Opinnäytetyö (AMK)

Liiketalouden koulutusohjelma

Tietoverkot ja Tietoturva

2016

Mikko Haapasaari

JULKISEN AVAIMEN INFRASTRUKTUURIN KÄYTTÖÖNOTTO PK-YRITYKSESSÄ

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Liiketalouden koulutusohjelma | Tietoverkot ja Tietoturva

2016 | 39

Jarkko Paavola

Mikko Haapasaari

JULKISEN AVAIMEN INFRASTRUKTUURIN KÄYTTÖÖNOTTO PK-YRITYKSESSÄ

Kesällä 2014 Liedon kunnan oppilasympäristössä päätettiin ottaa käyttöön kaksi uutta ympäristön hallintaa helpottavaa järjestelmää. Järjestelmien turvallinen ja sujuva käyttö vaati, että niillä on esittää niitä käyttävien ohjelmien ja laitteiden luottama X.509-varmenne. Tämän työn tavoitteena oli edellä mainittuun ympäristöön soveltuvan julkisen avaimen infrastruktuurin (PKI) suunnittelu ja rakentaminen, sekä sen aikana esille nousseiden kysymyksien ja ratkaisuvaihtoehtojen tutkiminen.

Työ tehtiin konstruktiivisella tutkimusotteella, jota pyrittiin peilaamaan myös työn rakenteessa. Tällä tavoiteltiin selkeyttä ja helppoa hyödynnettävyyttä työn lukijalle. Ensin tutkittiin PKI:n historiaa ja kehitystä, työn kannalta riittävän perustiedon saamiseksi. Tämän jälkeen tutkittiin niitä valintoja ja valintojen vaihtoehtoja, joita toimeksiantajan PKI-ympäristön käyttöönotossa tuli vastaan. Lopuksi käytiin läpi, mihin näistä vaihtoehdoista toimeksiantajan ympäristössä päädyttiin ja millä perustein.

Työn lopputuloksena syntyi toimeksiantajan vaatimukset täyttävä kaksitasoinen, Microsoft Active Directory Certificate Servicesillä toteutettu PKI-ympäristö. Palvelimien käyttöjärjestelmiksi valikoitui Windows Server 2012 R2 ja ne asennettiin toimeksiantajan olemassa olevaan virtuaaliympäristöön.

Työtä tehdessä esille nousi jokaisen organisaation yksilöllisten tarpeiden ja alan nopean kehityksen vaikutukset ympäristön rakenteen muodostumiseen. Niistä johtuen luotettavastakaan lähteestä saadun ohjeen suora seuraaminen ei automaattisesti takaa oikeanlaista lopputulosta. Tämän takia valintojen vaikutusten kattava tutkiminen koettiin PKI-ympäristön käyttöönotossa erittäin tärkeäksi.

ASIASANAT:

Varmenteet, tietoturva, salaus, infrastruktuuri

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Business | Data Networks and Information Security

2016 | 39

Jarkko Paavola

Mikko Haapasaari

DEPLOYING A PUBLIC KEY INFRASTRUCTURE IN A SME ENVIRONMENT

In the summer of 2014 the municipality of Lieto decided to utilize two new management systems for their educational environment. One of the requirements for these systems was that they needed to be able to present a trusted X.509 certificate to the programs and devices using them. The purpose of this thesis was to plan and deploy a suitable public key infrastructure (PKI) for this environment as well as to study the questions and different deployment options that were encountered during this process.

The research approach was constructive which has been reflected on the structure of the thesis in order to provide the information better future use. First the thesis studies the history and development of PKI in order to provide the reader with sufficient background knowledge of the subject. Then it covers the results of the research in to the different deployment options and issues that were encountered during the planning and deployment of the PKI. Finally, it presents the options that were chosen in this particular case and the reasons for the chosen options.

The result was a two-tier PKI relying on Microsoft Active Directory Certificate Services which met the requirements set for the environment. The servers were built using Windows Server 2012 R2 operating system, and were deployed to the existing virtual environment of the municipality.

During the writing of this thesis, the effects of organisation specific requirements as well as the fast speed of technological development became obvious. Therefore, when it comes to PKI deployment, following ready-made instructions from even the most reliable sources does not guarantee a suitable outcome and a proper study in to the effects of the choices made was found to be extremely important.

KEYWORDS:

Certificates, data security, encryption, infrastructure

SISÄLTÖ

KUVAT

1 JOHDANTO

Kesällä 2014 Liedon kunnan oppilasympäristössä päätettiin ottaa käyttöön kaksi uutta ympäristön hallintaa helpottavaa järjestelmää. Järjestelmien turvallinen ja sujuva käyttö vaatii, että niillä on esittää niitä käyttävien ohjelmien ja laitteiden luottama X.509-varmenne.

Julkisen avaimen infrastruktuuri (PKI) koostuu laitteistoista, ohjelmistoista, käytännöistä ja standardeista, joita tarvitaan varmenteiden luomiseen, jakamiseen, hallintaan ja sulkemiseen (Rouse 2014). Tapoja PKI:n toteuttamiseen on monia ja sen suunnitteluvaiheessa yrityksen tulee valita niistä itselleen sopivin kokonaisuus.

Koska varmenteiden rakentamiseen ja jakeluun ei oppilasympäristössä aiemmin ollut kiinnitetty huomiota, sain 17.9.2014 toimeksiannon ympäristöön sopivan ratkaisun suunnittelun ja rakentamisen. Työ on konstruktiivisella tutkimusotteella tehty, ja pyrin peilaamaan sitä myös työn rakenteessa. Samalla toivon saavuttavani sillä selkeyttä ja helppoa hyödynnettävyyttä työn lukijalle. Ensin tutkin PKI:n historiaa ja kehitystä työn kannalta riittävän perustiedon saamiseksi. Tämän jälkeen tutkin niitä valintoja ja valintojen vaihtoehtoja, joita toimeksiantajan PKI-ympäristön käyttöönotossa tuli vastaan. Tapaus-osiossa käyn läpi, mihin näistä vaihtoehdoista toimeksiantajan ympäristössä päädyttiin ja millä perustein.

1.1 Salassapito

Osa toimeksiantoa oli ympäristön rakentaminen ja valmiista ympäristöstä on luovutettu toimeksiantajalle sen tekniset yksityiskohdat kattava dokumentointi. Kyseistä dokumentointia ei kuitenkaan tietoturvasyistä julkaista osana tätä opinnäytetyötä. Samasta syystä itse ympäristön asennukseen liittyviä toimenpiteitä ei myöskään ole työssä kuvattu. PKI ei itsessään ole uusi asia, minkä takia erilaisia ohjeita sen rakentamiseen on saatavilla useista eri lähteistä. Tässä työssä ympäristön rakentamista käsittelevässä kappaleessa esitellään ne läh-

teet, jotka tapauksen suunnittelussa tehtyjen valintojen perusteella koettiin helposti ja luotettavasti sovellettaviksi.

1.2 Työn rajaus

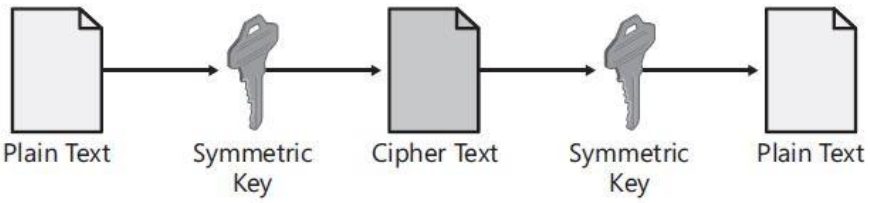
Omia varmentajia rakentaessa organisaation tulee ottaa huomioon sen tietoturvaan liittyvät kysymykset. Tässä työssä käsitellään vain suoraan varmenteisiin ja niiden hallintaan liittyviä tietoturvakysymyksiä, kuten avaimien pituuksia ja hallinnan jakamista mahdollistavia rooleja.

Työssä ei käsitellä yleisiä IT-palveluille yhteisiä tietoturvakysymyksiä, kuten alustapalvelimien koventamista, laitteiden fyysistä tietoturvaa, verkon tietoturvaa, riittävän käytettävyysasteen varmistamista tai varmuuskopiointia. Nämä asiat määräytyvät organisaation tietoturvapolitiikan sekä normaalien katastrofi- ja riskienhallintaprosessien mukaan.

2 PKI:N HISTORIA JA KEHITYS

2.1 Symmetrinen ja asymmetrinen salaus

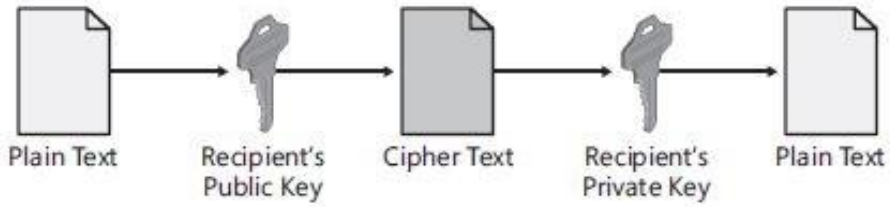
Symmetrisessä salauksessa tietoa salataan ja avataan samalla salausavaimella ja sen erilaisia muotoja on ollut olemassa jo tuhansia vuosia (Carlisle & Lloyd 2002, 7).



Kuva 1. Symmetrinen salaus (Komar 2008, 5).

Tämän takia kommunikaatiokäytössä symmetrisen salauksen ongelmaksi muodostuu siinä käytettävän avaimen turvallinen toimittaminen viestin vastaanottajalle (Carlisle & Lloyd 2002, 10). Tunnettuja symmetrisiä salausalgoritmeja ovat muun muassa DES, 3DES ja AES (Komar 2008, 8).

Vuonna 1976 Whitfield Diffie ja Martin Hellman julkaisivat ajatuksensa asymmetrisestä salauksesta. Toisin kuin symmetrisessä salauksessa, asymmetrisessä salauksessa tiedon salaamiseen ja avaamiseen käytetään avainparia yhden salausavaimen sijaan. Avainten luomiseen käytettävän matemaattisen kaavan takia tieto, joka on salattu yhdellä parin avaimella, voidaan avata vain parin toisella avaimella. Tämä mahdollistaa sen, että vastaanottaja voi laittaa avainparin toisen avaimen julkisesti saataville. Lähettäjä voi salata viestin käyttämällä vastaanottajan julkista avainta ja varmistua, että sisältö on avattavissa vain kyseisen avainparin toisella, vastaanottajan yksityisenä pitämällä avaimella. Tämän ominaisuutensa takia asymmetristä salausta alettiin kutsua termillä julkisen avaimen salaus (public-key cryptography). (Carlisle & Lloyd 2002, 10.)



