TAMPERE UNIVERSITY OF TECHNOLOGY

**Riku Itäpuro**

# Luottamuksen etsintä ja varmenteiden julkaisu DNSSECillä tai ilman

Kandidaatintyö

Tarkastaja: Marko Helenius
Jätetty tarkastettavaksi 8.12.2013

# TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
**RIKU ITÄPURO: Luottamuksen etsintä ja varmenteiden julkaisu DNSSECillä tai ilman**
Kandidaatintyö, 31 sivua
Joulukuu 2013
Pääaine: Tietoliikenne
Tarkastaja: Marko Helenius
Avainsanat: DANE, DNS, DNSSEC, PKI, luottamusankkuri

Julkisen avaimen järjestelmä tarvitsee hakemiston varmenteille ja tavan kiinnittää luottamus varmenteisiin. Varmenne sitoo julkisen avaimen nimeen, mutta nimen yhteys varsinaiseen identiteettiin ei ole taattu. Nykyisin luottamuksen varmenteelle tuo ennakkoon luotettu varmentaja. Varmenne esitetään yhteyden alussa ja vastapuoli tarkistaa luottamusketjun varmentajalle asti. Toisaalta jo nyt palvelee DNS nimien ja IP-osoitteiden uniikkina ja globaalina hakemistona. Sen turvalaajennus DNSSEC estää hakemistotietojen muuttumisen ja varmentaa lähteen oikeellisuuden. Tutkimustyö esittelee nykyisiä luottamusmalleja, niiden puutteita ja sitä, miten nimipalvelu voi jakaa julkisia avaimia eli varmenteita. Työ rajaa ulkopuolelle DNS:n, DNSSECin, salausmenetelmien, algoritmien ja PKI:n tarkan toiminnan.

# ALKUSANAT

Tämä kandidaatintyö on Tampereen teknillisen yliopiston opinnäyte, joka on tehty syksyn 2013 aikana. Tutkimuksen vastuulliseksi tekijäksi merkitään yksi henkilö, vaikka työssä apuna on ollut moni muu. Et al. joukkoon laittaisin, jos voisin, kandidaattiseminaariryhmäni Jaakko Ylisen ja Katariina Kannuksen, jotka auttoivat näkemään työtäni ulkopuolisen silmin. Joona Kannisto samanhenkisenä asiantuntijana toi sisäpuolista näkemystä ja ohjaajani Marko Helenius antoi tarkkoja vinkkejä rakenteelle. Kiitän heitä avusta.

Suurin apu kuitenkin tuli muusaltani, Päiviltä. Olet rakas.

Tamperella, 8.12.2013
Riku Itäpuro

# SISÄLLYS

# TERMIT JA NIIDEN MÄÄRITELMÄT

amplification factor
: DoS hyökkäyksen voimakkuus. Vastauksen koko jaettuna pyynnön koolla.

anycast
: Lähetystekniikka, jossa vastaanottaja valitaan sijainnin perusteella lähimmästä samannimisisistä osoitteista.

autentikointi
: Osapuolen tunnistus.

autoritäärinen
: Autoritääriseltä DNS-palvelimelta saadaan alkuperäiset, ajantasaisimmat DNS-tiedot. Tietojen ensisijainen päivityspaikka.

base32
: Merkkikoodaustapa, jossa bitit ryhmitellään viiden ryhmiin ($2^5 = 32$), aakkostona käytetään merkkejä A–Z ja 2–7. Sekaannuksen välttämiseksi numeroita 0 ja 1 ei käytetä.

CERT
: Computer Emergency Response Team. Keskus, jonka tehtävinä ovat tietoturvauhkista tiedottaminen, tietoturvaloukkausten ennaltaehkäisy, havainnointi ja ratkaisu. Myös DNS:n RR, joka ei ole yleistynyt.

CPS
: Certificate Policy Stamement. Mihin tarkoituksiin, kuten salaamiseen, mutta ei autentikointiin, varmennetta voi käyttää ja mikä on sen luojan hallinnointitapa varmenteille.

DANE
: DNS based Authentication of Named Entities, nimipalvelun avulla tapahtuva yksikön autentikointi. DANE on sekä RFC-luonnos, että IETF:n työryhmä, joka kehittää DANE-protokollaa ja sitä hyödyntäviä työkaluja.

DNS
: Domain Name System, nimipalvelu, tietokanta, joka yhdistää nimiosaan resurssityypin RR ja tieto-osan.

DNSSEC
: DNS:n laajennus, joka lisää eheyden ja todennuksen DNS RR:iin PKI:llä.

DNSSECbis
: DNSSECin nykyinen versio, joka sisältää DS RR:n.

DoS
: Denial of Service, myös palvelunestohyökkäys. Palvelun lamauttaminen ylikuormittamalla se turhilla pyynnöillä.

DS
: Delegation Signer. DNSSECbis:n käyttämä tietue alemman tason vyöhykkeen sitomiseen ylempään tasoon.

EDNS — Extension for DNS. DNS-parametrien kasvatusmahdollisuus, mikä on tärkeä DNSSECille.

fingerprint — Katso sormenjälki.

IANA — Internet Assigned Numbers Authority. Koordinoi DNS-juurien, IP-osoitteiston ja internetprotokollien numerointia.

IETF — Internet Engineering Task Force. Yhteisö, joka tuottaa dokumentteja, joilla internetin toimintaa parannetaan.

Internet-Draft — Myös I-D. Yleensä RFC-dokumenttiin tähtäävä luonnos erilaisista internetin standardeista ja protokollista. Tekijänä IETF (Internet Engineering Task Force).

julkinen avain — Salaisen avaimen vastinpari PKI:ssä. Käytetään tarkistamaan salaisella avaimella allekirjoitettu data tai salaamaan dataa salaisen avaimen purettavaksi.

juurinimipalvelin — DNS:n ympäri maailmaa hajautettu palvelin, joka jakaa palvelupyyntökuormaa DNS-hierarkian ylimmällä tasolla. Sisältää tietoja TLD-nimipalvelimista. Kutsutaan ”.”-palvelimeksi.

luottamusankkuri — Riittävän luotettavana pidetty julkinen avain, johon etsiminen lopetetaan, kun luottamusketjuja tarkistetaan hierarkiassa [1].

luottamusverkosto — Verkosto, joka on sopinut yhteisen politiikan luottamusparametreistä.

MIME — Multipurpose Internet Mail Extensions. Laajennos, jolla sähköpostissa voidaan välittää liitetiedostoja ja käyttää muita koodaustapoja kuin ASCIIta.

MITM — Man-In-The-Middle, myös välimies. Hyökkäys, jossa yhteyksien välillä oleva kolmas osapuoli teeskentelee kummallekin osapuolelle olevansa vastapuoli välittäen näin haluamansa viestit mahdollisesti muokattuna.

nimiosa — DNS-tietokannassa avaintieto, jonka mukaan tietoja haetaan.

PAM — Pluggable Authentication Module. Ohjelmakirjasto autentikointiin, jotta ohjelman ei tarvitse toteuttaa sitä itse.

| | |
|---|---|
| PKI | Public Key Infrastructure, julkisen avaimen infrastruktuuri. Julkisen ja salaisen avaimen hyödyntäminen yhteyksien sähköisessä salauksessa ja allekirjoittamisessa. |
| PKIX | X.509-varmenteita käyttävä PKI. Myös IETF:n työryhmä. |
| RFC | Request for Comments. ks. RFC5741 [2], joka määrittelee RFC prosesseja. |
| Resource Record | DNS:n tietuetyyppi, esimerkiksi A, SRV, NS, CNAME, TLSA, SRV, SSHFP. |
| RR | Katso Resource Record. |
| SAML | Security Assertion Markup Language. XML-dokumenttimuoto, joka on suunniteltu autentikointi- ja valtuutustietojen välitykseen. |
| SMTP | Simple Mail Transfer Protocol. Postipalvelimien välinen tiedonsiirtoprotokolla. |
| S/MIME | Secure MIME. MIME-datan salaus ja allekirjoitus PKI:llä. |
| sormenjälki | Myös tiiviste. Tiivistefunktiolla laskettu muoto datasta. |
| SSH | Secure Shell-protokolla. PKI:tä käyttävä salattu etäyhteys. |
| SSH-avain | SSH-protokollassa käytettävä avain. Tässä tarkoittaa julkista puolta. |
| SSHFP | DNS RR. Sisältää SSH:n julkisen avaimen sormenjäljen |
| tiivistefunktio | Yksisuuntainen algoritmi, joka tiivistää datasta piirteitä. |
| TOFU | Trust On First Use. Luottamusmalli, jossa varmistetaan luottamus käyttäjältä ensimmäistä kertaa yhteyttä luotaessa. |
| TLD | Top level domain. Täydellisen DNS-nimen kaikkein oikeanpuolimmainen osa, kuten maatunnukset (fi, se, uk) sekä org, info, com ja net. |
| TLS | Transport Layer Security. Tietoliikenteen turvaamisen käytetty protokolla, joka salaa päästä-päähän yhteyden ja autentikoi kohteet. |
| TLSA | DANEssa määritelty RR, joka rajoittaa TLS-liikenteen käytettäväksi kelpaavaa varmennetta. |

varmenne TLS-varmenne, muotona X.509. Yleisesti se on julkinen avain, joka sidotaan identiteettiin allekirjoituksella.

varmentaminen Allekirjoituksen liittäminen tietorakenteeseen, joka sisältää julkisen avaimen ja entiteetin muita tietoja.

vyöhyke DNS:n hierarkiataso, joka sisältää kaikki saman tason RR:t. Vyöhyke voidaan jakaa alivyöhykkeisiin.

X.509 X.500-hakemistojärjestelmän määritys varmenteen muodoksi.

Yksisuuntainen funktio

Funktio, jolle on laskennallisesti vaikea keksiä käänteisfunktio.

# 1. JOHDANTO

Tietoliikenne käyttää varmenteita salaukseen, tunnistukseen ja eheyteen. Liikenne olisi turvallista, jos osapuolille saataisiin jaettua oikeat varmenteet. Burrows, Abadi ja Needham tutkivat 1989 formaalilla todistusmenetelmällä varmenteiden tuomaa turvaa autentikoinnissa [3, s. 248], mutta olettivat, että varmenteet ovat etukäteen tunnettuja. Tämä tutkimus kyseenalaistaa sen nykyisen mallin, joka luottaa varmentajan olevan aina hyväntahtoinen ja toimivan aina oikein jakaessaan varmenteita. Tutkimus myös pohtii, miten varmenteiden jako toimii DNS:n avulla.

Nimipalvelu DNS on internetin tärkeimpiä protokollia, sillä vaikka ohjelmat ja laitteistot toimisivat pelkillä IP-osoitteilla, kääntyy nimi melkein aina kuitenkin IP-osoitteeseen [4, s. 1]. Ihmiset eivät pysty muistamaan pitkiä numerosarjoja, mutta toisaalta yksi keskitetty hakemistopalvelu ei pystyisi vastaamaan kaikkiin kyselyihin, sillä IP-osoitteita on liian paljon yhdessä paikassa säilytettäväksi ja hallinnon vyöhykehajauttaminen on osoittautunut hyvin skaalautuvaksi. DNS on altis tiedon muuntelulle ja virheille. Sen eheyden ja autenttisuuden tarkistukseen on kehitetty julkisen avaimen menetelmää käyttävä DNSSEC, jonka käyttö on vielä vähäistä. Tämä kandidaatintyö käyttää DNS:ää varmenteiden ja niiden rajoitteiden jakeluun sekä DNSSECiä suojaamaan tiedon eheys ja autenttisuus.

DNS on hakemistopalvelu ja sen vyöhykkeitä hallinnoi yksiköt, jotka julkaisevat vyöhykkeissä omien palvelimiensa tietoja ja voivat edelleen delegoida omaa vyöhykettään pienempiin alivyöhykkeisiin. Yksiköiden vastuulla on usein myös tietää, mitä varmenteita sen palvelimissa on käytettävä. Varmenteiden lisääminen DNS:ään yksinkertaistaa hallintoa, koska samaan palveluun liittyvää tietoa (IP-osoite, nimi ja varmenne) ylläpidetään keskitetyssä sijainnissa. Varmenne toisi vielä IP-osoitteeseen eli sijaintiin identiteettitunnisteen, joka on vahvempi kuin pelkkä DNS-nimi. Nykyisin DNS-palveluun lisätään erikseen palvelun nimi ja toisaalla varmennetoimittajalta tilataan palvelulle varmenne. Varmenteiden eliniän päätyttyä mahdollisesti eri varmennetoimittaja uusii varmenteen. Jos kohteen nimi ja sen käyttämä varmenne molemmat löytyisivät DNS:stä, vähenisi väärien varmenteiden riski. Lisäksi riippuvuus ulkoisista varmentajista vähentyisi, jos varmenteina voidaan julkaista omia, itse tehtyjä varmenteita. Tätä rajoittaa nykyisin riippuvuus esiasennettuihin varmentajiin. Kustannukset vähenevät, kun varmenteiden uusintaan kuluu vähemmän aikaa eikä varmenteista tarvitse maksaa aina rahaa varmentajalle.

