Maiju Virtanen

LUOTTAMUKSEN HALLITSEMINEN SEMANTTISESSA WEBISSÄ

Tietojärjestelmätieteen
kandidaatintutkielma
23.05.2002

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Virtanen, Maiju Anniina
Luottamus semanttisessa webissä/Maiju Virtanen
Jyväskylä: Jyväskylän yliopisto, 2002.
32 s.
Kandidaatintutkielma

Tässä tutkielmassa tarkastellaan luottamusta semanttisessa webissä. Tutkielmassa esitellään erilaisia näkökulmia ja ongelmia luottamuksen hallinnassa, mutta pääpaino on XML-pohjaisilla menetelmillä luottamuksen hallinnassa. Tavoitteena on esitellä neljä keskeistä tekniikkaa käytännönläheisesti, eikä teknisiin yksityiskohtiin oteta syvällisemmin kantaa. Luottamukseen otetaan tutkielmassa kaksi perusnäkökulmaa. Toinen on luottamuksellisten tietojen salassa pysyminen ja toinen se, että miten voimme tietää, mihin verkossa olevaan palveluun tai tietoon voimme yleensäkään luottaa.

Webin koko kasvaa koko ajan ja sen tietomäärää täytyisi pystyä hallitsemaan. XML on joustavuutensa ja muiden etujensa vuoksi tulossa keskeiseksi tekniikaksi webissä. Se asettaa omat vaatimuksensa myös tiedonsalausmenetelmiin hierarkisen rakenteensa takia. Elektroninen liiketoiminta on yleistymässä ja siihen liittyy paljon luottamuksellista tietoa, joka ei saa vuotaa ulkopuolisten käsiin. Käytetty lähdeaineisto tässä tutkielmassa on pääasiassa tieteellisiä artikkeleita, joista on saatu tutkijoiden näkökulmia aiheeseen ja World Wide Web Consortiumin (W3C) spesifikaatioita, joita on käytetty tekniikoihin liittyen niiden kuvaamiseen ja yksityiskohtien tarkistamiseen.

Tutkielman keskeisenä tuloksena on, että luottamuksen hallitsemiseen löytyy monia potentiaalisia tekniikoita. Metatietoa voidaan joustavasti esittää Annotea-projektin tekniikoiden avulla. XML-muotoista tietoa voidaan salata XML-salakirjoituksella ja –allekirjoituksella. Luottamuksellisien tietojen salassa pitämistä voi hallita myös P3P-yhteyskäytännön tietosuojamäärittelyillä.

AVAINSANAT: luottamus, P3P, XML-allekirjoitus, XML-salakirjoitus, Annotea.

SISÄLLYS

1 JOHDANTO

Internetin ja webin tietomäärä ja käyttäjämäärä kasvavat koko ajan kovaa vauhtia. Siitä on paljon hyötyä, että tietoa kerääntyy kattavasti eri asioista ja eri lähteistä. Webissä on kuitenkin valtavasti myös sellaista tietoa ja palvelua, joka on harhaanjohtavaa tai väärää. Web on tullut kaikkien saataville ja julkaisukynnys on madaltunut. Miten voimme tietää mihin palveluun voimme luottaa ja mihin emme? Koska kaikenlaisia käyttäjiä tulee koko ajan lisää ja erityisesti kun elektroninen liiketoiminta yleistyy, täytyy myös verkossa liittyvät luottamukselliset tiedot, kuten henkilötiedot ja luottokortin numerot, pystyä pitämään salassa.

Jotenkin Webin valtavaa informaatiokimppua täytyisi pystyä hallitsemaan. *Semanttinen web* tuo ratkaisuja tähän ongelmaan. Esimerkiksi webissä oleviin dokumentteihin liittyen pystytään semanttisen webin tekniikoilla tallentamaan kuvaavaa metatietoa niiden sisällöstä, dokumentin tekijästä, julkaisupäivästä ja kohderyhmästä sekä dokumenttien välisistä suhteista. Tämä tieto tallennetaan niin formaaliin muotoon, että koneetkin pystyvät sitä ymmärtämään ja automaattisesti tulkitsemaan. Hyvönen (2001) on kuvaillut semanttista webiä ”merkitysten internetiksi” ja ”älykkääksi internetiksi”. Hänen mukaansa semanttinen web on visio, jonka käyttäjänä ovat ihmisten ohella koneet ja sen työkaluja ja standardeja ovat muun muassa XML ja ontologiat.

Tässä tutkielmassa keskitytään siihen, millaisia uusia semanttisen webin XML-pohjaisia tekniikoita on olemassa luottamuksen hallitsemiseksi. Taustaksi esitellään myös näkökulmia luottamukseen ja ongelmia luottamuksen hallinnassa. XML on kovasti tulossa webin oleelliseksi tekniikaksi joustavuutensa ja muiden etujensa ansiosta. Näin XML-muotoiselle tiedolle tarvitaan myös omia tietosuojamenetelmiä. Tavoitteena tutkielmassa on tehdä kartoittava selvitys tulevaisuuden XML-pohjaisista tekniikoista luottamuksen varmistamiseen. Neljä keskeistä tekniikkaa on tarkoitus esitellä käytännönläheisesti, eikä tarkkoihin teknisiin yksityiskohtiin pureuduta syvällisemmin.

Luottamus on laaja käsite. Tässä tutkielmassa siihen on kaksi perusnäkökulmaa. Ensinnäkin webissä liikkuu luottamuksellista tietoa ja täytyy varmistua siitä, etteivät

asiattomat tahot pääse käsiksi tähän tietoon. Web-sivut voivat tarjota tietosuojamäärittelyjä, joilla ilmaistaan esimerkiksi, kuinka käyttäjän henkilötietoja tullaan palvelutapahtumissa käyttämään. Näin käyttäjälle tarjotaan mahdollisuus hyväksyä tai hylätä palvelun käyttäminen ja hän voi varmistua siitä, että luottamukselliset tiedot eivät vuoda. Tällaisia tietosuojamäärittelyjä voidaan tehdä tietosuojaprotokolla P3P:n ja sen kuvauskielen APPEL:n avulla. Luottamuksellisen tiedon hallinnassa tarvitaan usein myös tiedon salausta. Tässä tutkielmassa esitellään julkisen avaimen järjestelmä sekä XML-pohjaiset digitaalinen allekirjoitus ja salakirjoitus. Digitaalinen allekirjoitus ratkaisee sellaisia luottamukseen liittyviä seikkoja, kuten käyttäjän tunnistus (authentication) eli varmistetaan, että viestin lähettäjä on se, joka väittää olevansa, tiedon eheys (data integrity) eli tieto säilyy muuttumattomana lähettäjältä vastaanottajalle ja kiistämättömyys (non-repudiation), joka tarkoittaa sitä, että viestin lähettäjä ei voi kieltää tiedon lähettämistä.