Kuva 2. Asymmetrinen salaus (Komar 2008, 7).

Asymmetrinen salaus poistaa symmetrisessä salauksessa olevan avaimen toimittamiseen liittyvän ongelman, sillä viestin avaavaa avainta ei koskaan tarvitse erikseen toimittaa vastaanottajalle (Carlisle & Lloyd 2002, 10). Tunnettuja asymmetrisiä algoritmeja ovat muun muassa DH, RSA ja DSA (Komar 2008, 9).

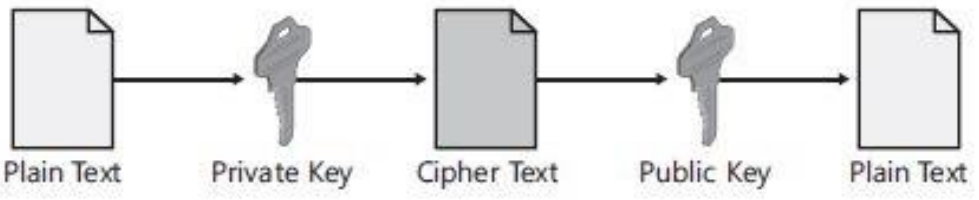
Asymmetrinen salaus on puolestaan huomattavasti symmetristä salausta hitaampaa. Tästä syystä suurin osa sovelluksista käyttääkin näitä molempia. Asymmetristä salausta käyttäen lähettäjä ja vastaanottaja pystyvät luotettavasti sopimaan yhteisestä avaimesta. Tätä yhteistä avainta voidaan sitten käyttää nopeampaan symmetriseen salaukseen, jolla itse välitettävä tieto suojataan. (Komar 2008, 9.)

Salauksen vahvuus määräytyy käytetyn algoritmin vahvuuden ja siinä käytettävän avaimen pituuden mukaan (Oracle 2010). Näillä on myös suuri vaikutus salauksen suorituskykyyn, eli siihen kuinka nopeasti salattua tietoa on mahdollista tuottaa tai avata. Esimerkiksi kun RSA algoritmia käytettäessä salausavaimen pituus tuplataan, salatun tiedon avaamiseen kuluva aika nousee yli kuusinkertaiseksi. (Coffey 2012.)

2.2 Allekirjoittaminen

Asymmetrisen salauksen käyttämällä avainparilla on mahdollista myös todentaa lähettäjän alkuperä ja varmistaa, ettei tieto ole muuttunut lähetyksen aikana. Tätä prosessia kutsutaan asymmetriseksi allekirjoittamiseksi. Allekirjoittaessa lähettäjä salaa viestin omalla yksityisellä avaimella. Vastaanottaja voi tällöin avata viestin lähettäjän julkisella avaimella ja varmistua, että viesti on todella

luotu vain lähettäjän hallussa olevalla yksityisellä avaimella. Jos avaaminen onnistuu, ei viesti ole myöskään muuttunut lähetyksessä, sillä pienikin muutos sisällössä aiheuttaisi avaamisprosessin epäonnistumisen. (Komar 2008, 8.)



Kuva 3. Allekirjoittaminen (Komar 2008, 8).

Suurin osa digitaalisia allekirjoituksia luovista sovelluksista, käyttävät sen tekemiseen sekä asymmetristä allekirjoitusta että hajautusfunktiota (Komar 2008, 11).

Hajautusfunktio (hash function) ottaa sille annetun syötteen ja laskee siitä tiivisteen. Samasta syötteestä samalla funktiolla laskettu tiiviste on aina sama. Jos syöte muuttuu vähänkin, myös siitä laskettu tiiviste muuttuu. Lisäksi vahvaa hajautusfunktioalgoritmia käyttäessä on erittäin epätodennäköistä, että kahdesta eri syötteestä muodostuu sama tiiviste. (Komar 2008, 11.)

Digitaalista allekirjoitusta luotaessa lähetettävä tieto annetaan ensin syötteeksi valitulle hajautusfunktiolle. Hajautusfunktio laskee tästä tiedosta tiivisteen. Tämän jälkeen lähettäjä salaa tiivisteen omalla yksityisellä avaimellaan. Tämän prosessin tulosta kutsutaan digitaaliseksi allekirjoitukseksi, jonka lähettäjä liittää lähetettävän tiedon mukaan. (Komar 2008, 12.)

Vastaanottaja voi puolestaan avata allekirjoituksen salauksen lähettäjän julkisella avaimella. Jos avaaminen onnistuu, on varmaa että allekirjoitus on tehty vain lähettäjän tiedossa olevalla yksityisellä avaimella. Tämän jälkeen vastaanottaja voi itse laskea samalla hajautusfunktiolla saapuneesta tiedosta tiivisteen ja verrata sitä lähettäjän laskemaan, allekirjoituksessa tulleeseen tiivisteeseen. Mikäli tulokset ovat samat, vastaanottaja voi olettaa, ettei tieto ole muuttunut sen allekirjoittamisen jälkeen. (Komar 2008, 12.)

2.3 Varmenteet

Asymmetrinen salaus mahdollistaa tiedon turvallisen välittämisen ilman tietoa suojaavan avaimen vaihtoa. Lisäksi digitaalisilla allekirjoituksilla voidaan varmistua, että tieto on todella peräisin julkisen avaimen tarjonneelta osapuolelta, eikä se ole muuttunut allekirjoittamisen jälkeen. Pelkän julkisen avaimen toimittaminen lähettäjälle ei kuitenkaan riitä. Lähettäjän tulee myös tietää, mitä algoritmia avaimen kanssa tulee käyttää, jotta viestin purkaminen vastaanottajalla onnistuu. Digitaalista allekirjoitusta käytettäessä, tulee sitä lukevan tahon myös tietää millä hajautusfunktiolla siihen sisällytetty tiiviste on laskettu, jotta tiivisteiden vertaaminen olisi mahdollista.

Yksi tapa julkisen avaimen ja sen käsittelyyn tarvittavien tietojen tarjoamiseen ovat varmenteet, joiden hallintaan termillä julkisen avaimen infrastruktuuri (PKI) viitataan (Rouse 2014). Kun lähettäjä tavalla tai toisella ilmaisee haluavansa kommunikoida turvallisesti, vastaanottaja lähettää hänelle oman varmenteensa, joka sisältää julkisen avaimen lisäksi sen käsittelyyn tarvittavat tiedot (Komar 2008, 21). Varmenteiden toinen tärkeä ominaisuus on pystyä todentamaan, että sen lähettäjä todella on, kuka hän väittää olevansa (Microsoft 2003). Varmenteen sisältö ja sen lähettäjän todentamiseen käytetty tapa vaihtelevat varmennetyypeittäin. Erilaisia varmennetyyppejä ovat muun muassa X.509, PGP ja SKIP. (Gerck 2000.)

2.4 X.509-varmennestandardi

X.509 on alun perin International Telecommunications Unionin (ITU-T) vuonna 1988 kehittämä varmennestandardi, joka kehitettiin osaksi X.500-standardissa määritettyjä hakemistopalveluita (ITU 1988; Rhee 2005, 223).

Kun vuonna 1994 Netscape Communications kehitti SSL-protokollan suojaamaan liikennettä internetissä, se valitsi X.509-standardin käytettäväksi sen kanssa (Hwang 2012; Comodo 2016). Kyseessä oli vielä keskeneräinen X.509-standardin kolmas versio, jonka ISO/IEC/ITU ja ANSI X9 kehittivät vastaamaan

varmenteiden uusien käyttötarkoituksien tuomia vaatimuksia (Rhee 2005, 223). Vuotta myöhemmin Internet Engineering Taskforce (IETF) kiinnitti huomionsa näiden varmenteiden tärkeyteen ja perusti Public-Key Infrastructure X.509 (PKIX) -työryhmän kehittämään standardeja, jotka tukisivat niiden käyttöä internetissä (Oppliger 2009, 211; IETF 2016).

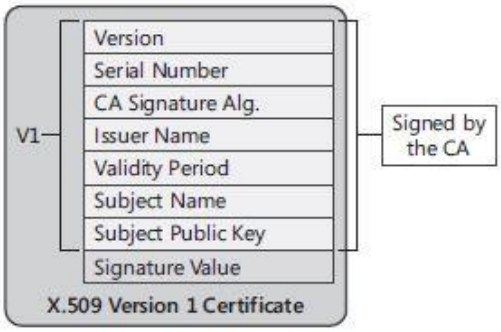
X.509 lähestyy varmenteen lähettäjän todentamista hierarkkisesti. Varmenteiden myöntämisestä vastaavat varmentajat (Certificate Authority, CA), jotka lisäävät niihin digitaalisen allekirjoituksensa (Komar 2008, 21). Kun ohjelma vastaanottaa X.509-varmenteen, se tarkistaa siitä kuka varmenteen on allekirjoittanut ja mistä allekirjoittajan varmenne on noudettavissa. Noutopiste on merkittynä varmenteen Authority Information Access (AIA) -kentässä. Tämän jälkeen ohjelma noutaa allekirjoittajan varmenteen ja toistaa edellä mainitun prosessin. Prosessia toistetaan niin kauan, kunnes noudettu varmenne on allekirjoitettu siinä määritetyn tahon omasta toimesta. Ketjun viimeistä, varmenteensa itse allekirjoittavaa varmentajaa kutsutaan juurivarmentajaksi (Root CA). (Komar 2008, 235.) Tyypillisessä varmenneketjussa on kahdesta neljään varmentajaa (Komar 2008, 73).

Kun ohjelma on kerännyt ketjun kaikki varmenteet, se aloittaa allekirjoituksien tarkistamisen. Se avaa ketjun ensimmäisen varmenteen allekirjoituksen noutamallaan allekirjoittajan varmenteessa olevalla julkisella avaimella ja vertaa siinä olevaa tiivistettä sen varmenteesta itse laskemaan tiivisteeseen. Jos allekirjoitus aukeaa ja tiivisteet vastaavat, voi ohjelma todeta, että varmenteen on todella allekirjoittanut siinä ilmoitettu taho, eikä se ole muuttunut allekirjoituksen jälkeen. Tämän jälkeen prosessi toistetaan kaikissa allekirjoittajien varmenteissa, aina juurivarmentajaan asti. (Komar 2008, 235.)

