

Madonreiät langattomissa ad hoc -verkoissa

Jan Wikholm

Kandidaatintutkielma
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 4. toukokuuta 2014

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Jan Wikholm			
Työn nimi — Arbetets titel — Title			
Madonreiät langattomissa ad hoc -verkoissa			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma		4. toukokuuta 2014	21
Tiivistelmä — Referat — Abstract			
<p>Langattomat ad hoc -verkot mahdollistavat laitteiden autonomisesti muodostaa verkon ja sen reitityksen. Näitä verkkoja voidaan käyttää esimerkiksi pelastustoiminnassa infrastruktuurin puuttuessa. Verkon reitityksen itseohjautuvuus altistaa sen erityisen vakavalle hyökkäystavalle, jota kutsutaan madonreikähyökkäykseksi. Tässä hyökkäyksessä paha-aikaiset laitteet eivät tarvitse edes verkon salausavaimia voidakseen aiheuttaa häiriötä reititykseen tarjoamalla näennäisesti nopeimman ja lyhimmän polun verkon läpi. Tätä hyökkäystä vastaan on yli vuosikymmenen ajan kehitetty eri puolustuskeinoja, joita tässä tutkielmassa kootaan yhteen ja tyypitetään kahteen kategoriaan: puhtaasti protokollapohjaiset sekä erillislaitteistoa vaativat. Kirjoitus on tyyliltään katsausartikkeli, jossa käsitellään ja kootaan aiheen kirjallisuudesta tärkeitä puolustusmekanismeja, ja täten sopii jatkotutkimuksen pohjaksi esimerkiksi jonkin uuden puolustusmekanismin kehitykseen.</p> <p>ACM Computing Classification System (CCS): C.2.2 [Network Protocols] K.6.5 [Security and Protection]</p>			
Avainsanat — Nyckelord — Keywords			
ad hoc -verkot, wlan, hyökkäys, puolustus, havainnointi, madonreikähyökkäys, tietoliikenneprotokollat			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

Sisältö

1	Johdanto	1
2	Hyökkääjä- ja hyökkäystyypit	2
2.1	Hyökkääjätyypit	2
2.2	Hyökkäystyypit	2
3	Laitteistoriippuvaiset puolustusmekanismit	4
3.1	Pakettihinnat ja TIK-protokolla	5
3.1.1	Aikahinnat	5
3.1.2	Geohinnat	5
3.1.3	TIK-protokolla	6
3.1.4	Resurssivaativuus	7
3.2	Suunta-antennin käyttö madonreikähyökkäysten estämisessä .	8
3.2.1	Naapurien löytäminen	8
3.2.2	Todentajalaite	9
3.2.3	Worawannotai-hyökkäys ja sen estäminen	10
3.3	SECTOR ja MAD	10
3.3.1	MAD-protokolla	11
3.3.2	Kohtaamisen tuoreus- ja aikatakuu	12
3.3.3	Hyökkääjien ja hyökkäysmallien tyypitys	14
4	Puhtaasti protokollapohjaiset puolustukset	15
4.1	DeWorm	15
4.2	DelPHI	16
4.3	LiteWorp	17
4.3.1	Paikallinen valvonta	17
4.3.2	Eistäminen	18
4.4	MobiWorp	19
4.4.1	SMP-protokolla	19
4.4.2	CAP-CV-protokolla	20
5	Yhteenveto	20
	Lähteet	21

1 Johdanto

Langattomat päätelaitteet – kuten matkapuhelimet, PDA:t ja kannettavat tietokoneet – voivat muodostaa langattoman ad hoc -verkon, jonka avulla ne voivat kommunikoida ilman erillistä verkkoinfrastruktuuria [CL06]. Sensori- ja ad hoc -verkot voivat toimia viestintäalustana monissa erilaisissa käyttötarkoituksissa, kuten pelastus-, armeija- [HPJ03] sekä siviilikäytössä [KBS05]. Esimerkiksi luonnonkatastrofin jäljiltä perinteiset langattomat tukiasemat voivat olla tuhoutuneet, jolloin pelastuslaitosten työntekijät ovat viestinnässään ad hoc -verkkojen varassa [HPJ03].

Ad hoc -verkkojen suurimpia etuja ovat käyttöönoton nopeus ja kustannustehokkuus [CL06, HPJ03], sillä laitteisto on usein edullista ja päätelaitteet osaavat itsenäisesti luoda verkon. Vaikka ad hoc -verkkoja voi muodostaa myös langallisesti, on useimmiten käytössä langattomat teknologiat [HPJ03] ja siksi keskitymme niihin.

Pääosa ad hoc -verkkojen alkuvaiheen tutkimuksesta on keskittynyt verkkojen käyttöönoton autonomisuuden parantamiseen ja laitteistovaatimusten pienentämiseen. Tutkimus on tuottanut reititysprotokollia ja muita välttämättömiä viestinnän osia [KBS05]. Ad hoc -verkkojen avoimuuden ja autonomisuuden seurauksena ne ovat erityisen haavoittuvia monille erilaisille hyökkäyksille: *salakuuntelu* (eavesdropping), *väärennys* (spoofing) ja *toistaminen* (replay) [HPJ03]. Näiden lisäksi hyökkääjä voi tahallisesti olla välittämättä paketteja, mitä kutsutaan *musta aukko -hyökkäykseksi* (blackhole attack), tai syöttää niitä verkkoon paljon tukehduttaakseen sen järkevän käytön, mitä kutsutaan nimellä *valkoinen aukko -hyökkäys* (white hole attack) [CL06]. *Madonreikähyökkäys* (wormhole attack) on erityisen vakava hyökkäys ad hoc -verkoissa [KBS05].

Madonreikähyökkäyksessä yleensä kaksi tai useampi paha-aikeista tahoa toimivat yhteistyössä saadakseen liikenteen ohjautumaan niiden välillä kulkevaa reittiä pitkin, jotta voivat toteuttaa yllä lueteltuja hyökkäyksiä. Hyökkääjät välittävät kaikki kuulemansa paketit toisille hyökkääjille toistettavaksi. Tämä pakettien välitys voidaan toteuttaa hyökkääjien omalla suurinopeuksisella tiedonsiirtokanavalla, pakettien kapseloinnilla normaalia verkkoa pitkin tai vaikka suuritehoisella lähettimellä [KBS05]. Tunnelin ollessa toiminnassa se häiritsee reititysprotokollia tarjoten lyhimmän ja yleensä nopeimman reitin, joten muut verkon laitteet päätyvät lähettämään suuren osan paketeista sen läpi.

Madonreikähyökkäyksen toiminta perustuu nimenomaan ad hoc -verkkojen autonomiseen reititykseen. Tukiasemiin perustuvassa verkossa vastaava ei voi toimia, koska kaikki liikenne ohjataan aina tukiasemien kautta.

Erityisen salakavalan hyökkäyksestä tekee se, että hyökkääjien ei tarvitse murtaa mitään salausta, koska koko hyökkäys perustuu pakettien kopiointiin (salakuunteluun ja sen jälkeiseen toistoon) verkon osasta toiseen.

Esittelen luvussa 2 madonreikähyökkäysten hyökkääjä- [CL06] ja hyökkäystyypit [KBS05], minkä jälkeen kerron laitteistoriippuvaisista puolustuskeinoista luvussa 3 ja protokollapohjaisista puolustuksista luvussa 4.

2 Hyökkääjä- ja hyökkäystyypit

Madonreikähyökkääjiä on kahta eri tyyppiä: *piilotettu* ja *avoin* [CL06] ja hyökkäyksiä on viittä eri tyyppiä [KBS05]. Kaikkia hyökkäystyyppejä ei ole käsitelty kaikissa puolustuskeinoissa, joten käsittelemättömien hyökkäystyyppien torjumisen toimivuus on erinomainen kohde jatkotutkimukselle. Hyökkäys- ja hyökkääjätyyppejä on kuvattuna kuvassa 1.