Toinen tämän tutkielman perusnäkökulma luottamukseen on se, että mihin verkosta löytyviin tietoihin voimme luottaa. Esimerkiksi tuntemamme luotettavan henkilön tai yrityksen antamiin metakuvauksiin on helpompi luottaa kuin joihinkin satunnaisiin ja tuntemattomiin sivuihin. Tutkielmassa esitellään tähän liittyen Annotea-projektia, jonka tavoitteena on kehittää tekniikat, joiden avulla metatietoa ja kommentteja webin dokumenteista voidaan joustavasti ja helposti esittää ja hallita.

2 NÄKÖKULMIA JA ONGELMIA LUOTTAMUKSEN HALLITSEMISESSA

Tässä luvussa käsitellään erilaisia näkökulmia luottamukseen ja sen hallintaan. Käsitellään luotettavia lähteitä, luottamuksellisten tietojen käsittelyä ja luottamusta agenttien välillä. Määritellään mitä luottamus oikeastaan on ja kerrotaan luottamuksen syntymisestä. Pohditaan ongelmia luottamuksen hallinnassa ja mitä ongelmia perinteisiin menetelmiin luottamuksen varmistamiseksi liittyy.

2.1 Luottamuksen osa-alueita

Perusongelmana luottamuksen luomisessa webissä on, että käyttäjät eivät uskalla käyttää verkkopalveluja, koska he pelkäävät, että henkilökohtaisia tietoja käytetään väärin. Verkkopalveluiden täytyisi jollain tapaa osoittaa asiakkaalle luotettavuutensa. Tähän tarjoaa ratkaisuja ainakin tietosuojatekniikka nimeltä P3P, josta kerrotaan lisää myöhemmin.

Eräs näkökulma on luottamuksen jakaminen. Eri yhteisöille ja ryhmille jotkut tietolähteet voivat olla luotettavampia, kiinnostavampia ja tärkeämpiä kuin toiset. Web-dokumentteihin voidaan liittää metatietoa, millaisille käyttäjäryhmille kukin aineisto on tarkoitettu. Näitä käyttäjäryhmiä voisivat olla esimerkiksi lapset, tietyn tieteenalan harjoittajat tai tietystä asiasta tai harrastuksesta kiinnostuneet. Muu metatieto voi olla esimerkiksi kommentteja ja huomautuksia web-dokumenteista muille yhteisön jäsenille.

Luottamus on elintärkeää joissain järjestelmissä, esimerkiksi elektronisen kaupankäynnin palveluissa. Luottamuksellisten tietojen, esimerkiksi luottokortin numeron tai liikesalaisuuksien vuotaminen ulkopuolisten käsiin saattaa aiheuttaa katastrofin. Esimerkiksi jos luottokortti joutuu väärille teille, siitä voi tulla todella kallis lasku luottokortin omistajalle. Toisaalta taas henkilökohtaistenkin tietojen leviäminen voi aiheuttaa ongelmia. Esimerkiksi puhelinnumeron joutuminen väärille

teille, voi aiheuttaa häiriösoittoja varsinkin julkisuuden henkilöille, jotka yleensä haluavat pitää numeronsa salassa.

Näkökulmiksi luottamukseen (trust) ovat Bigham ym. (2001) esittäneet reiluuden (fairness), luotettavuuden (reliability), maineen (reputation) ja uskollisuuden (lojalty). Heidän ajatuksensa luottamuksesta liittyy sekä agenttien väliseen luottamukseen että luottamukseen koskien systeemiä itseään. Luotettavuus liittyy yleensä suoraan johonkin kohteeseen. Luotettavuus on luottamuksen alakäsite ja hyvin lähellä käsitettä luottamus. Jokin www-dokumentti esimerkiksi voi olla jonkun käyttäjän mielestä luotettava. Maine taas luo luotettavuutta henkilöön tai palveluun. Omalla yhteisöllä tai organisaatiolla on yleensä hyvä maine yhteisön jäsenen silmissä ja näin siihen on helppo luottaa. Jos metatietoa on liittänyt web-resursseihin joku hyvämaineinen tunnettu henkilö tai yhteisö, resursseihin on paljon helpompi luottaa. Reiluus voisi tarkoittaa verkkokaupassa sitä, että verkkokaupan omistaja tai palveluntarjoaja kohtelee asiakastaan hyvin. Tällöin palveluntarjoaja ei ajattele vain omaa etuaan, vaan antaa asiakkaalle mahdollisuuden esimerkiksi tuotteen palauttamiseen ja kohtuulliseen maksuaikaan. Uskollisuus voisi liittyä asiakkaan puolelta vaikkapa siihen, että asiakas muistaa yrityksen, puhuu hyvää siitä ja ostaa tuotteet siitä tietystä samasta verkkokaupasta. On tärkeää, että asiakas voi luottaa yritykseen, mutta yhtä tärkeää on, että yritys voi luottaa asiakkaaseen. Yritykselle voi koitua suuria vahinkoja, jos sen palveluja käytetään väärin, tai se saa uskottomien asiakkaiden pahojen puheiden takia huonon maineen. Näin nämä neljä ihmisten välisen luottamuksen käsitettä ilmentyvät myös webissä. Tosin siellä luottamuksen syntyminen ja sen saavuttaminen tapahtuvat usein eri tavalla, koska henkilökohtaista suhdetta ostajan ja myyjän kanssa ei yleensä synny.