Varmenteita käyttäviin ohjelmiin kuten selaimiin ja käyttöjärjestelmiin on ohjelmistovalmistajan toimesta määritetty luotettujen juurivarmentajien säiliö. Tähän säiliöön lisätään kaikkien niiden juurivarmentajien varmenteet, joihin ohjelma saa luottaa. Kun allekirjoituksien tarkistus etenee juurivarmentajan varmenteeseen, tarkistaa ohjelma ensin, löytyykö vastaava varmenne sen luotettujen var-

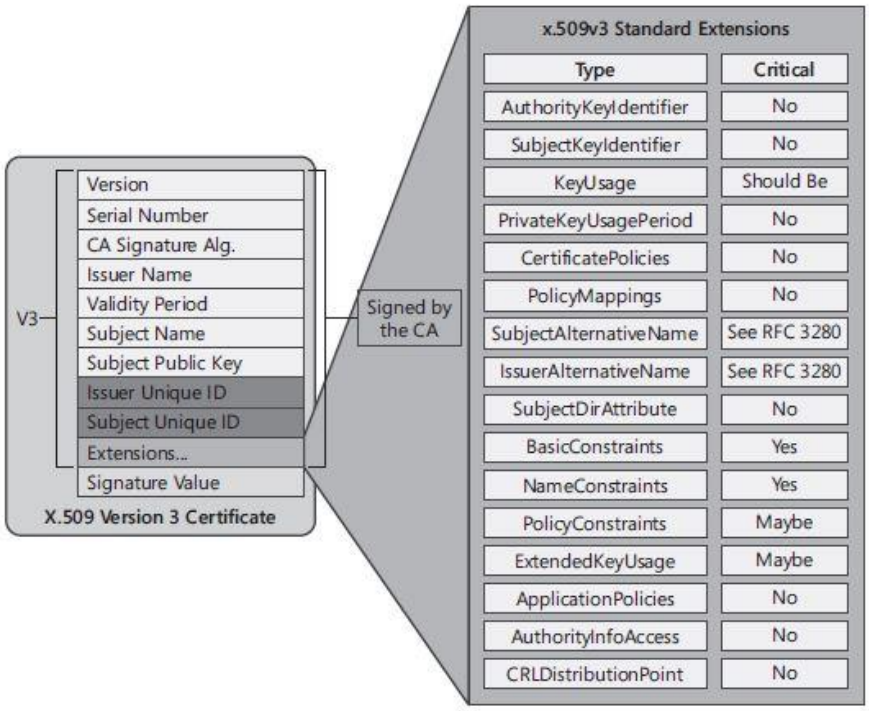
mentajien säiliöstä. Jos löytyy, tarkistaa ohjelma sillä ketjua ladatessaan saaneensa juurivarmenteen allekirjoituksen. (Komar 2008, 31.)

Alkuperäisen vuonna 1988 määritetyn X.509-standardin mukaan varmenteista tarkistetaan allekirjoituksien lisäksi ainoastaan niiden voimassaolo, joka on merkitty varmenteen Validity Period -kenttään. Ohjelma luottaa sille tarjottuun varmenteeseen, jos kaikkien siihen liitetyn ketjun varmenteiden allekirjoituksien varmentaminen onnistuu, juurivarmentajan varmenne löytyy luotettujen varmentajien säilöstä ja kaikki varmenteet ovat tarkistushetkellä voimassa. (Komar 2008, 22.)



Kuva 4. X.509-varmenteen ensimmäinen versio (Komar 2008, 22).

Vuonna 1996 valmistunut varmenteen kolmas versio toi mukanaan useita muutoksia. Yksi niistä oli mahdollisuus varmenteen kumoamiseen (revocate) ennen sen voimassaolon päättymistä (Komar 2008, 28). Tämä ominaisuus on tärkeää, jos esimerkiksi varmenteeseen sidottu yksityisen avaimen epäillään vaarantuneen (Komar 2008, 208). Yksi tapa kumoamisen toteuttamiseen on käyttää sulkulistoja (Certificate Revocation List, CRL), johon kaikki kumotut varmenteet ovat merkitty. Jokaisen myönnetyn varmenteen CRLDistributionPoint attribuuttiin voidaan tällöin määrittää URL-muotoinen jakelupiste, josta sulkulista on noudettavissa. Varmenteen luotettavuutta tarkistaessa ohjelma hakee ajantasaisen sulkulistan jakelupisteestä ja varmistaa, ettei varmennetta ole siinä merkitty kumotuksi. (Komar 2008, 28.)



Kuva 5. X.509-varmenteen kolmas versio (Komar 2008, 22).

Internetiin kytkettyjen laitteiden räjähdysmäisen kasvun myötä PKIX-työryhmän luomat standardit hyväksyttiin alalla yleiseen käyttöön. Tästä syystä nykyisin X.509-varmenteilla viitataankin yleensä alkuperäisen ITU-T:n standardin sijaan tähän PKIX-työryhmän luomaan varmennestandardiin (Harvey 2011). Sen ajantasaisin kuvaus löytyy dokumentista RFC 5280 (IETF 2008).

3 PKI YRITYKSESSÄ

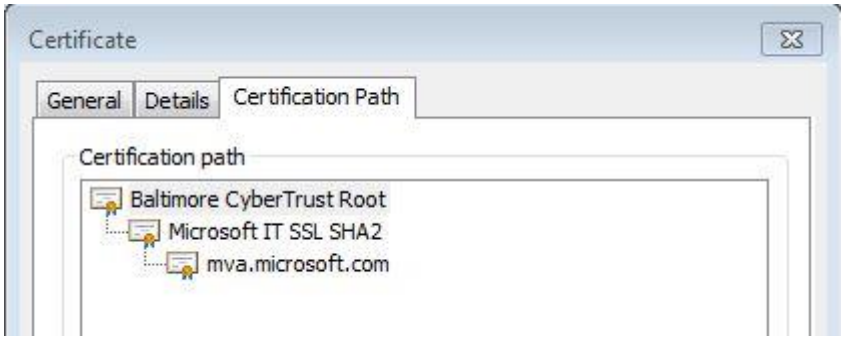
3.1 Luottamusmallin valinta

Kun varmenteiden käyttö tulee organisaatiossa ajankohtaiseksi, yksi ensimmäisistä valinnoista on päättää, tuleeko niiden olla julkisesti luotettuja (Microsoft 2012, 6-9).

Varmenteita on mahdollista ostaa joltain markkinoilla olevista julkisesti luotetuista varmentajista. Niistä maksetaan yleensä varmennekohtaisesti, vaikka muitakin maksumalleja varsinkin suuremmille asiakkaille on olemassa. Näiden varmenteiden suurin etu on hallinnallinen helppous (Microsoft 2012, 6-9). Ne eivät vaadi oman ympäristön rakentamista tai varmenteiden erillistä jakelua, sillä niiden juurivarmentajien varmenteet ovat jo valmiiksi useimpien käyttöjärjestelmien ja selainten luottamia. Haitaksi puolestaan varsinkin useampaa varmennetta tarvittaessa muodostuu hinta. (Microsoft 2012, 6-9.)

Organisaatiolle voidaan myös luoda oma sisäinen varmentaja tai varmentajien hierarkia (Microsoft 2012, 6-9). Vaihtoehtoisesti jotkut varmenteita tarvitsevat palvelut ja kolmannen osapuolen sovellukset mahdollistavat itse allekirjoitettujen varmenteiden luomisen palvelukohtaisesti, jotka toimivat juurivarmenteiden tavoin. Tämän tyyppiset varmenteet ovat yksityisiä ja niiden jakelu niitä käyttävien ohjelmien luotettujen varmentajien varmennesäiliöihin pitää tavalla tai toisella ottaa huomioon (Microsoft 2012, 6-9). Jakelusta muodostuu hankalaa tilanteissa, joissa varmenteita lukevat ohjelmat eivät ole organisaation omassa hallinnassa. Mallin selkein etu on kuitenkin sen hinta, sillä itse varmenteista ei muodostu organisaatiolle kustannuksia (Microsoft 2012, 6-9).

Jotkut yritykset ovat ottaneet käyttöönsä niin sanotun hybrid-mallin. Tässä mallissa yrityksen sisäisen varmentajan varmenne on allekirjoitettu julkisesti luotetun varmentajan toimesta. Siinä yhdistyvät julkisen ja sisäisen varmentajan edut, sillä tällöin organisaation itse myöntämät ja hallinnoimat varmenteet ovat julkisesti luotettuja. (Microsoft 2012, 6-9.)



Kuva 6. Microsoftin hybrid-mallin varmenne.

Julkisesti luotetut varmenteet sisäisille varmentajille ovat kuitenkin hyvin kalliita (Microsoft 2012, 6-9). Lisäksi koska niihin luottavat automaattisesti myös organisaation ulkopuoliset laitteet, täytyy sellaisen hankkimista suunnittelevan organisaation pystyä täyttämään julkisille varmentajille asetetut vaatimukset (GlobalSign 2016a). Näistä syistä ne ovat käytännössä vain erittäin suurien organisaatioiden saatavilla.

3.2 Toteutusvaihtoehdot

Jos sisäisen varmentajan rakentamiseen päädytään, tulee ensimmäiseksi eteen arkkitehtuurilliset kysymykset. Markkinoilla on saatavilla useita erilaisia sisäisen varmentajan ja siihen liittyvien palveluiden rakentamisen vaihtoehtoja, joissa kussakin on omat etunsa ja haittansa.

Microsoft Active Directory Certificate Services (AD CS)

Microsoftin AD CS on globaalisti käytetyin PKI-ratkaisu ja se tulee Microsoft Server käyttöjärjestelmien mukana. Palvelu on täysin integroitavissa Microsoftin aktiivihakemistoon (Active Directory, AD), joka mahdollistaa muun muassa varmenteiden automaattisen jakelun aktiivihakemistoon liitettyihin työasemiin ja palvelimiin. (Komar 2008, xxv.)

AD CS -palvelun hyödyntäminen on integroituna myös moniin muihin Microsoftin palveluihin, kuten System Centeriin, Internet Information Servicesiin ja Identity Manageriin.

Windows Server 2008 toi myös mukanaan Cryptography Next Generation -ohjelmointirajapinnan, joka mahdollistaa organisaatioiden omien algoritmien käyttämisen varmenteiden kanssa (Komar 2008, xxv).

Microsoft Windows Server -käyttöjärjestelmä on maksullinen. Mikäli organisaatiolla ei ole olemassa olevia sopimuksia Microsoftin kanssa, PKI:n rakentaminen AD CS:n kanssa voi maksaa toteutustavasta riippuen pelkästään lisenssimaksuina useita satoja euroja. (Microsoft 2016a.)