2.1 Hyökkääjätyypit

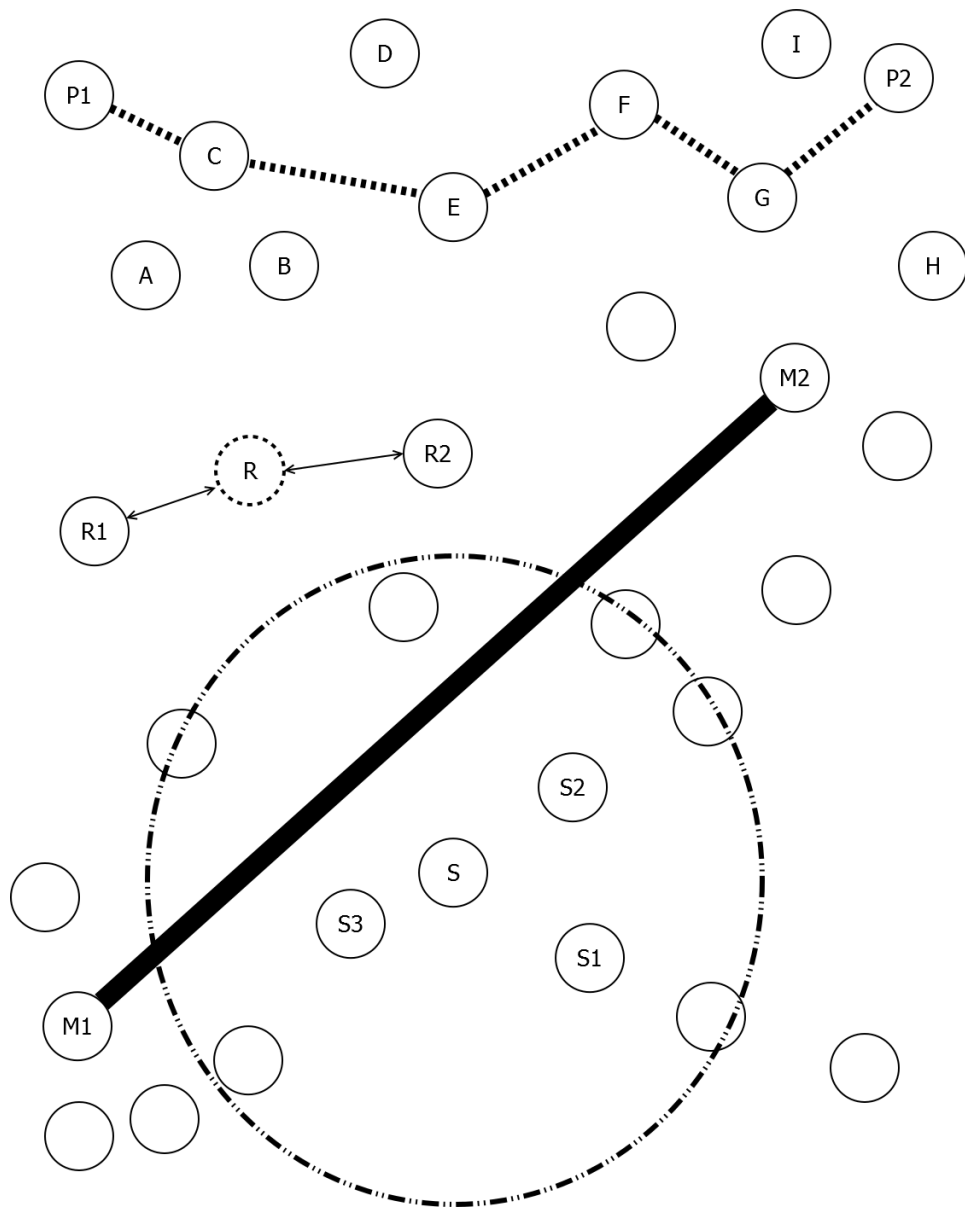
Piilotetut hyökkääjät toimivat verkossa kertomatta muille verkon laitteille omasta olemassaolostaan. Ne kuuntelevat liikennettä ja siirtävät paketteja madonreiän läpi täysin muokkaamatta. Tällöin kaukanakin olevat laitteet voivat luulla madonreiän läpi tulevia paketteja naapureilta tuleviksi, koska eivät tiedä välissä olevan toistimen toimivan madonreikänä. Katso kuvan 1 solmu R.

Esimerkiksi pakettihihnat, jotka esitellään luvussa 3.1, toimivat piilotettuja hyökkääjiä vastaan.

Avoimet hyökkääjät rekisteröityvät verkkoon kuten muutkin laitteet. Muille verkon laitteille näyttää siltä, että nämä laitteet ovat ensimmäisen asteen naapureita ja siten niiden kautta löytyvä lyhyt polku on täysin käypä vaihtoehto. Tapa, jolla madonreikä on muodostettu, on jokin alla kuvailluista viidestä hyökkäystavasta.

2.2 Hyökkäystyypit

Erilliskaistahyökkäys on se hyökkäystyyppi, johon yleisimmin viitataan nimellä *madonreikähyökkäys*. Mikäli hyökkääjillä on normaalin lähetykskaistan – esimerkiksi wlan-verkon – lisäksi käytössä vaikkapa yksinkertaisesti kytketty ethernetverkko, voivat ne kommunikoida yleistä verkkoa nopeammin ja sen kantamaa pidemmälle. Tämä sotkee reititysprotokollia ja suuri osa paketeista päättyy kulkemaan hyökkääjien linkin läpi. Kuvassa 1 laitteiden M1 ja M2



Kuva 1: Eri madonreikähyökkäystyypit yhdessä kuvassa.

välillä on erilliskaistaa kuvaava tumma viiva. Kaikki M1 lähistöllä olevat laitteet luulevat olevansa myös M2:n lähistöllä olevien laitteiden naapureita.

Pakettikapseloinnissa kuvan 1 hyökkääjät P1 ja P2 voivat käyttää jo olemassa olevaa verkkoa: P1 kuulee paketin ja luo uuden P2:lle suunnatun paketin, jonka sisältönä on sellaisenaan P1:n kaappaama paketti. Kun tämä kapseloitu paketti saavuttaa P2:n, se toistaa alkuperäisen sisällön sellaisenaan

verkkoon, joten P2:n naapurit luulevat olevansa P1:n naapureiden lähellä. Tämä ei vaadi hyökkääjien osalta mitään muista verkon laitteista poikkeavia resursseja. Täten esimerkiksi laite C luulee, että sen lyhin reitti laitteeseen I on $C - P1 - I$ (2 hyppyä) vaikka todellisuudessa paketti vain kapseloidaan ja sen oikea reitti on $C - P1 - C - E - F - G - P2 - I$ (7 hyppyä), joka on pidempi matka kuin sen alkuperäinen $C - E - F - G - P2 - I$ reitti (5 hyppyä).

Suurteholähetys on malliesimerkki siitä, että hyökkääjille voi olettaa loputtomat resurssit toisin kuin muille verkon laitteille, joita nimenomaan yleensä yhdistää resurssien vähyys. Tässä hyökkäystyypissä kuvan 1 hyökkääjälaitte S lähettää kaappaamansa reitityspaketit huomattavan suurella lähetysteholla (katkoviivalla S:n ympärillä) ja täten saa aikaan sen itsensä lävitse kulkevan lyhimmän reitin. Tämä hyökkäys ei siis vaadi kahta osapuolta. Täten kaikki S:n lähetyksen kuuluvuusalueella olevat laitteet luulevat olevansa S:n naapureita ja yhden hypyn päässä laitteista S1, S2 ja S3. Mikäli nämä huijatut laitteet yrittävät lähettää viestiä S:n kautta, ne epäonnistuvat siinä, koska niiden lähetysteho ei riitä lähettämään S:lle asti.

Pakettivälitys – kuten suurteholähetys – ei vaadi hyökkääjäparia vaan sen voi suorittaa yksikin vihamielinen laite. Tässä kuvan 1 hyökkääjä R on kahden laitteen, R1 ja R2, välissä välittäen paketteja, jolloin nämä laitteet luulevat olevansa naapureita ja hyökkääjä niiden välissä voi esimerkiksi salakuunnella tai suorittaa palvelunestohyökkäyksen. Tämä on perinteinen ns. välimeshyökkäys, mutta se on tyypiltään samanlainen kuin madonreikähyökkäys ja siksi mukana tässä.

Protokollapoikkeamat ovat paha-aikeisten laitteiden tahallista verkko-protokollan rikkomista: esimerkiksi pakettitörmäysten estämiseksi tietyt protokollat vaativat reitityspakettien lähetyksessä pientä viivettä - paha-aikeinen laite voi täten lähettää paketit heti ja aiheuttaa tavallisille laitteille haittaa törmäyksillä. Toinen vaihtoehto on reitityspakettien lähettämättä jättäminen, jolloin verkon reititys ja polunetsintä eivät toimi oikein. Kummassakin tapauksessa hyökkääjä voi junailla toimensa siten, että suuri osa verkon liikenteestä päätyy reitittymään sen läpi.

3 Laitteistoriippuvaiset puolustusmekanismit

Alla kuvatut puolustusmekanismit eivät vaadi reititysprotokolliin muutoksia, mutta niillä on laitteistovaatimuksia.

