

Madonreiät langattomissa ad hoc -verkoissa

Jan Wikholm

Kandidaatintutkielman aineversio
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 4. maaliskuuta 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	
Kandidaatintutkielman aineversio	4. maaliskuuta 2014	11	
Tiivistelmä — Referat — Abstract			
Madonreikä-hyökkäysten ja niiden vastatoimien tyypitys.			
Avainsanat — Nyckelord — Keywords			
ad hoc -verkot, wlan, hyökkäys, puolustus, havainnointi			
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	3
3.1	Pakettihihnat ja TIK-protokolla	3
3.2	Suunta-antenni	6
4	Puhtaasti protokollapohjaiset puolustukset	8
4.1	DeWorm	8
4.2	DelPHI	9
4.3	LiteWorp	10
	Lähteet	11

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 käyttö-tarkoituksissa kuten pelastus-, armeija- [HPJ03] sekä myös siviilikäytössä [KBS05]. Esimerkiksi luonnonkatastrofin jäljiltä perinteiset langattomat tukiasemat voivat olla tuhoutuneet ja täten pelastuslaitosten työntekijät ovat viestinnässään ad hoc -verkkojen varassa [HPJ03].

Näiden verkkojen suurimpia etuja ovat käyttöönoton nopeus ja kustannus-tehokkuus [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 teknologian alkuvaiheen tutkimuksesta on keskittynyt näiden lupautusten toteuttamiseen luoden 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, *musta aukko -hyökkäys* (blackhole attack), tai syöttää niitä verkkoon paljon tukehduttaakseen sen järkevän käytön, *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 edellä mainittuja hyökkäyksiä. Nämä tahot välittävät kaikki kuulemansa paketit toiselle osapuolelle, joka toistaa ne omassa päässään. Tämä pakettien välitys voidaan toteuttaa dedikoidulla suurinopeuksisella linkillä, 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. 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.

Esittelemme luvussa 2 madonreikähyökkäysten hyökkääjä- [CL06] ja hyökkäystyyppit [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.

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 toistimena toimiva madonreikä.

Esimerkiksi pakettihihnat 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 lähetyskaistan - esim. wlan-verkko - lisäksi käytössä vaikkapa yksinkertaisesti kytketty ethernetverkko, voivat ne kommunikoida yleistä verkkoa nopeammin ja sen kantamaa pidemmälle. Tämä sotkee reititysprotokollia ja suuri osa paketeista päätyy kulkemaan hyökkääjien linkin läpi.

Pakettikapseloinnissa hyökkääjät H1 ja H2 voivat käyttää jo olemassa olevaa verkkoa: H1 kuulee paketin ja luo uuden H2:lle suunnatun paketin, jonka sisältönä on sellaisenaan H1:n kaappaama paketti. Kun tämä kapseloitu paketti saavuttaa H2:n se toistaa alkuperäisen sisällön sellaisenaan verkkoon, joten sen naapurit luulevat H1:n naapureiden olevan lähellä. Tämä ei vaadi hyökkääjien osalta mitään muista verkon laitteista poikkeavia resursseja.

Suurteholähetys on malliesimerkki siitä, että hyökkääjille voi olettaa loputtomat resurssit toisin kuin muille verkon laitteille, joita nimenomaan yleensä yhdistää resurssien vähyys. Tässä hyökkäystyypissä hyökkääjälaitte lähettää kaappaamansa reitityspaketit huomattavan suurella lähetysteholla

ja täten saa aikaan sen itsensä lävitse kulkevan lyhimmän reitin. Tämä hyökkäys ei siis vaadi kahta osapuolta.

Pakettivälitys – kuten suurteholähetys – ei vaadi hyökkääjäparia vaan sen voi suorittaa yksikin vihamielinen laite. Tässä hyökkääjä on kahden laitteen välissä välittäen paketteja jolloin nämä laitteet luulevat olevansa naapureita ja hyökkääjä niiden välissä voi suorittaa esimerkiksi salakuuntelua tai palvelunestohyökkäystä.

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 junaila toimensa siten, että suuri osa verkon liikenteestä päätyy reitittymään sen läpi.

3 Laitteistoriippuvaiset puolustusmekanismit

Nämä puolustusmekanismit eivät vaadi reititysprotokoliin 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 dataa, jolla paketin enimmäiskantamaa voidaan rajoittaa.

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

Aikahihnat

Aikahihnojen edellytyksenä on tarkka kellojen synkronointitarkkuus: muutaman 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.

Aikahihnaa muodostaessa lähettäjä päättää enimmäispituuden lähetykselle

ja laskee kauanko valonnopeudella kulkevalla radiosignaalilla kestää sinne päätyä – tässä tietenkin otetaan huomioon synkronoinnin virhemarginaali – ja tämän avulla lähettäjä laskee paketille vanhenemisajan, jonka jälkeen paketti on hylättävä. Tunnelointi aiheuttaa pakosti viivettä, koska se kulkee kauemmaksi, joten madonreiän toisella puolella olevat laitteet pudottavat vanhentuneet paketit.

