidaan kuitenkin käyttää torjumaan tulvitukseen perustuvia palvelunestohyökkäyksiä, joskin siihen liittyy ongelmia. Koska jokaiseen tukiasemaan on käsin asetettava sallitut MAC-osoitteet, niin suodatuksen käyttämisestä tulee käytännössä hyvin hankalaa, jos tukiasemia ja asemia on paljon ja asemien vaihtuvuus on suurta. Hyökkääjä voi kiertää MAC-osoitteisiin perustuvan suodatuksen väärentämällä kehyksiinsä jonkin tukiaseman salliman MAC-osoitteen [LYB10]. Saattaa herättää kuitenkin epäilyksiä, jos yhdestä ja samasta MAC-osoitteesta tulee jatkuvasti esimerkiksi useita authentication requesteja. Hyökkäys on siis aika helppo havaita ja sitä vastaan voidaan ryhtyä tarvittaessa toimenpiteisiin.

Tukiaseman käsittelemien authentication- ja association requestien määrää voidaan rajoittaa. Normaalisti näitä tulee harvakseltaan silloin tällöin eikä kerralla kovin suuria määriä. Rajaamalla hyväksyttyjen kehysten määrä enintään viiteen per sekunti voidaan sitä useammin tulevat kehykset jättää kokonaan huomiotta [LiY07]. Ennen rajoittamisen ulottamista association requestiin tukiasema tarkistaa onko lähettäjä todennettu. Kehysten määrän rajoittamisen ongelmana on myös oikeutettujen käyttäjien lähettämien kehysten suodattuminen hyökkäyksen aikana. Hyökkääjän

lähettäessä jatkuvasti runsaasti authentication requesteja on pieni todennäköisyys, että oikeutetun käyttäjän lähettämä kehys on niiden vain viiden joukossa, jotka tukiasema enintään hyväksyy per sekunti.

Tulvitushyökkäyksen vaikuttavuutta voidaan pienentää vähentämällä tukiaseman tekemiä uudelleenlähetyksiä. Normaalisti tukiasema lähettää kehyksen korkeintaan seitsemän kertaa ellei saa siihen kuittausta [IEE07]. Näin ollen yksikin hyökkääjän lähettämä väärennetyllä MAC-osoitteella varustettu authentication request saa tukiaseman lähettämään siihen vastauksensa useita kertoja. Koska mihinkään ei kuitenkaan tule kuittausta, lähettää tukiasema kehyksen uudestaan kuusi kertaa ennen kuin luopuu yrityksistä. Rajoittamalla uudelleenlähetyksien määrää voidaan pienentää tulvitushyökkäyksen tehokkuutta [LYB10]. Uudelleenlähetysten rajoittamisessa on kuitenkin ongelmana, että muutkin kuin hyökkäystarkoituksessa lähetetyt kehykset kohtaavat saman rajoituksen. Jos esimerkiksi yhteys on hetkellisesti huono, niin tukiaseman oikeutetulle asemalle lähettämä authentication response ei tavoita asemaa. Tukiasema ei saa kuittausta kehyksen perille menosta, mutta ei silti lähetä kehystä enää uudestaan rajoituksen vuoksi. Hetken päästä asema joutuu lähettämään authentication requestin uudestaan, jolloin yhteys on ehkä jo parempi. Uudelleenlähetysten määrän rajoittaminen ennemminkin vain hidastaa kuin estää oikeutettuja asemia liittymistä tukiaseman verkkoon. Mitä vähemmän tehdään uudelleenlähetyksiä sitä nopeammin tukiasema tulkitsee yhteyden muodostamisen epäonnistuneen ja poistaa puskuristaan yhteyden muodostamisen ajaksi tallennettuja tilatietoja. Jos tulvitus on tarpeeksi voimakasta, niin uudelleenlähetysten rajaamisesta huolimatta puskuri saattaa kuitenkin täyttyä.

Hyökkäysten tehokkuutta voidaan myös vähentää rajoittamalla identtisten kehysten vastaanottoa [LYB10] tukiasemassa. Hyökkääjä voi lähettää esimerkiksi samalta näyttävää authentication requestia useampia. Tukiasema hylkää kehykset, jotka ovat täsmälleen samanlaisia, kuin tukiasemalla jo käsittelyssä oleva kehys. Toisin sanoen ennen kuin saapunut authentication request on johtanut joko todennuksen onnistumiseen tai epäonnistumiseen, ei tukiasema ota toista täsmälleen samanlaista kehystä käsiteltäväksi. Hyökkääjä voi yrittää kiertää identtisiin kehyksiin perustuvan torjunnan vaihtamalla lähettäjän MAC-osoitetta jokaiseen kehykseen, jotta ne näyttäisivät tulevan eri lähettäjältä.

Tukiasema saattaa priorisoida hallintakehysten lähettämisen datakehysten edelle [LYB10]. Joutuessaan authentication- tai association requestilla tehtävän tulvituksen kohteeksi tukiasema käyttää kaiken kapasiteettinsa niihin vastaamiseen, vaikka

myös datakehyksiä olisi odottamassa lähettämistä. Eri tyyppisten kehysten tasapuolisella lähettämisellä dataliikenne ei tyrehtyisi täysin runsaankaan tulvituksen aikana, mutta hallintakehyksiin vastaaminen hidastuisi jonkin verran [LYB10].