Luottamusta voi ajatella myös laajemmin ja loppujen lopuksi kaikki luottamuksen alakäsitteet ja määrittelyt nivoutuvat toisiinsa ja liittyvät jollain tapaa yhteen. Periaatteessa luottamus on kahden tai useamman osapuolen välinen asia. WWW-dokumentin kanssa ei yleensä millään tavalla jaeta luottamusta, kun taas esimerkiksi agentille käyttäjä voi antaa puhelinnumeronsa tai muita henkilötietoja. Tällöin käyttäjän ja agentin välillä vallitsee luottamus ja ne jakavat luottamuksen keskenään. Toisaalta agenttiinkin voi liittyä luotettavuus. Agentti on luotettava, kun agentti toimii oikealla, käyttäjän määrittelemällä tavalla. Kun puhutaan tiedon siirtämisestä

osapuolten välillä, luottamukseen kuuluu olennaisesti myös käyttäjän tunnistus (authentication - voidaan varmistua, kuka vastapuoli on ja että hän on varmasti se, joka väittää olevansa), tiedon kiistämättömyys (lähettäjä ei voi kiistää tiedon lähettämistä) ja tiedon eheys (tieto säilyy muuttumattomana lähettäjältä vastaanottajalle).

2.2 Luottamuksen syntyminen

Bickmore ja Cassell (2001) ovat käsitelleet artikkelissaan luottamuksen syntymistä ihmiselle. Heidän ajatuksiaan esitellään tässä tutkielmassa mielenkiintoisina näkökulmina ja tulevina potentiaalisina tutkimusaiheina. Bickmoren ja Cassellin (2001) mukaan ihminen käyttää monenlaisia ihmissuhteisiin liittyviä keskustelustrategioita, kuten kevyttä jutustelua (small talk), luomaan luottamusta toistensa välille. He väittävät, että tällaisia strategioita voivat käyttää myös käyttöliittymäagentit. He käsittelevät esimerkkinään ostoagenttia (REA, real-time multimodal, life-sized Embodied Conversational Agent), joka on graafinen ihmisen näköinen hahmo heijastettuna isolle kankaalle ja jonka kanssa voi käydä keskusteluja. Tätä voisi käyttää elektronisessa liiketoiminnassa esimerkiksi silloin, kun panokset ovat kovia ja suhde ostoagentin kanssa on tarkoitus jatkua useita viikkoja. Tällöin tarvitaan todellista luottamusta agentin ja käyttäjän välillä. Ehkä ison kankaan levittäminen jokaisen ostoskerran takia tuntuu turhalta, mutta keskusteluidea on mielestäni hyvä.

Perusidea lähtee siitä, että ihminen aloittaa luottamuksellisen suhteen tekemisen varovasti – kevyellä jutustelulla ja vasta vähitellen paljastamalla itsestään jotain. Malli sosiaalisesta dialogista käyttäjän luottamuksen saavuttamiseksi liittyy siihen, että vuorovaikutteiset suhteet mitataan monilla ulottuvuuksilla, kuten intimiteetti, solidaarisuus, läheisyys, tuttuus ja yhteys. Tapoja luottamuksen tuottamiseksi on täten antaa tietoa itsestään, ja kehottaa kuuntelijaa tekemään samoin sekä kevyt jutustelu, jolla voidaan liikkua vähitellen lähemmäs arkaluontoisia aiheita tai etsiä sopivaa kommunikointitapaa tai slangia juuri tämän kuulijan kanssa. Jos kone alkaisi kertoa itsestään kesken sukkahousujen oston verkkokaupasta, itsestäni ainakin se voisi tuntua aika teennäiseltä, naurettavalta ja jopa pelottavalta. Mutta kone saattaa osata tehdä

tämän myös kohtuullisen luonnollisesti. Tässä katkelma Bickmoren ja Cassellin (2001) esittämästä viehättävästä keskusteluesimerkistä (lähes kaikki agentin puhumaa):

> Tämä mikrofoni on kauhea, inhoan käyttää näitä.
> Anteeksi muuten ääneni. Tämä on jonkun insinöörin mielipide luonnollisesta äänestä.
> Oletko meidän sponsoreitamme? *Käyttäjä: Kyllä.*
> Olitko viimeksi sponsorikokouksessa?
> Minä tulin niin uupuneeksi viime kokouksessa, että minulta meinasi jo lähteä ääni lopussa.
> Niin, missä haluaisit asua?
> Kuinka monta makuuhuonetta tarvitset?
> Jne.

Kyse on siis talon ostamisesta. Kun asiaa alkaa ajatella, niin ei ehkä kuitenkaan sukkahousujen ostamista, mutta juuri tällaisia suuria hankintoja varten, on elintärkeää, että saavutetaan ostajan luottamus. Tarkoitus ei ole millään tavalla harhauttaa käyttäjää, vaan antaa hänelle turvallisempi tunne tehdä ostopäätöstä. Tällaisessa keskustelussa nimittäin ”oppii tuntemaan” myyvän agentin. Koska kyseessä ovat suuret päätökset, tunnelmaa voidaan välillä keventää, ettei ostaja tuntisi oloaan ahdistuneeksi. Tietysti keskustelut vaikuttavat erilaisiin käyttäjiin eri tavalla riippuen heidän persoonallisuudestaan, joten agentin tulisi jollain tavalla tuntea käyttäjänsä tai jotenkin kontrolloida sitä, ettei pitkä jaarittelu ala kyllästyttää häntä. Bickmoren ja Cassellin mallissa (2001) on mielestäni kehityskelposia ideoita luottamuksen synnyttämiseen webissä.

2.3 Ongelmia agenttien toiminnassa

On hyvin ongelmallista, jos agenttiohjelmat toimivat automaattisesti käyttäjänsä hyväksi, eivätkä tiedä käyttäjästä tarpeeksi. Joissain tapauksissa agentti voi olla niin kehittynyt, että osaa esimerkiksi tilata lounaan käyttäjälle automaattisesti. Jos käyttäjä ei kuitenkaan sattuisikaan olemaan nälkäinen ja lounaan hinta olisi kallis, tulisi käyttäjälle agentin toiminnasta ongelmia. Agenttien virheelliset toiminnat voivat siis tulla hyvin kalliiksi käyttäjälle.