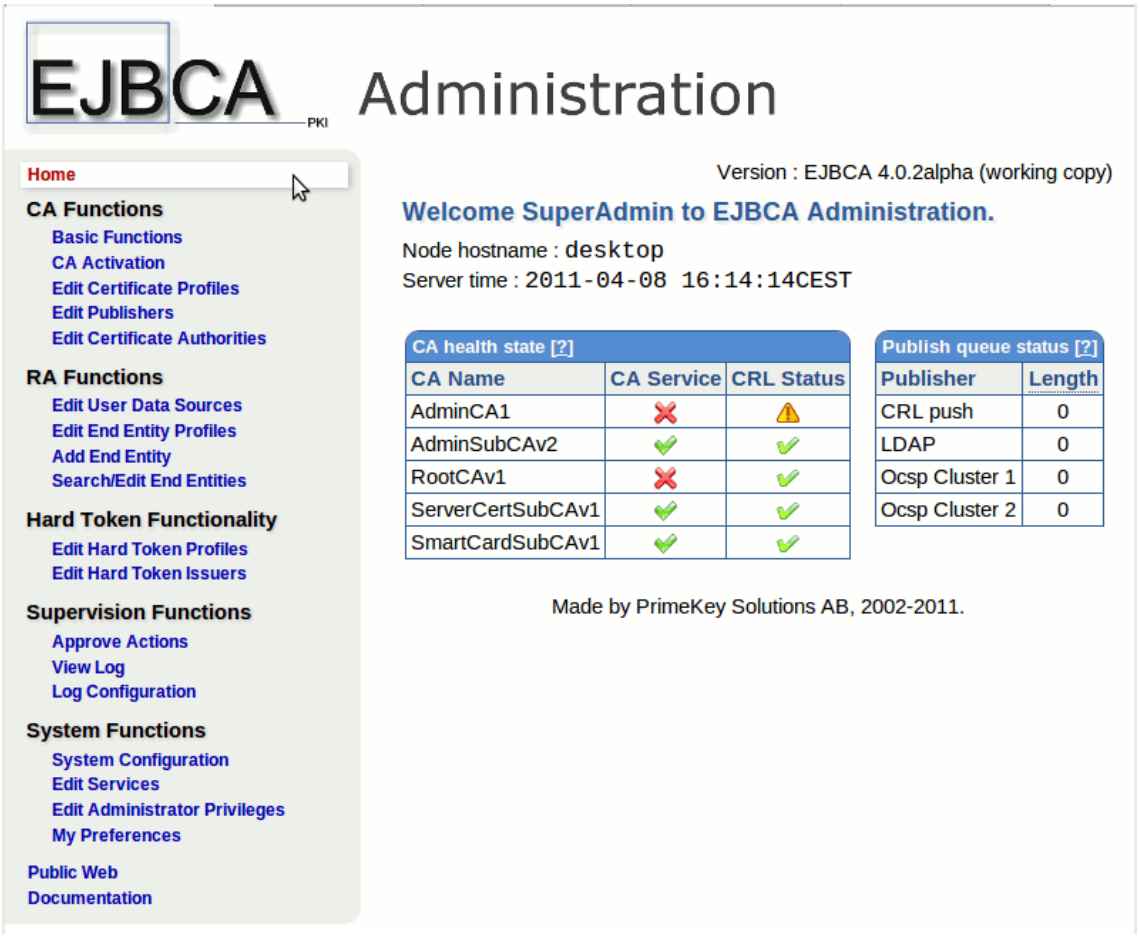
Microsoftin tarjoaa tuotteilleen tukea globaalisti ja sitä on saatavilla myös suomeksi (Microsoft 2016b). Suoran tuotetuen lisäksi sen maailmanlaajuisen kumppaniverkoston kautta on mahdollista hankkia erilaisia asennus ja ylläpitopalveluita ja pelkästään Suomessa näitä kumppaneita on satoja (Microsoft 2016c).

Enterprise Java Bean Certificate Authority (EJBCA)

EJBCA on ainakin joidenkin lähteiden mukaan suosituin suuryrityskäyttöön soveltuva avoimen lähdekoodin PKI-ratkaisu (Schwier 2014; Gustavsson 2016). Sen suunnittelun lähtökohtina on ollut muun muassa suorituskyky ja skaalautuvuus. Koska kyseessä on nimenomaan suuryrityskäyttöön suunniteltu ratkaisu, sen käyttöönotto saattaa tuntua turhan raskaalta vain muutamaa varmennetta tarvittaessa. (EJBCA 2016.)

EJBCA tukee kaupallisista PKI-ratkaisusta tuttuja ja suurempien ympäristöjen tarvitsemia ominaisuuksia, kuten varmenteiden automaattista jakelua Windows ja Linux -ympäristöissä. AD CS:n tavoin sillä on myös mahdollista julkaista varmenteita ja sulkulistoja suoraan Microsoftin aktiivihakemistoon. Se on kirjoitettu Javalla, jonka takia se on asennettavissa mille tahansa Java sovelluspalvelimia tukevalle käyttöjärjestelmälle (EJBCA 2016.)

EJBCA:n mukana tulee myös monipuolinen käyttöliittymä, jolla ympäristöä on mahdollista hallita (EJBCA 2016).



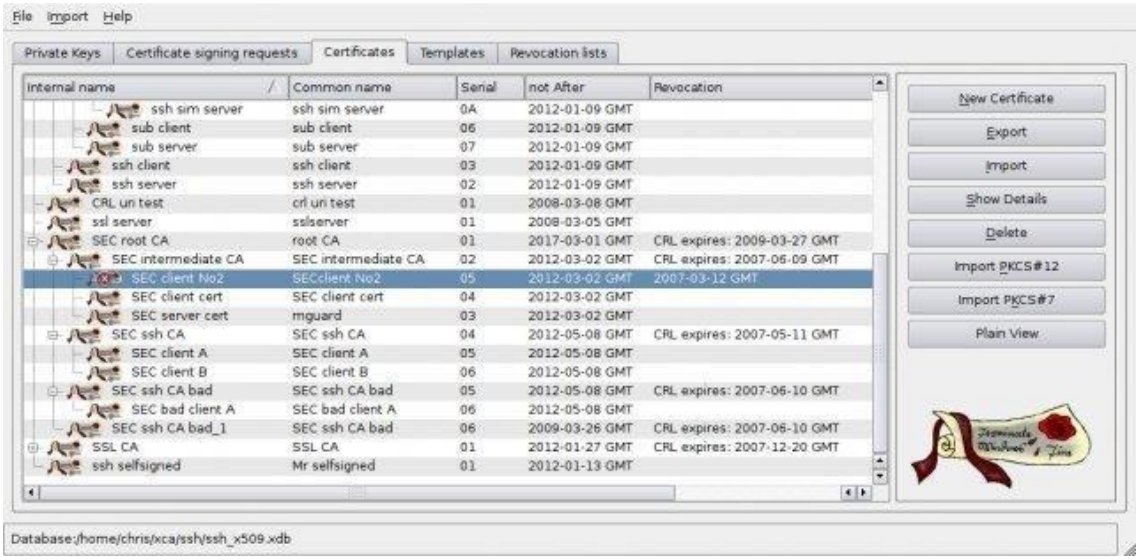
Kuva 7. EJBCA-käyttöliittymä (EJBCA 2016).

EJBCA on avoimen lähdekoodinsa takia ilmainen, mutta suurimman osan sen koodista omistaa ruotsalainen Primekey AB. Primekey pitää oikeudet EJBCA:n kaupallisiin käyttötarkoituksiin ja siltä on ostettavissa siihen ohjelmistoon tukea, integrointi- ja kehitysapua sekä konsultointia. Primekeyllä ei kuitenkaan sen verkkosivujen perusteella ole toimitiloja tai kumppaneita Suomessa. (Primekey 2016.)

OpenSSL ja XCA

Toinen tunnettu avoimen lähdekoodin ratkaisu on OpenSSL, joka perustuu Eric Youngin ja Tim Hudsonin SSLeay-kirjastoon, ja jonka jatkokehityksestä vastaa OpenSSL-yhteisö. Se on avoimen lähdekoodinsa ansiosta erittäin muokattavissa ja asennettavissa muun muassa Linux, Windows ja OS/2 alustoille. (OpenSSL 2016.)

OpenSSL ei pidä sisällä käyttöliittymää. Se ei myöskään tarjoa valmiita ominaisuuksia esimerkiksi varmenteiden automaattiseen jakeluun tai niiden julkaisuun aktiivihakemistossa, vaan vastaava toiminnallisuus tulee rakentaa itse. (OpenSSL 2016.) Avoimen lähdekoodinsa ansiosta OpenSSL on synnyttänyt muita projekteja kuten X Certificate and Key Managementin (XCA), joka tarjoaa käyttöliittymää OpenSSL-kirjastojen käyttöön (XCA 2015).



Kuva 8. XCA-käyttöliittymä (XCA 2015).

XCA on pieni, selkeä ja helposti asennettava käyttöliittymällinen ohjelma varmenteiden luontiin. Koska XCA on tarkoitettu pienen ympäristön varmentajaksi, sen ominaisuudet rajoittuvat lähinnä varmenteiden ja sulkulistojen helppoon luontiin ja seurantaan. Se ei tuo mukanaan isompien ympäristöjen kaipaamia lisäominaisuuksia, kuten varmenteiden automaattista jakelua. (XCA 2015.)

3.3 Varmentajien roolit ja hierarkia

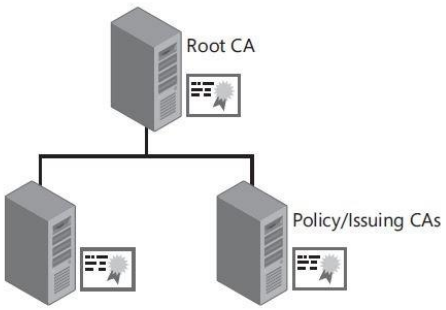
Varmentajien roolit määräytyvät niiden tehtävän ja paikan mukaan hierarkiassa. Useimmissa varmentajien hierarkioissa on kahdesta neljään tasoa, mutta yksitasoinenkin malli saattaa sopia pienempiin organisaatioihin. (Komar 2008, 73.) Vaikka tasojen määrällä ei olekaan rajoitusta, yli neljän tason käyttäminen ei ole suositeltua (Komar 2008, 76).