Tutkimus kertoo syyt, jotka ovat johtaneet vaihtoehtoisiin malleihin löytää luottamus varmenteisiin ja näistä malleista esitellään tarkemmin DNS:n varmennejulkaisu. Edelleen se testaa, kuinka paljon työtä ratkaisun käyttöönotto vaatisi ja vertaa sitä olemassaoleviin luottamusmalleihin. Tuloksena esitetään käyttöympäristö, jossa varmenteita noudetaan DNSSEC-suojatusta palvelusta. Tärkeimpänä tuloksena näytetään, että virallinen varmenne ei välttämättä ole aina luotettava. Jos varmenteen tarkistus laajentuu DNS-hakemistoon, voidaan estää osa ongelmista, kuten varmentajien väärille kohteille myönnetyt varmenteet. DNSSECin merkitys kasvaa uusilla käyttökohteilla.

Työssä kerrotaan riittävä määrä DNS:n, DNSSECin ja PKI:n toiminnasta, jotta voidaan selittää luottamusketjun tarkistus, mutta muuten nämä oletetaan tunnetuksi yleisellä tasolla. Työ käsittelee varmennejulkaisua sekä DNSSECin kanssa että ilman. DNSSECin käyttöönoton jälkeen DNS oletetaan tässä tutkimuksessa täysin luotetuksi. Lähteinä käytetään runsaasti RFC-dokumentteja, joihin normaalisti ei ole soveliasta viitata, mutta tässä tutkimuksessa ne perustelevat kehityksen vaiheita. Työ on kirjoitettu suomeksi, sillä esimerkiksi DANEn käyttämiin suomenkielessä vakiintumattomiin termeihin halutaan ehdottaa vastineita.

Aiemmin on tutkittu kriittisesti juurivarmentajien roolia ja vastuun siirtoa muualle. sekä luottamusankkureiden tärkeyttä ja hallinnointia [5]. Muuntuneiden varmenteiden havaintoon on ehdotettu varmennelaajennoksessa välitettävää pientä lisämäärettä (”pin” tai ”tack”) [6; 7]. Maineeseen perustuvissa malleissa käyttäjä voi luottaa ennakkoon annettujen juurivarmentajien sijaan omavalintaisiin notaaripalveluihin [8; 9; 10]. RFC 4398 [11] (”Storing Certificates in the DNS”) vuodelta 2006 käsittelee CERT RR:n käyttöä X.509- ja OpenPGP- varmenteille, mutta CERT-tietueita ei käsitellä tässä työssä.

Painotus tässä työssä on varmennuksen tarkistus viimeisellä linkillä eli käyttäjän loppupäätös ja eroaa siinä aiemmista tutkimuksista. Tutkimus on luonteeltaan kvalitatiivinen, pyrkimyksenä ymmärtää ilmiötä ja perustella hyötyisivätkö nykyiset mallit varmenteiden julkaisemisesta DNS:ssä. Tätä tukeva testausympäristö käyttää DNSSEC-palvelinympäristöä, joka rakennettiin Laukan (2013) kandidaatintyön [12] ohjeiden pohjalta, koska tämän työn painopiste ei ole DNSSEC.

Luku 2 käsittelee nykyisin käytössä olevaa varmenteisiin luottamista ja mahdollisia käytännön haavoittuvuuksia TLS CA -mallissa. Luvussa 3. esitellään DNS:n käyttöä varmennehakemistona käyttäen SSHFP:ää ja DNS based Authentication of Named Entities -menetelmää eli DANEa. Sovellusten tuki varmenteiden hakuun DNS:stä kuvataan luvussa 4. Testiympäristö esitellään luvussa 5. Luvussa 6 pohditaan DNS:n sopivuutta luottamuksen turvaajaksi ja lopuksi luvussa 7 esitetään johtopäätökset ja suositukset.

# 2. LUOTTAMUKSEN SIJAINTI JA TARKISTUS

Varmenne on sähköisesti allekirjoitettu todiste, joka sitoo julkisen avaimen ja sen esittäjän nimitiedot toisiinsa. Julkista avainta tarvitaan vahvaan autentikointiin ja salattuun liikennöintiin. Varmenteeseen luottamus liittyy luottamiseen varmenteen tekijään . Tutkimus vertaa seuraavassa varmenteeseen sidotun luottamuksen nykyistä etsintää luonnosvaiheessa olevaan, DNS:ää hyödyntävään menetelmään ja kertoo, kuinka X.509-varmenteiden [13], SSH-avainten ja DNSSECiin kuuluvien julkisten avainten luottamus tarkistetaan.

## 2.1 Taustaa varmenteista

Varmenne sisältää tiedon varmentajasta, käyttötarkoituksesta, eliniästä, mitätöintimenetelmästä ja haltijan nimestä. X.509-varmenteen myöntäjää kutsutaan varmentajaksi (Certificate Authority, CA). Varmenne voi toimia itsensä varmentajana (self-signed certificate). Varmentajalla itselläänkin voi olla oma varmentaja, jolloin alempaa varmentajaa kutsutaan välivarmentajaksi (Subroot-CA). Luotettu varmentaja kuuluu luottamusankkureiden [5, s. 61] joukkoon.

Luottamuksen tarkistus tarkoittaa yhteyskumppanien autenttisuuden tarkistusta. Käyttäjille tämä näkyy ajoittain tuntemattoman varmenteen hyväksymispyyntönä tai varoituksena varmenteen virheellisyydestä. Usein käyttäjät kuitenkin ohittavat varoitusviestit, minkä on tutkittu liittyvän käytettävyysongelmiin. Käytettävyyttä vaikeuttaa puolestaan se, että tehtävän ratkaiseminen on peruskäyttäjälle yleisesti vaativa. Käyttäjä voi jättää huomioimatta vihjeet siitä, että liikennettä on voitu salakuunnella tai muuttaa. Katseenseurantatutkimukset [14] näyttävät, että käytäjä ei huomioi passiivisia vihjeitä, kuten selaimen osoiterivillä olevaa suljettua lukkoa tai vaihtunutta väriä [15, s. 60]. Turhiin varoituksiin turtuminen [15, s. 59] puolestaan on vienyt käyttäjän huomion siitä, että joskus voisi olla oikea syy varovaisuuteen [16].

Jos mietitään kuinka kauas luottamusta voidaan siirtää, niin jossain vaiheessa löytyy riittävän luotettuna pidetty tai tarkistettu piste, jota nimetään luottamusankkuriksi. Luottamuksen tarkistus alkaa vastapuolen väitöksestä ja sen todisteista, joita kuvaa vastapuolen lähettämä varmenne. Automaattinen tekninen tarkistus paljastaa varmenteen mahdolliset puutteet. Varmenteen tarkistaja päättää tämän ja varmentajan perusteella luottaako hän varmenteeseen. Myös varmentajan luotettavuus on tarkistettava, jos se ei itse ole luottamusankkuri. Syntyy yksittäisten

luottamusketjujen tarkistuspolku luottamusankkuriin.

## 2.2 Luottamuksen löytäminen eri malleissa

Kohde ja lähde autentikoituvat yhteyden avauksessa. Riippumatta siitä, ketkä osapuolet esittävät varmenteen, molemmat tekevät luottamuspäätöksen erikseen. Ideaalitapauksessa luottamus toisen varmenteeseen on syntynyt jo ennakkoon, muussa tapauksessa se täytyy tarkistaa.

Käyttöjärjestelmän tai www-selainohjelmiston tarjoamat varmenteet ovat luottamusankkureita ilman erillistä tarkistusta. Tarkistuksen on käyttäjän puolesta tehnyt näiden järjestelmien toimittaja. Käyttäjä tietää luottamusankkureista vähän tai ei yhtään mitään [5, s. 1].

### 2.2.1 Hierarkinen luottamus

Taulukko 2.1 esittää varmenteiden ja tiivisteiden esiintymistä DNS:ssä. Hierarkinen varmennejärjestelmä on malliltaan puu ja sen luottamusankkureina toimivat juurivarmenteet. Ulkopuolisena todistajana ne varmistavat varmennetta hakevan yksikön oikeellisuuden. Jos juurivarmentaja on vakuuttunut, allekirjoittaa se hakemuksen (Certificate Signing Request) kryptografisesti omalla salaisella avaimellaan. Lopputulosta kutsutaan varmenteeksi. Kuka vain voi tarkistaa allekirjoituksen liittymisen varmentajan julkiseen avaimeen. Sekä PKIX että DNSSEC ovat hierarkisia järjestelmiä.

Hallinnollisten syiden takia [17; 18] juurivarmentajan ja lopullisen kohteen välillä voi olla välivarmentajia eri käyttötarkoituksiin. Hierarkia kulkee puun juuresta mahdollisten välivarmentajien kautta lopulliseen varmistettavaan kohteeseen. Välivarmentajan uusiminen on helpompaa kuin juurivarmenteen vaihtaminen, sillä välivarmenteen uudelleenluomisen jälkeen sillä on kuitenkin sama luottamusankkuri.

Taulukko 2.1: Julkisten avainten ja tiivisteiden käyttö DNS:ssä.

| RR-tyyppi | Selite |
|---|---|
| SSHFP | palvelimen tai käyttäjän SSH-avain |
| TLSA | TLS-varmenteen tarkistus |
| SMIMEA | S/MIME-varmenteen tarkistus |
| OpenPGP | PGP-avaimen tarkistus |
| DKIM | vyöhykkeen tunnustama oma SMTP-palvelin |
| RRSIG | DNSSEC-tietueen sormenjälki |
| DNSKEY | ZSK- ja KSK-RRSIG varmistukseen |
| DS | alivyöhykkeen KSK:n sormenjälki delegoivassa vyöhykkeessä |
| CERT | tapa julkaista X.509 varmenne |
| TXT | yleiskäyttöinen tapa julkaista mitä vain |

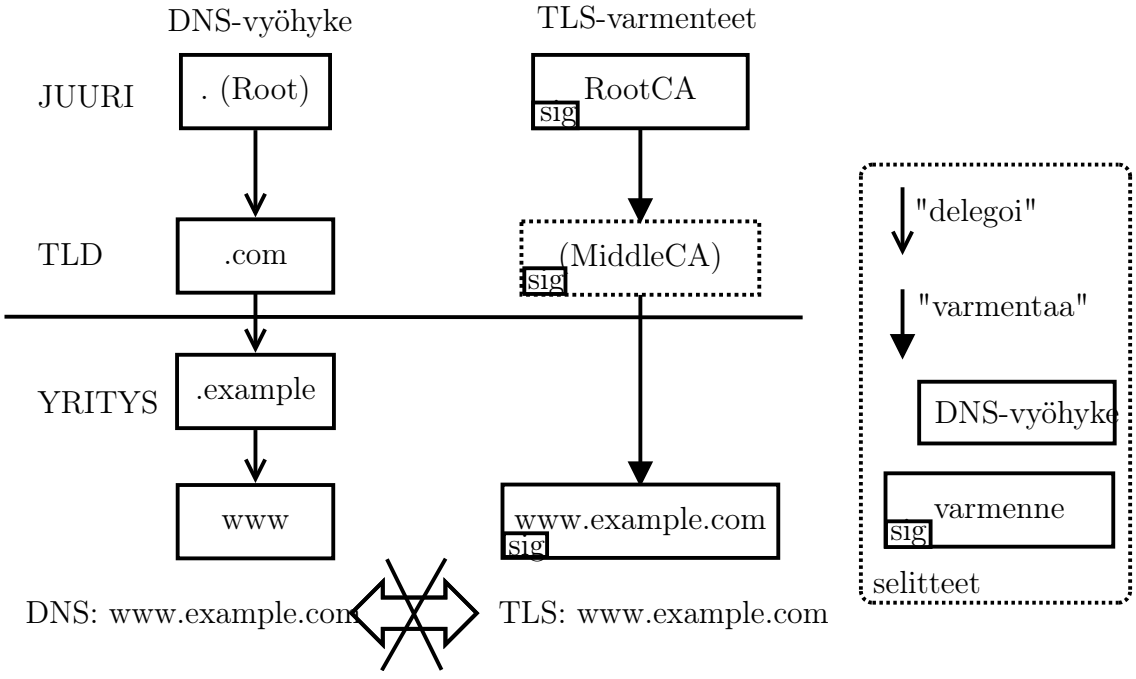
Juurivarmenteella puolestaan ei ole luottamusankkuria, joten se täytyy varmistaa erikseen. Poikkeuksena ovat sellaiset toisensa ristiinvarmentavat (BridgeCA) juurivarmentajat, joita tämä työ ei käsittele.

DNS-nimellä on hierarkia DNS-juureen, samoin varmenteella luottamusankkurille. Kuvassa 2.1 verrataan DNS- ja PKIX-hierarkiaa palvelimelle `www.example.com`. Huomattavaa on, että DNS-nimellä ja TLS-varmenteella ei ole keskenään yhteistä hierarkiaa. DNS-hierarkiasta puuttuu luottamussuhteet ja TLS-hierarkiasta puuttuu DNS:n yksikäsitteisyys. Samalla DNS-nimellä voi olla varmenteita eri varmentajilta [19], joten `www.example.com` varmenteita voi olla olemassa monta.