6.3.3 Tehtäviin perustuvia tulvituksen torjuntakeinoja

Tulvitushyökkäyksiä vastaan on esitetty myös runsaasti erilaisia tehtäviin perustuvia torjuntakeinoja. Tehtävien käyttäminen palvelunestohyökkäysten torjunnassa ei ole kuitenkaan mikään uusi idea ja ajatus tehtävien käyttämisestä tasaamaan osapuolten resurssien kulutusta on ensimmäisen kerran esitetty 1992 hillitsemään roskapostia [DwN92]. Vuonna 1999 tehtäviä esitettiin ensimmäisen kerran käytettäväksi myös internetissä havaittujen tulvitusten torjumisessa [JuB99]. WLANissa tehtävät ovat usein joko prosessoria tai muistia kuluttavia ongelmia, joista tukiasemaan yhteyden ottavan aseman on suoriuduttava ennen kuin protokollan mukainen yhteydenotto voi jatkua. Tehtävällä on tarkoitus varmistaa, että tukiasema ei joudu kuluttamaan asemaa enempää resursseja yhteyttä muodostettaessa. Normaalin verkon käyttäjän on ratkaistava vain yksi tehtävä, jolloin sen aiheuttama resurssien kulutus on varsin pientä. Hyökkääjä tekee kuitenkin useita yhteydenottoyrityksiä, mihin koko tulvitus perustuu, ja joutuu suoriutumaan useista tehtävistä, mikä kuluttaa hyökkääjän resursseja runsaammin ja rajoittaa laajamittaisen tulvituksen tekemistä. Toisin kuin kaapeliverkossa langattomassa verkossa laitteiden kirjo on kuitenkin huomattavasti suurempi ja on haastavaa määritellä sellainen tehtävä, että se huomioisi erilaisten laitteiden kapasiteetin. Helposti jokin tehtävä voi olla esimerkiksi kannettavalle tietokoneelle hyvinkin helppo, mutta toisaalta matkapuhelimen heikommalta prosessorilta sen ratkaisemiseen saattaa kulua kohtuuttoman kauan. Tukiaseman olisi pystyttävä säätämään helposti ja laitekohtaisesti tehtävien vaativuutta tai vaihtoehtoisesti tehtävien suorittamisen olisi perustuttava johonkin muuhun kuin laitteiden muistin tai prosessoritehon käyttämiseen. Tehtävien muodostaminen ja tarkistaminen eivät myöskään saa kuluttaa liikaa tukiaseman resursseja. Muuten on vaarana, että hyökkääjä voi käyttää tulvitushyökkäyksen torjuntakeinoa palvelunestohyökkäyksen tekemiseen.

Tiivistefunktiot soveltuvat erinomaisesti käytettäviksi tehtävissä, koska annetusta merkkijonosta voi muodostaa nopeasti tiivisteen, mutta käänteinen toimenpide eli merkkijonon, joka tuottaa jonkin tietyn tiivisteen, löytäminen on huomattavasti hitaampaa. Tästä antaa hyvän esimerkin Dongin ja kumppaneiden [DGL10] ehdottama tehtävä. Siinä tukiasema lähettää jokaisessa beaconissa keksimänsä nonssin N_{AP}

ja tehtävän haastavuuteen vaikuttavan luvun L. Riippuen tukiaseman kohtaamasta ruuhkasta se voi muuttaa nonssin kestoaikaa lyhyemmäksi tai pidemmäksi sekä vaihdella L:ää. Jos verkossa ei ole ollenkaan ruuhkaa, L voi olla jopa nolla. Jos beaconin kuullut asema haluaa päästä tukiaseman verkkoon, sen on muodosteattava nonssi N_{STA} ja ratkaistava seuraava tehtävä:

$$Hash(X, N_{STA}, N_{AP}, MAC_{AP}, L) = 0^L,$$

missä $Hash$ on jokin sopiva tiivistefunktio (esim. MD5), X tarkoittaa tehtävän ratkaisua, MAC_{AP} tukiaseman MAC-osoitetta sekä 0^L L:n mittaista pelkistä nollista koostuvaa bittijonoa. Aseman on toisin sanoen löydettävä sellainen X, että se yhdessä muiden parametrien kanssa tuottaa tiivisteen, jonka alussa on ainakin L:n osoittama määrä pelkkiä nollabittejä. Oikea ratkaisu löytyy vain raa'alla laskemisella, johon asema joutuu käyttämään resurssejaan. Tehtävän ratkaistuaan asema lähettää authentication requestissa tukiasemalta aiemmin saamansa N_{AP}:n, L:n, X:n sekä itse muodostamansa nonssin N_{STA}. Saatuaan kehyksen tukiasema tarkistaa ensin, että siinä oleva N_{AP} on vielä voimassa ja että L on sama minkä tukiasema lähetti aiemmin asemalle. Seuraavaksi tukiasema tarkistaa, ettei se ole jo saanut samaa ratkaisua nykyisten voimassa olevien N_{AP}:n ja L:n aikana. Vasta tämän jälkeen tukiasema muodostaa itse tiivisteen X:stä,N_{STA}:sta, N_{AP}:stä, MAC_{AP}:stä ja L:stä ja tarkistaa, että sen alussa todellakin on L nollabittiä. Jos näin on, tukiasema tallettaa ratkaisun ja hyväksyy aseman todennuksen.

Edellä esitetty tehtävä estää tehokkaasti authentication requestilla tehtävää tulvitusta. Jotta hyökkääjä ei voisi tulvittaa probe requesteilla, niitä ei käytetä ollenkaan [DGL10]. Probe-kehykset ovat WLANissa valinnaisia ja asemat saavat kaiken oleellisen informaation myös säännöllisesti tukiaseman lähettämistä beaconeistakin. Association requesteilla tehtävä tulvitus estetään jättämällä todentamattomilta tai tukiasemaan liittymättömiltä asemilta saapuvat kehykset kokonaan huomiotta. Niihin ei lähetetä takaisin edes kuittausta tai purkukehystä [DGL10].

Prosessoritehoa kuluttavien tehtävien lisäksi on esitetty myös muistiin liittyviä tehtäviä [ABM05]. Muistiin liittyvät tehtävät perustuvat muistista tehtäviin hakuihin, jossa eri laitteiden väliset erot eivät ole yhtä suuria kuin prosessoritehoissa. Tehtävien vaativuuteen vaikuttaa kuitenkin oleellisesti välimuistien määrä jo koko. Sekä prosessoriin että muistiin liittyvät tehtävät voivat hyvin heterogeenisessä laiteympäristössä osoittautua joillekin laitteille liian vaikeiksi. Joutuessaan hyökkäyksen kohteeksi tehtävien vaikeusastetta kasvatetaan ja tehtävät auttavat torjumaan hyökkäystä, mutta ne voivat hidastaa kohtuuttomasti heikompitehoisen laitteiden yhtey-

denottoa niiden joutuessa ratkaisemaan kohtuuttoman vaikeaa tehtävää.

Aseman resursseista, kuten prosessoritehosta tai muistista, riippumattoman tehtävän ratkaisuun kuluva aika kohtelisi erilaisia asemia tasa-arvoisesti. Riippumatta asemien ominaisuuksista jokainen käyttää yhteisesti jaettua langatonta siirtotietä, minkä hyödyntämiseen perustuukin Martinovicin ja kumppaneiden [MZW08] esittämä tehtävä. Ennen authentication requestin lähettämistä aseman STA_U on kuunneltava ympäristöään, ja jokainen tukiaseman verkkoon jo aiemmin liittynyt asema, jonka signaalin voimakkuus on vähintään kynnysarvon (Neighborhood Signal Threshold, NST) suuruinen, on STA_U:n *naapuri*. Naapureiksi tulkittavien asemien määrä riippuu toki käytetystä NST:stä, jonka tukiasema kertoo jokaisessa beaconissa. Koska signaalin voimakkuus heikkenee etäisyyden kasvaessa, eivät kauempana olevien asemien signaalien voimakkuudet yllä NST:hen asti. Naapureiksi tulkitut asemat ovat siis pääsääntöisesti STA_U:ta lähimpänä olevat asemat. STA_U:n kaikki naapureiksi tulkitsemat asemat muodostavat *alueen*, jonka STA_U lähettää authentication requestin mukana tukiasemalle.