Yksitasoinen malli

Yksitasoisessa mallissa on vain juurivarmentaja, joka toimii samalla myöntäjävarmentajana. Se on yksinkertainen rakentaa ja helppo hallita. Yksitasoisen mallin ongelmia ovat vikasietoisuuden ja skaalautuvuuden puute. Jos juurivarmentaja ei ole saatavilla, eivät varmennepalvelut ole käytettävissä. Koska juurivarmentaja rakennetaan kokoajan päällä olevaksi ja verkkoon sekä mahdollisesti toimialueeseen liitetyksi, mallia on hankala muuttaa palveluiden käyttöasteen kasvaessa. (Komar 2008, 74.) Tämän takia sitä käytetäänkin lähinnä pienissä, alle kolmensadan käyttäjän ympäristöissä (Komar 2008, 73).

Kaksitasoinen malli

Kaksitasoisessa mallissa ylimpänä on juurivarmentaja, jonka alla on yksi tai useampi myöntäjävarmentaja (Issuing CA) (Komar 2008, 74).



Kuva 9. Kaksitasoinen hierarkia (Komar 2008, 74).

Kaksitasoisessa mallissa juurivarmentaja allekirjoittaa vain myöntäjävarmentajien varmenteet, jotka puolestaan hoitavat palveluihin tai laitteisiin liitettyjen varmenteiden allekirjoittamisen (Komar 2008, 74). Koska uusia myöntäjävarmentajia tarvitaan todella harvoin, voidaan niiden varmenteiden siirto hoitaa esimerkiksi ulkoista mediaa apuna käyttäen (Komar 2008, 128). Tämän takia juurivarmentajaa ei koskaan tarvitse kytkeä verkkoon tai liittää toimialueelle ja palvelinta on mahdollista säilyttää sammutettuna. Jos juurivarmentajan fyysinen tietoturva on vahva, tekee tämä sen yksityisen avaimen varastamisesta erittäin hankalaa. (Komar 2008, 74.)

Kaksitasoinen malli kasvattaa ympäristön tietoturvaa, sillä jos yhden myöntäjävarmentajan yksityisen avaimen epäillään vaarantuneen, vain sen oma ja sen myöntämät varmenteet menettävät luotettavuuden. Luotettavuuden menettäneet varmenteet on mahdollista kumota ja palveluille voidaan myöntää uudet varmenteet joltain edelleen luotettavista myöntäjävarmentajista. On myös aina mahdollista rakentaa uusia myöntäjävarmentajia olemassa olevien rinnalle, jolloin tilanne saadaan takaisin normaaliksi. Jos puolestaan juurivarmentajan yksityinen avain vaarantuu, ei yhteenkään ympäristön varmenteeseen voida luottaa. (Microsoft 2016d.)

Kaksitasoisella mallilla on myös mahdollista rakentaa vikasietoisuutta ja parantaa palveluiden saatavuutta. Useampi varmentaja on mahdollista asettaa jakamaan samanlaisia varmenteita, jolloin niiden jakelu jatkuu vaikka jokin varmentajista vikaantuisikin. (Komar 2008, 75.) Päinvastoin jakamalla tietyillä varmentajilla vain tietyn tyyppisiä varmenteita, voidaan pienentää niihin kohdistuvaa kuormaa (Komar 2008, 78). Maantieteellisesti suurille alueille jakaantuneissa organisaatioissa, saatavuutta voidaan parantaa tuomalla varmentajia fyysisesti lähemmäksi niitä tarvitsevia laitteita. (Komar 2008, 77).

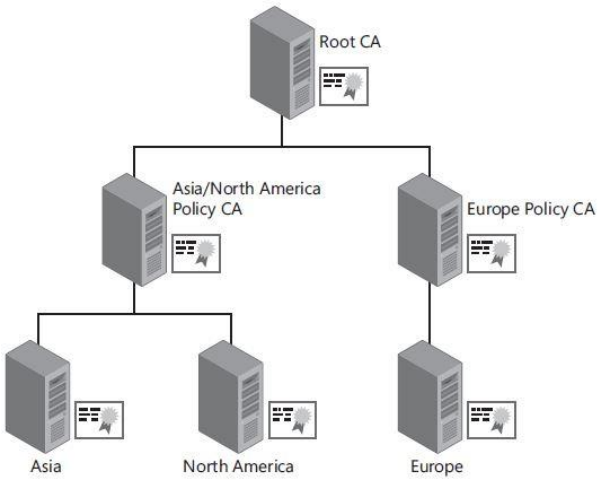
Malli mahdollistaa myös hallinnan jakauttamisen ja erilaisten käytäntöjen pakottamisen, joka saattaa olla tarpeen varsinkin suuremmissa organisaatioissa tietoturvan tai organisaatiorakenteen takia. Esimerkiksi työasemiin varmenteita myöntävän varmentajan hallinta voidaan käyttöoikeuksin rajoittaa työasematiimille ja palvelinten varmentajan hallinta palvelintiimille. (Komar 2008, 77.)

Myöntäjävarmentajan varmenteeseen on myös mahdollista asettaa rajoitteita, jolloin sillä voidaan allekirjoittaa vain tietyntyyppisiä varmenteita (Microsoft 2016e).

Varmentajalle on myös mahdollista määrittää erilaisia varmennepoliitikkoja (Certificate Policy, CP) ja varmennekäytäntöjen lausumia (Certificate Practice Statement, CSP), joissa ympäristön suojaamiseen ja varmennettavan tunnistamiseen liittyviä poliitikkoja ja käytäntöjä kuvataan. Näitä käytäntöjä ja poliitikkoja määrittäviä varmentajia kutsutaan käytäntövarmentajaksi (Policy CA). (Komar 2008, 32.)

Kolmetasoiset ja sitä suuremmat mallit

Hierarkiaan on mahdollista lisätä tasoja juuri- ja myöntäjävarmentajien väliin. Yksi syy tason lisäämiseksi voi olla erillisen käytäntövarmentajien tason tarve hierarkiassa. (Komar 2008, 76; Microsoft 2016e.) Tällöin käytäntövarmentajat myöntävät varmenteita vain myöntäjävarmentajille ja kunkin myöntäjävarmentajan tulee noudattaa omalla käytäntövarmentajallaan määritettyjä poliitikkoja ja käytäntöjä (Komar 2008, 31).



Kuva 10. Kolmitasoinen hierarkia (Komar 2008, 75).

Tämä saattaa olla tarpeen esimerkiksi erittäin suurissa ympäristöissä, joissa myöntäjien suojaamiseen ja varmennettavan tunnistamiseen kohdistuu erilaisia vaatimuksia esimerkiksi organisaation rakenteesta tai voimassa olevasta lainsäädännöstä johtuen (Komar 2008, 76). Tason lisääminen saattaa olla tarpeen myös hallinnan riittäväksi jakauttamiseksi (Komar 2008, 76; Microsoft 2016e).

Kuten juurivarmentajaa, myös käytäntövarmentajia voidaan säilyttää suljettuina, mikä lisää ympäristön tietoturvaa (Komar 2008, 75; Microsoft 2016e).

3.4 Algoritmit, avainten pituudet ja eliniät

Edistysaskeleet kryptoanalyysissä ja tietokoneiden laskentatehon jatkuva kasvu luovat tarpeen algoritmien parantamiselle ja avainten pituuksien kasvattamiselle (GlobalSign 2016b). PKI-ratkaisua suunnitellessa organisaation hyvä tutkia, mitkä sillä hetkellä olevat algoritmien ja avaintenpituuksien yhdistelmät voidaan luokitella turvallisiksi. Alan yleisten suositusten lisäksi valintoihin voi vaikuttaa myös muut tekijät.

Yhteensopivuus ja suorituskyky

Kaikki ohjelmat tai laitteet eivät tue kaikkia algoritmeja ja avainten pituuksia. Esimerkiksi Cisco 3000 -sarjan VPN-keskittimet ja versiota 1.5 vanhemmat Java ohjelmat eivät tue yli 2048-bittisiä avaimia. (Komar 2008, 89.) Myös monilla älykorteilla ja niiden lukijoilla on ongelmia pidempien avaintenpituuksien kanssa (Pocock 2013). Algoritmilla ja avaimen pituudella on myös vaikutusta suorituskykyyn, joka saattaa kasvaa merkittäväksi varsinkin suuria tietomääriä salattaessa (Coffey 2012).

Alaan tai käyttötarkoitukseen sidotut vaatimukset

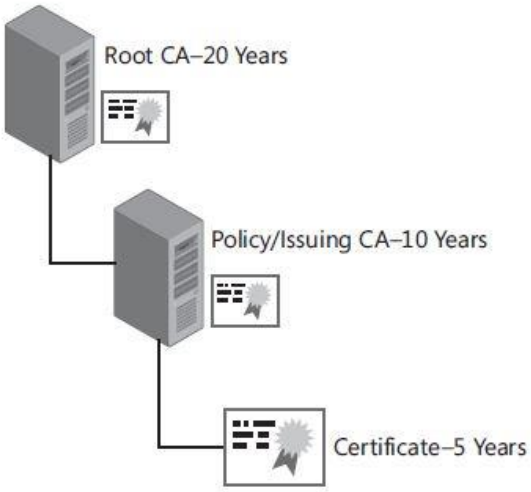
Algoritmeja ja avainten pituuksia valittaessa organisaation tulee myös tunnistaa mahdolliset alaan tai käyttötarkoitukseen sidotut vaatimukset. Esimerkiksi

Payment Card Industry Data Security Standard (PCI DSS) koskee kaikkia korttimaksuja vastaanottavia yrityksiä ja maksukorttitietoa käsitteleviä palveluntarjoajia. (Nixu 2016; PCI 2016.) Standardissa määritetään muun muassa algoritmien ja avaintenpituuksien minivaatimuksen maksukorttien PIN-koodin lähetyksenaikaisen suojaamiseen (PCI 2016).

Elinikä

Alan yleisesti suunnitteluhetkellä turvallisiksi tunnistamien algoritmi ja avaimenpituus yhdistelmien lisäksi PKI-ympäristöä suunnitellessa varmenteille annettava elinikä on myös syytä ottaa huomioon. Mitä pidempi elinikä, sitä enemmän hyökkääjällä on aikaa varmenteeseen sidotun yksityisen avaimen selvittämiseen. Lyhyt elinikä puolestaan johtaa siihen, että varmennetta on uusittava useammin. (Komar 2008, 88.) Tämä saattaa olla hankalaa esimerkiksi kaksitasoisessa mallissa myöntäjävarmentajien varmenteiden kohdalla, sillä uusimista varten juurivarmentaja on käynnistettävä ja uusien varmenteiden siirtäminen tavalla tai toisella hoidettava.