Toisenlainen ongelma voi tulla, kun agentit kommunikoivat keskenään. Verkossa on paljon salaista ja arkaluoteista tietoa esimerkiksi elektroniseen liiketoimintaan liittyen, ja jos nämä tiedot vuotavat agenttien välillä, voi seurauksena olla suuria ongelmia. Tällöin täytyy tiedon salauksen olla kunnossa ja agentille täytyy olla selvästi määriteltynä, mitä tietoja se saa käyttää tapahtuman ulkopuolella ja mitkä tiedot sen täytyy pitää visusti itsellään.

2.4 Salausmenetelmien historiaa

Seuraavat tiedot SSL-tietosuojamenettelystä (Secure Sockets Layer) ovat peräisin Simonin, Madsenin ja Adamsin (2001) artikkelista. SSL suojaa arkaluonteisen tiedon siirron selaimen ja WWW-palvelimen välillä. Tieto jätetään kuitenkin liian usein suojaamattomana palvelimelle, josta hakkeri metsästää sitä jopa todennäköisemmin kuin kesken tiedon siirtämisen. Jos haluaa käsiinsä salaista tietoa, on paremmat mahdollisuudet saada sitä kasapäin palvelimelta, kuin murtautumalla yhteen tiedonsiirtotapahtumaan. Simon ym. (2001) esittävät, että parempi tapa olisi suojata itse tieto. Tähän kuuluu olennaisena osana pitkäaikainen osapuolten tunnistus, tiedon eheys ja kiistämättömyys, joita perinteiset internetin tietoturvamenetelmät, kuten SSL ja käyttäjätunnus/salasana eivät varmista yksinään.

Simon ym. (2001) esittelevät maailmanlaajuisesti tunnustettua mallia, joka turvaa tapahtumat käyttämällä digitaalista sertifikaattia ja digitaalista allekirjoitusta. Julkisen avaimen järjestelmä on arkkitehtuuri, johon sisältyy standardit ja prosessit, jotka mahdollistavat digitaalisten sertifikaattien ja allekirjoitusten käytön. Sitä esitellään seuraavassa luvussa tarkemmin.

Mactaggart (2001) väittää, että perinteiset metodit luottamuksen ilmaisemisessa osapuolten välillä eivät riitä Internetissä. Se pitää varmasti paikkansa, koska Internet on julkinen eli sinne voi päästä kuka tahansa. Arkaluontoiset tiedot täytyy tämän takia salata erityisen tarkasti. XML:n yleistyminen tuo uusia haasteita tiedon salaamiselle. Myös XML-muotoinen arkaluontoinen tieto pitäisi pystyä pitämään salattuna ja sen hierarkisen rakenteen vaatimukset täytyisi ottaa huomioon. Mactaggartin mukaan XML-muotoinen tieto voidaan salakirjoittaa perinteisilläkin menetelmillä, mutta

tapaus, jossa eri osapuolet tarvitsevat erilaiset oikeudet saman dokumentin eri osiin, onkin sitten ongelmallisempi. Esimerkki tällaisesta tapauksesta elektronisessa liiketoiminnassa: Verkkokaupan myyjän ei tarvitse tietää ostajan luottokortin numeroa, kunhan maksu tulee perille ja toisaalta taas pankin ei tarvitse tietää yksityiskohtaisesti ostetuista tavaroista. Tällaisen dokumentin yksityiskohtaisen osiin pilkkomisen mahdollistavat XML-muotoiset dokumentit ja asiakirjat yhdessä XML allekirjoituksen kanssa.

3 TEKNIIKAT LUOTTAMUKSEN HALLITSEMISESSA

Tässä luvussa esitellään neljä keskeistä XML-pohjaista tekniikkaa luottamuksen esittämiseen ja hallitsemiseen. Tekniikoita käsitellään esimerkkien avulla ja käytännönläheisesti, eikä teknisiin yksityiskohtiin pureuduta tarkasti.

3.1 Tietosuojaprotokolla P3P

Tässä luvussa kerrotaan aluksi perusperiaatteet tietosuojaprotokollasta P3P. Toisessa aliluvussa esitellään P3P-protokollan tietosuojamäärittelyt kuvaava APPEL-kieli.

3.1.1 P3P

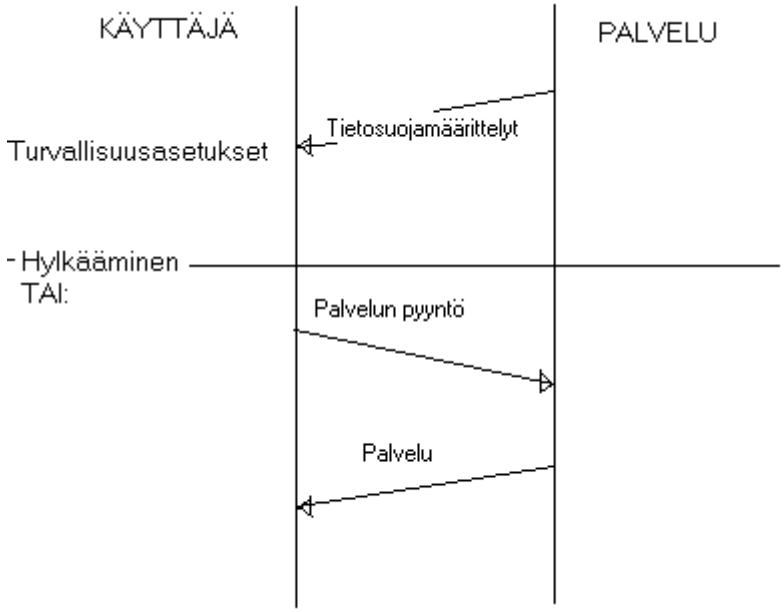
P3P:n kehitys lähti PICS:stä (Platform for Internet Content Selection) vuonna 1996. PICS:n, kuten myös P3P:n, perusajatus on suodattaa sellaiset web-sivut ja -sisällöt pois, joita käyttäjä ei halua tai tarvitse. Alkuperäinen tarkoitus PICS:lle oli suojella lapsia pornografialta ja väkivaltakuvauksilta webissä. (Grimm ja Rossnagel, 2000)