Kaikki STA_U:n kantaman sisällä olevat tukiasemaan jo aiemmin liittyneet asemat tarkistavat, että authentication requestissa ilmoitettu alue ei ole ristiriidassa niiden oman käsityksen kanssa [MZW08]. Toisin sanoen, jos tukiaseman verkossa jo oleva asema STA_1 havaitsee STA_U:n lähettämän signaalin olevan vähintään NST:n vahvuinen, niin STA_U:n pitäisi tehdä sama havainto ja STA_1:n pitäisi löytyä STA_U:n ilmoittamasta alueesta. Jos STA_1 ei löydä itseään alueesta, STA_1 lähettää tukiasemalle varoituksen. Myös päinvastaisessa tapauksessa, jossa STA_U:n signaali ei STA_1:n havaitsemana yllä NST:hen, mutta STA_1 on mainittu STA_U:n alueessa, lähettää STA_1 tukiasemalle varoituksen. Jos tukiasema saa yhdenkin varoituksen se ei hyväksy STA_U:n todennuspyyntöä. Aseman annettua väärän alueen ja sen todennuspyynnön tultua evätyksi voi se yrittää uudestaan todennusta vasta sen jälkeen, kun tukiasema on vaihtanut käytettävää NST:tä. Tukiasema valitsee uuden NST:n esimerkiksi muutaman sekunnin välein satunnaisesti väliltä [-95 dBm, -55 dBm] ja lähettää sen beaconissa ympäristöönsä. Tukiasema siis saatuaan tuntemattoman aseman authentication requestin ei hyväksy sitä heti, vaan odottaa, saako se muilta jo verkkoonsa liittyneiltä asemilta varoituksen. Vain jos yhtään varoitusta ei tule, hyväksytään todennus ja lähetetään authentication response. Jos useassa authentication requestissa on sama alue, voi tukiasema hylätä ne sallien kuitenkin eri alueita sisältävät authentication requestit [MZW08].

NST:n vaihtaminen ei välttämättä hidasta tulvitushyökkäyksen tekemistä. Hyök-

kääjä voi aluksi tarkkailla verkkoa pitemmän aikaa selvittääkseen kaikkien kuuluvuusalueella olevien verkkoon jo liittyneiden asemien signaalien voimakkuudet. Sen jälkeen aina NST:n vaihduttua hyökkääjä voi muodostaa nopeasti oikean alueen ilman, että jokaisen NST:n vaihtumisen jälkeen on tutkittava ympäristöä uudestaan. Alueen koostumuksessa saattaa tapahtua muutoksia, kun asema poistuu tai uusi asema liittyy verkkoon. Myös verkossa jo olevan aseman liikkuminen, lähetystehon vaihtuminen tai ympäristössä tapahtuvat muutokset, jotka vaikuttavat radioaaltojen etenemiseen, voivat muuttaa kelvollista aluetta. Jatkuvasti muuttuvassa ympäristössä hyökkääjä joutuu tarkkailemaan ympäristöään usein voidakseen määritellä alueensa oikein, mutta hyvin staattisessa ympäristössä hyökkääjä voi tehdä NST:n vaihtumisesta huolimatta jatkaa tulvitushyökkäystä ellei tukiasema hylkää saman alueen sisältäviä authentication requesteja. Hylkääminen voi joissain tilanteissa johtaa kuitenkin myös aivan oikeutetunkin aseman lähettämän authentication requestin torjumiseen, jos siinä sattuu olemaan sama alue, jota hyökkääjä lähettää jatkuvasti. Hyökkääjä voi kiertää samaan alueeseen liittyvän torjunnan lisäämällä alueeseen välttämättä mukaan otettavien asemien lisäksi muita keksittyjä asemia. Keksittyjä asemia vaihtelemalla hyökkääjä voi lähettää useita eri alueilla varustettuja authentication requesteja ilman, että tukiasema saisi mitään varoitusta muilta asemilta. Tämäkin on toki helppo estää, jos tukiasema hyväksyy alueeseen vain sellaisia asemia, jotka jo kuuluvat sen verkkoon eikä mitä tahansa keksittyjä asemia.

Koska radioaaltojen eteneminen ei rakennetussa ympäristössä ole aina symmetristä osapuolten välillä, saattaa signaalin voimakkuus vaihdella eri suuntiin. Erityisesti lähellä kynnysarvoa olevien asemien päätös varoituksen lähettämisestä voi muuttua pienestäkin sopivaan suuntaan tapahtuvasta muutoksesta signaalin voimakkuudessa. Tästä johtuvien väärien varoitusten vähentämiseksi sallitaan jo liittyneille asemille pieni toleranssi niiden havaitessa uuden aseman signaalia [MZW08]. Esimerkiksi, jos kynnysarvo on -80 dBm, toleranssi 1 dBm, STA_1 on STA_U:n ilmoittamassa alueessa ja STA_1 havaitsee STA_U:n signaalin voimakkuudeksi -81 dBm, niin toleranssi huomioiden signaali [-80 dBm, -82 dBm] yltää kynnysarvoon eikä STA_1 lähetä varoitusta. Jos STA_1 ei olisi löytänyt itseään STA_U:n ilmoittamasta alueesta, se ei olisi myöskään lähettänyt varoitusta, koska toleranssin sallimissa rajoissa on mahdollista myös signaalin jääminen kynnysarvon alle.

Toleranssi vähentää aiheettomien varoitusten määrää, mutta tarjoaa myös hyökkääjälle parempia mahdollisuuksia tulvitushyökkäyksen tekemiseen [MZW08]. Hyökkääjän ympäristössä olevat asemat, jotka annetulla NST:llä ja toleranssilla sallivat sekä mukana- että poissaolonsa hyökkääjän ilmoittamassa alueessa, antavat hyökkääjälle

mahdollisuuden useiden erilaisten hyväksyttävien aluiden luomiseen. Jos esimerkiksi tällaisia asemia on k kappaletta, niin kaikki näiden kombinaatiot eli 2^k erilaista yhdistelmää muodostavat hyväksyttävän alueen, kunhan jokaiseen lisätään vielä ne asemat, jotka toleranssista huolimatta on pakko sisällyttää alueeseen. Jos tukiasema sallii samaa aluetta useammassakin authentication requestissa, niin varoitusta aiheuttamattomien authentication requestien määrä kasvaa moninkertaiseksi. Kuten aiemmin todettiin, pelkkä NST:n vaihtaminen ei vaikeuta hyökkäystä, mutta käytettäessä toleranssia sillä on vaikutusta. Muutos NST:ssä muuttaa huomattavasti niiden asemien joukkoa, jotka toleranssin johdosta eivät lähetä varoitusta. Hyökkääjä joutuu aina NST:n vaihtumisen jälkeen selvittämään nämä asemat uudestaan [MZW08].