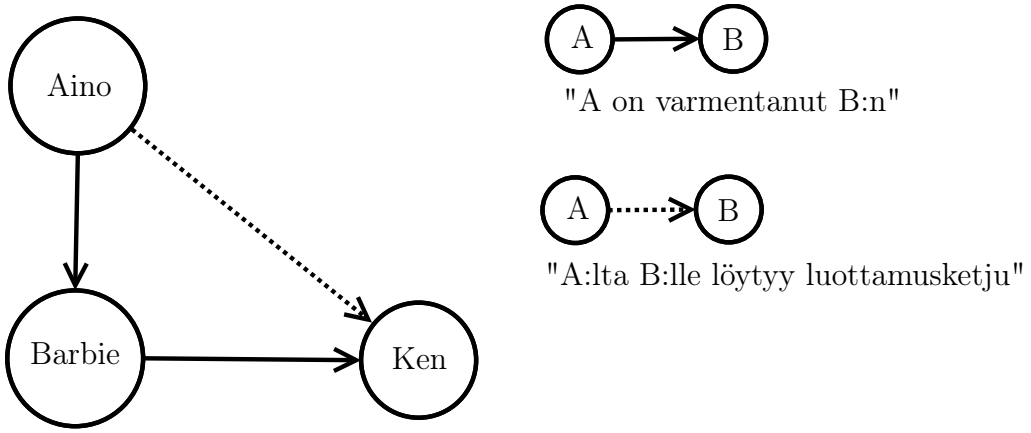
### 2.2.2 Luottamusverkosto

Luottamusverkosto (Web of Trust) on malliltaan graafi, jossa luottamusyhteyksiä voidaan luoda moneen suuntaan. Varmentaja ja varmennettava voivat varmentaa toisensa ristiin, jolloin saadaan molemminpuolinen varmennus. Kuvassa 2.2 Aino varmentaa Barbien ja Barbie varmentaa Kenin. Ainolla on siten luottamusketju Keniin.

OpenPGP on luottamusverkosto ja sen julkisiin avaimiin voidaan liittää useita allekirjoituksia. OpenPGP-mallissa puhutaankin avaimista, ei varmenteista. Luottamusverkostolle on ominaista, että verkosto kasvaa henkilökohtaisesti varmentamalla muiden avaimia. Käyttäjä määrittää muiden avaimille omat painoarvot sen mukaan, kuinka paljon hän luottaa toisen avaimenhaltijan kykyyn varmentaa muita avaimenhaltijoita. Tämä tieto säilytetään käyttäjällä itsellään. Kaikille omaan verkostoon




Kuva 2.1: DNS- ja PKIX-hierarkia.




Kuva 2.2: Luottamusverkosto.

kuuluville avaimille voidaan näin laskea luottamusarvo, jolloin tietyn kynnyksen ylittävät avaimet muodostavat luottamusankkuriston, vaikka tätä mainetta ei levitetä muille käyttäjille.

Ennakkoon ei ole olemassa mitään luottamusta. Avaimen luonnin jälkeen oma avain kiinnittää ensimmäinen luottamusankkurin allekirjoittamalla itsensä. Omaan verkkoon kuulumattoman avaimen luottamus selviää luottamuspolkuja tarkistamalla. Luottamus siis lähtee omasta avaimesta kohti muita avaimia luottamusverkostossa. Hierarkisessa järjestelmässä luottamus rakentuu ennakkoon luotetuista ulkoisista juurivarmenteista kohti varmennettavaa kohdetta.

### 2.2.3 TOFU ja maineeseen perustuvat mallit

Kun käyttäjältä hyväksytty luottamus tallentuu ensimmäisellä yhteyskerralla, eikä luottamusta kysytä enää jatkossa, on kyseessä TOFU-malli (Trust on first use). Jos tulevaisuudessa kohteen varmenne on muuttunut, huomauttaa sovellus tästä käyttäjälle [20, s. 15–16]. Koska käyttäjä käynnistää yhteyden, on täsmähyökkäyksen riski pieni, mutta kuitenkin olemassa. Malli sallii, muttei vaadi, että käyttäjä on hankkinut ennakkoon toista yhteyskanavaa pitkin luotettavasti varmenteen tunnusmerkit kuten sormenjäljen. SSH ja varmennepoikkeusten lisääminen selaimeen käyttävät TOFU-mallia.

Luottamus voi perustua myös maineeseen. Esimerkiksi Perspectives [9] käyttää monipolkuista TOFU-mallia. Kuka tahansa voi ylläpitää omaa notaaripalvelua, joka seuraa varmenteiden muuttumista. CA:n sijaan käyttäjä voi valita ryhmän luotettuja notaareja, ja varoitus näkyy vain silloin, kun varmenne on muuttunut. Luottamuspolkuja syntyy siis monta [21], minkä ansiosta muuttunut varmenne paljastuu helpommin kuin yksipolkuisessa TOFUssa. Convergence [8] on tästä varioitu malli. Electronic Frontier Foundationin (EFF) Sovereign Keys [10] on puoliksi keskitetty malli, joka ei tarvitse ulkoisia luottamuksen myöntäjiä sen jälkeen, kun palvelin on

luonut erityisen avaimen. Käytetyistä varmenteista tallennetaan jälki erilliseen historiatietoja tallentavaan palvelimeen. DNS-nimen hallintaoikeus on silti osoitettava CA:lla tai DNSSEC-avaimella. Varmennehistorian tallentamista juoksevaan listaan on tutkittu myös tarkoituksena selvittää, millä varmenteella salaus on suoritettu menneisyydessä [22].

## 2.3 Virallisten juurivarmenteiden luottamusharha

Juurivarmentaja vähentää erikseen tehtävien luottamuspäätösten määrää. Selaimet esiasentavat juurivarmentajia luottamusankkurisäiliöönsä (Trust Anchor Store, TAS) noin 160 kappaletta [24, s. 1]. Microsoftin selaimissa oli alunperin tilaa noin 100-200 varmenteelle [5, s. 61]. Käyttöjärjestelmät tarjoavat lisäksi omia varmenteita, jotka ovat osin samoja edellisten kanssa. Debian GNU/Linux 7.0 käyttöjärjestelmä antaa käyttäjän valita haluamansa 450 varmentajan joukosta. Käyttöohjeissa [25] sanotaan selkeästi, ettei käyttöjärjestelmä itse vahvista minkään varmenteen oikeellisuutta, joten käyttäjä saa itse päättää luottamuksestaan. EFFin projekti SSL Observatory [26] antaa kuvan maailmalla käytössä olevasta noin 650 varmentajasta, jotka ovat Microsoftin tai Mozillan tai molempien ohjelmistojen luottamusankkurisäiliössä.

On huomioitava, että termi ”virallinen varmenne” on harhaanjohtava. Usein tällä tarkoitetaan vain sitä, että jokin esiasennetuista juurivarmentajista on taannut yhteyden ja ketju luottamusankkuriin on ehjä. Se ei kuitenkaan tarkoita, etteikö toinen virallinen juurivarmentaja toisaalla olisi voinut tehdä aivan samoin samannimiselle hyökkääjän hämäyspalvelimelle. Juurivarmentaja ei ole sidottu mihinkään nimiavaruuteen, jonne se voi teknisesti varmenteita myöntää. Toisin sanoen, kuka tahansa esiasennetuista varmentajista voi myöntää virallisen varmenteen nimelle `secure.example.com` ja ”virallinen” menettää merkityksensä. Yksikin näistä juurivarmentajista voi tahallisesti tai virheellisesti myöntää väärälle kohteelle varmenteen. Silloin ehjältä vaikuttava luottamusketju vie hyökkääjän palvelimelle. Tämä riski toteutui koko 2000-luvun alun ajan [27], mutta yleisesti tunnetuksi se tuli vasta vuonna 2011, jolloin juurivarmentajat Comodo [17] ja DigiNotar [18] erehtyivät myöntämään väärälle hakijalle varmenteita ja kärsivät siitä. Comodo selvisi tästä vielä, mutta DigiNotar ajautui lopulta konkurssiin. Selaimet ja käyttöjärjestelmät poistivat päivityksinä vialliset varmenteet ja asettivat osan myönnetyistä varmenteista ennakkoon kovakoodatulle sulkulistalle. DigiNotar oli toiminut muun muassa Hollannin hallituksen käyttäjähallinnon (PKIoverheid) varmentajana. Comodon osalta asiaa lievensi se, että siltä hävisi vain välivarmentaja (MiddleCA tai Subordinate CA), kun DigiNotarin tapauksessa luottamus katosi juureen asti. Viestintäviraston CERT-FI-yksikkö teki koosteen sivuillaan juurivarmentajiin luottamisesta [28].

Ulkoisia varmentajia on näin ollen liikaa. Pienempi määrä juurivarmentajia vähentäisi osaltaan DigiNotarin kaltaisia tapauksia. Vielä helpompaa olisi, jos luotta-

musankkureita löytyisi rinnakkaisilla tavoilla ja itse hallinnoitu DNS kertoisi, mikä varmenne on luotettava. Pelkkä varmenteen julkinen avain ei riitä, sillä siitä puuttuu Certificate Policy Stamement (CPS). CPS esimerkiksi kertoo, kuinka varmenteen myöntäjä huolehtii omasta tietoturvastaan. Varmenteen käyttötarkoitukselle on varattu omat bitit X.509-varmenteessa. Yksi tapa havaita moninkertaisen luotetut varmentajat on Googlen kehittämä Certificate Transparency [23]. Siinäkin myönnetyistä varmenteista kertyy lokia. Tavoite on, että myönnetyistä lokiin ilmestyneistä varmenteista kulkeutuu tieto DNS-nimen omistajalle nopeasti, eikä virheellinen varmennemyöntö jää huomaamatta.

### 2.3.1 Yhteenveto luottamuksen tarkistuksesta

Ennakkoonpäätetty luottamus tarkoittaa, että käyttäjän ei tarvitse tehdä erikseen työtä, koska luottamusankkurit ovat jo luotettuja. Varmennehierarkiassa luottamusankkureina toimivat tunnetut juurivarmentajat ja luottamusverkostossa ne naapurit, joiden kynnysarvo on ylittynyt. Jälkimmäisissä kahdessa järjestelmässä luottamusankkuri on varmennettava etukäteen, jotta luottamusketjun tarkistus onnistuisi.

Luottamusankkureiden hallinta on vaativinta, sillä murtunut ankkuri rikkoo luottamuksen kaikkiin siihen tukeutuviin ketjun osiin. Luottamuspolkujen tarkistus on mekaanista, kun luottamusankkuri on kiinnitetty. TOFUssa luotetaan ensimmäisen yhteydenoton tuottamaan luottamusankkuriin, mikä parantaa käytettävyyttä. Maineen vahvuus on siinä, että tieto maineen menetyksestä leviää nopeasti.

## 2.4 Käytettävyysongelmat

Ennakkoon annetun varmenteen pitäisi olla luotettu. Sen todistamisessa tärkeää on vaihtoehtoinen viestikanava (Out-of-band). Kanavaksi käy esimerkiksi paperilla tai puhelimitse saatu tieto varmenteen sormenjäljestä, jonka ihminen varmistaa käsin alle minutissa. Pääasia on, että tiedonvälitys ei ole käyttänyt samaa kanavaa, kuin mistä varmenne alunperin tuli, sillä tämä kanava voi väittää mitä tahansa varmenteesta. Tässä tutkimuksessa DNS on vaihtoehtoinen viestikanava.

Ymmärrettävästi moni ei halua vaivautua toisen kanavan tarkistukseen, vaan ulkoistaa luottamuksen suoraan kaikille luottamusankkureille ennakkoon. Lisäksi moni luottaa uusiin varmenteisiin oletuksena, sillä työmäärä kaikkien näiden yksittäiseen varmentamiseen on liian suuri. Vaikka käyttäjä saa varoituksen varmenteen epäluotettavuudesta, ohittaa hän usein sen. Syynä voi olla se, että hän on oppinut tavan muilta, jos oppimistapahtuma ei ole liittynyt turvallisuuteen. Suurin syy voi kuitenkin olla se, että käyttäjä ei ymmärrä annettua varoitustekstiä [16, s. 14] ja ehdollistuminen [15, s. 68; 20, s. 1] aiheettomiin varoituksiin vaarantaa turvallisen käytön. Aiheettomat varoitukset ovat lieviä kuten varmenteelle annetun eliniän päät-

tyminen tai sellaisen turvallisen luottamusankkurin käyttö, jota päätelaite ei tunne. Harvoin kyseessä on oikea reagoimista vaativa tietoturvarikkomus. Käyttäjät oppivat ohittamaan liian paljon huomiota vievät virheilmoitukset [15, s. 59–60; 20, s. 1]. Epäselväksi jää se, millaiseen varoitukseen käyttäjä reagoisi oikein [16, s. 14].

# 3. LUOTTAMUKSEN SIIRTO DNS:ÄÄN

Galvin [29] esitti DNS:ää mahdollisena globaalina varmennejakelukanavana vuonna 1996. DNSSEC oli tuolloin vasta suunnitteilla, joten ehdotus ei saanut toteutusta. Kerrataan seuraavaksi DNS:n ja DNSSECin perustoiminta ja tutustutaan luottamusankkureiden julkaisuun DNS:ssä sekä yleiseen avainten jakeluun ja tarkistukseen ehdotettuun DANE-protokollaa.