P3P on protokolla, joka on suunniteltu ilmoittamaan webin käyttäjille web-sivun tai -palvelun tietosuojamenettelyt[1], jotka kuvaavat sen, mitä tietoa ne keräävät käyttäjistä ja miten tätä tietoa käytetään. Nämä web-sivun turvallisuutta koskevat käytännöt toteutetaan XML-muodossa. Käyttäjien agentit voivat hakea ja tulkita sivujen tietosuojamenettelyjä automaattisesti, eikä käyttäjien edes tarvitse joka kerta sivulle tullessaan lukea niitä, koska P3P agentit ilmoittavat käyttäjälle ne vain tarvittaessa. (Cranor ym. 2002, P3P1.0)

[1] Vastedes sanaa “privacy practises” käytetään tässä tutkielmassa vastaamaan sanaa tietosuojakäytännöt ja sanaa “policy” sanaa tietosuojamenettely. (Atk-sanakirja) Sanasta ”policy” käytetään joissain yhteyksissä myös sanaa tietosuojamääritykset.

Ideana siis on, että käyttäjä tekee selaimelleen omat turvallisuusasetukset. Selaimen pitää tukea P3P:tä tai siinä pitää olla P3P-agentti, ja kohteena olevalla web-sivulla oletetaan olevan P3P tietosuojamenettelymäärittely. Esimerkki käyttäjän asetuksista: ”Tietojani saa luovuttaa vain siinä tapauksessa, että niitä luvataan käyttää vain kyseessä olevaan tapahtumaan”. Jokaisella sivulla selain tarkistaa sivun tietosuojamenettelymääritykset ja vertaa niitä omiin asetuksiinsa. Käyttäjää varoitetaan silloin, kun tehdyt asetukset ovat ristiriidassa sivun tai palvelun tietosuojamäärityksien kanssa. Cranorin ym. kirjoittaman spesifikaation (2002, P3P1.0) mukaan P3P-agentit voidaan toteuttaa myös muuten kuin selaimeen: selaimen lisäohjelmaan (plug-in – ATK-sanakirja), proxy-palvelimelle tai sitten Java-applettina, JavaScriptillä, elektroniseen lompakkoon tai muuhun käyttäjän tiedonhallintatyökaluun.

Kuvio 1 esittää P3P-yhteyskäytännön toimintaa. Kuvan mukaisesti palvelu ilmoittaa käyttäjälle tietosuojamäärittelyt. Seuraavaksi käyttäjän tehtävänä on valita, hyväksyykö vai hylkääkö hän palvelun käytön. Usein hyväksyminen hoidetaan automaattisesti selaimen turvallisuusasetusten avulla.



KUVIO 1. P3P-yhteyskäytännön toiminta Grimmia ja Rossnagelia (2000) mukaillen.

3.1.2 Tietosuojamäärittelyjen kuvailukieli APPEL

APPEL eli A P3P Preference Exchange Language on kieli, jolla voidaan kuvailla P3P tietosuojamäärittelyjä. Tällä kielellä käyttäjä voi tehdä omat asetuksensa eli niin kutsutun sääntöjoukon (ruleset), jonka pohjalta agentti pystyy automaattiseen tai puoliautomaattiseen päätöksentekoon. W3C:n spesifikaatiossa (Cranor ym. 2002, APPEL1.0) kerrotaan, että joissain tapauksissa automaattinen päätöksenteko voi mennä jopa niin pitkälle, että jos käyttäjän asetukset käyvät yhteen sivun käytäntöjen kanssa, tässä tapauksessa agenttina toimiva elektroninen lompakko on valtuutettu julkaisemaan tietoja palvelulle. Koska voi olla kyse yksityisestäkin tiedosta, kuten puhelimen tai luottokortin numerosta, täytyy agentin ja käyttäjän välillä tällöin vallita vahva luottamus.

Tavalliselle käyttäjälle turvallisuusasetusten tekeminen on liian vaikeaa. Cranor ym. (2002, APPEL1.0) esittävät tähän ratkaisuksi sääntöjoukkojen jakamisen. Täten yritys voi luoda yhteisen suosituksen sääntöjoukosta, jota yksittäiset käyttäjän voivat halutessaan muuttaa. Uskoisin, että organisaatio, johon käyttäjä kuuluu, on varmasti tarpeeksi luotettava tekemään sopivan sääntöjoukon.

Saattaa olla, että kun WWW tulee yhä yleisempään ja laajempaan käyttöön, koko turvakulttuuri muuttuu. Erityisesti asenteet muuttuvat sillä tavalla, että ei edes haluta käyttää palveluja, joita ei ole suojattu standardoidulla tai asianmukaisella tavalla. Etsitään ja käytetään vain luotettavia palveluja. Standardin mukaisiin palveluihin (esimerkiksi P3P) voidaan luottaa. Näin uskalletaan käyttää palvelua ja voidaan varmistua siitä, että henkilökohtaiset tiedot eivät vuoda ulkopuolisten käsiin. Grimm ja Rossnagel (2000) puhuivat kansainvälisistä tietoturvalaeista ja se herätti itselleni ajatuksia siitä, että tulevaisuudessa varmasti tällaisia tarvitaan yhä enemmän. Ne ja yhteiset standardit mahdollistavat sen, että on helppo liikkua palvelusta toiseen luottavaisin mielin.

3.2 XML- allekirjoitus ja XML-salakirjoitus

Tämän luvun ensimmäisessä aliluvussa lukija johdatellaan julkisen avaimen järjestelmään ja digitaaliseen allekirjoitukseen, jonka jälkeen esitellään varsinainen XML-allekirjoitus. Seuraavaksi kerrotaan XML-salakirjoituksesta sekä XML-allekirjoituksen ja –salakirjoituksen välisestä suhteesta.