3.1 Pakettihihnat ja TIK-protokolla

Yih-Chun Hu et al [HPJ03] kuvailevat uuden mekanismin - *pakettihihnat* (packet leashes) - ja sen kaksi eri varianttia: *aikahihnan* (temporal leash) ja *geohihnan* (geographic leash). Tässä hihnalla tarkoitetaan sellaista tietoa, jolla paketin enimmäiskantamaa voidaan rajoittaa. Tällaista tietoa ovat esimerkiksi paketin vanhenemista ilmaiseva aikaleima tai lähettäjän paikkatiedon ja paketin maksimietäisyyden yhdistelmä.

Pakettihihnojen lisäksi he luovat uuden TIK-verkkoprotokollan, joka käyttää aikahihnoja madonreikiä vastaan.

3.1.1 Aikahihnat

Aikahihnojen edellytyksenä on tarkka kellojen synkronointitarkkuus: muutamien mikrosekunnin tai jopa satojen nanosekuntien tarkkuus. Kaikkien laitteiden pitää myöskin olla tietoisia virhemarginaalin suuruusluokasta kahden laitteen välillä. Tällainen synkronointitarkkuus onnistuu esimerkiksi GPS:n avulla. Vaikka tarkkuusvaatimus on erittäin tiukka, on se kirjoittajien mukaan täysin hyväksyttävä ottaen huomioon madonreikähyökkäyksen vakavuuden.

Aikahihnana muodostaessaan lähettäjä päättää enimmäiskantaman lähetykselle ja laskee kellojen synkronoinnin virhemarginaalin huomioiden, kauanko valonnopeudella kulkevalla radiosignaali kestää sinne päätyä. Tämän avulla lähettäjä asettaa paketille vanhenemisajan, jonka jälkeen paketti on hylättävä. Vaikka radiosignaali ei etenekään valonnopeutta ilmassa ja voi hidastua esteistä johtuen, voi sen perusteella laskea absoluuttisen enimmäiskantaman signaalin etenemiselle tietyssä ajassa. Viestien tunnelointi aiheuttaa viivettä niiden kulkiessa fyysisesti kauemmaksi. Tämän viiveen takia viesti ehtii vanhentua ja madonreiän toisella puolella olevat laitteet hylkäävät sen. Mikäli kellot on synkronoitu satojen nanosekuntien tarkkuudella on virhemarginaali n. 30–300 metriä.

Vaikka aikahihna on hihnatyypeistä tarkempi, on sen haasteena se, ettei lähettäjä tiedä aina tarkalleen omaa lähetysaikaansa, koska fyysisen kerroksen lähetysmekanismi voi joutua odottamaan vuoroaan. Täten on vaikeaa etukäteen luoda lähetysajankohtaan perustuvaa digitaalista allekirjoitusta. Tämän kirjoittajat ovat ratkaisseet kohta esiteltävässä TIK-protokollassa.

3.1.2 Geohihnat

Geohihnoja käytettäessä kaikkien verkon laitteiden pitää tietää oma sijain-

titietonsa sekä niiden pitää pystyä synkronoimaan kellonsa muiden kanssa. Kellojen synkronointitarkkuus ei ole niin tärkeä seikka, koska laitteiden liikkumisvauhti suhteessa valonnopeuteen on marginaalinen.

Geohihnan toiminta on hyvin yksinkertainen: laite tarkistaa onko sen oma sijainti alkuperäisessä paketissa määritellyn sallitun matkan päässä. Koska paketit allekirjoitetaan digitaalisesti, voi laite luottaa paikannustiedon olevan alkuperäiseltä lähteeltä. Toisin kuin pelkkä matkan pituuteen perustuva tarkistus geohihnat toimivat myös siinä tilanteessa, että madonreikä kuljettaa paketin jonkin esteen ohi.

Geohihnoissa laitteet tietävät toistensa oletetun maksimiliikenopeuden, ja täten madonreikä-tunnettu paketti voidaan tunnistaa ilkeävaltaisen laitteen lähettämäksi, jos se lähettää verkkoon kaksi pakettia, joiden sijainnissa on tapahtunut maksimiliikenopeutta nopeampaa liikettä edellyttävä muutos.

3.1.3 TIK-protokolla

Tärkeimpänä ominaisuutena koko aikahihnafunktion toiminnassa on aikaleimojen luotettavuus: leimat pitää pystyä todentamaan. Kirjoittajat hylkäävät jaetut avaimet todentamismekanismina, koska niiden hallinta on liian raskasta. Toisena ideana on digitaalisen allekirjoituksen liittäminen jokaiseen pakettiin, jolloin jokaiselle laitteelle riittää yksi julkinen-yksityinen-avainpari ja jokaisen laitteen tarvitsee tietää vain tämä kaikkien julkisten avainten joukko. Digitaaliset allekirjoitukset kuitenkin yleensä perustuvat raskaaseen asymmetriseen kryptografiaan, joka ei sovi ad hoc -verkkojen oletettuun resurssivähyyteen.

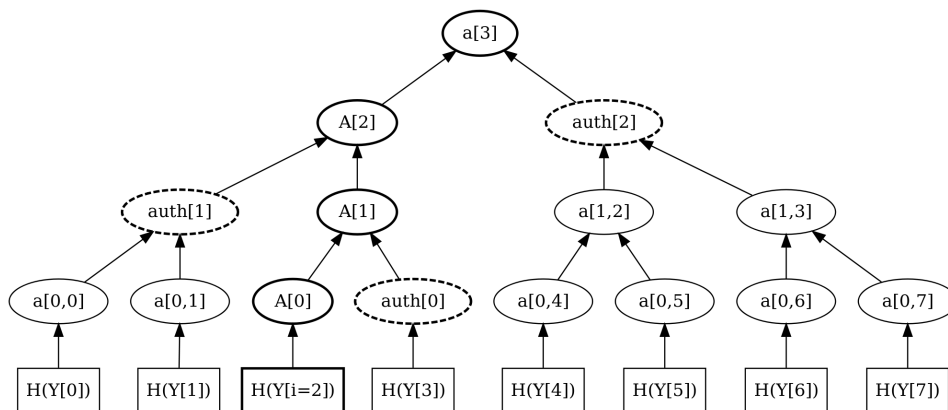
Symmetriseksi vaihtoehdoksi tarjotaan tiivisteistä koostuvaa binääripuuta (Merklen puu). Tiivistekäyttöön hyväksi puoleksi kerrotaan niiden erittäin tehokas laskeminen ja yksisuuntaisuus. Merklen puussa on 2^n lehteä, joista jokainen sisältää arvonaan kertakäyttöisen avaimen. Lehtien lukumäärä riippuu siitä, kuinka monta kertakäyttöistä avainta halutaan pitää valmiiksi laskettuna. Puun kaikkien vanhempien arvoksi lasketaan tiiviste sen lapsien arvoista. Tätä lapsista vanhemman arvoksi -tiivistelaskemista jatketaan kunnes saadaan puun juurisolmun arvo laskettua. Tämä juuritiiviste on laitteen julkinen avain ja lehtien arvot ovat yksityinen avain.

Lähetäjän avainnippu koostuu satunnaisluvuista, jotka on tallennettu Merklen puun pohjalle siten että ne on indeksoitu juoksevaan järjestykseen. Tätä järjestyslukua vastaa ajanhetkien järjestysluku. Ajanlasku alkaa laitteiden keskenään sopimasta hetkestä ja etenee sovituin lisäyksin (esimerkiksi 11,5 mikrosekunnin välein).

Vastaanottaja tietää jo etukäteen lähettäjän julkisen avaimen, eli puun juuritiivisteen, ja jokaisen paketin yhteydessä se saa neljä uutta datapalaa:

1. tiivisteen viestistä ja avaimesta K_i ,
2. viestin,
3. tiedot, joiden avulla avaimesta K_i saadaan laskettua juuritiivistä, sekä
4. avaimen K_i .

Kohtaa 3. varten lähettäjä paljastaa Merklen puustaan niiden solmujen arvot, joita tarvitaan, että kohdassa 4. paljastettavasta avaimesta saadaan laskettua tiivistet lehdessä juureen asti (katso kuva 2). Nämä osat tulevat tarkasti tässä järjestyksessä. Koska avainta ei ole viestin tiivistelaskennan aikaan vielä lähetetty, vastaanottaja voi nyt olla varma, ettei kukaan ole voinut väärentää tiivistettä. Vastaanottajan tarkistettua viestin aitouden se tarkistaa vielä aikahihnan rajat ja laskee, onko paketti vanhentunut.