Vaikka aikahihna on hihnatyypeistä tarkempi on sen haasteena se, ettei lähettäjä tiedä aina tarkalleen omaa lähetyssaikaansa, koska fyysisen kerroksen lähetysmekanismi voi joutua odottamaan; täten on vaikeaa etukäteen luoda lähetyssajankohtaan perustuvaa digitaalista allekirjoitusta.

Geohihnat

Geohihnoja käytettäessä kaikkien verkon laitteiden pitää tietää oma sijaintitietonsa sekä niiden pitää pystyä synkronoimaan kellonsa muiden kanssa; joskin 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-tunneloitu paketti voidaan tunnistaa ilkeävaltaisen laitteen lähettämäksi, jos se lähettää verkkoon kaksi pakettia joiden lokaatiotiedoissa on tapahtunut maksimiliikenopeutta nopeampaa liikettä edellyttävä muutos.

TIK-protokolla

Tärkeimpänä ominaisuutena koko aikahihnafunktion toiminnassa on aika-leimojen luotettavuus; leimat pitää pystyä todentamaan. Todentamisen toteuttamiseen kirjoittajat hylkäävät jaetut avaimet suoraan todeten niiden hallinnan olevan liian raskas operaatio. 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 (Merkle-tiivistepuu). Tiivistysten hyväksi puoleksi kerrotaan niiden erittäin

tehokas laskeminen ja yksisuuntaisuus. Merkle-puussa jokainen tiiviste muodostuu kahden lapsensa tiivisteistä aina juureen saakka. Tämä juuritiiviste on laitteen julkinen avain ja alimman tason lapset ovat yksityinen avain.

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

Vastaanottaja tietää jo etukäteen lähettäjän julkisen avaimen, eli Merkle-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 juuritiiviste, sekä
4. avaimen K_i .

Nämä osat tulevat nimenomaan 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 katsoo, onko paketti voimassa.

Resurssivaativuus

Resurssien niukkuuteen kirjoittajat mainitsevat päätarkaisuna symmetrisen kryptografian käyttö asymmetrisen sijaan – operaatioiden nopeuserot voivat olla kolmesta neljään suuruusluokkaa (100–1000-kertaisia) hitaampia jälkimmäisessä.

TIK-protokollaa arvioidessaan kirjoittajat tarkastelivat vuoden 2001 mobiililaitteiden laskentatehoa ja muistimäärää. He toteavat protokollan laskenta- ja muistivaatimukset mahdollisiksi saatavilla olevilla laitteilla; joskin huomauttavat ettei TIK sovi resursseiltaan aivan kaikkein köyhimpiin laitteisiin, joista esimerkiksi sensoriverkot muodostuvat.

Allekirjoitusta varten muodostettavan tiivistepuun muodostusta ja tallennusta voi optimoida siten että vain osa puusta säilötään muistissa ja osa lasketaan tarvittaessa. Tällaisen optimoinnin takia yhden vuorokauden yksityiset avaimet sisältävä osittainen puu saadaan mahtumaan 2,5 megatavuun kokonaisen puun 170 gigatavun sijaan.

3.2 Suunta-antenni

Lingxuan Hu ja David Evans kuvaavat suunta-antennin käyttöä madonreikien 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 lähetyssektoreista 360 asteen kattavuuden. Näitä antenneja sitten kytketään päälle sitä mukaan kun halutaan tiettyyn suuntaan lähettää; suuntauksessa käytetään laitteen sisäistä kompassia siten, että sektori numero yksi osoittaa aina itään.

Protokolla perustuu naapurilistoihin ja siihen että verkon laitteet voivat luottaa ilmoitetun naapurin olevan siellä suunnalla, mitä se väittää. Protokolla olettaa, että verkossa on mekanismi, jolla laitteet voivat muodostaa turvallisen linkin keskenään ja että kriittiset viestit ovat kryptattu.

Naapurien löytäminen

Kun laite A haluaa liittyä verkkoon, se käy läpi jokaisen suunta-antenninsa sektoreista ja lähettää siihen suuntaan yleisen liittymisviestin, joka sisältää A:n oman tunnisteen sekä tiedon ko. sektorin numerosta – kaikki laitteethan ovat suunnattu samansuuntaisesti. Kun laite B vastaanottaa A:n viestin omalla sektorillaan X se ensimmäisenä tarkistaa, että A:n ilmoitettu sektori on X:stä vastakkainen pudottaen paketin muussa tapauksessa.