Edellä kuvatussa ympäristön asemien signaalien vahvuuksien kuuntelemiseen perustuvassa tulvitusta torjuvassa tehtävässä on kuitenkin mielestäni eräs ongelma. Verkkoon aiemmin liittyneistä asemista osa voi olla juuri silloin hiljaa, kun STA_U kuuntelee ympäröivien asemien signaalien voimakkuuksia. Toisen aseman signaalin voimakkuus voidaan mitata vain aseman lähettäessä. Jos esimerkiksi STA_1 ei lähetä juuri silloin mitään, kun STA_U selvittää ketkä ovat sen naapureita ja ketkä siis muodostavat alueen, ei STA_U huomioi STA_1:tä ollenkaan. STA_1 kuulee kuitenkin STA_U:n lähettämän authentication requestin ja huomaa puuttuvansa STA_U:n alueesta, vaikka sen pitäisi olla siinä, jolloin STA_1 varoittaa tukiasemaa, eikä se hyväksy STA_U:n todennusta. Lisäksi havaitussa kehyksessä on oltava lähettäjän MAC-osoite, jotta signaalin taso voidaan yhdistää oikeaan asemaan. Lähettäjän MAC-osoite kuitenkin puuttuu joistakin MAC-kehyksistä, kuten esimerkiksi CTS:stä ja kuittauksista.

6.3.4 Sekvenssinumeroihin perustuva tulvituksen torjuminen

Aiemmin luvussa 6.2.2 esitettyjä sekvenssinumeroihin perustuvia väärennettyjen deauthentication- ja dissassociation-kehysten torjuntatapoja voidaan käyttää tulvitushyökkäystenkin torjuntaan. Lähettäessään deauthenticationin tai dissassosiationin hyökkääjä yrittää MAC-osoitteensa väärentämällä naamioitua joksikin toiseksi verkossa jo olevaksi asemaksi. Tulvituksessa hyökkääjän ei tarvitse naamioitua miksikään verkossa jo olevaksi asemaksi, vaan päinvastoin väärentämällä jokaiseen kehykseen eri MAC-osoitteen saada tukiasemaa luulemaan usean eri aseman lähettävän sille kehyksiä.

Normaalisti saman aseman lähettämissä hallinta- tai datakehyksissä olevat sekvens-

sinumerot kasvavat yhdellä. Tästä on joitakin poikkeuksia ja tukiaseman samalta asemalta vastaanottamien kehysten numerot saattavat normaalistikin hiukan poiketa kasvavasta numerojärjestyksestä kuten luvussa 6.2.2 kerrottiin. Tukiaseman saamassa deauthenticationissa tai dissassociationissa olevan sekvenssinumeron selvä poikkeaminen sekvenssinumeroiden aiemmasta sarjasta viittaa vahvasti väärennettyyn kehykseen, vaikka MAC-osoite olisikin oikea.

Tulvitusta tehdään probe-, authentication- tai association request -kehyksillä. Jos tukiaseman verkossa on vain normaalia ruuhkaa, niin jokaisessa sen saamassa kehyksessä on hyvin todennäköisesti täysin eri sekvenssinumero, koska ne tulevat eri asemilta. Jos yksi asema tekee tulvitusta, niin sekvenssinumerot muodostavatkin suhteellisen säännönmukaisen peräkkäisten numeroiden sarjan siitäkin huolimatta, että MAC-osoitteen perusteella kehykset näyttäisivät tulevan eri asemilta. Sekvenssinumerot siis paljastavat hyökkäyksen, mutta niitä on tulkittava täysin päin vastoin kuin havaittaessa purkukehyksillä tehtävää palvelunestohyökkäystä. Purkukehyksiä torjuttaessa oleellista on huomata purkukehyksen sekvenssinumeron poikkevan selvästi samasta MAC-osoitteesta saapuneiden purkukehystä edeltävien ja sen jälkeen tulevien sekvenssinumeroiden sarjasta. Tulvitushyökkäyksen paljastuttua voidaan väärennettyjä kehyksiä suodattaa. Ei MAC-osoitteen, vaan niissä olevien sekvenssinumeroiden perusteella.

Tulvitusta ei kuitenkaan havaita aivan heti hyökkäyksen alettua, mutta kun kehyksiä on seurattu hetken aikaa, voidaan niiden sekvenssinumeroissa havaita hyökkäyksen paljastavaa säännönmukaisuutta. Hyökkäyksen paljastuttua hyökkääjän lähettämät kehykset voidaan jättää huomiotta eikä tukiaseman resursseja kulu niiden käsittelyyn. Torjunta ei myöskään aiheuta juurikaan haittaa muille asemille. On erittäin pieni todennäköisyys, että oikeutetun aseman lähettämä probe, authentication- tai association request tulee väärin perustein torjutuksi. Jos tukiasema on tunnistanut hyökkäyksen ja aloittanut tiettyjä säännönmukaisesti eteneviä sekvenssinumeroita sisältävien kehysten suodattamisen, joutuu myös oikeutetun aseman lähettämä kehys suodatetuksi, jos sen sekvenssinumero on liian lähellä hyökkääjän lähettämien kehysten sekvenssinumeroita.

Hyökkääjä voi kiertää sekvenssinumeroihin perustuvaa tulvituksen paljastumista vaihtelemalla lähettämiinsä kehyksiin MAC-osoitteiden lisäksi myös sekvenssinumeroita. Sekvenssinumeroihin vaikuttaminen ei kuitenkaan ole helppoa, koska niitä käsittelee suoraan verkkokortin laiteohjelmisto, jota ei päästä helposti ohjaamaan. Joissain tapauksissa voi kuitenkin olla mahdollista vaikuttaa sekvenssinumeroihin

[BeS03]. Edellä esitettyä sekvenssinumeroihin perustuvaa tulvituksen torjumista ei tiettävästi ole esitetty aiemmin. Idea vaatii siis kokeita ja lisätutkimuksia ennen kuin sen tehokkuutta ja käyttökelpoisuutta voidaan tarkemmin arvioida.