## 3.1 DNS hajautettuna hakemistopalveluna

DNS on rakenteeltaan hierarkinen hakemisto, joka sitoo nimiä resurssitietueisiin (RR). DNS jakaantuu tietokantaan ja nimenselvitykseen (selvitin, englanniksi resolver). Ensimmäinen on vastuussa (autoritäärinen) nimistä ja jälkimmäinen ohjaa kyselyn eteenpäin hierarkiassa. DNS ei sisällä eheyden eikä autenttisuuden tarkistusta. Nämä sisältyvät DNSSEC-laajennukseen.

Saatavuus tarkoittaa tässä sitä, että palvelu on käytettävissä. Saatavuus paranee, kun DNS-palvelimia hajautetaan. DNS-vyöhyke hajautuu monistamalla autoritäärisiä DNS-palvelimia. Ylläpidettäväksi jää näiden palvelimien sisäisen tiedon synkronointi, joka voi tapahtua esimerkiksi vyöhykesiirroilla. Hajautus on maailmanlaajuinen, kun DNS:n eri juuripalvelimet sijaitsevat eri maissa. Lisäksi anycast-osoitteistus reitittää pyynnöt lähimmälle kopiolle samannimisistä juuripalvelimista.

Tiedonsiirtoon DNS on käyttänyt alusta asti User Datagram protokollaa (UDP). Se toisaalta pienentää vähän pyyntöjen ja vastausten kokoa, mutta ei varmista perillepääsyä, sillä UDP ei ole määritelmän mukaisesti luotettava siirtokerroksen protokolla. DNS voi tarvittaessa käyttää TCP:tä esimerkiksi suuriin vyöhykesiirtoihin tai rajoittamaan osaa palvelunestohyökkäystyypeistä.

DNS:n laajentaminen tapahtuu tietuetyyppejä (eng. Resource Records, myös RR) lisäämällä. Tavanomaisten A, PTR, NS, MX ja SOA joukkoon on kehitetty SSHFP, TLSA ja SMIMEA, jotka esitellään tässä. Myös CERT RR:ää käytetty X.509-varmenteisiin ja tekstimuotoista TXT RR:ää yleisiin laajennuksiin, mutta tässä työssä niitä ei esitellä. CERT RR:n luonnin aikaan (2006) ei maa-vyöhykkeitä eikä varsinkaan juuria oltu vielä allekirjoitettu DNSSEC-hierarkiassa. Koska tarvetta uusille tyypeille syntyy jatkuvasti, on `kitchen sink` -tietuetta (Code 40) esitetty RR:ille, joita ei ole tarkoitus rekisteröidä yleiseen käyttöön [30].

## 3.2 DNSSEC

DNS:n haavoittuvuuksia ovat datan vioittuminen, dynaamiset päivitykset vääristä lähteistä, DNS-palvelinten kaappaukset, DNS-kyselyjen tai vyöhykesiirtojen kaapaus ja muokkaus (MITM-hyökkäys), DNS-välimuistin saastuttaminen ja palvelunestohyökkäys (DoS). Yleisin DoS on väärällä lähdeosoitteella tehty massiivinen pyyntökuormitus (spoofing-based DoS) [4]. DNSSEC korjaa osan DNS:n puutteista allekirjoittamalla digitaalisesti vyöhykkeen ja sitomalla alemman tason vyöhykkeet ylempään tasoon. Ylempi taso tarkoittaa tässä DNS-vyöhykettä, jonka alla alempi DNS-vyöhyke sijaitsee. Esimerkiksi '.fi'-vyöhykkeen alla on '5w.fi' niminen alemman tason vyöhyke, jota organisaatio 5w hallinnoi. Allekirjoitusten tarkastuksella asiakas voi varmistua tietojen muuttumattomuudesta ja autenttisuudesta. NSEC-tietueen avulla pystytään vastaamaan, että jotain tietoa ei ole olemassa.

Koska DNSSEC suojaa vain DNS-palvelinten välisen eheyden, on pyyntö asiakkaalta lähimmälle DNS-palvelimelle suojaamaton, mikäli asiakkaalla itsellään ei ole DNSSEC-selvitintä. Jos DNS on toimiva, lähellä ja käyttäjä luottaa siihen, sieltä saatu tieto varmenteista ei ainakaan huononna nykyistä tilannetta.

DNSSEC-palvelimen ylläpitoa lisää oikean tietuerakenteen luonti, vyöhyketietueiden allekirjoitus aina tietueita muokattaessa ja kahden avaimen hallinta. Normaalin vyöhyketarkistuksen ja uudelleenlatauksen lisäksi vyöhyke allekirjoitetaan vyöhykeavaimella (ZoneSigningKey, ZSK). ZSK puolestaan täytyy allekirjoittaa säännöllisesti avaimen allekirjoitusavaimella (KeySigningKey, KSK). KSK:n julkinen avain vaihtuu vain, jos sen käyttämä avain vaihdetaan. Vaihtosyynä voi olla yksityisen avaimen paljastuminen, hukkuminen tai tarve sen kryptoparametrien vaihtamiseen. DNSKEY-tietue kertoo sekä ZSK:n että KSK:n julkisen avaimen. Tiiviste ZSK:n julkisesta avaimesta kopioidaan turvallisella menetelmällä ylävyöhykkeelle Delegation Signer -tietueeksi (DS). Tämä ketjuttaa alivyöhykkeen ylävyöhykkeeseen vahvasti.

DNS-palvelimissa on välimuisti toistuvien kyselyjen nopeuttamista varten. Jos tiedot muuttuvat DNS-palvelimella, muutosten eteneminen viivästyy välimatkalla olevien välimuistien keston verran. Näin kyselijä voi saada samaan aikaan useaa eri tietoa riippuen nimipalvelimen sijainnista. Jotta monikäsitteisyyttä ei tulkittaisi väärennökseksi, on muutosikkunan aikana mahdollista julkaista useampaa varmennetta samalle kyselylle. Kaukaa kyselyn suorittava saa vanhan varmenteen tiedon, mutta läheltä kysyvä saa sekä vanhan että tulevan varmenteen tiedon. Kun vaihdosaika on ohi ja uusi varmenne on levinnyt myös välimuistien taakse, voidaan uusi varmenne ottaa käyttöön palvelimessa ja vanha varmenne poistaa autoritäärisesta DNS:stä.

DNSSECissä DNS-tietueelle on olemassa autoritäärisen nimipalvelimen ZSK:n allekirjoittama vastaava RRSIG-tietue. Luottamusketjun tarkistus etenee DNSSEC-selvittimessä samankaltaisesti kuin muissakin malleissa, ketjun lenkki kerrallaan

alkaen luottamusankkurin tarjoamasta DS:stä. Ennen varsinaisen tietueen allekirjoituksen varmistamista selvitin tarkistaa, onko vyöhyke oikea ylempien vyöhykkeiden mielestä. Tähän tarvitaan tietueet DS ja sen RRSIG ylemmästä vyöhykkeestä sekä KSK:n DNSKEY ja sen RRSIG alemmasta vyöhykkeestä. Vaikka maatasolla DNSSECin käyttö on mahdollista jo 61 maassa, joiden TLD on allekirjoitettu juurinimipalvelimilla, ei sen käyttö ole vielä (2013/09) yleistynyt [31]. Koska DNSSEC on kuitenkin taaksepäin yhteensopiva DNS:n kanssa, ei sen käyttöönotto riko vanhoja sovelluksia. Yksi käyttöönottoa helpottava ohjelma on Dan Kaminskyn Phreebird [32], joka tekee valmiista DNS-hakemistosta DNSSEC-palvelimen. DNS-palvelin asetetaan kuuntelemaan Phreebirdiltä tulevia pyyntöjä ja Phreebird asetetaan varsinaiseksi DNS-edustapalvelimeksi. Phreebird lisää pyyntöihin automaattisesti DNSSEC-allekirjoitukset ja toimii lisäksi kaistanrajoittimena. Tällaiset ratkaisut helpottavat uusien tekniikoiden tunnettavuutta ja käyttöönottoa siirtymävaiheessa.

## 3.3 SSHFP

SSH:ssä käyttäjän henkilökohtainen avainvarasto täydentyy TOFUlla aina otettaessa yhteyttä uuteen SSH-palvelimeen. Vaihtoehtoisesti käyttäjä voi luottaa myös ennakkoon kiinnitettyyn, käyttöjärjestelmälaajuiseen listaan luotetuista palvelinavaimista. RFC4255 [33] kuvaa, kuinka Secure Shell Key Fingerprint -tarkistus eli SSHFP toimii. SSHFP:tä käytettäessä SSH-asiakas pyytää DNS:ltä SSHFP RR:ää, joka kertoo SSH-palvelimen avaimen sormenjäljen, muodostustavan ja tyypin. Jos tämä vastaa sormenjälkeä, joka otetaan palvelimen lähettämästä julkisesta avaimesta, on palvelin autentikoitu oikein.

Taulukossa 3.1 `kone` edustaa SSH-palvelinta. `IN` on yleisin DNS-luokka (Internet), `SSHFP` RR ja `alg` SSH-palvelimen julkisen avaimen käyttämä algoritmi. IANA on määritellyt algoritmeille numeroarvot, jotka vastaavat tällä hetkellä RSA-, DSA- ja ECDSA-algoritmeja. `tyyppi` kentän algoritmi liittyy sormenjälkeen (1:SHA1, 2:SHA256). Viimeinen sarake kertoo itse sormenjäljen. Jälkimmäisen rivin SSH-avain on tarkoitettu geneerisenä esimerkkimuotona sellaiselle DNS-palvelimelle, joka sisäisesti ei tunne SSHFP-tietuetyyppiä. 22 on sormenjäljen koko heksadesimaalina. Huomioitavaa on, että DNS on suunniteltu laajennettavaksi, ja IANA voi määritellä uusia algoritmeja [34].

Taulukko 3.1: DNS-tietue SSHFP.

| kone | luokka | DNS RR | alg | tyyppi | sormenjälki |
|---|---|---|---|---|---|
| host | IN | SSHFP | 2 | 1 | 253d9636219be6a1342acb |
| host | IN | TYPE44 # 22 | 02 | 1 | 253d9636219be6a1342acb |

## 3.4 DANE

Uusi ehdotus DANE (DNS Authenticated Named Entities) tavoittelee nimien autentikointia DNS-palvelussa. Tämä tutkimus näyttää, miten DANEn mukaan julkaistaan varmenteita. DANE tuo DNS-operaattorille määräysvaltaa siitä, miten domainin varmenteiden luottamus pitää tarkistaa lisäämällä erilaisia rajoitteita DNS RR:ään.

DANE-määrittelyistä pisimmällä ovat TLSA [35] ja SMIMEA [36], jotka kertovat varmenteen julkaisutavan ja rajoitteet. Alkuvaiheessa ovat DANE-määrittelyt OpenPGP-avaimille vuodelta 2013 [37], SMTP:lle ja SAML:lle. SMTP DANE on tarkoitettu postipalvelimien väliseen tiedonsiirtoon eli se on M2M-liikennettä (Machine-to-Machine). SAML DANEa voidaan käyttää esimerkiksi luottamusverkostojen, kuten HAKA (Suomi), Feide (Norja) ja AAI (Tanska) välisissä viesteissä [38].

Normaali TLS-yhteys alkaa sovelluksen DNS-kyselyllä palvelimen IP-osoitteesta. DNSSEC-kyselyssä liikennettä syntyy enemmän ja IP-osoitteen oikeellisuus tarkistetaan ennen sen luovuttamista osoitteen pyytäjälle. Seuraavaksi sovellus avaa yhteyden IP-osoitteella palvelimeen, joka palauttaa jossain vaiheessa oman tai mahdollisesti kaikki varmenteet juurivarmentajaansa asti. Nyt sovellus tekee luottamustarkistuksen perustuen varmenteen myöntäjään ja ennalta tiedettyihin varmenteisiin. Tässä vaiheessa DANE-rajoitteet määrittävät, kuinka varmenteeseen pitää luottaa. Sovelluksen pitäisi pyytää jo ensimmäisen DNS-kyselyn yhteydessä lisätietoina mahdollisia DANE-rajoitteita. Ne olisivat tällöin nopeammin käytössä, eikä pyynnön koko kasvaisi merkittävästi tällä.

TLS- ja S/MIME-varmenteiden nouto DNS:stä vähentää kaapatun identiteetin hyökkäysvektoria, sillä oikeaa DNS:stä saatua tietoa voidaan verrata kaappaajan tarjoamaan tietoon. Jos DNS tarjoaa TLSA- tai SMIMEA-tietueen, täytyy sitä verrata palvelimen ehdottamaan varmenteeseen ennen vastaavan TLS- tai S/MIME-liikenteen aloittamista. S/MIME-varmenteet leviävät myös manuaalisesti, hakemistopalvelun kautta tai sähköpostin liitteenä. Tärkeää tarkistuksessa on varmistaa oikeellisuus vaihtoehtoista kanavaa pitkin. Jos DNSSEC on käytössä, voidaan väärän varmenteen hyökkäys [24] havaita. Varmenteen kuoletustarkistus hoidetaan nykyisin CRL-kyselyllä (Certification Revocation List) tai OCSP:llä (Online Certificate Status Protocol). Näiden rinnalle DANE tarjoaa DNS-tarkistuksen, sillä jos varmennetta ei löydy sieltä, on se joko kuoletettu tai muuten väärä. DANE ei tällä hetkellä tue kuoletustietoa muuten, mikä tullee aiheuttamaan ongelmaa, jos varmenteen allekirjoitus täytyy tarkistaa tulevaisuudessa.