Kuva 2: Avaimen $A[0]$ allekirjoituksen tarkistukseen tarvittavat tiivistepuun osat: $auth[0]$, $auth[1]$, $auth[2]$. Näillä voidaan laskea juuritiiviste $a[3]$ [Tsu14]

3.1.4 Resurssivaativuus

Resurssien niukkuuteen kirjoittajat mainitsevat päätarkaisuna edellä kuvattun symmetrisen kryptografian käytön asymmetrisen sijaan. Asymmetrisessä kryptografiassa operaatiot voivat olla kolmesta neljään suuruusluokkaa hitaampia (100–1000-kertaisia).

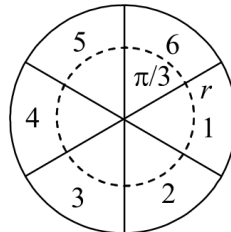
TIK-protokollaa arvioidessaan kirjoittajat tarkastelivat silloisten, vuoden

2001, mobiililaitteiden laskentatehoa ja muistimäärää. He toteavat laitteiden täyttävän protokollan laskenta- ja muistivaatimukset. He huomauttavat, ettei TIK sovi resursseiltaan aivan kaikkein rajallisimpiin ympäristöihin kuten sensoriverkkoihin.

Allekirjoitusta varten muodostettavan tiivistepuun muodostusta ja tallennusta voidaan optimoida säilömällä vain osa puusta muistissa ja laskemalla loput tarvittaessa. Tällaisen optimoinnin takia yhden vuorokauden yksityiset avaimet sisältävä osittainen puu saadaan mahtumaan 2,5 megatavuun kokonaisen puun 170 gigatavuun sijaan.

3.2 Suunta-antennin käyttö madonreikähyökkäysten estämisessä

Lingxuan Hu ja David Evans kuvaavat suunta-antennin käyttöä madonreikäklien estämisessä [HE04]. Suunta-antenneja on kahdentyyppisiä: ohjattuja ja kytkettyjä. Ohjatut suunta-antennit tarjoavat tarkempaa suuntausta, mutta ne ovat yleensä liian kalliita sensoriverkkoihin. Kytkeyty suunta-antenni pitää sisällään useamman staattisesti suunnatun antennin, jotka muodostavat yhtä suurista lähetysspektoreista 360 asteen kattavuuden. Näitä antenneja kytke-tään päälle tarvittaessa. Suuntauksessa käytetään laitteen sisäistä kompassia sektorin numero yksi osoittaessa aina itään.



Kuva 3: Suunta-antennin sektorit sekä lähetysteho suunnattuna (ulompi kaari) sekä suuntaamattomana (katkoviivalla) [HE04]

Protokolla perustuu naapurilistoihin ja niiden ylläpitoon. Protokolla olettaa, että verkossa on mekanismi, jolla laitteet voivat muodostaa turvallisen linkin keskenään ja että kriittiset viestit ovat salattu.

3.2.1 Naapurien löytäminen

Kun laite A haluaa liittyä verkkoon, se lähettää jokaiselle sektorille yleisen liittymisviestin, joka sisältää A:n oman tunnisteen sekä tiedon ko. sektorin numerosta. Tämä sektori on suunnattu kaikilla laitteilla samaan suuntaan.

Kun laite B vastaanottaa A:n viestin omalla sektorillaan X, se ensimmäisenä tarkistaa, että A:n ilmoitettu sektori on X:stä vastakkainen. Mikäli sektori ei ole vastakkaisella puolella, kalibrointivirheen takia tai muusta syystä, pudottaa vastaanottaja paketin eikä vastaa siihen.

Mikäli A:n paketti on oikeellinen, B lähettää A:lle A:n tunnisteen salattuna näiden yhteisellä jaetulla avaimella sekä oman tunnistensa. Jos B:n vastaus tulee A:n lähetyssektorin suunnalta ja B on ilmoittanut vastakkaisen sektorin omaksi lähetyssektorikseen sekä A onnistuu purkamaan B:n vastauksen ja saamaan vastaukseksi oman tunnistensa, A lisää B:n tunnisteen omaan naapurilistaansa. Tässä vaiheessa A ei ole välttämättä vielä B:n naapurilistalla, vaan B hoitaa oman naapuriensa löytöyrityksen itse valitsemanaan satunnaisena ajankohtana.

Tämä yksinkertainen naapurietsintä ei vielä estä madonreikiä. Esimerkiksi alla oleva lineaarinen tapaus on edelleen mahdollinen:

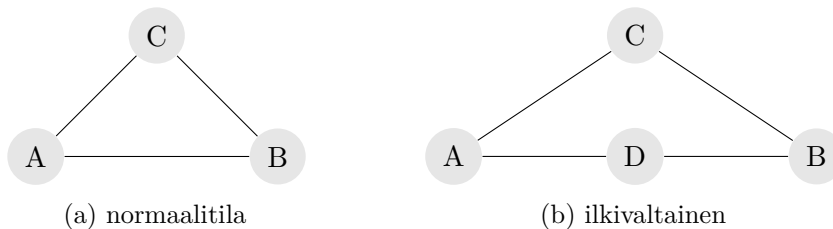
$$A- > M1 - - - - - > M2- > B$$

Nyt A:n ja B:n välissä on madonreiän muodostavat laitteet M1 ja M2. Tässä tapauksessa madonreikä voisi olla mielivaltaisen pitkä ja silti laitteet luulisivat olevansa naapureita, koska A:n lähetysspektori on vastakkainen B:n vastaanottosektorille.

Tätä ongelmaa vastaan Hu ja Evans ehdottavat ylimääräisen todentajalaitteen vahvistusta A:n ja B:n suunnille.

3.2.2 Todentajalaite

Jotta A voi varmistua siitä, ettei B ole madonreiän toisella puolella, tarvitsee A avukseen todentajalaitteen, C, joka pystyy todentamaan B:n naapurikseen. Tämän lisäksi B:n pitää olla kuulua A:lle ja C:lle eri sektoreilla. Katso kuva 4a.

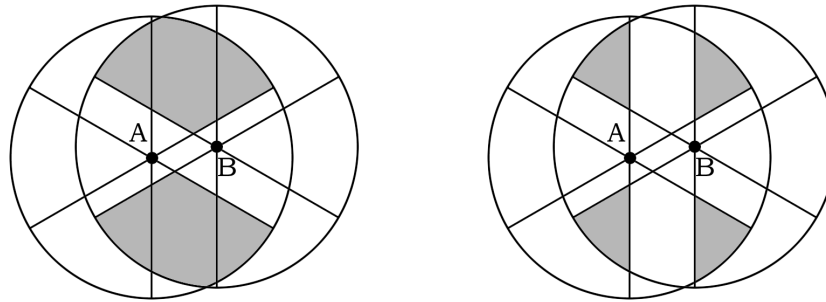


Kuva 4: Todentajalaite normaalitilassa (a) ja ilkivaltainen todentajalaite piilotetun hyökkääjän kanssa (b)

Mikäli B on C:n naapuri ja se on eri sektorilla C:lle kuin A:lle niin A voi luottaa B:n olevan oikeasti vieressä eikä madonreikä-tunnelin toisella puolella.

3.2.3 Worawannotai-hyökkäys ja sen estäminen

Erityistapaus todentaja-aseman hyväksikäytöstä on, jos kuvassa 4b laitteet A ja B eivät oikeasti kuule toisiaan ja C on hyökkääjä ja se saa asetettua A:n ja B:n väliin viestejä välittävän *piilotetun hyökkääjän* D. A tällöin pyytää C:tä todentamaan, että B on tämän naapuri eikä välissä ole madonreikää. Hyökkääjälaitte C vakuuttaa A:n ja B:n olevan naapureitaan ja piilotettu hyökkääjälaitte D näiden välissä pääsee suorittamaan jo aiemmin kuvattuja madonreikähyökkäyksen eri keinoja – esimerkiksi salakuuntelua. Tätä hyökkäystä kutsutaan Worawannotai-hyökkäykseksi keksijänsä mukaan.



(a) Löyhällä paikkavaatimuksella (b) Worawannotai-hyökkäyksen esteenä

Kuva 5: Todentajalaitteen mahdolliset sijainnit

Tätä erityistapausta varten todentajalaitteen sijainnille pitää asettaa tiukemmat rajat siten, ettei todentaja saa olla kuuluvuusalueita yhdistävällä alueella. Todentajalaitteen löysät ja tiukat sijaintirajaukset on kuvattu kuvassa 5.