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

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

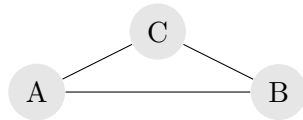
$$A - > M1 - - - - - > M2 - > B$$

Missä 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 lähetyssktori on vastakkainen B:n vastaanottosektorille.

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

Todentajalaite

Todentaja laitteen, C , rooli on toimia todistajana A:lle B:n suhteen – ja päinvastoin – että B kuuluu C:lle ja että se kuuluu eli sektorilta kuin A kuulee B:n. Esimerkki:

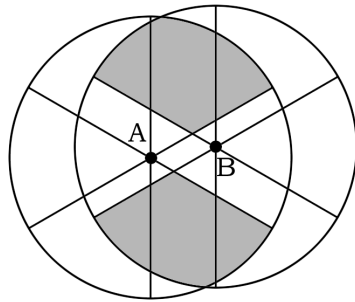


Mikäli B on C:n naapuri ja se on eri sektorilla C:lle kuin A:lle niin A voi luottaa B:n olevan oikeasti vieressä eikä madonreikä-tunnelin toisella puolella.

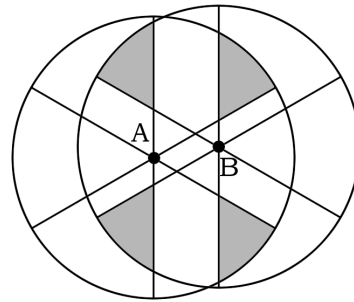
Worawannotai-hyökkäys ja tiukka naapurin löytäminen

Erityistapaus todentaja-aseman hyväksikäytöstä on, jos yllä olevassa kuvassa 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*. 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 näiden välissä pääsee suorittamaan jo aiemmin kuvattuja madonreikähyökkäyksen eri keinoja - esimerkiksi salakuuntelua.

Tätä erityistapausta varten todentajalaitteen positiolle pitää asettaa tiukemmat rajat siten ettei todentaja saa olla kuuluvuusalueita yhdistävällä alueella. Tämä koko asia selvenee kirjoittajien kuvilla:



(a) Löyhällä paikkavaatimuksella



(b) Worawannotai-hyökkäyksen esteenä

Kuva 1: Todentajalaitteen mahdolliset sijainnit

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 vaihtoehdoisen reittihaun: jokaiselle noodille polulla A–B katsotaan, että sitä seuraavasta seuraavaan noodiin vievän reitin pisimmän ja lyhyimmän 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 poikkeampia reittien pituuksissa, voi A olla varma ettei matkalla ole madonreikää.

Ongelmat

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

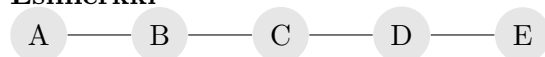
Toinen ongelma on niin sanotut kriittiset solmut, eli laitteet jotka legitiimisti yhdistävät kaksi verkkoa ainoana solmukohtana. Tällainen solmukohta näyttää DeWorm-protokollan silmin madonreiältä ja siksi verkon pitää pystyä ottamaan tällainen huomioon esimerkiksi antamalla kriittiselle solmulle ja sen naapureille erivapauden DeWorm-protokollan ohitukseen. Tällainen erivapaus on mahdollinen 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 madonreiän voi löytää havaitsemalla reitin, jonka lähetysaika per hyppyjen määrä on suuri. Normaali reitti verkon poikki etenee tasaisin hyppäyksin, joita voi olla melko 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: 1 hyppy jos M1 ja M2 ovat *piilotettuja hyökkäjiä*. Aikaa kuluu: T_2 .

Koska molemmissa esimerkeissä signaali kulkee maksimissaan valonnopeutta ei kokonaisaika todennäköisesti ole kovin paljon erilainen, eli $T_1 \approx T_2$ ja täten $(T_1/4) < (T_2/1)$ vaikka tämä reitti olisi pituutensa puolesta muuten parempi.

Protokollan eteneminen

1. Broadcast “reitti A–B” kaikille naapureille rekursiivisesti
2. Päivitetään “edellinen noodi” ja “pituus” kenttiä paketissa kun sitä reititetään eteenpäin
3. B lähettää kaikki DelPHI-polunetsintäpaketit suoraan unicastina takaisin lähettäjälle
4. A saa vastaukset ja analysoi pituudet vs aika/hypyt

5. Jos kaksi aika/hypyt arvoa eroaa toisistaan raja-arvon T verran \Rightarrow madonreikä

4.3 LiteWorp

Khalil et al esittelevät naapurilistoihin ja vartiointiin perustuvan LiteWorp-protokollan [KBS05]

- vartijat
- väärät syytökset vs verkkopakettien törmäilyt
- vahtilistojen ja -puskurien tilavaativuudet
- protokollan heikkeneminen tiheissä verkoissa (naapuri-lkm >20)

Lähteet

- [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: *LITEWOP: 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.