### 3.4.1 TLSAn esitystapa DNS:ssä

DANEn TLSA-rajoite liittää julkaistavaan varmenteeseen neljä kenttää: käyttötavan (usage), valitsimen (selector), yhteensopivuuden (match) ja varsinaisen rajoitedatan.

Rajoitteella verrataan varsinaista palvelimen lähettämää varmennetta. Käyttötapa kertoo, miten varmennetta tulee käsitellä luottamusketjussa ja sen sanamuodon selvyyttä pidetään DANE-työryhmässä tärkeänä. Käytettävyyden parantamiseksi DANE-työryhmä on ehdottanut mahdollisuutta käyttää numerokolmikon sijaan tekstimuotoisia muistikkaita eli lyhyitä, muistettavia sanoja. Selkeänä, joskin hiukan juurivarmentajien roolia painottavana ehdotuksena on tauluun 3.2 lisätty Viktor Dukhovnin DANE-postilistalla [40] 2.12.2013 ehdottamat sanamuodot. SSHFP on kehitetty ennen DANEa. Samankaltaisuus näkyy, koska tekijöinä näillä on samoja henkilöitä.

Valitsimen mukainen osa palvelimen lähettämästä varmenteesta otetaan verrattavaksi. Vaihtoehdot ovat joko koko varmennetta tai verrataan DER-koodattua SPKI:tä (SubjectPublicKeyInfo). Sopivuus (matching) kertoo, onko rajoitedatassa koko varmenne vai vain sormenjälki siitä. Koska rajoite voi olla moniarvoinen eli TLSA-tietueita voi olla samalla kohteelle monta, niin varmenteiden vaihdon yhteydessä DNS voi sallia sekä vanhan että tuoreen varmenteen. Mikäli DNSSEC on käytössä, se allekirjoittaa TLSA-tietueet ja julkaisee vastaavan RRSIG-tietueen.

Taulukossa 3.3 on TLSA-tietue palvelimelle `www.` Palvelin vastaa TCP-portissa 443. Merkintä `_portti._protokolla_.nimi` sallii samalle nimelle useita palveluita eri porteissa ja protokollilla. Käyttötarkoitus=3 merkitsee, ettei luottopolun tarkistusta tarvita. Valitsin=0 tarkoittaa, että tiivistettä verrataan koko varmenteeseen, jonka TLS-suojattu www-palvelin lähettää. Sovitin=1 valitsee SHA256-tiivisteen.

### 3.4.2 SMIMEAn esitystapa DNS:ssä

Tämä kappale esittää tämän hetkistä (2013/09) luonnosta SMIMEAn esitystavasta. S/MIME varmenne on X.509 muotoa ja sidottu käyttäjän sähköpostiosoitteeseen. SMIMEAn rajoitteet ovat samoja kuin TLSA-tietueessa kuvassa 3.2, mut-

Taulukko 3.2: Käyttötavan, valitsimen ja yhteensopivuuden koodaus DANE-tietueessa.

| # | Käyttötapa (usage) | lyhenne-ehdotus |
|---|---|---|
| 0 | Sallitut (ulkopuoliset) juurivarmentajat. | CA-LOVER |
| 1 | Tietty varmenne DATA osassa, luottamus tarkistettava. | CA-SLAVE |
| 2 | Tietty luottamus-ankkuri (oma juurivarmentaja). | CA-OWNER |
| 3 | Tietty varmenne, ei luottopolun tarkistusta tarpeen. | CA-HATER |
| | Valitsin (selector) | |
| 0 | Koko varmenne binäärirakenteena [39] | |
| 1 | Varmenteen SPKI (SubjectPublicKeyInfo) DER-koodattuna [39, s. 17-18] | |
| | Sopivuus (matching) | |
| 0 | Täysi sopivuus varmenteeseen | |
| 1 | Sopivuus SHA256 tiivisteeseen varmenteesta [RFC6234] | |
| 2 | Sopivuus SHA512 tiivisteeseen varmenteesta [RFC6234] | |

Taulukko 3.3: DNS-tietue TLSA

| omistaja | Luokka | RR | käyttö | valitsin | sopivuus | data |
|---|---|---|---|---|---|---|
| _443._tcp.www | IN | TLSA | 3 | 0 | 1 | ab11c53ccec39de34... |

ta nimiosana sähköpostiosoite täytyy esittää sopivasti DNS:ssä, joka rajoittuu 7-bittiseen ASCII-merkistöön. Sähköpostiosoite koostuu lokaalista osasta (Left-Hand-Side, LHS) ja domain-osasta (RHS), joita erottaa merkki '@'. DNS:ään koodataan muoto "LHS@RHS" siten, että

1. LHS muunnetaan base32-koodauksella [41].
2. Lisävihjeeksi Scott Rose esitti 6.11.2013 dane-postilistalla [40] merkkijonoa "_sign" tai "_encr".
3. Varmennetyyppi ilmaistaan merkkijonolla "_smimecert".
4. LHS osuudet liitetään toisiinsa piste-merkillä.
5. '@' korvataan '.'-merkillä, kuten SOA-tietueen omistajan kohdalla.
6. RHS julkaistaan sellaisenaan.

Taulukossa 3.4 on kuvattu osoitteen `etunimi.sukunimi@example.com` koodaus. Rajoitteet ovat samankaltaisia kuin TLSA:n yhteydessä, mutta S/MIME on pysyvä varmenne. Edellisvuonna lähetetty, S/MIME:llä varmennettu sähköposti on voitava varmentaa myös tulevaisuudessa [42]. Kuoletusmerkintää ei ole ja on huomattavaa, että SMIME-viestit ovat pysyviä, kun TLS-sessiot vaihtuvat nopeasti. Koska base32-koodaus on merkkikokoriippuvainen, sähköposti "Etunimi.Sukunimi" olisikin "`IV2HK3TJNVUS4U3VNN2W42LNNE======`".

Taulukko 3.4: Ehdotus (2013/09) käyttäjän `etunimi.sukunimi@example.com` varmenteen nimiosaksi SMIMEA RR:ssa.

| |
|---|
| MV2HK3TJNVUS443VNN2W42LNNE======[1]._encr[2]._smimecert[3].example.com |
| MV2HK3TJNVUS443VNN2W42LNNE======._sign.[4]_smimecert.[5]example.com[6] |

# 4. SSHFP- JA DANE-TIETUEITA HYÖDYNTÄVÄT OHJELMISTOT

Jotta DNS:ään tallennetuista varmenteista olisi hyötyä, tulee sovelluksissa olla tuki niihin. Käytännössä ohjelman tulee osata pyytää tietoa DNS:stä sekä lisäksi tarjota käyttäjälle palaute tarkistuksesta. Seuraavassa esitellään tuettuja sovelluksia ja ohjelmakirjastoja, joilla tuki saadaan myös muihin ohjelmiin. Kooste sovelluksista ja niiden versioista on taulussa 4.1.

## 4.1 OpenSSH

Normaalin TOFU-menetelmän lisäksi OpenSSH voi etsiä luottoavainta DNS:stä, mikäli SSHFP-haku on tuettu. Asetuksen `VerifyHostKeyDNS` arvo

**yes** vertaa SSHFP:tä SSH-palvelimen avaimen tiivisteeseen,

**ask** kysyy käyttäjältä hyväksynnän, vaikka SSHFP on oikein ja

**no** ei tee lainkaan SSHFP kyselyä. Tämä on oletuksena asetettu.

Oletuksena SSH-palvelimelta saatu uusi, hyväksytty julkinen avain tallennetaan asiakkaan henkilökohtaisiin asetuksiin. `StrictHostKeyChecking` asetuksella käyttäytymistä voidaan muuttaa, jolloin TOFU ei ole käytössä. Mikään ei estäisi julkaisemasta samalla tavalla käyttäjäkohtaisia olevia SSH-avaimia, jolloin myös käyttäjä autentikoitaisiin DNS:n kautta. OpenSSH:ssä ei ole tukea tälle vielä suoraan, joten tunnistus tapahtuisi erillisen autentikaatiomoduulin (PAM) kautta.

## 4.2 Selainlaajennokset

Selaimille on olemassa laajennuksia (plugin), joilla DNSSEC- ja TLSA-varmennusta voidaan testata. DNSSEC Validator ilmoittaa osoiterivillä, jos kohde löytyy DNSSECin kautta. Extended DNSSEC Validator ilmoittaa lisäksi, jos kohteen varmenne pystytään todentamaan TLSA:n avulla. Se rajoittuu kuitenkin tällä hetkellä vanhempiin DANE määrityksiin (draft11) ja vain SHA1-tiivisteisiin, joten siitä eí ole apua testauksessa. Toisaalta laajennos on avoin, joten sitä olisi voinut itse korjata, jos aikaa olisi ollut enemmän.

DANE varmennuksiin laajennoksilla toimii nyt parhaiten DANE Patrol [43] ja sitä käytetään testausympäristössä. DNSSEC-tools projekti [44] olisi tarjonnut lisäkoodin usealle ohjelmistolle, jos ohjelmia kääntää itse. Tätä ei kuitenkaan käytetä tässä työssä.

## 4.3 Kirjastot

Omat ohjelmistot voivat käyttää valmiita yhteisiä ohjelmakirjastoja toimiakseen DANEn tai DNSSECin kanssa. Testiympäristössä käytetään ldnsutils:ia vyöhykkeiden allekirjoituksiin, sshfp-ohjelmaa SSH-palvelimista sormenjäljen laskemiseen ja tulostamiseen valmiiksi SSHFP-tietueeksi. Tuotantoympäristössä vyöhykkeiden allekirjoitus tapahtuu automaattisemmalla tavalla. GNUtls- ja OpenSSL-kirjastoissa on kehitystukea DANElle. Perl-moduuleista löytyi tuki Base32-koodaukselle, jota tarvitaan SMIMEAn kanssa, mutta se ei toiminut oikein täytemerkkien (englanniksi padding) kanssa. Tilalla käytettiin python-kielen base64-moduulia, joka nimestään huolimatta tukee myös base32-koodausta.

Taulukko 4.1: käytettyjen ohjelmistojen versiot

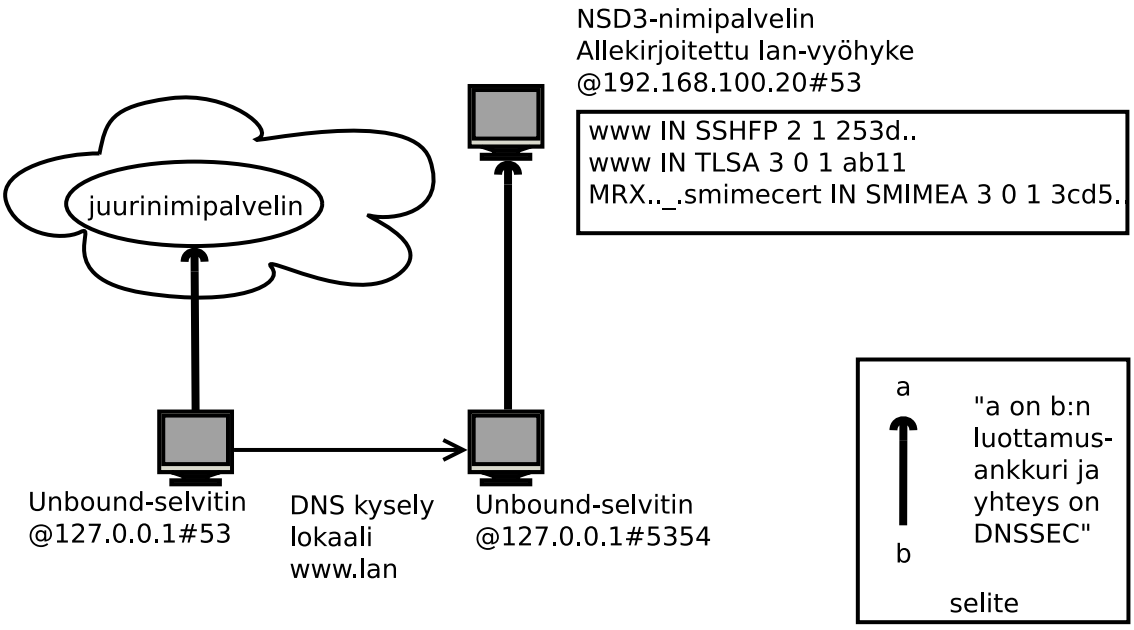
| Nimi | Versio | Käyttökohde |
|---|---|---|
| | **Sovellukset** | |
| Chromium | 31.0.1650.5 | www-selain |
| Iceweasel | 17.0.10esr-1 | www-selain (firefox) |
| Mutt | 1.5.21-6.2+deb7u1 | sähköpostiohjelma |
| NSD | 3.2.12-3 | autoritäärinen DNS |
| Openssh | 1:6.0p1-4 | SSH asiakas ja palvelin |
| Python | 2.7.3-4+deb7u1 | ohjelmointikieli |
| Unbound | 1.4.17-3 | selvittävä DNS |
| | **Kirjastot ja työkalut** | |
| dnssec-tools | 2.0-1 | DNSSEC ylläpitoon |
| ldnsutils | 1.6.13-1 | DNS(SEC):n hallintaan |
| OpenSSL | 1.0.1e-2 | SSL-työkaluja |
| sshfp | 1.2.2-4 | SSHFP ja DANE luonti |
| tlsa_rdata | 0.1 | TLSA-RR:n luonti |
| | **Selainlaajennokset** | |
| DNSSEC Validator | 2.0.1 | DNSSEC-tarkistin(Chromium) |
| DANE Patrol | 0.2.3 | DANE-tarkistin (Firefox) |
| Extended DNSSEC-Validator | 0.5 | DNSSEC-tarkistin(Firefox) |

# 5. TESTAUSYMPÄRISTÖ

Tavoitteena oli todentaa laboratorioympäristössä internet-luonnoksien mukaan varmenteiden ja avaimien tarkistusta ja noutoa DNS:stä. Toteutettavat tehtävät olivat 1. hae tietoja DNSSECistä, 2. käytä DNS:ää SSH-yhteyden varmistukseen, 3. käytä selainlaajennosta TLSA-varmennukseen ja 4. varmista S/MIME sähköpostin SMIMEAlla.