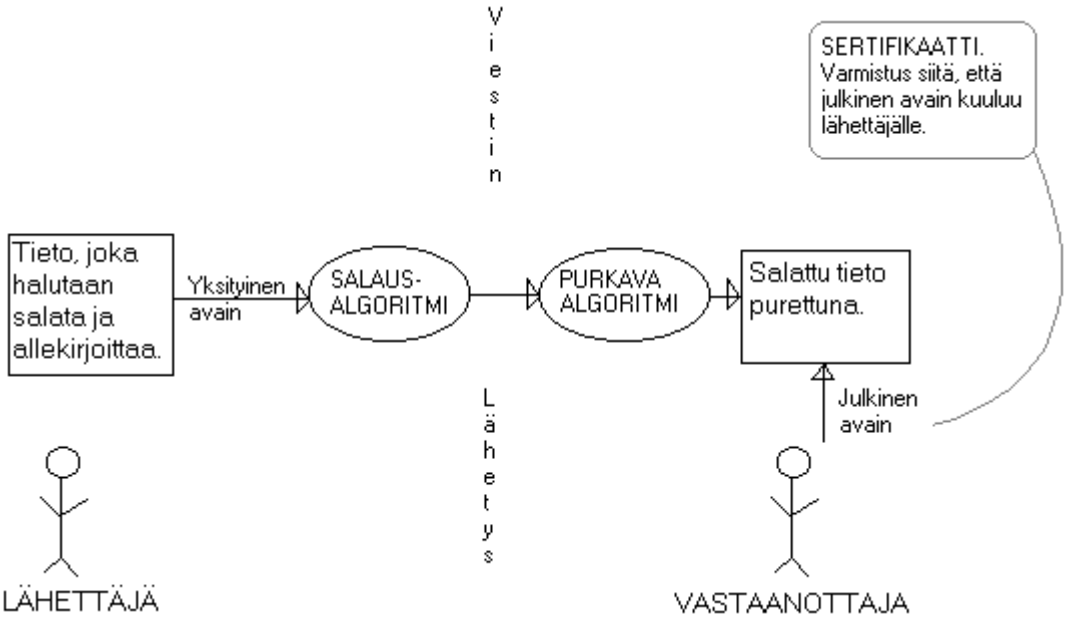
3.2.1 Digitaalinen allekirjoitus ja julkisen avaimen järjestelmä

Tämän luvun tarkoitus on johdatella lukija digitaalisen allekirjoituksen perusteisiin niin, että seuraavia lukuja ja varsinaista XML-allekirjoitusta olisi helpompi ymmärtää. Digitaalisen allekirjoituksen tarkoitus on suojata osapuolten välillä siirrettävä tieto. Viestintäviraston lähdeaineiston mukaan se takaa käyttäjän tunnistuksen eli varmistetaan, että tiedon lähettäjä on se, joka väittää olevansa; tiedon eheyden eli muuttumattomuuden ja kiistattomuuden eli sen, että tiedon lähettäjä ei voi jälkeenpäin kiistää lähettäneensä viestiä.

Public Key Infrastructure (PKI), joka usein suomennetaan julkisen avaimen järjestelmäksi, kokoaa yhdeksi tietoturva-arkkitehtuuriksi digitaalisen allekirjoituksen vaatimukset, kuten salakirjoitusmenetelmät, varmenneviranomaisen ja sertifikaatit. Perusperiaate julkisen avaimen järjestelmässä on se, että kaikilla viestien lähettämiseen osallistuvilla on käytössään sekä yksityinen että julkinen avain. Tällaista kahden avaimen salausta kutsutaan epäsymmetriseksi. Yksityisen avaimen henkilö pitää visusti itsellään, mutta julkisen avaimen hän antaa kaikille, joiden kanssa hän kommunikoi. Lähettäjä salaa viestin yksityisellä avaimellaan ja vastaanottaja voi tarkistaa viestin julkisella avaimella. (Viestintävirasto)

Kuviossa 2 on yksinkertaistettu digitaalisen allekirjoituksen luominen Simonin ym. (2001) kuvauksen mukaan. Kuvan mukaisesti salausalgoritmi, joka saa lähettäjältä parametrinaan yksityisen avaimen, salaa lähetettävän tiedon. Jos tiedon vastaanottamisvaiheessa lähettäjän yksityinen ja vastaanottajan lähettäjältä saama

julkinen avain täsmäävät toisiinsa, viestin alkuperä ja väärentämättömyys on varmistettu.



KUVIO 2. Digitaalisen allekirjoituksen luominen.

Julkisen avaimen infrastruktuuriin kuuluu, että avaimet voidaan varmentaa sertifikaatilla, joka koostuu mm. asiakkaan nimestä ja julkisesta avaimesta. Tämän varmenteen on allekirjoittanut varmenneviranomainen. Viestintäviraston lähdeaineiston mukaan ennen viestin salausta yksityisellä avaimella, viestistä lasketaan tiiviste. Se on toisin sanoen lyhennetty versio viestistä. Salauksen purkava algoritmi siis purkaa salatun viestin takaisin tiivisteeksi ja sitä verrataan vastaanottajan ohjelman itse laskemaan saman viestin tiivisteeseen (Viestintävirasto).

3.2.2 XML-allekirjoitus

XML-muotoinen tieto on yleistymässä WWW:ssä ja täten tällaisen tiedon turvallisuuden varmistaminen tulee ensiarvoisen tärkeäksi. XML-allekirjoitus (XML Signature) on World Wide Web Consortiumin (W3C) ja Internet Engineering Task Forcen (IETF) yhteinen projekti. Se ei määrittele spesifikaation (2001) mukaan koko