7 Yhteenveto

Langattomassa verkossa on kaapeliverkkoa vaikeampaa saavuttaa hyvää tietoturvaa, koska laitteet jakavat yhteisen siirtotien ja kaikki radioaallot leviävät normaalisti joka puolelle ympäristöönsä. Laitteet ovat myös hyvin erilaisia ja niillä on käytettävissä vaihtelevasti resursseja. Tämä hankaloittaa kaikille sopivien, mutta riittävän tietoturvan takaavien, ominaisuuksien määrittelemistä. Langattomuus tuo toki etujakin, mutta se asettaa vaikeuksia luotettavalle todennukselle, salaukselle, eheydelle sekä palvelujen saatavuudelle.

Vuonna 1997 julkaistiin ensimmäinen WLANin määrittävä 802.11-standardi ja sen 15-vuotisen historian aikana WLAN onkin levinnyt räjähdysmäisesti. WLANin kehittäminen ja tietoturva on ollut jatkuvassa muutoksessa standardien tekijöiden, tietoturva-aukkojen hyödyntäjien, akateemisen tutkimuksen ja laitevalmistajien välisessä kilpajuoksussa. Nopeaa kehitystä on usein vielä rasittanut vaatimus taaksepäin yhteensopivuudesta, joka on rajoittanut verkkojen nopeuksia tai pakottanut käyttämään verkossa jo vanhentuneita tietoturvaratkaisuja.

WLANin tietoturva on parantunut vaiheittain. Aluksi käytössä oli WEP, mutta se osoittautui pian tietoturvaltaan lähes täysin epäonnistuneeksi. Se ei taannut luotettavaa todennusta, salausta eikä lähetysten eheyttä. WEPin puutteita korjaavaa 802.11i-standardia saatiin kuitenkin odottaa vuoteen 2004 saakka. Laitevalmistajat lisäsivät jo sitä ennen laitteisiinsa tulevaan standardiin suunniteltuja ominaisuuksia. WPA:ssa oli parannettu todennusta ja avaintenhallintaa. Todennusta ei enää WEPin tapaan tehty epäluotettavasti toteutetulla haaste-vaste-periaatteella, vaan käyttöön otettiin 802.1X:ään, ja joko EAPiin ja ulkoiseen RADIUS-palvelimeen tai vahtoehtoisesti PSK:hon perustuva todennus. Toisin kuin WEPissä, WPA:ssa myös tukiasema todentaa itsensä asemalle ja käytössä on nelivaiheinen kättely, jolla osapuolet varmistavat, että kumpikin on muodostanut saman PMK:n.

Salauksesta ja eheydentarkistuksesta vastasi WPA:ssa TKIP. Se käytti edelleen samaa RC4-jonosalausalgoritmia kuin WEP, mutta ennen salausta avainvirta muodostetaan turvallisemmin. Eheydentarkistukseen käytetty Michael-algoritmi tekee lähetysten huomaamattomasta muuttamisesta vaikeampaa. Parannuksista huolimatta

TKIP:ssäkin oli edelleen heikkouksia ja salaus oli helposti purettavissa. Eräs TKIP:n heikkous on vaatimus sen yhteensopivuudesta vanhojen vain WEPin tunnistavien laitteiden kanssa.

WPA2:n myötä WLANin salaus ja eheydentarkistus saatiin luotettaviksi. Taaksepäin yhteensopivuudesta luovuttiin kokonaan ja TKIP korvattiin CCMP:llä, joka huolehtii salauksesta ja eheydestä. Salaukseen käytetään AES-lohkosalausalgoritmia ja CBC-MACia eheydentarkistukseen. WLANin tietoturva on parantunut askel kerrallaan: aluksi WEPissä tietoturva oli täysin olematonta, WPA:ssa todennus oli jo luotettavaa, mutta salaus ja eheydentarkistus eivät olleet riittäviä joskin WEPiä parempia ja WPA2:ssa myös salaus ja eheydentarkistus oli toteutettu tarpeeksi laadukkaasti.

Vielä WPA2:n jälkeenkin WLANissa ovat edelleen ongelmana saatavuutta uhkaavat palvelunestohyökkäykset, joiden torjuntaan WPA2 ei tuonut mitään parannusta. Radiohäirintä käyttää hyväksi radioaalloilla tapahtuvan tiedonsiirron perusrajoitusta: siirtotiellä voi olla vain yksi lähettäjä kerrallaan. Häirintäsignaalia on hyvin helppo muodostaa. Sitä voidaan tuottaa joko täysin käytetystä protokollasta mitään tietämättä tai sitten älykkäämmin ajoittamalla häirintä vain tiettyihin tarkoin valittuihin protokollan kohtiin. Radiohäirinnän torjuminen on vaikeaa. Jossain määrin lähetystehon kasvattaminen tai kanavan vaihtaminen voi auttaa, mutta usein ainoiksi torjuntakeinoksi jäävät fyysiset toimenpiteet: suunnatun antennin käyttäminen, etäisyyden kasvattaminen hyökkääjään tai hyökkääjän lähettimen vaientaminen.

WPA:n myötä todennus WLANissa on ollut luotettavaa, mutta se keskittyy vain asemien todentamiseen niiden halutessa liittyä tukiasemaan. Kehyksiä ei todenneta WLANissa mitenkään. Ainoastaan hyvin helposti väärennettävissä oleva MAC-osoite kertoo kehyksen lähettäjän. Kehysten olemattoman todennuksen vuoksi hyökkääjä voi MAC-osoitteensa väärentämällä naamioitua helposti joksikin toiseksi verkon laitteeksi ja aiheuttaa väärennetyillä hallintakehyksillä erilaisia palvelunestohyökkäyksiä. Virtuaalihäirinnässä hyökkääjä hämää muut verkon laitteet asettamaan NAVinsa liian isoksi, jolloin ne pidättäytyvät siirtotielle pyrkimisestä. Hyökkäys vielä tehostuu, jos hyökkääjä yhdistää sen epäreilun lyhyeen odotusaikaan. Virtuaalihäirintää voidaan torjua tarkkailemalla siirtotietä ja pyrkimällä sille, jos se havaitaan vapaaksi, vaikka NAVin mukaan pitäisikin vain odottaa. Odotusajan arpominen voidaan siirtää asemilta tukiasemalle. Tällöin tukiaseman on myös valvottava, että asemat noudattavat saamiaan odotusaikoja. Väärennetyillä hallintakehyksillä voidaan myös häiritä laitteiden välistä käsitystä todennuksen ja liittymisen

tilasta. Tilojen purkukehyksiä on helppo väärentää ja vastaanottaja hyväksyy ne aina ehdoitta. Purkamisen jälkeen on suoritettava mahdollisesta hidas todennustai liittymisprosessi kokonaan uudestaan. Puutteellisesta kehysten todentamisesta aiheutuvia ongelmia voidaan torjua seuraamalla data- hallintakehyksissä olevien sekvenssinumeroiden säännönmukaisuutta. Tarvittaessa voidaan käyttää kehyksissä näennäissatunnaisia sekvenssinumeroita. Hyökkääjä voi väärennetyllä PS-Poll-kehyksellä huijata WLANissa käytössä olevaa virransäästötilaa ja saada tukiaseman poistamaan puskuroimansa datan. Tätä voidaan torjua salaamalla PS-Pollin AID-kenttä, joka osoittaa mille asemalle puskuroitu data on lähetettävä.