Myös valituilla teknologioilla voi olla vaikutusta elinikiin. Esimerkiksi Microsoftin varmentajat eivät pysty myöntämään varmenteita, joiden elinikä ylittää myöntäneen varmentajan varmenteen eliniän. Tämän takia se suositteleekin, että jokaisen varmentajan varmenteen eliniäksi asetetaan vähintään kaksi kertaa niin pitkä aika, kun sen myöntämien varmenteiden halutaan olevan. Tällä kaavalla jos kaksitasoisessa mallissa halutaan myöntää laitteille viiden vuoden eliniällä olevia varmenteita, tulee juurivarmentajan varmenteella olla 20 vuoden elinikä. (Komar 2008, 87.)



Kuva 11. Varmenteiden elinikä (Komar 2008, 88).

Mitä pidempään varmenteen on tarve olla voimassa, sitä vahvempi sen suojauksen on oltava. Esimerkiksi National Institute of Standards and Technology (NIST) arvioi 2048-bittisen RSA avaimen olevan turvallinen vuoteen 2030 saakka, määrittäen sen liian heikoksi käytettäväksi kahdenkymmenen vuoden eliniän kanssa (NIST 2016.)

Varmenteiden lisäksi myös sulkulistoille asetetaan elinikä. Kun varmenneketjua tarkistetaan, sulkulista ladataan paikalliseen välimuistiin. Tämä sulkulista on voimassa koko sen eliniän, eikä tarkistaja hae uutta listaa ennen sen vanhentumista ilman välimuistin manuaalista tyhjentämistä. Tämän takia lyhyellä sulkulistan eliniällä voidaan varmistaa, että varmenneketjua tarkistavat tahot saavat tiedon mahdollisista uusista kumotuista varmenteista nopeammin. Sulkulistojen lataaminen puolestaan lisää liikennettä verkossa ja saattaa vaikuttaa varmennetta käyttävän palvelun nopeuteen. (Komar 2008, 211.) Lisäksi, kuten varmenteita uusiessakin, sulkulistan julkaisu voi olla hankalaa, mikäli julkaisija on suljetussa tilassa säilytetty juurivarmentaja.

3.5 Prosessit ja roolit

PKI-ympäristön suojaamisen kannalta on oleellista, että organisaatio määrittää siihen liittyvät politiikat, prosessit, roolit ja vastuut selkeästi (Komar 2008, 39; Microsoft 2016f). Tämä on erityisen tärkeää varsinkin ulkoisille tahoille varmenteita jakaville ympäristöille, joissa nämä asiat tulisi olla tarkoin kuvattuina. Kuvauksien saatavuustiedot tulisi olla merkittyinä varmenteiden Certificate Policies -kentässä. Ilman näitä tietoja varmenteita lukevan tahon on vaikea tehdä päätöstä varmenteen luotettavuudesta. Pienissä sisäisiin käyttötarkoituksiin rakennetuissa varmentajien ympäristöissä ei CP tai CPS tiedoille välttämättä ole tarvetta. (Microsoft 2016f.)

Soveltuvien prosessien ja politiikkojen määrittelyyn voi apuna käyttää esimerkiksi jotain PKI:n suojaamiseen tai tietoturvaan yleisesti keskittyvää standardia. Kaksi yleistä tietoturvapoliitikkojen määrittelyyn apuna käytettyä standardia ovat ISO 27002 Code of Practice for Information Security Management ja RFC 2196 The Site Security Handbook (Komar 2008, 41). PKI-ympäristön poliitikkoihin ja käytäntöjen määrittämiseen voi hakea apua IETF:än RFC 3647 Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Frameworkista, jossa määritellään minkälainen CP ja CPS dokumenttien tulee rakenteeltaan ja sisällöltään olla (IETF 2003).

Yksi tunnettu PKI-ympäristön suojaamiseen keskittyvä standardi on Common Criterian Certificate Issuing and Management Components Protection Profile (CIMC PP). Osana sitä määritellään PKI-ympäristön hallintaan liittyvä roolien erittely (Role Separation). CIMC PP version 1.5 mukaan PKI-ympäristön hallinnalliset oikeudet tulisi jakaa neljään rooliin: Administrator, Operator, Officer ja Auditor. Jokaisella roolilla on oikeudet eri tehtäviin, eikä yhdelläkään yksilöllä saa olla enempää kuin yksi rooli. Tämän tarkoituksena on poistaa oikeuksien väärinkäytön mahdollisuus ympäristön ylläpitäjiltä. (Common Criteria 2011.)

CIMC PP:n mukaiseen roolien eriyttämisen käytännön toteutuksissa on ohjelmistovalmistajakohtaisia eroja. Esimerkiksi Microsoftin AD CS määrittää ja nimeää roolit seuraavasti:

- CA Administrator voi tehdä varmentajan ylläpitoon liittyviä toimenpiteitä, kuten julkaista sulkulistoja, muokata varmennepohjien parametreja ja käynnistää tai sammuttaa varmentajan palveluita.
- Certificate Manager voi hyväksyä, myöntää ja kumota varmenteita sekä palauttaa varmenteita tai avaimia varmuuskopioista.
- Backup Operator voi ottaa varmuuskopioita varmennetietokannasta, varmentajan asetuksista ja varmentajan avainparista.
- Auditorilla on pääsy varmentajan tapahtumienvalvonnan lokiin. (Komar 2008, 288.)

Jos roolien eriyttäminen pakotetaan päälle, AD CS valvoo varmentajan käyttäjien oikeuksia automaattisesti. Jos käyttäjälle myönnetyt oikeudet esimerkiksi oikeuttavat myöntämään varmenteita ja mahdollistavat pääsyn tapahtumienvalvonnan lokiin, AD CS estää tililtä kaikki Certificate Manager ja Auditor rooleihin sidotut toimenpiteet. (Komar 2008, 295.)

4 TAPAUS LIEDON KUNTA

4.1 Toimeksianto ja ympäristön kuvaus

Kesällä 2013 Liedon Kunta alkoi rakentamaan erillistä, keskitetysti hallittua työasemaympäristöä Liedon oppilaitosten opiskelijoiden käyttöön. Aiemmin näiden työasemien ylläpito ja hallinta oli järjestetty erilaisin ratkaisuin koulukohtaisesti. Tämän muutoksen myötä syntyi toimialue ja sitä tukeva aktiivihakemistoympäristö edu.lieto.fi.

Ympäristö rakennettiin alustavasti tukemaan opiskelijakäytössä olevia noin kahtasataa työasemaa, mutta suuri kasvu näihin lukuihin oli odotettavissa. Tämän takia ympäristön skaalautuvuus oli yksi suunnittelun keskipisteistä.

Kesän 2014 aikana ympäristöön päätettiin ottaa käyttöön kaksi uutta hallintaa helpottavaa järjestelmää: käyttäjien salasanojen nollaamiseen rakennettu verkkoportaali ja langatonta verkkoa käyttävien laitteiden tunnistautumisesta vastaavaa radius-palvelin. Näiden järjestelmien käyttöönoton yhteydessä syntyi tarve ympäristöön sopivan PKI-ratkaisun suunnittelulle ja rakentamiselle.

4.2 Rajoitteet ja riippuvuudet

Vaikka toimialueen nimi edu.lieto.fi viittaakin lieto.fi-toimialueen alitoimialueeseen, rakennettiin se tietoturvasyistä omaksi, täysin eristetyksi ympäristöksi. Tämän takia myös sen varmennepalvelut tulee toteuttaa omanaan, eikä lieto.fi-toimialueella mahdollisesti oleva ympäristö ole hyödynnettävissä.

Toimialueen sisäisiä palveluita käyttävät vain toimialueelle liitetyt, kunnan hallinnassa olevat työasemat. Sisäisten palveluiden lisäksi oppilaille tullaan tulevaisuudessa tarjoamaan ulkoisia palveluita, joiden käyttö on mahdollista myös kotikoneilta. Nämä palvelut on suunniteltu toteutettavaksi kaupallisten pilvipalveluiden avulla.

Edellä mainittujen rajoitteiden lisäksi ratkaisua valittaessa tulee ottaa huomioon myös

- hallittavuus ja ylläpidon helppous
- kustannustehokkuus
- skaalautuvuus
- tietoturvalle asetetut vaatimukset.

Järjestelmä tulee toteuttaa niin, että sen hallinta ja ylläpito on mahdollisimman helposti hoidettavissa organisaation oman henkilökunnan toimesta. Tarvittavat ylläpitotyöt halutaan pitää mahdollisimman vähäisinä, vaarantamatta kuitenkaan ympäristön tietoturvaa kohtuuttomasti. Lisäksi on tärkeää, että ratkaisu on toteutettu niin yleisellä ja toimivaksi todetulla tavalla, että ongelmatilanteissa tukea on helposti saatavilla myös kolmansilta osapuolilta.

Ratkaisun tulisi olla mahdollisimman kustannustehokas. Tämän takia ratkaisun toteutuksen tulee käyttää hyväksi jo olemassa olevia laitteita, järjestelmiä ja sopimuksia aina kun mahdollista. Myös mahdolliset ylläpidon kustannukset tulee ottaa huomioon.

Vaikka käyttöönottovaiheessa toimialueella on vain kaksi varmenteita tarvitsevaa palvelua, on ratkaisussa otettava huomioon kasvuennusteet. Tulevaisuudessa varmenteita voidaan tarvita muun muassa oppilaitosten tableteille verkkoon tunnistautumista varten tai työasemien tietoliikenteen salaamiseen. Tämän takia ratkaisun pitää pystyä tuottamaan varmenteita useaan eri käyttötarkoitukseen ja niiden jakelu tulee tarvittaessa olla mahdollista myös automatisoida työn helpottamiseksi.

Ratkaisulle halutaan kymmenen vuoden elinkaari, mikä täytyy ottaa huomioon palvelun tietoturvaan liittyviä valintoja tehtäessä. Tietototurvaan liittyvät toimenpiteet rajoittuvat toimeksiannossa rakennettavien palveluiden asennusvaiheeseen. Muut tietoturvaan liittyvät toimenpiteet, kuten alustapalvelimien riittävä koventaminen, verkon tietoturva ja varmuuskopiointi toteutetaan organisaation omasta toimesta.

4.3 Mallin valinta

Ympäristöön päätettiin rakentaa oma sisäinen varmentaja käytettäväksi toimeksiantajan itse hallinnoimien palveluiden ja työasemien kanssa. Mahdollisiin ulkoisiin palveluihin varmenteet tullaan ostamaan joltain julkisesti luotetulta varmentajalta. Näin pystytään rakentamaan ympäristö, jossa ei varmenteiden sisäisten käyttötarpeiden noustessa kustannukset kasva.

Vaikka myös hybrid-malli olisikin täyttänyt kaikki organisaation vaatimukset teknisessä mielessä, oli se hintansa ja sille asetettujen vaatimusten johdosta selkeästi organisaation hankintamahdollisuuksien ulkopuolella.