Ympäristöön kuului palvelinsovelluksina SSH-palvelin ja DNS, jonka sisällä oli TLSA-, SSHFP-, SMIMEA-tietueita DNS eriytettiin selvittimeen ja autoritäärisen nimipalveluun. Selvittimeksi valittiin Unbound ja autoritääriseksi DNS-palvelimeksi NSD3 niiden keveyden ja tuoreuden takia. Molemmat tukevat DNSSECiä. Muita mahdollisuuksia olisivat olleet PowerDNS, djbdns tai BIND. BIND:iä ei haluttu käyttää, koska sen eriyttämisominaisuudet ovat heikot ja siinä on laajuutensa takia ollut vuosittain tietoturvaongelmia kirjoittajan oman kokemuksen perusteella. djbdns puolestaan ei tue DNSSECiä, vaan omaa DNSCurveansa [45]. Valintaan vaikutti myös se, että valmiin ohjeistuksen [12] avulla DNSSEC-ydinympäristön rakentaminen kesti vain muutaman tunnin. Asiakassovelluksina toimivat OpenSSH, Chromium-selain laajennusosilla ja Mutt-sähköposti.

Kuva 5.1 hahmottaa testausympäristön. Autoritäärinet nimipalvelin asennettiin vastaamaan paikallisesta, ei virallisesta vyöhykkeestä `lan.` Autoritäärinen nimipalvelin sisälsi testitapaukset julkisista avaimista. Selvittimiä tehtiin kaksi. Ensimmäinen palveli kaikkia koneen tekemiä pyyntöjä normaalisti käyttäen ulkoisia, operaattorin tarjoamia nimipalvelimia, tarvittaessa DNSSEC-tuella. Toinen selvitin vastasi lan-vyöhykettä koskeneisiin pyyntöihin. Näitä varten selvitin oli kiinnittänyt luottamusankkurin DNSSEC-tarkistusta varten lan-vyöhykkeen DNSKEY:hin. Varsinainen nimikysely välittyi siis lopulta autoritääriselle palvelimelle kahden selvitinpalvelimen kautta, mutta kakkosselvitin tarkisti DNSSECin avulla luotettavuuden. Tämä vastasi käyttötapausta, jossa luottamus on siirretty ulkoisen operaattorin nimipalvelimelle. Menettely yksinkertaisti vyöhykkeiden DNSSEC-testausta, sillä lan-vyöhykkeen vieminen viralliseen haaraan olisi ollut testin kannalta epäolennaisen työlästä. Mahdollista olisi ehkä ollut käyttää yhtäkin selvitintä, jos sille voidaan määritellä monta luottamusankkuria.




Kuva 5.1: Testiympäristö.

## 5.1 Testitapaus 1. DNSSEC-kysely omalle vyöhykkeelle

Oman vyöhykkeen autoritääriselta palvelimelta kysyttiin DNSSEC-tarkistuksella SOA-tietue. Tämä näkyy koodissa 5.2. Vastauksessa tuli mukana allekirjoitukset (RRSIG), mutta flags-rivillä ei näkynyt ensin DNSSEC-kyselyn onnistumisesta ilmoittavaa AD-lippua, joten tarkistusosuus epäonnistui. Tämä johtui sekä sidoksen puuttumisesta juuripalvelimeen että autoritäärisen nimipalvelimen kyvyttömyydes-

```
$  dig @ns1.lan -t SOA ns1.lan +dnssec
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @ns1.lan -t SOA ns1.lan +dnssec
[..]
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34315
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
;; WARNING: recursion requested but not available
[..]
; EDNS: version: 0, flags: do; udp: 4096
[..]
;; AUTHORITY SECTION:
lan.  86400   IN  SOA     lan. dnsmasteri.lan. 2013102701 28800 7200 864000 86400
lan.  86400   IN  RRSIG   SOA 7 1 86400 20131222215905 20131124215905 55919 lan.\
      hassOeCzcs8Dl2cmT0rRDN99wb5+bUHzBNM7ydslPToyCKDva/16Vdgx u5PyAJl2yCBdK0Syy\
      e85sr4tE7dJpE2O/1u68wgxC0gso77vPRZ0WEqo 2XhWEMKOSSr3uz2PB8JIoJmPuviq/QsozU\
      iU3kdDXA4RU9Ro+r4UrfZs zNA=
;; Query time: 0 msec
;; SERVER: 192.168.100.32#53(192.168.100.32)
;; WHEN: Mon Nov 25 00:36:51 2013
;; MSG SIZE  rcvd: 438
```

Kuva 5.2: DNSSEC-pyyntö omalle vyöhykkeelle.

tä selvittää DNSSEC-hierarkiaa, koska autoritäärinen nimipalvelin ei ole selvitin. Vastaus oli odotettu. Huomattavaa ovat rivit flags ja DNS-Extensio (EDNS). Kun selvittimen konfiguraatioon oli lisätty

```
trust-anchorfile = "/etc/unbound.fake/fakeroot.key"
```

sama pyyntö onnistui:

```
$ dig @127.0.0.1 SOA ns1.lan +dnssec
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> -t SOA ns1.lan +dnssec
[..]
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48502
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
```

## 5.2 Testitapaus 2. SSH-yhteys SSHFP:llä ja ilman

Testauksessa annettiin komento ensimmäisellä yhteyskerralla

```
# ssh -o VerifyHostKeyDNS=yes  ns1.lan.
The authenticity of host 'ns1.lan. (192.168.100.32)' can't be established.
RSA key fingerprint is c4:f1:26:80:af:d0:b9:3c:d6:96:46:13:2b:c3:a8:d0.
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)?
```

SSHFP löytyi DNS:stä ja käyttäjälle ilmoitettiin. Muuten virheilmoitus olisi ollut `No matching host key fingerprint found in DNS.`
SSH-asiakas valitettavasti tekee DNS-kyselyn siihen osoitenimeen, joka on annettu komennossa. Tämä osoite voi olla eri kuin mihin itse DNS-haku kohdistuu. Jos hakupolku on "search lan" niin `ssh ns1` hakee SSHFP tietuetta ns1:lle eikä ns1.lan:lle.

## 5.3 Testitapaus 3. TLSA selainlaajennoksella

TLSA-tietue voidaan luoda valmiilla työkaluilla. Jos kohde on verkkoyhteydessä, voidaan käyttää komentoa dane, joka tulee sshfp-ohjelmiston mukana:

```
$ dane --rfc --sha256 www.had-pilot.com
_443._tcp.www.had-pilot.com IN TLSA 1 1   \
  FA35F5C5E20CED93B6DDDE7E7F65F15D940EDA24886E0C26B035E09CB1609BD6
```

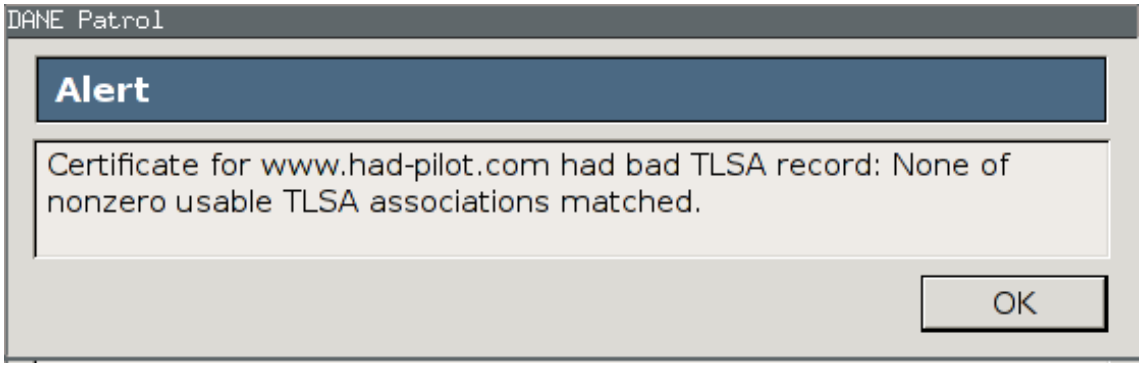
Jos varmenne on jo käsillä, niin dataosuus voidaan saada tlsa_rdata-ohjelmalla:

```
$ tlsa_rdata 3 0 1 varmenne.pem
3 0 1 2b0f672d2fbd5ffa27bf50534f6d92714973b5a19913d45c41f9712591ec3ef7
```

TLSA voidaan myös tehdä käsin. Olemassa olevasta varmenteesta saadaan tiiviste muuntamalla se ensin binääriseen muotoon (DER), ellei se jo ole siinä ja ottamalla tuloksesta SHA256 tiiviste. Alkupuoli TLSA-tietuetta on sama kuin yllä.

```
$ openssl x509 -outform DER -in www.lan.crt | openssl sha256
(stdin)= ab11c53ccec39de34a40ad22ee23dd5cd9172d7a256e88fb05e722...
```

Testissä käytettiin Chromium-selaimen laajennosta DANE Patrol [43], joka lisää ensimmäisellä yhteyskerralla varmenteen omaan luotettujen varmenteiden säiliöön. Kuvassa 5.3 näkyy yritys tarkistaa TLSA `https://www.had-pilot.com.` kohteesta. Verkosta käytettävä dane-tarkistin `https://check.sidnlabs.nl/dane/`, kertoi tapauksesta hiukan enemmän.




Kuva 5.3: DANE Patrolin antama varoitusviesti.

```
Something went wrong...
129.6.100.200 did not dane-validate, because: The TLSA record(s)
   did not match with the server certificate (chain)
2610:20:6005:100::200 did not dane-validate, because: The TLSA
   record(s) did not match with the server certificate (chain)
```

## 5.4 Testitapaus 4. SMIMEAn nouto Mutt-postiohjelmalla

Mutt on tekstipohjainen sähköpostiohjelma, johon on mahdollista ohjelmoida lisätoiminnallisuutta skripteillä. Muttissa on tuki sekä OpenPGP- että S/MIME-allekirjoitukselle ja -salaukselle. Tämän testin tarkoituksena oli tarkistaa S/MIME-allekirjoitettu sähköposti SMIMEAlla.

Ensin viestin digitaalisesta allekirjoituksesta eritellään allekirjoittajan sähköpostiosoite (kuva 5.4). Muuttujat %f ja %s viittaavat Muttin sisällä väliaikaisiin tiedostoihin. Tämä putkitetaan edelleen python-skriptille (kuva 5.5), joka koodaa sähköpostin base32-muotoon ja tuottaa SMIMEAn. Kuva 5.6 simuloi tätä muunnosta. Tulos kysytään nimipalvelusta ja ilmoitetaan käyttäjälle.

```
openssl smime -verify -in %f -noverify -signer %s 2>/dev/null \
  -out /dev/null; openssl  x509 -in %s -noout -email
```

Kuva 5.4: Sähköpostiosoitteen poimiminen S/MIME-viestistä.

```
#!/usr/bin/python
# smimea.py
# 20131206RI
# muuntaa STDINiin annetun merkkijonon SMIMEA-formaattiin
import base64, sys
# luetaan STDIN ja erotetaan LHS ja RHS @-merkin kohdalta
splitmail = sys.stdin.read().split('@')
# LHS base32-koodattuna SMIMEAn määrittelyin.
print base64.b32encode( splitmail[0] )+"_.smimecert." + splitmail[1]
```

Kuva 5.5: Sähköpostin muuntava python-skripti.

```
$ cat 'etunimi.sukunimi@tut.fi' | smimea.py
MV2HK3TJNVUS443VNN2W42LNNE======_.smime.tut.fi
```

Kuva 5.6: Sähköpostiosoitteen muokkaaminen SMIMEAn nimiosaksi.