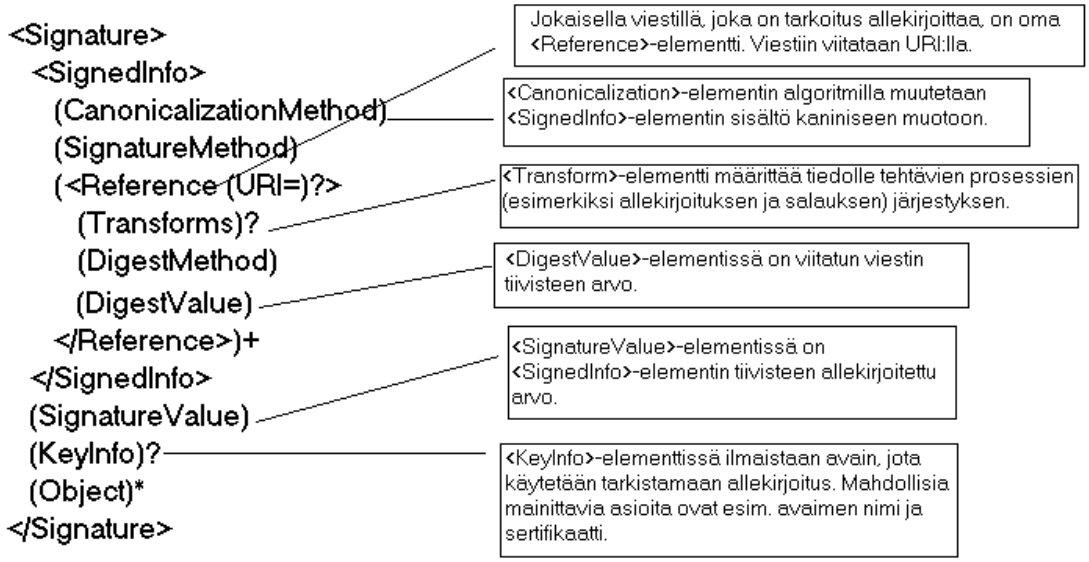
tietoturva-arkkitehtuuria, vaan käyttää apunaan julkisen avaimen järjestelmää ja viestin tiivistämistä. Sillä voidaan allekirjoittaa kaikkea digitaalisessa muodossa olevaa tietoa: niin tekstimuotoista (esim. HTML), binaarista (esim. GIF-kuva) kuin XML-muotoistakin tietoa.

XML-allekirjoitus takaa W3C:n spesifikaation mukaan (Fox ym. 2001) viestin ja lähettäjän autentikoinnin sekä viestin eheyden. XML-allekirjoituksen merkittävä etu on, että sen avulla voidaan allekirjoittaa dokumenttia osa kerrallaan. Tätä ominaisuutta tarvitaan silloin, kun eri dokumenttiin liittyvät osapuolet haluavat allekirjoittaa vain osat, jotka ovat merkityksellisiä heille ja kun nämä allekirjoitukset luonnollisesti voivat tapahtua eri aikaan. (Simon ym. 2001)

XML-allekirjoituksen kelpoisuus vaatii, että allekirjoitettu tieto on saatavilla. Viite alkuperäisestä allekirjoitetusta tiedosta sisältyy XML allekirjoitukseen ja voi:

1) olla viitattuna URI:n avulla XML-allekirjoituksessa,
2) olla samassa tiedostossa XML-allekijoituksen kanssa rinnakkain (sisarelementtinä),
3) olla upotettuna XML-allekirjoitukseen lapsielementtinä tai
4) sisältää XML-allekirjoituselementti upotettuna itseensä (XML-allekirjoituselementti lapsielementtina). (Simon ym. 2001)

XML-allekirjoituksen kompontit ovat esiteltynä kuviossa 3. Tässä alkuperäiseen tietoon viitataan <Reference>-elementissä URI:n avulla. Jokaisella tietoalkiolla on <DigestMethod>-elementti, joka sisältää algoritmin, joka laskee tiivisteen viestille. <CanonicalizationMethod>-elementissä oleva algoritmi muuttaa <SignedInfo>-elementin sisällön kanoniseen muotoon. Kolmas allekirjoituksessa tarvittava algoritmi on <SignatureMethod>-elementissä. Se tekee lopullisen allekirjoituksen kanonisessa muodossa olevasta <SignatureInfo>-elementistä ja sijoittaa tästä saadun arvon <SignatureValue>-elementtiin. <Transform>-elementin ja –algoritmin avulla voidaan määrittää allekirjoitettavalle viestille tehtävien prosessien järjestys. Oletuksena on, että aluksi viestistä lasketaan tiiviste, mutta järjestys saa olla myös toinen (Fox ym. 2001).



KUVIO 3. XML-allekirjoituksen komponentit Simonin ym. (2001) kuvaa mukaillen.

Kanonisointia tarvitaan (Mactaggart, 2001), koska salakirjoituksessa tarvittava hash-algoritmi reagoi pieneenkin muutokseen dokumentissa, ja tällöin, vaikka dokumentit olisivat hierarkialtaan ja elementeiltään samanlaiset, johtuen pienistä eroista puhtaassa tekstimuodossa, dokumentit tulkitaan erilaisiksi. Pienet dokumenttien erot johtuvat tyhjistä elementeistä, eroista kommenteissa tai erilaisten työkalujen (esimerkiksi jäsentäjien) käytöstä XML-dokumentin muokkaamisessa. (Mactaggart, 2001) Kanonisen XML:n spesifikaatiossa (Boyer, 2001) on tarkat määrittelyt metodista, jolla XML-dokumentti voidaan muuttaa sellaiseen muotoon, jossa tietyt sallitut muutokset ovat hallittavissa. Käytännössä se tarkoittaa sitä, että kun kahdella dokumentilla on sama kanoninen muoto, ne ovat loogisesti samanlaiset ja myös niitä käyttävä sovellus tulkitsee ne samanlaisiksi pienistä eroista huolimatta. Tämä on välttämätöntä XML-allekirjoituksen kelvolliseksi osoittamiseksi.

XML-allekirjoituksen kelvolliseksi osoittaminen mukaillen Foxin ym. 2001 kirjoittamaa spesifikaatiota:

1) Lasketaan alkuperäiselle tiedolle tiiviste (<DigestMethod>:lla) ja verrataan tätä <DigestValue>-elementin arvoon.

2) Lasketaan allekirjoituksen arvo <SignatureMethod>-elementin ja käytetyn avaimen (<KeyInfo>) avulla ja verrataan tätä <SignatureValue>-arvoon.

Jos sekä tiedon tiivisteet että allekirjoitukset täsmäävät, allekirjoitus on kelvollinen. (Huom. arvoja laskettaessa täytyy <SignedInfo>-elementti sisältäen alkuperäisen tiedon ja <SignarureMethod>-elementti olla kanonisessa muodossa.)