Aivan kuten internetissä myös WLANissa voidaan tehdä palvelunestohyökkäystä tulvittamalla. Hyökkääjän runsas hallintakehysten tulvitus kuluttaa tukiaseman resursseja eikä se voi enää palvella muita asemia. Tulvitusta voidaan tehdä ainakin probe request, authentication request ja association request -kehyksillä. Tulvitus saattaa onnistua myös reassociation request- ja PS-Poll-kehyksellä, mutta niiden käyttöä ei ole tutkittu. Tulvituksen torjuntaan on esitetty useita tehtäviin perustuvia tapoja. Niissä hyökkääjä joutuu jokaista lähettämäänsä tulvituskehystä kohti suorittamaan ensin jonkin hyökkääjän resursseja kuluttavan tehtävän. Voidakseen tehdä tulvitusta tällaisia tehtäviä on ratkaistava useita, mikä rajoittaa tulvitusta. Tehtävät perustuvat usein tiivistefunktioiden käyttöön, jolloin tukiasema voi nopeasti sekä muodostaa että tarkistaa tehtävän. Tehtäviä käytettäessä ongelmaksi voivat kuitenkin muodostua eri tasoiset laitteet. Tehtävä voi olla toiselle laitteelle hyvin helppo ja toiselle vähäisemmillä resursseilla varustetulle laitteelle turhan hankala. Ympäristön tarkkailuun ja naapureiden signaalien tasoon perustuvassa tehtävässä ei ole tätä ongelmaa. Siitä selviytyminen ei ole riippuvainen laitteen resursseista, vaan ympäristön tarkkailuun käytetystä ajasta. Ongelmaksi voivat kuitenkin muodostua verkkoon kuuluvat hiljaiset asemat, joita uusi asema ei havaitse, mutta jotka varoituksillaan kuitenkin estävät uutta asemaa liittymistä tukiaseman verkkoon.

Tulvitusta voidaan torjua myös kehysten sekvenssinumeroiden avulla. Niitä on aiemmin esitetty käytettäväksi vain purkukehysten havaitsemiseen, mutta ne soveltuvat myös tulvituksen torjuntaan. Purkukehysten havaitsemisessa hyökkäyksen paljasti sekvenssinumeroiden poikkeaminen säännönmukaisesta sarjasta. Tulvituksessa tilanne on juuri päinvastoin: eri MAC-osoitteista tulevien kehysten sekvenssinumeroiden säännönmukainen kasvaminen paljastaa hyökkääjän, kun taas niiden satunnainen jakaantuminen kertoo vain normaalista ruuhkasta.

Lähteet

ABA05 Aboba, B., Beadles, M., Arkko, J. ja Eronen, P., *The Network Access Identifier*, 2005. RFC 4282.

ABM05 Abadi, M., Burrows, M., Manasse, M. ja Wobber, T., Moderately hard, memory-bound functions. *ACM Transactions on Internet Technology*, 5,2(2005), sivut 299–327.

ABV04 Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. ja Levkowetz, H., *Extensible Authentication Protocol (EAP)*, 2004. RFC 3748.

Ahm10 Ahmad, S., WPA too, http://www.airtightnetworks.com/fileadmin/pdf/WPA-Too-Hole196-Defcon18-Presentation.pdf, 2010. [Viitattu 4.5.2012].

AIK06a Aslam, B., Islam, H. M. ja Khan, S. A., 802.11 disassociation DoS attack and its solutions: A survey. *Proc. of the first Internat. Conf. on Mobile Computing and Wireless Communication*, USA, 2006, sivut 221–226.

AIK06b Aslam, B., Islam, H. M. ja Khan, S. A., Pseudo randomized sequence number based solution to 802.11 disassociation denial of service attack. *Proc. of the first Internat. Conf. on Mobile Computing and Wireless Communication*, USA, 2006, sivut 215–220.

AST04 Acharya, M., Sharma, T., Thuente, D. ja Sizemore, D., Intelligent jamming in 802.11b wireless networks. *Proc. of the OPNETWORK Conf.*, USA, 2004.

AcT05 Acharya, M. ja Thuente, D., Intelligent jamming attacks, counterattacks and (counter)^2 attacks in 802.11b wireless networks. *Proc. of OPNETWORK Conf.*, USA, 2005.

Bec10 Beck, M., Enhanced TKIP Michael attacks, http://download.aircrack-ng.org/wiki-files/doc/enhanced_tkip_michael.pdf, 2010. [Viitattu 4.5.2012].

Ber04 Berghel, H., Wireless infidelity I: War driving. *Communications of the ACM*, 47,9(2004), sivut 21–26.

BFV08 Bernaschi, M., Ferreri, F. ja Valcamonici, L., Access points vulnerabilities to DoS attacks in 802.11 networks. *Wireless Networks*, 14,2(2008), sivut 159–169.

BGM01 Borisov, N., Goldberg, I. ja Wagner, D., Intercepting mobile communications: The insecurity of 802.11. *Proc. of the 7th annual Internat. Conf. on Mobile Computing and Networking*, USA, 2001, sivut 180–189.

BHL06 Bittau, A., Handley, M. ja Lackey, J., The final nail in WEP's coffin. *Proc. of the 2006 IEEE Symposium on Security and Privacy*, USA, 2006, sivut 386–400.

BeS03 Bellardo, J. ja Savage, S., 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. *Proc. of the 12th Conf. on USENIX Security Symposium*, USA, 2003, sivut 15–28.

BeT09 Beck, M. ja Tews, E., Practical attacks against WEP and WPA. *Proc. of the second ACM Conf. on Wireless Network Security*, USA, 2009, sivut 79–86.

Car03 Cardenas, E. D., MAC spoofing - an introduction, http://www.giac.org/paper/gsec/3199/mac-spoofing-an-introduction/105315, 2003. [Viitattu 4.5.2012].