4.4 Arkkitehtuurit, eliniät, algoritmit ja avainten pituudet

Toimeksiantajan henkilökunnan osaaminen on keskittynyt Microsoftin tuotteisiin. Lisäksi toimeksiantajan olemassa oleva lisenssisopimus mahdollistaa Windows Server -palvelinten asentamisen ilman lisäkustannuksia. Koska AD CS:n integroituminen aktiivihakemistoon mahdollistaa myös varmenteiden ja sulkulistojen automaattisen jakelun, koettiin se sopivimmaksi toteutusvaihtoehdoksi.

Pitkän elinkaarensa takia palvelinkäyttöjärjestelmäksi valikoitui Windows Server 2012 R2, ja palvelimet sovittiin asennettavaksi toimeksiantajan olemassa olevaan virtuaaliympäristöön kustannustehokkuuden ja riittävän korkean käytettävyyden saavuttamiseksi.

Vaikka yksitasoinen hierarkia olisikin käyttöönottohetkellä ollut riittävä, ympäristöstä päätettiin rakentaa kaksitasoinen sen mahdolliset tulevaisuuden käyttötarpeet huomioiden. Samalla kaksitasoinen malli parantaa ympäristön tietoturvaa ja mahdollistaa vikasietoisuuden rakentamisen tarvittaessa. Lisäpalvelimesta ei myöskään muodostunut kustannuksia, ja yhden sammutetussa tilassa säilytettävän juurimyöntäjän lisäämisen ei koettu merkittävästi vaikeuttavan ympäristön hallintaa. Kaksitasoinen malli on myös Microsoftin suosittelema useimpien yritysten tarpeisiin (Microsoft 2016e).

Ympäristöstä haluttiin pystyä myöntämään päätelaitteille ja palveluille varmenteita viiden vuoden voimassaoloajalla seuraavan kymmenen vuoden ajan, ilman että varmentajien varmenteita tarvitsee välissä uusia. Koska AD CS:llä rakennetun myöntäjän myöntämät varmenteet eivät voi ylittää sen oman varmenteen elinikää, täytyi myöntäjävarmentajan varmenteelle asettaa 15 vuoden elinikä. Microsoftin ohjeistuksen mukaan tässä tapauksessa juurivarmentajan varmenteelle tulee asettaa 30 vuoden elinikä.

Samasta syystä myös juurivarmentajan julkaisemalle sulkulistalle asetettiin pitkä viiden vuoden elinaika. Pitkä sulkulistan elinikä koettiin soveltuvaksi, koska kaikki varmenteita käyttävät päätelaitteet ovat toimeksiantajan hallinnassa. Tällöin ne on keskitetysti mahdollista pakottaa hakemaan uusi sulkulista, vaikka vanha olisikin vielä voimassa.

Ympäristössä ei ollut olemassa olevia sovelluksia, jotka asettaisivat rajoitteita avainten pituuksiin tai algoritmeihin. Tämän takia niitä valittaessa seurattiin National Institute of Standards and Technologyn (NIST) suosituksia. Sen mukaan suojauksen vahvuuden tulee olla vähintään 112 bittiä ollakseen turvallinen käytettäväksi vuoteen 2030 saakka (NIST 2016). Vuoden 2030 jälkeen vahvuuden tulisi olla vähintään 128 bittiä. RSA algoritmia käytettäessä 112-bittinen suojaus tarkoittaa vähintään 2048-bittistä avaimen pituutta ja 128-bittinen suojaus vaatii vähintään 3072-bittisen avaimen. Allekirjoituksissa käytettävistä hajautusfunktioista muun muassa SHA-224 on vahvuudeltaan 112 bittiä ja SHA-256 128 bittiä. (NIST 2016.)

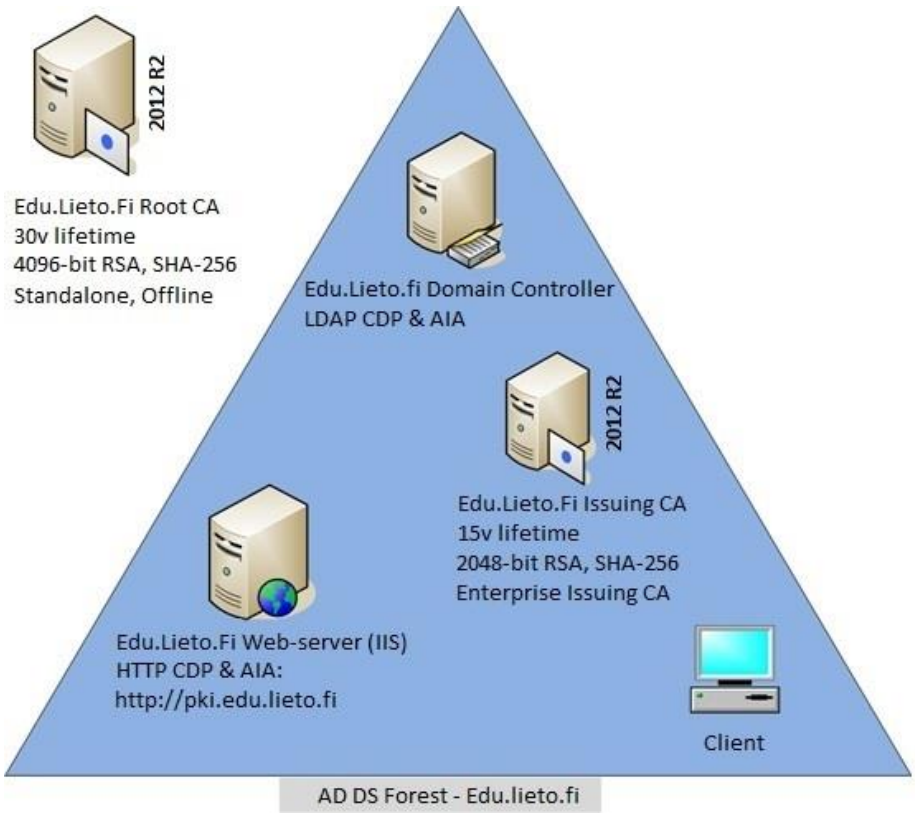
4.5 Ympäristön rakentaminen ja lopputulos

Ympäristön rakentamisessa sovellettiin kahta ohjetta. Ensimmäinen oli Brian Komarin kirjoittaman Windows Server 2008 PKI and Certificate Security -kirjan asennusohje. Asennuksen toimenpiteet ovat Windows Server 2012 R2 palvelimella samat, vaikkakin käyttöliittymä onkin hieman muuttunut. Windows Server 2012 ja 2012 R2 tuovat kuitenkin mukanaan uusia ominaisuuksia AD CS -palveluun, joihin on hyvä tutustua (Microsoft 2014). Erityisen hyvän ohjeesta

tekee tapa, jolla Brian avaa tehtyjen valintojen syitä ja seurauksia. Kirjassa kuitenkin suositaan komentopohjaista lähestymistä, joka saattaa tuntua hankalalta siihen tottumattomalle.

Toinen sovelletuista ohjeista oli Microsoftilla työskentelevän Yung Choun kirjoittama blogi Enterprise PKI with Windows Server 2012 R2 Active Directory Certificate Services. Blogi on luettavissa Microsoftin Technetissä (Chou 2013). Tämän etu on selkeät kuvat käyttöliittymällä tehdyistä valinnoista ja se, että se on tehty Windows Server 2012 R2 palvelimilla. Ohje ei kuitenkaan selitä tehtyjen valintojen taustoja ja esimerkiksi varmenteiden hajautusfunktioiksi on ohjeessa valittuna SHA-1 algoritmi, jota ei enää nykykäytäntöjen mukaan pidetä turvallisena (Palmer & Sleevi 2014).

Soveltaen Komarin ja Choun ohjeita organisaation tarpeisiin, lopputuloksesta rakentui alla olevan kuvan 12 mukainen.



Kuva 12. Lopputulos.

Juurivarmentajan avaimet luotiin 30 vuoden eliniällä, käyttäen 4096-bittistä avainta RSA-algoritmin kanssa. Varmenteen digitaalisen allekirjoituksen hajautusfunktioksi valittiin SHA-256. Palvelinta säilytetään sammutettuna (offline), eikä sitä ole liitetty toimialueelle (standalone).

Myöntäjävarmentajan avaimet luotiin 15 vuoden eliniällä. Lyhyemmän eliniän takia 2048-bittisen avaimen koettiin olevan riittävä käytettäväksi RSA-algoritmin kanssa. Allekirjoituksien hajautusfunktioksi valittiin myös SHA-256, vaikka NIST:n ohjeistuksen mukaan SHA-224 olisi ollut riittävä. Koska Sha-224 perustuu SHA-256 -funktioon, ei tällä muutoksella ole suurta vaikutusta suorituskykyyn (IETF 2004).

Ympäristöön perustettiin kaksi jakelupistettä sulkulistojen ja allekirjoittajan varmenteiden jakelua varten. Toinen jakelupiste on HTTP-muotoinen ja se sijoitettiin toimeksiantajan olemassa olevalle WWW-palvelimelle. Toisena jakelupisteenä toimii suoraan aktiivihakemisto, joka on varmenteille määritettynä LDAP-muotoisena jakelupisteenä. Näin toimialueelle liitetyt koneet voivat tarkistaa sulkulistan ja allekirjoittajan varmenteen suoraan aktiivihakemistosta ja muiden, toimialueen ulkopuolisten koneiden on mahdollista hakea nämä tiedot WWW-palvelimen kautta.

Koska juurivarmentajaa ei tietoturvasyistä haluttu liitettäväksi verkkoon, sen varmenne ja sulkulista kopioitiin jakelupisteisiin manuaalisesti. Sulkulistan osalta tämä täytyy toistaa hieman ennen sen viiden vuoden eliniän päättymistä. Myöntäjävarmentajan sulkulistan elinikä säilytettiin oletusasetuksissa, joka on yksi viikko. Koska myöntäjävarmentaja on verkkoon ja toimialueelle liitetty laite, voitiin se määrittää julkaisemaan sulkulistansa jakelupisteisiin automaattisesti.

Käyttötarpeiden rajallisuuden ja organisaation pienen koon takia, toimeksiantaja ei kokenut varmennepoliitikkojen, varmennekäytäntöjen lausumien, ylläpidon hajauttamisen tai erillisten hallinnallisten roolien ja prosessien käyttöönoton olevan tässä vaiheessa tarpeen.