3.3 SECTOR ja MAD

SECTOR on kokoelma toimia, joilla voidaan todentaa laitteiden tapaaminen, ja MAD on protokolla, jolla voidaan suojautua madonreikähyökkäyksiä vastaan. Nämä molemmat kuvaillaan Capkunin, Buttyanin ja Hubaux'n artikkelissa "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks" [CBH03].

SECTORin oletukset järjestelmän ja verkon suhteen

Käyttöympäristönä voi toimia sekä puhdas mobiilien päätelaitteiden ad hoc-verkko että hybridimalli, jossa on paikoittain paikallaan olevia tukiasemia.

SECTORiin liittyy aikasidonnaisia osia, joten oletetaan, että kaikilla laitteilla on kello. Koska operaatiot eivät ole yhtä tarkkoja kuin edellisessä protokollassa, riittää kellojen synkronointitarkkuudeksi alle sekunnin tarkkuus – vertaa luvun 3.1 aikahihnojen satojen nanosekuntien tarkkuus. Vaikka synkronointi suoritetaan melko pienellä tarkkuudella, oletetaan ajanmittauksen tarkkuudeksi kuitenkin nanosekuntiskaalaa, jotta voidaan laskea kahden tapahtuman välissä kulunutta aikaa tarkasti ja tehdä sen perusteella päätöksiä.

MAD-protokolla olettaa päätelaitteissa olevan lisälaitteen, joka osaa vastata verkon yli tulevaan bittihaasteeseen itsenäisesti ohittaen laitteen prosessorin.

SECTOR olettaa myös, että verkossa on jokin keskitetty auktoriteetti, joka kontrolloi verkkoon liittymistä ja antaa laitteille uniikin identiteetin. Lisäksi oletetaan, että laitteilla on parikohtaiset salaiset avaimet tai toistensa todennetut julkiset avaimet. Tämä avainten jakaminen voidaan saada aikaan esimerkiksi tallentamalla avaimet verkkoa rakentaessa manuaalisesti laitteisiin tai käyttämällä jotakin avainvaihtoprotokollaa.

SECTORin määrittämät roolit ja niiden tehtävät

SECTOR määrittelee kolme eri roolia, joilla on omat tehtävänsä. Hyökkääjien vaikutus verkon toimintaan riippuu siitä, missä roolissa se toimii kulloinkin.

Kohtaaja on laite, joka väittää kohdanneensa jonkin toisen laitteen ennen, jälkeen tai tarkalleen jonakin tietynä ajanhetkenä.

Todistaja on laite, joka voi todistaa joko kohtaamisen ajanhetken tai sijainnin. Ajanhetken se voi todistaa, jos todistaja itse oli kohdattu laite. Sijainnin tietynä ajanhetkenä se voi todistaa, jos todistaja on havainnut laitteen.

Varmentaja on laite, jolle annetaan tiedot kohtaajan väitteestä ja todistajan tarjoama lisäinformaatio ja se pystyy tarkistamaan tietojen yhteensopivuuden ja täten varmentaa kohtaamisen tapahtuneen.

Puhtaissa ad hoc -verkoissa, joissa laitteet ovat yhdenvertaisia, kaikki laitteet voivat toimia kaikissa rooleissa ja täten laitteiden väliset kohtaamiset voidaan joustavasti varmentaa minkä tahansa laitekolmikon kesken. Mikäli käytössä on tukiasemallinen hybrdiverkko – kuten monihyppyinen matkapuhelinverkko – voi varmentajana toimia vain osa verkon laitteista: tukiasemat.

3.3.1 MAD-protokolla

MAD-protokolla, eli molemminpuolinen todennus etäisyysrajauksella (Mutual Authentication with Distance-Bounding, MAD), on hyvin yksinkertainen

rakenteeltaan ja samalla varma tapa todentaa kahden laitteen tapaaminen. Protokollan ydinkomponentit ovat laitekohtainen satunnainen bittisarja sekä laite, joka voi autonomisesti ottaa vastaan yksibittisen haasteen, lähettää takaisin XOR-bittioperaation tuloksen tästä bittihaasteesta ja laitteen satunnaisen bittisarjan seuraavasta bitistä.

Protokollan alustusvaiheessa molemmat laitteet luovat l -bittisen satunnaisen bittijonon sekä l' -bittisen satunnaisen vahvistusbittisarjan. Aloitteen tehnyt laite, A , lähettää tiivisteiden näiden bittisarjojen yhdisteestä ja vastaanottaja, B , lähettää vastaavan tiivisteensä takaisin. Bittisarjojen luonnin voi hoitaa jo ennen laitteiden kohtaamista, jotta laskentavaatimukset protokollalle kohtaamistilanteessa saadaan pienennettyä.

Alustusvaiheen jälkeen laitteet aloittavat tuon l -bittisen jonon läpikäynnin bitti kerrallaan. A lähettää ensimmäisen bittinsä sellaisenaan ja tallentaa lähetysajan kellonsa maksimitarkkuudella talteen. B :n bittihaastelaite laskee XOR-bittioperaation vastaanotetusta bitistä sekä B :n bittijonon seuraavasta bitistä, lähettää sen A :lle sekä tallentaa oman lähetysaikansa. A vastaanottaa B :ltä tulleen vastauksen ja, kuten edellä, laskee XOR-bittioperaation seuraavan bittijononsa bitin kanssa ja lähettää sen. Tämän bittivaihdon lisäksi sekä A että B vertailevat paketin saapumisaikaa edellisen oman pakettinsa lähetysaikaan ja saavat täten sisäisen kellonsa tarkkuudella maksimietäisyyden välimatkalleen, koska paketti ei voi kulkea valonnopeutta nopeammin. Periaate on sama kuin luvun 3.1 aikahihnojen kanssa.

Kun l -bittinen jono on kulutettu loppuun, seuraa protokollan varmennusvaihe, jossa molemmat osapuolet laskevat XOR-bittioperaatioilla alkuperäisen vastapuolen bittijonon ja laskevat siitä tiivisteiden omien tunnistetietojensa kanssa sekä luovuttavat toisilleen alustusvaiheessa lasketun l' -bittisen jonon. Näiden tietojen avulla molemmat laitteet voivat päästä varmuuteen, että paketteja ei ole välissä muokattu.

3.3.2 Kohtaamisen tuoreus- ja aikatakuu

SECTOR sisältää viisi erilaista tapaa todistaa kohtaamisen ajankohtaa. Nämä tavat on luokiteltu kahteen kategoriaan todistettavan ajankohdan tarkkuuden mukaan: kohtaamisen tuoreustakuu ja kohtaamisen aikatakuu. Tuoreustakuu takaa kohtaamisen tapahtuneen ennen tiettyä ajankohtaa, mutta ei todista ajankohtaa tarkasti. Kohtaamisen aikatakuu pystyy todistamaan ajankohdan halutulla tarkkuudella, joka on ennalta määrätty.

Kohtaamisen tuoreustakuu (Guaranteeing Encounter Freshness, GEF)

Tuoreustakuuta varten SECTOR määrittelee kaksi takuutasoa: todistaja-

takuun sekä todistaja–kohtaaja-takuun. Ensimmäisessä jokainen verkon laite on etukäteen luonut satunnaisen luvun V ja ottanut siitä N rekursiivista tiivistettä. N on etukäteen määritelty luku, jonka pitäisi kattaa kaikki tarvittavat todistuskerrat. Julkisenä allekirjoituksenaan laitteet julkaisevat N :nnen rekursiivisen tiivisteen ja paljastavat edellisiä tiivisteitä ketjussa aika- tai pyyntöpohjaisesti. Koska protokolla olettaa tiivistefunktion olevan alkukuvaresistentti, tämä on yksinkertainen tapa todentaa laitteen olevan tiivisteketjun omistaja.

Kohtaaja, A , saa siis tällaisen kohdatun laitteen, B , vielä paljastushetkellä kaikille tuntemattoman edellisen alkukuvan ja pystyy täten todistamaan varmentajalle kohtaamisen tapahtuneen. Tämä kuitenkin vain todistaa, että joku laite verkossa on tavannut B :n. A on voinut vain kaapata tuon tiedon matkalla ja käyttää sitä hyväkseen nyt.