Pohdintaa herätti se, mihin DNS-vyöhykkeeseen SMIMEA-tietue on tallennettu, sillä siinä on mukana vyöhykeosa. DNS:ssä nimiosan perässä oleva piste merkitsee, että nimeä ei laajenneta enää, mutta ilman pistettä nimiosan perään lisätään vyöhykeosuus. (Vedos-)RFC pakottaa MUA:n SMIMEA-toteutukselle tuen varmennekäytöille 0–3, käyttäen valitsinta 0 tai 1 ja yhteensopivuutta 1 (SHA256). SHA512-tuki on suositeltava, vaan ei pakollinen. Tämä prototyyppi luo ainoastaan SMIMEA-kyselyn eli vastauksen tulkintaa ei ole toteutettu. Muttiin kysely kiinnitetään määrittelemällä se muuttujan `smime_verify_opaque_command` arvoksi. Testin jatkaminen tästä eteenpäin vaikeutui huomattavasti, sillä nimipalvelin ei tukenut SMIMEA-tietueen lisäämistä. Tietue julkaistiin kuitenkin vaihtoehtoisella tavalla IANA:n yksityiskäyttöön varatulta alueelta RR-numeroa 65500. Varmennedatan pituus heksadesimaalina on 35 ja SMIMEAn rajoittekolmikko (3, 0, 1) ilmoitetaan heksadesimaalina 030001 datan edessä. Lopulliseksi riviksi DNS-palveluun muodostui

```
MRXHG3LBON2GK4TJ_.smimecert  IN  TYPE65500  \# 35   \
 0300013cd57cc9233685b28e0dddc7c61227328774af4e99a53b10aa7a5b4abcbd6c71
```

Testauksessa havaittiin joitain ongelmia. ZSK:llä on elinaika, mikä täytyy muistaa DNSSEC ylläpidossa. Kun vyöhykkeeseen yritettiin lisätä ulkopuolinen SSHFP (kaarne.cs.tut.fi), allekirjoitus onnistui, mutta nimipalvelin ei hyväksynyt tätä, kuten ei pitänytkään. SSH tarjosi ensin käytettäväksi ECDSA-avainta, jota ei oltu lisätty DNS:ään. Järjestystä muuttamalla RSA-avain löytyi. SSHFP:tä noudetaan eri osoitteelle, johon ssh kohdistetaan. Juurinimipalvelimen väärennös tehtiin omalla selvittimellä ja oli työläs. Muuten DNSSEC olisi toiminut vain joko paikallisesti tai ulkomaailmaan. SMIMEA varmenteen tarkistus oli työläs ja toteutus jäi kesken.

# 6. POHDINTAA DNS:N KÄYTÖSTÄ VARMENNEJULKAISUSSA

## 6.1 DNS-paketin koon vaikutukset

DNS-vastauksen koko kasvaa, kun siirrettävänä on mahdollisesti kokonaisia julkisia avaimia. Onko kasvu liian suuri DNS:n toiminnalle? Koska varmennedata ilman tiivistettä on suurempi kuin yksittäinen normaali DNS-vastaus, niin UDP-paketista voi hävitä matkalla osia siirtokerroksella. Havainto tapahtuu vasta ajastimen lauetessa, jolloin kysyjän täytyy lähettää pyyntö uudestaan, mikä hidastaa vasteaikoja. Ongelmia voi syntyä myös välimuistien täyttyessä nopeammin suurista tietueista. [29].

Ratkaisuksi käy DNS:n laajennosta Extension for DNS (EDNS), jolla voidaan laajentaa DNS-viestin kokoa alkuperäisestä 512 tavusta. Tutkimus [46, s. 9] vuodelta 2006 havaitsi, että 512 tavun koko saattoi aiheuttaa vastausviestin ennenaikaisen katkaisun. Kaikki selvittimet tutkimuksessa siirtyivät tällöin käyttämään TCP-protokollaa UDP:n sijaan ja kysely onnistui. Jos EDNS on käytössä nimipalvelimessa, UDP-puskurin oletuskoko on 4096. Se on voitu käsin pienentää 512 tavuun, jotta mikään palomuuri ei estäisi sen kulkua, koska alunperin DNS paketin kooksi oli määritelty 512 tavua. Toisaalta suureksi mainostettu DNS paketin koko nostaa palvelunestohyökkäyksen (DoS) vahvistuksen (amplification factor) 40:ään 4096 tavun DNS-pakettikoolla. UDP:n kanssa DoS on houkutteleva, sillä UDP:n lähdeosoite on helppo väärentää ja avoimia DNS-selvittimiä löytyy maailmalta. [47]. Todellisissa testeissä Vaughn mainitsee palvelunestohyökkäyksen vahvistuskertoimeksi jopa 60 [48], mutta tästä ei ole saatavilla tieteellistä julkaisua.

Itseaiheutettu epäsynkronisuus, mikä aiheutuu normaalisti vyöhykkeen siirrossa ensisijaiselta DNS-palvelimelta toissijaisille palvelimille, ei ole enää hyväksyttävää alkuperäisessä DNSSECissä. Tilannetta pahentaa vyöhykekoon kasvu. Korjauksen tai ainakin nopeutuksen tähän toteuttaa DNSSECbis ja sen DS. Seurauksena ylätason vyöhykkeelle tarvitaan enemmän dataa ja vyöhyketarkastukseen menee kaksi pyyntöä, mutta vyöhykesiirtoon kuluu vähemmän liikennettä ja vyöhykkeet pysyvät paremmin synkronissa keskenään. DNSSECin nykyisellä versiolla tarkoitetaan DNSSECbis:iä, mutta selkeyden vuoksi ”bis” jätetään usein pois nimestä.

## 6.2 Yksityisyys

DNS-vyöhykkeen sisäiset tiedot paljastuvat, jos käytössä on pelkkä NSEC. NSEC vihjaa, mitä tietoja vyöhykkeessä on, vaikka varsinaista kysyttyä tietoa ei löydykään, minkä avulla vyöhyke voidaan selata läpi, mitä usein ei haluta sallia. NSEC3 korjasi vuonna 2008 [49] tämän estämällä vyöhykkeen läpiselaamisen. NSEC3 ei paljasta liikaa tietoa vastatessaan tietueen poissaolosta. Tämä liittyy myös sähköpostiosoitteiden julkaisemiseen, jotta organisaation henkilötiedot eivät olisi vapaasti selailtavissa roskapostitukseen.

DNSSEC on tietoisesti suunniteltu salaamattomaksi [50, s. 16], joten pyynnöt kulkevat selkokielisinä. TLSA- ja SMIMEA-nimipalvelupyyntöjen kysely voi loukata kysyjän yksityisyyden suojaa. Tämän kiertämiseksi olisi käytettävä anonymisoivaa kerrosta. Kyselyjen salauksen tarjoavia protokollia ovat DNSCurve [45] ja DNSCrypt [51], jotka molemmat käyttävät elliptisten käyrien kryptografiaa.

## 6.3 Maa-vyöhykkeet, organisaatiovarmentajat ja luottamus

Voidaanko valtion hallinnassa olevaan TLD-vyöhykkeen allekirjoitukseen luottaa? Jos valtiollista TLD:tä pidetään luottamusankkurina, eikä juurinimipalvelimia ole varmistettu, maan hallinnon olisi helppoa valvoa kansalaisten tietoliikennettä. Jos kyselyt juurinimipalveluun ohjataan myös väärään kohteeseen, ei edes DNSSEC voi turvata yhteyksiä tapuksessa, jossa luottamusankkuria ei ole varmistettu.

Toisaalta, jos hallinto pystyy suoraan toimimaan luotettuna CA:na, se pystyy MITM:nä purkamaan yksityisten liikenteen yksinkertaisesti luomalla dynaamisesti halutun varmenteen, purkamalla liikenteen ja avaamalla yhteyden oikeaan kohteeseen TLS:llä. Tämä on mahdollista teknisesti ja useiden maiden laki myös sallii sen, jopa pakottaen CA:na toimivia yrityksiä avunantoon, kuten Soghoian ja Stamm toteavat aihetodisteissaan [52, s. 5-6]. Vaikka DNSSECiä voidaan käyttää varmentamaan koko ketju juuresta lehteen asti tai luottaa vain nimipalveluoperaattorin vastaukseen, voi joko operaattori tai maakohtainen TLD (ccTLD) välittää kysyjälle väärän juurinimipalvelinvarmenteen ja ohjata väärään juurinimipalvelimeen.

Ne organisaatiot, jotka purkavat ja uudelleensalaavat käyttäjiensä liikennettä, ovat voineet saada viralliselta juurivarmentajalta välivarmenteen, jolla luoda dynaamisesti varmenteita. Organisaatio on myös voinut sisällyttää oman varmentajansa käyttäjiensä luottamusankkureihin, mikä on helppoa, jos organisaatio ylläpitää käyttäjien laitteita. Esimerkiksi Trustwave-juurivarmentaja myönsi tehneensä yhden välivarmentajan organisaatiolle [53; 54, s. 5], minkä seurauksena Trustwave oli vaarassa poistua Mozillan tarjoamista luotetuista varmentajista [55]. Samoin kuin Comodon tapauksessa vain myönnetty välivarmentaja poistettiin Mozillan listoilta.

Selainvalmistaja Opera tarjosi mobiilikäyttöön optimoituja web-yhteyksiä Opera

Mini Mobile selaimessaan. Yhteys kiersi Operan palvelimen kautta, mutta purkiko yhtiö myös TLS-yhteyden vai välitti liikenteen vain eteenpäin? Nokia purki Gaurang Pandyan mukaan Asha-malliston puhelimessaan TLS-yhteyksiä alkuvuodesta 2013 [56]. Tämä korjattiin selainpäivityksen yhteydessä, mutta yhteydet kulkevat yhä puhelinvalmistajan palvelimen kautta, kuten Opera Mini Mobilessa.

## 6.4 Tekniikan kypsyys

DNSSECin käyttöönotto on ollut hidasta, mutta ilman aikaisia käyttöönottajia ei olisi löydetty parannusehdotuksia kuten NSEC3 ja DS. RFC-ehdotusten odottaminen standardeiksi asti ei aina kannata, sillä standardoiminen on hidasta. Sama ilmiö näkyy esimerkiksi WLAN-standardien (802.11x-protokollat) kehityksessä. Laitevalmistajat eivät aina odota protokollan valmistumista tuodessaan uusia laitemalleja markkinoille, vaan toteuttavat ominaisuudet kulloisenkin parhaan käsityksen mukaan. Tällä hetkellä tällaista edustavat RFC-dokumentit. Koska WLAN-laitteiden elinikä on lyhyt, kuluttaja hyötyy uutta ostaessaan uudemmasta tekniikasta, joka voi olla taaksepäin yhteensopiva.

# 7. JOHTOPÄÄTÖKSET

Tutkimuksessa esitettiin, että viralliset varmenteet eivät aina takaa turvaa, ja varmenteita voidaan julkaista DNS:ssä. Lisäksi tutkimus osoitti, että yhdistämällä varmennejakeluun DNS ja CA-hierarkiaan DNSSEC, nykyiset luottamusankkurin hakumenetelmät paranevat. Varmenteiden julkaisemiseen tarvittava hakemisto on oltava saatavilla, eheä ja autenttinen. Siihen sopii DNSSEC, mutta myös pelkkä DNS voi olla parannus nykyiseen käytäntöön, sillä vaikka se ei tarjoa eheyttä eikä autenttisuutta, tarjoaa se kuitenkin vaihtoehtoisen kanavan.

Varmennejulkaisuun liittyy tässä tutkimuksessa kaksi erillistä turvallisuusparannusta: varmenteiden julkaisu DNS:ssä DANEn ja SSHFPn avulla ja DNSSECin käyttö. Alkuoletus oli, että varmennejulkaisu DNS:ssä on vain yksi uusi tapa levittää varmenteita ja siirtää luottamuksen hallinta DNS:n hallitsijalle. Testit kuitenkin osoittavat, että pienellä vaivalla voidaan parantaa olemassa olevaa tapaa. Sekä DNSSEC, SSHFP että DANE täydentävät luottamusankkurin etsintämenetelmiä. Vaikka DNSSEC tuo eheyden ja autenttisuuden varmennejakeluun, ei sen ole tarkoitus kokonaan syrjäyttää olemassa olevia jakelujärjestelmiä, vaan täydentää niitä ja tuoda lisäturvaa sitä haluaville.

Koska DNSSEC on vielä vuonna 2013 harvinainen palvelimissa ja selvittimissä, sen käyttöönotto toisi kaikkein suurimman hyödyn. Yhteensopivuus DNS:n kanssa tukee tätä päätelmää. Helpointa on asentaa asiakaslaitteisiin selvitin, joka tukee DNSSECiä. Varmennejulkaisu DNS:ssä on syytä aloittaa palvelimen päästä, jotta asiakassovellukset pääsevät hyötymään siitä. Tekniikka on kokeiluvaiheessa varsinkin DANEn osalta ja sen standardointi onkin luonnosvaiheessa. Historia on kuitenkin osoittanut, että mitään standardia ei kannata jäädä odottamaan loputtomiin valmistumiseensa asti. DNSSEC on jo kypsä käyttöönotettavaksi.

Tutkimuksessa tuotettu makrosovellus SMIMEAn tarkistukseen on vain prototyyppi ja vaatisi paljon kehitystä ja muokkausta, kun SMIMEAn RFC:t muuttuvat. Esitetyt uudet tekniikat ovat selkeästi vielä testausvaiheessa, eikä niistä ole vielä tehty riittävästi tutkimusta. Toisaalta osa on vanhaa, paljon käytettyä ja valmista jo käyttöönotettavaksi. Jatkotutkimuksena voisi selvittää, miten varmenteita tuodaan DNS:ään ja kuka toimii varmenteen hallinnoijana. Onko se käyttäjä vai DNS:n hallinnoija? Millainen käytettävyys on käyttäjäpohjaisessa avainhallinnassa. DANEsta tullaan näkemään vielä tutkimuksia muidenkin PKI-sovellusten yhteydessä.