Asennuksen jälkeen ympäristö oli valmis jakamaan varmenteita ja sen toimintaa testattiin luomalla testivarmenne yhdelle toimeksiantajan WWW-palvelimista, mikä sujui ongelmitta.

5 POHDINTA JA JOHTOPÄÄTÖKSET

Saadessani toimeksiannon minulla oli hyvin suppea kuva PKI:sta ja salauksesta sekä niihin liittyvästä historiasta, tekniikasta ja käytännöistä. Tietoa etsiessäni en onnistunut löytämään pieneen ympäristöön soveltuvaa ohjetta, joka todella avaisi syitä tehtyjen valintojen takana. Tämän takia lähdinkin kirjoittamaan opinnäytetyötä sillä ajatuksella, että saisin aikaiseksi dokumentin, jonka olisin itse toivonut olevan saatavilla toimeksiannon saadessani.

Vaikka Brian Komarin kirjoittaman Windows Server 2008 PKI and Certificate Security keskittyy vahvasti Microsoftin tuotteisiin, sisältää se paljon yleistä tietoa PKI:sta ja sen historiasta. Suosittelen vähintäänkin kirjan silmäilyä kaikille, minkä tahansa PKI-ratkaisun rakentamista suunnitteleville. Kirja keskittyy kuitenkin hyvin suuren ympäristön tarpeisiin, ja sen yli 700 sivun sisällöstä vain pieni osa on sovellettavissa tässä työssä käsitellyn kokoluokan käyttöönotoissa.

Kun työssä tutkittujen valintojen vaihtoehdot ja vaikutukset olivat selvillä, itse asentaminen oli melko suoraviivaista. Ainut isompi vastoinkäyminen syntyi juuripalvelimen asennuksessa. Microsoftin Server Manager -käyttöliittymä tarjoaa varmentajien varmenteiden hajautusfunktion oletusarvona SHA-1 algoritmia, joka jäi epähuomiossa muuttamatta. Tämän takia jouduin virheen huomattuani tekemään ympäristön asennuksen lähes kokonaan uudestaan.

Vaikka SHA-1 algoritmissa on jo pitkään tiedetty olevan heikkouksia, oletusarvon säilyttäminen oli havaittavissa monessa internetissä löytyvässä ohjeessa, kuten myös Microsoftin omassa virallisessa kurssimateriaalissa, 20412B Configuring Windows Server 2012 Services. Juuri tällaisten virheiden välttämiseksi on tärkeää ymmärtää tehtyjen valintojen vaikutukset.

Koen päässeeni tavoitteeseeni niin toimeksiannon kuin siihen liittyvän kirjallisen osuudenkin suhteen. Rakennettu ympäristö täyttää toimeksiantajan tarpeet ja harkittuja poikkeuksia lukuun ottamatta, se on rakennettu alalla yleisesti tunnistettujen hyvien käytäntöjen mukaisesti. Lisäksi vaikka nämä tarpeet vaihtelevatkin suuresti varmenteita käyttävän ympäristön koon ja varmenteiden käyttötar-

koitusten mukaan, koen onnistuneeni työssä käsittelemään ne asiat, jotka joka käyttöönotossa on vähintään osattava ottaa huomioon.

Kuten johdannossa mainitsin, PKI ei itsessään ole uusi asia, minkä takia erilaisia ohjeita sen rakentamiseen on saatavilla useistakin eri lähteistä. Työn tekemisen jälkeen minulle oli entistä selkeämpää, että organisaatioiden vaihtelevien tarpeiden ja alan nopean kehityksen takia, luotettavastakaan lähteestä saadun ohjeen seuraaminen ei automaattisesti takaa oikeanlaista lopputulosta. Tästä syystä, vaikka IT-alalla ei olekaan tavatonta seurata valmiita ohjeita uutta palvelua käyttöönottaessa, PKI-ympäristöä rakentaessa haluan painottaa valintoihin perehtymisen tärkeyttä. Hyvin suunniteltu todella on puoliksi tehty.

LÄHTEET

Carlisle, A. & Lloyd, S. 2002. Understanding PKI: Concepts, Standards, and Deployment Considerations. 2. uudistettu painos. Boston: Addison-Wesley Professional.

Chou, Y. 2013. Enterprise PKI with Windows Server 2012 R2 Active Directory Certificate Services. Viitattu 21.2.2016 http://blogs.technet.com/b/yungchou/archive/2013/10/21/enterprise-pki-with-windows-server-2012-r2-active-directory-certificate-services-part-1-of-2.aspx.

Coffey, N. 2012. RSA Key lengths. Viitattu 21.2.2016 http://www.javamex.com/tutorials/cryptography/rsa_key_length.shtml.

Common Criteria 2011. Certificate Issuing and Management Components Protection Profile. Viitattu 21.2.2016 https://www.commoncriteriaportal.org/files/ppfiles/cert-issu-v15-sec-eng.pdf.

Comodo 2016. History of SSL Certificate. Viitattu 21.2.2016 https://www.evsslcertificate.com/ssl/ssl-history.html.

EJBCA 2016. EJBCA www-sivut. Viitattu 21.2.2016 https://www.ejbca.org/.

Gerck, E. 2000. Overview of Certification Systems: X.509, PKIX, CA, PGP & SKIP. Viitattu 21.2.2016 http://mcwg.org/mcg-mirror/certover.pdf.

GlobalSign 2016a. Trusted Root Certificates. Viitattu 21.2.2016 https://www.globalsign.com/en/certificate-authority-root-signing/.

GlobalSign 2016b. Choosing Safe Key Sizes & Hashing Algorithms. Viitattu 21.2.2016 https://www.globalsign.com/en/ssl-information-center/choosing-safe-key-sizes/.

Gustavsson, T. 2016. Primekey. Viitattu 21.2.2016 https://www.primekey.se/.

Harvey, M. 2011. DER vs. CRT vs. CER vs. PEM Certificates and How To Convert Them. Viitattu 21.2.2016 https://support.ssl.com/Knowledgebase/Article/View/19/0/der-vs-crt-vs-cer-vs-pem-certificates-and-how-to-convert-them/.

Hwang, J. 2012. The Secure Socet Layer and Transport Layer Security. Viitattu 21.2.2016 http://www.ibm.com/developerworks/library/ws-ssl-security/.

IETF 2003. Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. Viitattu 21.2.2016 https://www.ietf.org/rfc/rfc3647.txt.

IETF 2004. A 224-bit One-way Hash Function: SHA-224. Viitattu 21.2.2016 http://tools.ietf.org/rfc/rfc3874.txt.

IETF 2008. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Viitattu 21.2.2016 https://tools.ietf.org/html/rfc5280/.

IETF 2016. Public-Key Infrastructure (X.509). Viitattu 21.2.2016 https://datatracker.ietf.org/wg/pkix/charter/.

ITU 1988. X.509 : The Directory - Authentication framework. Viitattu 21.2.2016 https://www.itu.int/rec/T-REC-X.509-198811-S/en/.

Komar, B. 2008. Windows Server 2008 PKI and Certificate Security. Redmond: Microsoft Press.

Microsoft 2003. What Are Certificates?. Viitattu 21.2.2016 https://technet.microsoft.com/en-us/library/cc758348(v=ws.10).aspx.

Microsoft 2012. 20412B Configuring Advanced Windows Server 2012 Services. Microsoft Learning.

Microsoft 2014. What's New in Certificate Services in Windows Server. Viitattu 21.2.2016 https://technet.microsoft.com/en-us/library/dn473011.aspx.

Microsoft 2016a. How to buy Windows Server 2012 R2. Viitattu 21.2.2016 http://www.microsoft.com/fi-fi/server-cloud/products/windows-server-2012-r2/purchasing.aspx.

Microsoft 2016b. Global Customer Service phone numbers. Viitattu 21.2.2016 https://support.microsoft.com/fi-fi/gp/customer-service-phone-numbers/en-us/.

Microsoft 2016c. Maailmanlaajuinen luotettavien kumppanien verkosto. Viitattu 21.2.2016 http://www.microsoft.com/fi-fi/server-cloud/audience/partner.aspx.

Microsoft 2016d. Securing PKI: Compromise Response. Viitattu 21.2.2016 https://technet.microsoft.com/en-us/library/dn786435.aspx.

Microsoft 2016e. Securing PKI: Planning a CA Hierarchy. Viitattu 21.2.2016 https://technet.microsoft.com/en-us/library/dn786436.aspx.

Microsoft 2016f. Securing PKI: PKI Process Security. Viitattu 21.2.2016 https://technet.microsoft.com/en-us/library/dn786431.aspx.

NIST 2016. Recommendation for Key Management Part 1: General. Viitattu 21.2.2016 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.

Nixu 2016. Maksupäätteet ja PCI DSS –standardi. Viitattu 21.2.2016 https://www.nixu.com/fi/julkaisut/maksup%C3%A4%C3%A4tteet-ja-pci-dss-standardi/.

OpenSSL 2016. OpenSSL www-sivut. Viitattu 21.2.2016 https://www.openssl.org/.

Oppliger, R. 2009. SSL and TLS Theory and Practice. Norwood: Artech House.

Oracle 2010. Sun Directory Server Enterprise Edition 7.0 Reference. Viitattu 21.2.2016 https://docs.oracle.com/cd/E19424-01/820-4811/aakfw/index.html.

Palmer, C. & Sleevi, R. 2014. Gradually sunsetting SHA-1. Viitattu 21.2.2016 https://googleonlinesecurity.blogspot.fi/2014/09/gradually-sunsetting-sha-1.html.

PCI 2016. The PCI Security Standards Council www-sivut. Viitattu 21.2.2016 https://www.pcisecuritystandards.org/.

Pocock, D. 2013. RSA Key Sizes: 2048 or 4096 bits? Viitattu 21.2.2016 http://danielpocock.com/rsa-key-sizes-2048-or-4096-bits/.

Primekey 2016. Primekey www-sivut. Viitattu 21.2.2016 https://www.primekey.se/.

Rhee, M. 2005. Internet Security Cryptocraphic Principles, algorithms and protocols. Chishester: Wiley.

Rouse, M. 2014. PKI (public key infrastructure). Viitattu 21.2.2016 http://searchsecurity.techtarget.com/definition/PKI/.

Schwier, A. 2014. Accessing your SmartCard-HSM from EJBCA. Viitattu 21.2.2016 http://www.smartcard-hsm.com/2014/09/05/Accessing_your_SmartCard-HSM_from_EJBCA.html.

Vacca, J. 2009. Computer and Information Security Handbook. Burlington: Elsevier.

XCA 2015. X Certificate and Key management. Viitattu 21.2.2016 http://sourceforge.net/projects/xca/.