Tätä haastetta varten SECTOR tarjoaa todistaja–kohtaaja-takuun, jossa kaikki verkon laitteet luovat kaikkia muita verkon laitteita varten omat tiivisteketjunsä ja julkaisevat kaikkien tietoon omat juurensä jokaista eri laitetta varten. Täten laite M ei voi käyttää laitteen A saamaa alkukuvavastausta C :ltä, koska C :llä on omat julkiset juuritiivisteet sekä M :lle että A :lle. Tuosta A :n saamasta alkukuvavastauksesta päädytään nimenomaan A :ta varten luotuun juuritiivisteeseen, joka ei ole sama kuin M :lle luotu juuritiiviste. Täten M ei voi hyväksikäyttää salakuunneltua viestiä.

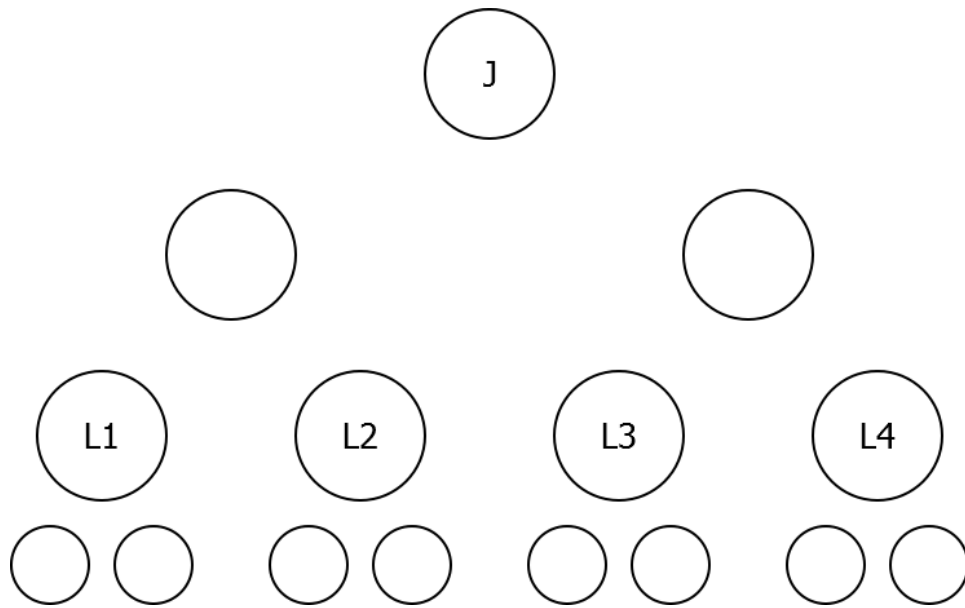
Tällaisen $n * (n - 1)$ tiivistelistan ylläpito ei ole tilavaatimuksiltaan järkevää ottaen huomioon laitteiden rajalliset resurssit. Tämän lisäksi tuoreustakuu takaa vain tapaamisen tapahtuneen ennen tietyn avaimen paljastusta. Tilarajoitteiden takia avaimia ei voi luoda esimerkiksi joka sekunnille, joten etäisyystarkkuus heikkenee.

Kohtaamisen aikatakuu (Guaranteeing the Time of Encounter, GTE)

Koska tuoreustakuun tarkkuus kohtaamisen ajankohdan suhteen ei ole riittävä, SECTOR kuvailee tähän tarkoitukseen paremman ratkaisun: samanlaiset Merkle-puut kuin luvun 3.1.3 TIK-protokollassa. Tässäkin Merkle-puun lehtisolmujen arvo on aikaan sidottu ja sen arvo muodostuu satunnaisluvusta. Kuten tuoreustakuussa on aikatakuussa samat takuutasot: todistaja-takuun sekä todistaja–kohtaaja-takuun. Nämä takuutasot oletuksena mukailevat tuoreustakuun toimintaa eli todistaja-takuuta varten jokaisella laitteella on yksi muodostettu Merkle-puu, jonka juuritiiviste toimii sen julkisena allekirjoituksena, ja todistaja–kohtaaja-takuuta varten jokainen laite luo jokaista verkon muuta laitetta varten oman tiivistepuun.

Aikatakuun tapauksessa SECTOR tarjoaa kuitenkin myös optimoidun tavan tallentaa kaikkien laitteiden tiivistepuut yhdeksi tiivistepuuksi per laite, jotta ei tarvitse jakaa kuin yksi juuritiiviste. Tällöin jokainen laite muodostaa oman tiivistepuunsa, mutta ne ovat isomman tiivistepuun lapsia, katso kuva 6.

Kuten luvun 3.1.3 TIK-protokollassa, varmennus tiivistepuita käyttäen perustuu siihen, että jokaisen varmennuksen yhteydessä puun omistaja luovuttaa kaikki ne tiivistepuun solmut, jotka tarvitaan juuritiivisteen laskuun.



Kuva 6: Kohtaamisen aikatakuuta varten muodostettava optimoitu tiivistepuiden tiivistepuu. L1-4 ovat eri laitteiden tiivistepuiden juuritiivisteet, joista lopulta saadaan tiivistepuiden tiivistepuun juuritiiviste solmuun J

3.3.3 Hyökkääjien ja hyökkäysmallien tyypitys

SECTORissa kuvaillaan kolmen eri roolin sekä kahden eri vaarannustason yhdistelmiä ja miten edellä kuvaillut eri menetöt niistä selviävät. Roolitus menee kuten yllä 3.3 kuvattu ja vaarannustasot ovat seuraavat: laite on *vihamielinen* (malicious) tai se on *kaapattu* (compromised).

Sekä tuoreus- että aikatakuun kanssa niiden yksinkertaisimmat versiot, eli pelkkä todistaja-takuu, ovat haavoittuvaisia tietyille hyökkäysmetodeille, mutta MAD-protokollan käyttö rajoittaa haavoittuvuutta paljon. Jos käytössä on MAD-protokolla ja todistaja-kohtaaja-takuutaso, hyökkäyspinta-ala rajoittuu siihen, että hyökkääjä jakaa saman identiteetin monen eri laitteen välillä. Tällöin hyökkääjä pystyy todistamaan sellaisen kohtaamisen, jota ei

ole oikeasti tapahtunut. Tähän SECTORin tekijät suosittelevat verkon topologiataarkistusta, jolla pystyttäisiin näkemään, mikäli solmu väittää olleensa kahdessa paikkaa lähes yhtä aikaa.

4 Puhtaasti protokollapohjaiset puolustukset

Seuraavilla ratkaisuilla on laajempi käyttöpotentiaali, koska ne eivät vaadi erityislaitteistoa.

4.1 DeWorm

Hayajneh et al kuvailevat DeWorm-protokollan [HKT09], jonka mainitaan erityisesti olevan tehokas *piilotettuja hyökkääjiä* vastaan – eli madonreikähyökkääjät M1 ja M2 siirtävät paketteja keskenään eivätkä itse osallistu verkon laitteina reititykseen.

Perusajatus

Kun laite A haluaa selvittää, onko sen ja laitteen B välissä madonreikä, se aloittaa DeWorm-protokollan mukaisen vaihtoehtoisen reittihaun: jokaiselle solmulle polulla A–B katsotaan, että sitä seuraavasta seuraavaan noodiin vievän reitin pisimmän ja lyhimmän siirtymän pituuksien erotus ei ole raja-arvoa n suurempi.

Esimerkki: A–B väli on kokonaisuudessaan A-1-2-3-4-B. A pyytää kaikilta naapureiltaan löytämään reitin noodiin 2 siten että reitti ei kulje nykyisen reitin kautta eli noodin 1 läpi. Reitti A–2 on kahden hypyn pituinen; jos löytyy yksi tai useampi reitti, jonka pituus on $2+n$ tai enemmän, A olettaa välillä A–2 olevan madonreikä. Mikäli madonreikää ei löydy, seuraavaksi tarkistetaan 1–3 ja taas pyydetään 1:n naapureita löytämään reitti 3:een kulkematta noodin 2 kautta. Tätä jatketaan kunnes erikoistapauksena käsitellään toiseksi viimeinen noodi (4), joka pyytää kaikkia naapureitaan löytämään reitin B:hen kulkematta itsensä läpi. Mikäli missään vaiheessa ei ole löytynyt raja-arvoa n suurempia poikkeamia reittien pituuksissa, voi A olla varma, ettei matkalla ole madonreikää.

Ongelmat

Raja-arvon n valinta on kriittistä, jotta voidaan tasapainotella virheellisten tunnistusten ja tunnistamatta jättämisen välillä. Kirjoittajien simulointien perusteella he suosittelevat n arvoksi 2–3.