# LÄHTEET


[1] R. Housley, S. Ashmore ja C. Wallace. *Trust Anchor Format*. RFC 5914 (Proposed Standard). Internet Engineering Task Force, 2010. URL: http://www.ietf.org/rfc/rfc5914.txt (ks. s. V)

[2] L. Daigle, O. Kolkman ja IAB. *RFC Streams, Headers, and Boilerplates*. RFC 5741 (Informational). Internet Engineering Task Force, 2009. URL: http://www.ietf.org/rfc/rfc5741.txt (ks. s. VI)

[3] M. Burrows, M. Abadi ja R. M. Needham. "A Logic of Authentication". *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 426.1871 (1989), s. 233–271. DOI: 10.1098/rspa.1989.0125. URL: http://rspa.royalsocietypublishing.org/content/426/1871/233.full.pdf+html (ks. s. 1)

[4] Fanglu Guo, Jiawu Chen ja Tzi-cker Chiueh. "Spoof Detection for Preventing DoS Attacks against DNS Servers". Teoksessa: *26th IEEE International Conference on Distributed Computing Systems, 2006. ICDCS 2006.* 2006, s. 37. DOI: 10.1109/ICDCS.2006.78 (ks. s. 1, 11)

[5] Massimiliano Pala ja Scott A. Rea. "Usable trust anchor management". Teoksessa: *Proceedings of the 8th Symposium on Identity and Trust on the Internet*. IDtrust '09. Gaithersburg, Maryland: ACM, 2009, s. 61–72. ISBN: 978-1-60558-474-4. DOI: 10.1145/1527017.1527025. URL: http://doi.acm.org/10.1145/1527017.1527025 (ks. s. 2–4, 7)

[6] Gabor Toth ja Tjebbe Vlieg. "Public Key Pinning for TLS Using a Trust on First Use Model" (2013). Online. Viitattu 6.9.2013, s. 13. URL: http://rp.delaat.net/2012-2013/p56/report.pdf (ks. s. 2)

[7] Moxie Marlinspike. *Trust Assertions for Certificate Keys draft-perrin-tls-tack-02.txt*. Toim. T. Perrin. Online. Viitattu 20.10.2013. 2013. URL: http://tack.io/draft.html (ks. s. 2)

[8] *Convergence project*. Online. Viitattu 17.11.2013. URL: http://convergence.io/ (ks. s. 2, 6)

[9] *Perspectives Project*. Online. Viitattu 17.11.2013. URL: http://perspectives-project.org/ (ks. s. 2, 6)

[10] *The Sovereign Keys Project*. Online. Viitattu 17.11.2013. URL: https://www.eff.org/sovereign-keys (ks. s. 2, 6)

[11] S. Josefsson. *Storing Certificates in the Domain Name System (DNS)*. RFC 4398 (Proposed Standard). Updated by RFC 6944. Internet Engineering Task Force, 2006. URL: http://www.ietf.org/rfc/rfc4398.txt (ks. s. 2)

[12] Lasse Laukka. *DNSSEC-laboratorioharjoitus*. Kandidaatin työ. 2013 (ks. s. 2, 18)

[13] S. Kent. *Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*. RFC 1422 (Historic). Internet Engineering Task Force, 1993. URL: http://www.ietf.org/rfc/rfc1422.txt (ks. s. 3)

[14] T. Whalen ja K. Inkpen. "Gathering evidence: use of visual security cues in web browsers". Teoksessa: *Proc. of Graphics Interface 2005*. 2005, s. 137–144 (ks. s. 3)

[15] Devdatta Akhawe et al. "Here's my cert, so trust me, maybe?: understanding TLS errors on the web". Teoksessa: *Proceedings of the 22nd international conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013, s. 59–70. ISBN: 978-1-4503-2035-1. URL: http://dl.acm.org/citation.cfm?id=2488388.2488395 (ks. s. 3, 8, 9)

[16] Joshua Sunshine et al. "Crying Wolf: An Empirical Study of SSL Warning Effectiven". Teoksessa: *Proceedings of the 18th USENIX Security Symposium*. Montreal,Canada, 2009 (ks. s. 3, 8, 9)

[17] Jacob Appelbaum. *Detecting Certificate Authority compromises and web browser collusion*. Online. Viitattu 15.10.2013. 2011. URL: https://blog.torproject.org/blog/detecting-certificate-authority-compromises-and-web-browser-collusion (ks. s. 4, 7)

[18] VASCO. *DigiNotar reports security incident*. Online. Viitattu 15.10.2013. 2011. URL: http://www.vasco.com/company/about_vasco/press_room/news_archive/2011/news_diginotar_reports_security_incident.aspx (ks. s. 4, 7)

[19] CCITT. *The Directory - Authentication Framework*. Draft Recommendation X.509. Version 7. 1987 (ks. s. 5)

[20] Simson L. Garfinkel ja Robert C. Miller. "Johnny 2: a user test of key continuity management with S/MIME and Outlook Express". Teoksessa: *Proceedings of the 2005 symposium on Usable privacy and security*. SOUPS '05. Pittsburgh, Pennsylvania: ACM, 2005, s. 13–24. ISBN: 1-59593-178-3. DOI: 10.1145/1073001.1073003. URL: http://doi.acm.org/10.1145/1073001.1073003 (ks. s. 6, 8, 9)

[21] Dan Wendlandt, David G. Andersen ja Adrian Perrig. ”Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing”. Teoksessa: *2008 USENIX Annual Technical Conference*. Online. Viitattu 20.10.2013. USENIX Association, 2008. URL: http://perspectivessecurity.files.wordpress.com/2011/07/perspectives_usenix08.pdf (ks. s. 6)

[22] Ties de Kock. ”Using Secure DNS To Associate Historic Certificates With Domain Names For TLS”. Teoksessa: *2012 17th Twente Student Conference on IT*. Online. Viitattu 3.10.2013. Enschede, The Netherlands: University of Twente, 2012, s. 9. URL: http://referaat.cs.utwente.nl/conference/17/paper/7327/using-secure-dns-to-associate-historic-certificates-with-domain-names-for-tls.pdf (ks. s. 7)

[23] Google. *Certificate Transparency project*. Online. Viitattu 18.11.2013. URL: http://www.certificate-transparency.org/ (ks. s. 8)

[24] Eric Osterweil et al. ”Reducing the X. 509 Attack Surface with DNSSEC's DANE”. *SATIN: Securing and Trusting Internet Names* (2012), s. 8 (ks. s. 7, 13)

[25] Debian.org. *Debian paketti ca-certificates.deb*. URL: file:///usr/share/doc/ca-certificates/README.Debian (ks. s. 7)

[26] Jesse Burns ja Electronic Frontier Foundation (EFF). *Map of 650-odd organizations that function as Certificate Authorities.* Online. Viitattu 3.12.2013. URL: https://www.eff.org/files/colour_map_of_CAs.pdf (ks. s. 7)

[27] Ulrich Schroeter et al. *History of Risks & Threat Events to CAs and PKI*. Online. Viitattu 2.12.2013. URL: https://wiki.cacert.org/Risk/History (ks. s. 7)

[28] Cert.fi. *Tietoturvakatsaus 3b/2011*. Online. Viitattu 18.10.2013. 2011. URL: https://www.cert.fi/katsaukset/2011/tietoturvakatsaus3b2011.html (ks. s. 7)

[29] James M. Galvin. ”Public key distribution with secure DNS”. Teoksessa: *Proceedings of the 6th USENIX Security Symposium*. Online. Viitattu 20.9.2013. 1996, s. 161–170. URL: https://www.usenix.org/legacy/publications/library/proceedings/sec96/galvin.html (ks. s. 10, 23)

[30] Donald E. Eastlake 3rd. *Internet-Draft, The Kitchen Sink DNS Resource Record.* Online. Viitattu 22.11.2013. 1999. URL: http://tools.ietf.org/html/draft-ietf-dnsind-kitchen-sink-02 (ks. s. 10)

[31] *DNSSEC Deployment Maps*. Online. Viitattu 17.11.2013. 2013. URL: http://www.internetsociety.org/deploy360/dnssec/maps/ (ks. s. 12)

[32] Dan Kaminsky. *Phreebird*. Online. Viitattu 23.11.2013. URL: http://dankaminsky.com/phreebird/ (ks. s. 12)

[33] J. Schlyter ja W. Griffin. *Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*. RFC 4255 (Proposed Standard). Internet Engineering Task Force, 2006. URL: http://www.ietf.org/rfc/rfc4255.txt (ks. s. 12)

[34] *Domain Name System Security (DNSSEC) Algorithm Numbers*. Online. Viitattu 17.11.2013. 2013. URL: http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml (ks. s. 12)

[35] P. Hoffman ja J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. RFC 6698 (Proposed Standard). Internet Engineering Task Force, 2012. URL: http://www.ietf.org/rfc/rfc6698.txt (ks. s. 13)

[36] P Hoffman ja J. Schlyter. *RFC, draft. Using Secure DNS to Associate Certificates with Domain Names For S/MIME.* 2012 (ks. s. 13)

[37] Paul Wouters. *RFC, draft. Using DANE to Associate OpenPGP public keys with email addresses.* 2013 (ks. s. 13)

[38] Sophia Bergendahl ja Christoffer Holmstedt. *Use of DANE to improve the security for identity federations*. Bachelor's Thesis. 2012 (ks. s. 13)

[39] D. Cooper et al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280 (Proposed Standard). Updated by RFC 6818. Internet Engineering Task Force, 2008. URL: http://www.ietf.org/rfc/rfc5280.txt (ks. s. 14)

[40] *DANE workgroup mailinglist*. Online. Viitattu 6.12.2013. URL: https://www.ietf.org/mailman/listinfo/dane (ks. s. 14, 15)

[41] S. Josefsson. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648 (Proposed Standard). Internet Engineering Task Force, 2006. URL: http://www.ietf.org/rfc/rfc4648.txt (ks. s. 15)

[42] Viktor Dukhovni. *Using Secure DNS to Associate Certificates with Domain Names For S/MIME*. URL: http://list-archives.org/2013/07/09/dane-ietf-org/i-d-action-draft-ietf-dane-smime-02-txt/f/4545714318 (ks. s. 15)

[43] *Dane Patrol*. Online. Viitattu 29.11.2013. URL: https://labs.nic.cz/page/1207/dane-patrol/ (ks. s. 17, 21)

[44] *DNSSEC-Tools Releases*. Online. Viitattu 29.11.2013. URL: http://www.dnssec-tools.org/ (ks. s. 17)

[45] Dan Bernstein. *Introduction to DNSCurve*. Online. Viitattu 23.9.2013. 2013. URL: http://dnscurve.org/ (ks. s. 18, 24)

[46] Jelte Jansen. "Measuring the effects of DNSSEC deployment on query load" (2006). URL: http://www.nlnetlabs.nl/downloads/dnssec-effects.pdf (ks. s. 23)

[47] Marios Anagnostopoulos et al. "DNS amplification attack revisited". *Computers & Security* 39, Part B.0 (2013), s. 475–485 (ks. s. 23)

[48] Evron G. Vaughn R. *DNS amplification attacks (preliminary release)*. Online. Viitattu 21.9.2013. 2006. URL: http://www.isotf.org/news/DNS-Amplification-Attacks.pdf (ks. s. 23)

[49] B. Laurie et al. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*. RFC 5155 (Proposed Standard). Updated by RFCs 6840, 6944. Internet Engineering Task Force, 2008. URL: http://www.ietf.org/rfc/rfc5155.txt (ks. s. 24)

[50] R. Arends et al. *DNS Security Introduction and Requirements*. RFC 4033 (Proposed Standard). Updated by RFCs 6014, 6840. Internet Engineering Task Force, 2005. URL: http://www.ietf.org/rfc/rfc4033.txt (ks. s. 24)

[51] *Home page of DNSCrypt, a protocol to authenticate DNS traffic*. Online. Viitattu 23.9.2013. 2013. URL: http://dnscrypt.org/ (ks. s. 24)

[52] Christopher Soghoian ja Sid Stamm. "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL" (2010). URL: http://files.cloudprivacy.net/ssl-mitm.pdf (ks. s. 24)

[53] SpiderLabs Anterior. *Clarifying The Trustwave CA Policy Update*. Online. Viitattu 2.12.2013. URL: http://blog.spiderlabs.com/2012/02/clarifying-the-trustwave-ca-policy-update.html (ks. s. 24)

[54] Jeff Jarmoc. "SSL/TLS Interception Proxies and Transitive Trust". Teoksessa: *blackhat Europe 2012*. Online. Viitattu 25.11.2013. 2012. URL: https://media.blackhat.com/bh-eu-12/Jarmoc/bh-eu-12-Jarmoc-SSL_TLS_Interception-WP.pdf (ks. s. 24)

[55] Sebastian Wiesinger. *Bug 724929 - Remove Trustwave Certificate(s) from trusted root certificates.* Online. Viitattu 2.12.2013. URL: https://bugzilla.mozilla.org/show_bug.cgi?id=724929 (ks. s. 24)

[56] Gaurang Pandya. *Nokia's MITM on HTTPS traffic from their phone*. 2013. URL: http://gaurangkp.wordpress.com/2013/01/09/nokia-https-mitm/ (ks. s. 25)