Cer00 Cert, CERT advisory CA-1998-01 smurf IP denial-of-service attacks, http://www.cert.org/advisories/CA-1998-01.html, 2000. [Viitattu 4.5.2012].

Che06 Cheung, S., Denial of service against the Domain Name System. *IEEE Security and Privacy*, 4,1(2006), sivut 40–45.

Cla88 Clark, D., The design philosophy of the DARPA internet protocols. *ACM SIGCOMM Computer Communication Review*, 18,4(1988), sivut 106–114.

DCA07 Dantu, R., Clothier, G. ja Atri, A., EAP methods for wireless network. *Computer Standards & Interfaces*, 29,3(2007), sivut 289–301.

DGL10 Dong, Q., Gao, L. ja Li, X., A new client-puzzle based DoS-resistant scheme of IEEE 802.11i wireless authentication protocol. *Proc. of the 3rd Internat. Conf. on Biomedical Engineering and Informatics*, USA, 2010, sivut 2712–2716.

DwN92 Dwork, C. ja Naor, M., Pricing via processing or combatting junk mail. *Proc. of the 12th Annual Internat. Cryptology Conf. on Advances in Cryptology*, UK, 1992, sivut 139–147.

DiR08 Dierks, T. ja Rescorla, E., *The Transport Layer Security (TLS) Protocol Version 1.2*, 2008. RFC 5246.

Dro97 Droms, R., *Dynamic Host Configuration Protocol*, 1997. RFC 2131.

Eco10 Economist, The 24-hour athenian democracy, http://www.economist.com/blogs/babbage/2010/12/more_wikileaks, 2010. [Viitattu 4.5.2012].

FuB08 Funk, P. ja Blake-Wilson, W., *Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)*, 2008. RFC 5281.

Fer02 Ferguson, N., Michael: an improved MIC for 802.11 WEP, https://openwiki.uninett.no/_media/gigacampus:mobilitet:11-02-020r0-i-michael-an-improved-mic-for-802.11-wep.doc, 2002. [Viitattu 4.5.2012] doc.: IEEE 802.11-02/02r0.

FMS01 Fluhrer, S., Mantin, I. ja Shamir, A., Weaknesses in the key scheduling algorithm of RC4. *Proc. of the 8th Annual Internat. Workshop on Selected Areas in Cryptography*, UK, 2001, sivut 1–24.

FeS00 Ferguson, P. ja Senie, D., *Network Ingress Filtering: Defeating Denial of Service Attacks which emply IP Source Address Spoofing*, 2000. RFC 2827.

FaT05 Fanglu, G. ja Tzi-cker, C., Sequence number-based MAC address spoof detection. *Proc. of 8th Internat. Conf. on Recent Advances in Intrusion Detection*, Saksa, 2005, sivut 302–329.

FBV04 Ferreri, F. Bernaschi, M. ja Valcamonici, L., Access points vulnerabilities to DoS attacks in 802.11 networks. *Proc. of IEEE Conf. on Wireless Communications and Networking*, USA, 2004, sivut 634–638.

Gar00 Garber, L., Denial-of-service attacks rip the internet. *Computer*, 33,4(2000), sivut 12–17.

GaS05 Gast, M. S., *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, USA, 2005.

Gli84 Gligor, Virgi, D., A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, 10,3(1984), sivut 320–324.

HaD05 Hardjono, T. ja Dondeti, L. R., *Security in Wireless LANs and MANs*. Artech House, USA, 2005.

HaH09 Halvorsen, Finn, M. ja Haugen, O., Cryptanalysis of IEEE 802.11i TKIP. Pro gradu, Norwegian University of Science and Technology, kesäkuu 2009.

HeM05 He, C. ja Mitchell, J., Security analysis and improvements for IEEE 802.11i. *Proc. of the 12th Annual Symposium on Network and Distributed System Security*, USA, 2005, sivut 90–110.

HPW02 Hissam, S., Plakosh, D. ja Weinstock, C., Trust and vulnerability in open source software. *IEE Proceedings Software*, 149,1(2002), sivut 47–51.

HoW01 Houle, Kevin, J. ja Weaver, George, M., Trends in denial of service attack technology, www.cert.org/archive/pdf/DoS_trends.pdf, 2001. [Viitattu 4.5.2012].

IEE04a *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless (LAN) Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Medium Access Control MAC Security Enhancements*, 2004. IEEE Std 802.11i-2004 (Amendment to IEEE Std 802.11, 1999 Edition).

IEE04b *IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control*, 2004. IEEE Std 802.1X-2004.

IEE07 *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless (LAN) Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2007. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999).

IEE09 *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless (LAN) Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 5: Enhancements for Higher Throughput*, 2009. IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007).

IsM07 Issac, B. ja Mohammed, Lawan, A., War driving and WLAN security issues-attacks, security design and remedies. *Information Systems Management*, 24,4(2007), sivut 289–298.

JuB99 Juels, A. ja Brainard, J., Client puzzles: A cryptographic countermeasure against connection depletion attacks. *Proc. of the 1999 Symposium on Network and Distributed System Security*, USA, 1999, sivut 151–165.

Ken96 Kenney, M., Ping of death, http://insecure.org/sploits/ping-o-death.html, 1996. [Viitattu 4.5.2012].

Kle08 Klein, A., Attacks on the RC4 stream cipher. *Designs, Codes and Cryptography*, 48,3(2008), sivut 269–286.

Kor04a Korek, chopchop (experimental WEP attacks), http://www.netstumbler.org/unix-linux/chopchop-experimental-wep-attacks-t12489.html, 2004. [Viitattu 4.5.2012].

Kor04b Korek, Next generation of WEP attacks?, http://www.netstumbler.org/news/next-generation-of-wep-attacks-t12277-30.html, 2004. [Viitattu 4.5.2012].

KyV03 Kyasanur, P. ja Vaidya, Nitin, H., Detection and handling of MAC layer misbehavior in wireless networks. *Proc. of the 2003 Internat. Conf. on Dependable Systems and Networks*, 2003, sivut 173–182.

Lip02 Lipson, Howard, F., Tracking and tracing cyber-attacks: Technical challenges and global policy issues, www.cert.org/archive/pdf/02sr009.pdf, 2002. [Viitattu 4.5.2012].

Luc00 Lucent Technologies, *ORiNOCO Manager Suite User's Guide*, 2000.

LiY07 Liu, C. ja Yu, J., A solution to WLAN authentication and association DoS attacks. *International Journal of Computer Science*, 34,1(2007), sivut 31–36.

LYB10 Liu, C., Yu, J. ja Brewster, G., Empirical studies and queuing modeling of denial of service attacks against 802.11 WLANs. *Proc. of the 2010 IEEE Internat. Symposium on A World of Wireless, Mobile and Multimedia Networks*, USA, 2010, sivut 1–9.