Toinen ongelma on niin sanotut kriittiset solmut, eli laitteet jotka legitimi yhdistävät kaksi verkkoa ainoana solmukohtana. Tällainen solmukohta

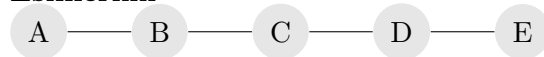
näyttää DeWorm-protokollan silmin madonreiältä ja siksi verkon pitää pysyä ottamaan tällainen huomioon esimerkiksi antamalla kriittiselle solmulle ja sen naapureille erivapauden DeWorm-protokollan ohitukseen. Tällainen erivapaus on mahdollista sallia, koska DeWorm-protokolla on suunniteltu ja testattu liikkumattomiin verkkoihin, joissa verkon muoto on tiedossa etukäteen.

Viimeisimpänä ongelmana on protokollan suuri liikennemäärä: jokainen verkon noodi voi päätyä tarkistamaan madonreikien olemassaolon jokaiseen noodiin, jonka kanssa se on yhteydessä. Kirjoittajat eivät kuitenkaan kuvaile, miten laitteen pitäisi päättää aloittaa DeWorm-protokollatarkistus.

4.2 DelPHI

Hon Sun Chiu ja King-Shan Lui kertovat viiveeseen perustuvasta DelPHI-protokollastaan [CL06]. Koko protokollan ydinajatus on hyvin yksinkertainen: verkon jokaisen hypyn pitäisi olla pituudeltaan suunnilleen yhtä pitkä ja täten madonreian voi löytää havaitsemalla reitin, jonka lähetysaika per hyppyjen määrä on suuri. Normaali reitti verkon poikki etenee tasaisin hyppäyksin, joita voi olla montakin verkon laidalta toiselle, mutta madonreiässä liikutaan pitkä fyysinen matka yhden hypyn aikana. Tällöin tuo yksi hyppy erottuu selkeästi hitaampana ja täten epäilyttävänä.

Esimerkki



Pituus A–E: 4 hyppyä. Aikaa kuluu T_1 .



Pituus A–E: 1 hyppy M1 ja M2 ollessa *piilotettuja hyökkäjiä*. Aikaa kuluu: T_2 .

Koska molemmissa esimerkeissä signaali kulkee maksimissaan valonnopeutta, ei kokonaisajassa ole suurta eroa, eli $T_1 \approx T_2$. Koska ensimmäisen reitin lähetysaika per hyppymäärä on selvästi jälkimmäistä parempi, $\frac{T_1}{4} < \frac{T_2}{1}$, on pidempi reitti DelPHI-protokollan mielestä parempi vaihtoehto.

Protokollan eteneminen

Vaihe I: Laite A aloittaa DelPHI-madonreikä-tunnistuksen lähettämällä DREQ-tietoliikennepaketin B:lle yleislähetystenä. Koska paketti lähtee yleislähetystenä, kaikki A:n naapurit kuulevat paketin ja tallentavat oman tunnis-

teensa paketin *edellinen hyppy* -kenttään, kasvattavat hyppymäärälaskuria ja lähettävät sen yleislähetystenä eteenpäin. Kun DREQ-paketti vihdoin saapuu B:lle, se lähettää vastauksena DREP-tietoliikennepaketin jokaista vastaanottamaansa DREQ-pakettia varten suoraan A:lle. Nämä DREP-paketit palaavat tulemaansa reittiä pitkin ja A saa tietää useita uniikkeja reittejä verkon läpi.

Vaihe II: Saatuaan vastaukset DREQ-pyyntöönsä laite A voi analysoida reittien pituuksien suhteita keskiarvoiseen hypyn viiveeseen. Mikäli näiden eri hyppyviiveiden välinen ero ylittää raja-arvon T , voi A olettaa madonreiän reitille. Kirjoittajien analyysin perusteella madonreikähyökkäyksen yhteydessä nämä hyppyviiveet jakautuvat kahteen ryhmään siten, että kahden ryhmän välillä oleva aikaero on aina suurempi kuin ryhmän sisäinen hajonta.

Esimerkkiarvoina tuolle raja-arvolle T annetaan 1-5 millisekuntia. Arvon ollessa 1 ms tunnistuu osa (15–20%) valideista reiteistä madonrei'iksi ja madonrei'istä tunnistetaan suurin osa (65–97%). Raja-arvolla 5 ms väriiden tunnistusten arvo on marginaalista (1–3%) ja madonrei'istä tunnistetaan simulaatioverkon koosta riippuen noin 10–89%. Simulaatioverkkojen koko vaikutti merkittävästi tunnistustarkkuuteen: raja-arvoilla 1–5 ms tunnistustarkkuus vaihteli pienemmässä verkossa 65–10,4% ja kaksinkertaisen kokoisessa verkossa 97,6–89%.

4.3 LiteWorp

Khalil et al esittelevät naapurilistoihin ja vartiointiin perustuvan LiteWorp-protokollan [KBS05]. Protokollan oletuksena on verkon staattisuus, eli laitteet eivät siis liiku. Tämän lisäksi jokaisella laitteella oletetaan olevan jaettu avain jokaista toista verkon laitetta varten.

Verkkoon liittyminen

A:n liittyessä verkkoon se lähettää yleislähetystenä tiedon olemassaolostaan. Kaikki viestin kuulijat vastaavat A:lle todentaen vastauksensa heidän yhteisellä jaetulla avaimellaan. A lisää kaikki todennetun vastauksen lähettäneet laitteet naapurilistaansa, R_A . Kun A:n yleislähetysten aikaraja on mennyt, se lähettää kaikille naapureilleen oman naapurilistansa todentaen viestinsä jokaiselle erikseen jaetulla avaimella. Täten liittymisen yhteydessä A tietää naapurinsa ja A:n naapurit tietävät kaikki A:n naapurit.

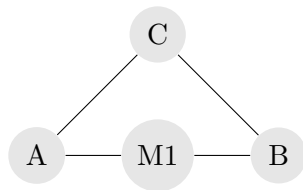
4.3.1 Paikallinen valvonta

Koko protokollan perusajatuksena on, että kaikki laitteet vartioivat kaikkea kuulemaansa liikennettä. Mikäli laite C saa laitteelta B paketin, jonka B

sanoo tulevan A:lta mutta C:n tietojen mukaan A ei ole B:n naapuri, C hylkää paketin ja kohottaa B:n paha-aikeisuuslaskuria Mal_B jonkin ennalta määrätyn *tekauslisäyksen* verran.

Toinen rike, jota verkossa olevat laitteet valvovat, on pakettien tahallinen pudottaminen. Tätä varten jokainen verkon laite pitää jokaista kuulemaansa naapuria varten erillistä vahtilistaa paketeista, jotka naapurin olisi tarkoitus lähettää eteenpäin.

Alla olevan kuvaa esimerkkinä käyttäen M1:n saadessa paketin A:lta B:lle tekee laite C omaan vahtilistaansa merkinnän tästä paketista ja odottaa hetken verran että M1 lähettää paketin normaalisti eteenpäin B:n suuntaan. Mikäli C ei havaitse M1:n lähettäneen sille tullutta välitettäväksi tarkoitettua pakettia eteenpäin, korottaa C nyt Mal_{M1} -paha-aikeisuuslaskuria *pudotuslisäyksen* verran.



4.3.2 Eristäminen

Kun C on havainnut raja-arvon, C_t , verran rikkeitä M1:n toiminnassa, se poistaa M1:n omalta naapurilistaltaan ja lähettää kaikille M1:n naapureille tiedon siitä, että epäilee M1:tä paha-aikiseksi laitteeksi. Kaikki M1:n naapurit tarkistavat, että C on ollut M1:n vartija, ja merkkäavat C:n omaan varoituspuskuriinsa M1:n suhteen. Laitteen saadessa luottamusrajan, γ , ylittävän määrän varoituksia M1:stä se poistaa M1:n omalta naapurilistaltaan. Tämän luottamusrajan asettaminen liian alhaiseksi voi luoda tilanteen, jossa verkon normaalit laitteet voivat joutua väärin leimatuksi ja verkosta hylätyksi. Liian korkea luottamusraja voi johtaa paha-aikeisten laitteiden huomaamatta jäämiseen. Varoitukset ovat kumulatiivisia eikä niissä ole aikarajaa.

Analyysia toimivuudesta

Yksi tämän protokollan toimintavarmuuteen vaikuttavista tekijöistä on verkoton pakettitörmäykset. Törmäyksistä voi aiheutua se, että vartijalaite ei havaitse vartioimalleen laitteelle tullutta välitettävää pakettia ja laitteen vartioitavan lähetettäessä tämän paketin normaalisti eteenpäin vartijalaite katsoo tämän tekaisseen paketin. Vaihtoehtoisesti pakettitörmäys voi peittää vartijalta näkyvyyden paketin oikeaan edelleenvälitykseen ja täten viattoman laitteen paha-aikeisuuslaskuria kohotetaan *pudotuslisäyksellä* vaikka se

oikeasti lähetti paketin.