MRH04 Moen, V., Raddum, H. ja Hole, K. J., Weaknesses in the temporal key hash of WPA. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8,2(2004), sivut 76–83.

MVS01 Moore, D., Voelke, G. M. ja Savage, S., Inferring internet denial-of-service activity. *Proc. of the 10th Symposium on USENIX Security*, USA, 2001, sivut 115–139.

MZW08 Martinovic, I., Zdarsky, F. A., Wilhelm, M., Wegmann, C. ja Schmitt, Jens, B., Wireless client puzzles in IEEE 802.11 networks: Security by wireless. *Proc. of the first ACM Conf. on Wireless Network Security*, USA, 2008, sivut 36–45.

Nee94 Needham, Roger, M., Denial of service: an example. *Communications of the ACM*, 37,11(1994), sivut 42–46.

NIS01 National Institute of Standards and Technology, *Advanced Encryption Standard (AES) (FIPS PUB 197)*, 2001.

NTN08 Nguyen, Thuc, N., Tran, Bao, N. ja Nguyen, D., H., A lightweight solution for wireless LAN: Letter-envelop protocol. *Proc. of the Third Internat. Conf. on Communications and Networking in China*, USA, 2008, sivut 17–21.

OhM09 Ohigashi, T. ja Morii, M., A practical message falsification attack on WPA. *Proc. of the Joint Workshop on Information Security*, USA, 2009.

Pax01 Paxson, V., An analysis of using reflectors for distributed denial-of-service attacks. *ACM SIGCOMM Computer Communication Rewiev*, 31,3(2001), sivut 38–47.

PaH02 Papadimitratos, P. ja Haas, Z. J., Securing the internet routing infrastructure. *IEEE Communications Magazine*, 40,10(2002), sivut 60–68.

PaL01 Park, K. ja Lee, H., On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. *ACM*

SIGCOMM Computer Communication Review, 31,4(2001), sivut 15–26.

PLR07 Peng, T., Leckie, C. ja Ramamohanarao, K., Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, 39,1(2007).

Plu82 Plummer, David., C., *An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*, 1982. RFC 826.

QAM07 Qureshi, Zaffar, I., Aslam, B., Moshin, A. ja Javed, Y., A solution to spoofed PS-poll based denial of service attacks in IEEE 802.11 WLANs. *Proc. of the 11th WSEAS Internat. Conf. on Communications*, USA, 2007, sivut 7–11.

Rap99 Rappaport, T. S., *Wireless communications: Principles & Practice.* Prentice Hall, USA, 1999.

RWR00 Rigney, C., Willens, S., Rubens, A. ja Simpson, W., *Remote Authentication Dial In User Service (RADIUS)*, 2000. RFC 2865.

SAH08 Simon, D., Aboba, B. ja hurst, R., *The EAP-TLS Authentication Protocol*, 2008. RFC 5216.

Sch82 Scholtz, Robert, A., The origins of spread-spectrum communications. *IEEE Transactions on Communications*, 30,5(1982), sivut 822–854.

Sec11 Securelist, Analysis - wardriving, http://www.securelist.com/en/analysis?topic=199380196. [Viitattu 4.5.2012].

SIR02 Stubblefield, A., Ioannidis, J. ja Rubin, A., Using the fluhrer, mantin, and shamir attack to break WEP. *Proc. of Internet Society Symposium on Network and Distributed System Security*, USA, 2002.

SKK97 Schuba, Christoph, L., Krsul, Ivan, V., Kuhn, Markus, G., Spafford, Eugene, H., Sundaram, A. ja Zamboni, D., Analysis of a denial of service attack on TCP. *Proc. of the 1997 IEEE Symposium on Security and Privacy*, USA, 1997, sivut 208–223.

SpL04 Specht, Stephenk, M. ja Lee, Ruby, B., Distributed denial of service: Taxonomies of attacks, tools and countermeasures. *Proc. of the 17 Internat. Conf. on Parallel and Distributed Computing Systems*, 2004, sivut 543–550.

Sta10 Stallings, W., *Cryptography and Network Security - Principles and Practice*. Prentice Hall, USA, 2010.

Tew07 Tews, E., Attacks on the WEP protocol. Pro gradu, Technical University of Darmstadt, joulukuu 2007.

Tra07 Traynor, I., Russian accused of unleashing cyberwar to disable Estonia, http://www.guardian.co.uk/world/2007/may/17/topstories3.russia, 2007. [Viitattu 4.5.2012].

TWP07 Tews, E., Weinmann, R.-P. ja Pyshkin, A., Breaking 104 bit WEP in less than 60 seconds. *Proc. of the 8th Internat. Conf. on Information Security Applications*, Saksa, 2007, sivut 188–202.

Vie11 Viestintävirasto, Määräys luvasta vapaiden radiolähettimien yhteistaajuuksista ja käytöstä, http://www.ficora.fi/attachments/suomiry/62anS1c3N/Viestintavirasto_15AC2011M.pdf, 2011. [Viitattu 4.5.2012].

VSS02 Vixie, P., Sneeringer, G. ja Schleifer, M., Events of 21-oct-2002, http://c.root-servers.org/october21.txt, 2002. [Viitattu 4.5.2012].

WHF03 Whiting, D., Housley, R. ja Ferguson, N., *Counter with CBC-MAC (CCM)*, 2003. RFC 3610.

Woo04 Wool, A., A note on the fragility of the "Michael" message integrity code. *IEEE Transactions on Wireless Communications*, 3,5(2004), sivut 1459–1462.

Wri03 Wright, J., Detecting wireless LAN MAC address spoofing, http://www.willhackforsushi.com/papers/wlan-mac-spoof.pdf, 2003. [Viitattu 4.5.2012].

WSZ10 Wang, P., Sparks, S. ja Zou, Cliff, C., An advanced hybrid peer-to-peer botnet. *IEEE Transactions on Dependable and Secure Computing*, 7,2(2010), sivut 113–127.

WZS02 Wang, H., Zhang, D. ja Shin, Kang, G., Detecting SYN flooding attacks. *Proc. of INFOCOM 2002. Twenty-First Annual Joint Conf. of the IEEE Computer and Communications Societies*, USA, 2002, sivut 1530–1539.

XTZ05 Xu, W., Trappe, W., Zhang, Y. ja Wood, T., The feasibility of launching and detecting jamming attacks in wireless networks. *Proc. of the 6th ACM Internat. Symposium on Mobile Ad Hoc Networking and Computing*, USA, 2005, sivut 46–57.