Kirjoittajien analyysin perusteella todennäköisyys madonreiän tunnistamiselle verkossa riippuu naapureiden määrästä. Alle 7 naapurin verkoissa protokolla ei toimi, mutta 7 naapurilla todennäköisyys madonreiän tunnistamiseen on noin 90% ja se kohoaa 100%:iin jo 11 naapurilla. Yhteentörmäysten takia protokollan teho alkaa heikentyä liian täydessä verkossa: yli 20 naapurin verkoissa tunnistuksen todennäköisyys putoaa lähes lineaarisesti siten että 35 naapurin verkossa todennäköisyys on enää noin 40%.

Vaikka väärin tunnistusten mahdollisuus on jo tuotu esille, ei se todellisuudessa kirjoittajien mukaan ole kovin suuri uhka. Pahimmillaan mahdollisuus legitiimin laitteen väärään tunnistukseen paha-aikaisena on 23 naapurin verkossa, jolloin todennäköisyys on 0,00003%.

4.4 MobiWorp

Khalil et al jatkavat LiteWorpin ideaa staattisista verkoista verkkoihin, joissa laitteet voivat liikkua, MobiWorp-protokollallaan [KBS06].

Perusidea vartijoista ja naapurilistoista yhä sama, nyt lisänä laitteiden liikkumiseen liittyvät protokollaosat. He esittelevät kaksi uutta protokollaa: SMP-protokollan (Selfish move protocol) sekä CAP-CV-protokollan (Connectivity Aided Protocol with Constant Velocity). Näiden protokollien lisäksi yksi uusi ja perustavanlaatuinen ero LiteWorpiin on vaatimus *keskitetystä auktoriteetista* (Central Authority, CA), joka pystyy valvomaan ilkeästi toimivia laitteita koko niiden elinkaaren ajan. Tällöin ilkeävaltainen laite ei voi paeta vastuuta liikkumalla uuteen verkon osaan.

4.4.1 SMP-protokolla

Jokainen laite, joka haluaa toimia verkon osana, hakee CA:lta valtuutettua lupaa liittyä verkkoon tiettyssä sijainnissa todennetulla naapuripäivitys- eli ANUM-pyynnöllä (Authentication Neighbor Update Message). CA tarkistaa tietokannastaan pyytävän laitteen nykyisten voimassaolevien tai evättyjen lupien tilan ja voi joko myöntää ANUM-luvan, lopettaa pyynnön käsittelyn tai lähettää vastauksena ANUM-hylkäyksen. Mikäli CA lähettää hylkäyksen, kaikki hylkäyspaketin lähetyksen kuulevat laitteet lisäävät hylätyn laitteen omalle estolistalleen ja tämä laite ei koskaan voi enää päästä verkkoon takaisin.

Laitteen siirtyessä paikasta toiseen se menettää oikeutensa toimia verkon välityssolmuna. Se voi yhä vastaanottaa sille tarkoitettuja paketteja ja lähettää muille viestejä, mutta pakettien välitys ei ole sallittua. Kun laite pysähtyy

uuteen paikkaan, se voi lähettää CA:lle ANUM-pyynnön uudella sijaintitiedolla ja luvan saatuaan se voi integroitua verkkoon siinä sijainnissa lähettämällä yleislähetyksenä naapureilleen tiedon itsestään, ANUM-luvastaan sekä omasta estolistastaan. Naapurit tarkistavat CA:n salaisella avaimella allekirjoitetun ANUM-luvan aitouden. Sen ollessa aito lisäävät uuden naapurin naapurilistaansa ja tallentavat sen estolistan talteen.

Kun laite siirtyy ANUM-lupansa sijainnista pois, sillä on *armonaikaa* (grace period) t_{grace} verran, jonka jälkeen se ei voi lähettää verkkoon enää mitään muita viestejä kuin CA:lle suunnattuja ANUM-pyyntöjä. Jos laitteen siirtymä pisteestä A pisteeseen B kestää kauemmin kuin t_{grace} sen pitää välissä pyytää uusi ANUM-lupa, koska muuten se ei voi edes vastaanottaa sille tarkoitettuja paketteja.

4.4.2 CAP-CV-protokolla

SMP-protokolla ei sovi hyvin verkkoihin, joissa laitteiden yhdenaikainen liikkuvuus on suurta, koska näissä tilanteissa isotkin verkon osat voivat koostua liikkeessä olevista laitteista, joilla ei ole oikeus välittää paketteja toisilleen. Toinen ongelma SMP-protokollassa on se, ettei liikkuva laite voi t_{grace} jälkeen viestiä enää mitenkään vaikka sille olisi tarve. Näitä ongelmia varten CAP-CV-protokolla kehitettiin. Se sallii laitteen pyytää CA:lta siirtymälupaa paikasta A paikkaan B ja kertoa CA:lle samalla sen arvioitu keskisiirtymänopeus. Mikäli laite noudattaa tätä CAP-CV-protokollan ANUM-luvan reittiä ja arvioitua nopeutta se voi viestiä verkkoon koko liikkeen ajan.

5 Yhteenveto

Madonreikähyökkäys on erittäin vakava uhka langattomissa ad hoc -verkoissa. Siitä tekee erityisen kavalan se, ettei hyökkääjien tarvitse murtaa minkäänlaista salausta vaan ne voivat saada haittaa aikaan puhtaasti verkkokerroksella paketteja kopioiden verkon osien välillä. Tästä seuraa se, että suuri osa puolustuskeinoista perustuu arviointiin toimivatko muut laitteet ja verkkopolut joidenkin ennalta asetettujen raja-arvojen puitteissa. Nämä raja-arvot päättävät sen löytyykö verkosta kaikkien hyökkääjälaitteiden lisäksi myös viattomia laitteita väärin tunnistettuina vai jääkö osa hyökkääjistä tunnistamatta.

Puolustuskeinoja omaan ad hoc -verkkoon valitessa pitää arvioida onko lisälaitteiden tuomat edut, esimerkiksi raskaan laskennan siirtäminen CPU:lta siihen erikoistuneelle lisäosalle, niiden lisäkustannuksen arvoisia. On myös puhtaasti laitteistoa vaativia etuja, joita ei voisi edes ohjelmistolla suorittaa: suunta-antennin tuomat suuntimatiedot eivät ilman siihen soveltuvaa lisälaitetta onnistu.

Lähteet

- [CBH03] Srdjan Capkun, Levente Buttyán ja Jean Pierre Hubaux: *SEC-TOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks*. Teoksessa *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN '03, sivut 21–32, New York, NY, USA, 2003. ACM, ISBN 1-58113-783-4. <http://doi.acm.org/10.1145/986858.986862>.
- [CL06] Hon Sun Chiu ja King Shan Lui: *DelPHI: wormhole detection mechanism for ad hoc wireless networks*. Teoksessa *Wireless Pervasive Computing, 2006 1st International Symposium on*, sivut 6 pp.–, Jan 2006.
- [HE04] Lingxuan Hu ja David Evans: *Using Directional Antennas to Prevent Wormhole Attacks*. Teoksessa *The 11th Annual Network and Distributed System Security Symposium, 2004. NDSS 2004. Proceedings.*, February 2004.
- [HKT09] T. Hayajneh, P. Krishnamurthy ja D. Tipper: *DeWorm: A Simple Protocol to Detect Wormhole Attacks in Wireless Ad Hoc Networks*. Teoksessa *Network and System Security, 2009. NSS '09. Third International Conference on*, sivut 73–80, Oct 2009.
- [HPJ03] Yih Chun Hu, A. Perrig ja D.B. Johnson: *Packet leashes: a defense against wormhole attacks in wireless networks*. Teoksessa *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, nide 3, sivut 1976–1986 vol.3, March 2003.
- [KBS05] I. Khalil, S. Bagchi ja N.B. Shroff: *LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks*. Teoksessa *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, sivut 612–621, June 2005.
- [KBS06] I. Khalil, S. Bagchi ja N.B. Shroff: *MOBIWORP: Mitigation of the Wormhole Attack in Mobile Multihop Wireless Networks*. Teoksessa *Securecomm and Workshops, 2006*, sivut 1–12, Aug 2006.
- [Tsu14] Tsuruya: *Wikipedia: A Merkle Tree with 8 leafs and authentication path*, Apr 2014. <http://en.wikipedia.org/wiki/File:MerkleTree2.svg>.