3.2.3 XML-salakirjoitus

XML-salakirjoitus on kehitetty XML-muotoisen tiedon salakirjoittamiseen ja se on olennainen osa XML-muotoisen tiedon suojaamista. Kun esimerkiksi elektronisessa liiketoiminnassa XML-muotoinen tiedonsiirto yleistyy, on tärkeää estää arkaluontoisen tiedon – esimerkiksi pankkitilin numeron tai henkilötunnuksen - vuotaminen ulkopuolisten käsiin. Dillawayn ym. (2002) mukaan salakirjoitettava tieto voi olla mielivaltaista XML-muotoista tietoa. Salakirjoituksen tulos on XML-muotoista - se sisällytetään <EncryptedData>-elementin sisälle, joka sisältää alkuperäisen tiedon yhtenä lapsielementtinsä sisältönä tai viittaa siihen URI:n avulla.

Mactaggart (2001) kuvaa, kuinka yksi XML:n eduista on etsimisen selkeys. Hänen mukaansa tämä etu ei ole enää voimassa, jos XML-dokumentti salakirjoitetaan kokonaan. Ratkaisuna on se, että elementtejä voidaan salakirjoittaa yksitellen. Juuri tämän takia XML-dokumenteille on tarvitaan oma salakirjoitusmenetelmä.

Seuraavaksi esitellään muutamia esimerkkejä XML-salakirjoituksen käytöstä, mukaillen W3C:n spesifikaation esimerkkejä. Niiden on tarkoitus selventää XML-salakirjoituksen ideaa, käyttötarkoituksia ja mahdollisuuksia. Esimerkki 1 on lyhyt XML-dokumentti Maiju Virtasen laskutustiedoista. Esimerkissä 2 koko dokumentti on salakirjoitettu. Tässä alkuperäinen xml-dokumentti on sisällytetty <EncryptedData>-elementin sisälle. Koko dokumentin salakirjoittaminen on käytännöllinen silloin, kun ei haluta paljastaa edes sitä, että kyse on laskutus- ja luottokorttitiedoista.

ESIMERKKI 1. XML-muotoinen kuvaus Maiju Virtasesen laskutustiedoista.

```
<?xml version=’1.0’?>
<Laskutustieto xmlns=’http://esimerkki.org/laskutus’>
   <Nimi>Maiju Virtanen</Nimi>
   <Luottokortti Raja=’500’ Currency=’Euro’>
      <Numero>8975 2222 1234</Numero>
      <Pankki>Oma pankki</Pankki>
      <Raukeamispäivä>10/03</Raukeamispäivä>
   </Luottokortti>
</Laskutustieto>
```

ESIMERKKI 2. Laskutustiedot kokonaan salakirjoitettuna.

```
<?xml version=’1.0’?>
<EncryptedData xmlns=’http://www.w3c.org/2001/04/xmlenc#’
Type=’http://www.isi.edu/in-notes/iana/assigments/media-
types/text/xml’>
   <CipherData>
      <CipherValue>D36K78L87</CipherValue>
   </CipherData>
</EncryptedData>
```

Esimerkissä 3 on salakirjoitettu vain osa dokumentista. Jossain tapauksessa vastapuolen on tarpeellista tietää lähettäjän nimi ja luottoraja, mutta ei luottokortin numeroa eikä raukeamispäivää. Tällainen tapaus voi olla esimerkiksi silloin, kun luottokortilla tehtävät ostokset ovat arvoltaan hyvin suuria. Tällöin se, että luottoraja ostajalla on tarpeeksi suuri, varmistaa myyjälle ostajan luotettavuuden maksukyvyn osalta.

ESIMERKKI 3. Laskutustiedot vain osittain salakirjoitettuna.

```
<?xml version=’1.0’?>
<Laskutustieto xmlns=’http://esimerkki.org/laskutus’>
   <Nimi>Maiju Virtanen</Nimi>
   <Luottokortti Raja=’3000’ Valuutta=’Euro’>
```

```
      <EncryptedData xmlns='
http://www.w3c.org/2001/04/xmlenc#'
Type='http://www.w3.org/2001/04/xmlenc#Content'>
         <CipherData>
            <CipherValue> D36K78L87</CipherValue>
         </CipherData>
      </EncryptedData>
   </Luottokortti>
</Laskutustieto>
```

<EncryptedData>-elementit eivät saa spesifikaation mukaan (Dillawayn ym. 2002) sisältää toisiaan. Kuitenkin salakirjoitettua tietoa <EncryptedData>-elementteineen saa salakirjoittaa edelleen. Joskus, kun samaa dokumenttia tai lomaketta täyttävät monta osapuolta eri aikaan, dokumentteja pitää pystyä salakirjoittamaan moneen kertaan.

3.2.4 XML-allekirjoituksen ja XML-salakirjoituksen suhteesta

Sekä XML-allekirjoituksessa että –salakirjoituksessa on vielä monia ongelmia, vaikka paljon kehitystä onkin tapahtunut. Varsinkin niiden käyttö yhtäaikaa voi aiheuttaa ongelmia niiden luotettavuuden varmistamisessa. Kun varmistetaan allekirjoituksen oikeellisuus, täytyy ainakin tietää, oliko allekirjoitus tehty salakirjoitetulle vai ei-salakirjoitetulle elementeille. (Dillaway ym. 2002)

On tilanteita, joissa samaa dokumenttia käyttää monia osapuolia eri aikaan. Kuvitellaan esimerkiksi, että vuokranantaja lähettää seuraavan sähköisen dokumentin Pekalle: “Vuokraa pitää maksaa 250 euroa viikon kuluessa.” Dokumentti sisältää myös vuokranantajan tilinumeron, joka on salakirjoitettu pankin julkisella avaimella. Seuraavaksi Pekka kirjoittaa tilinumeronsa tähän samaan dokumenttiin, allekirjoittaa sen ja lähettää takaisin vuokranantajalle. Myös Pekan tilinumero on salakirjoitettu. Pankkiin siis lähetetään sellainen dokumentti, joka on ensin salakirjoitettu, sitten allekirjoitettu ja taas salakirjoitettu.

Kuitenkaan, tällaisessa tapauksessa, jossa allekirjoituksen jälkeen dokumentin osaa vielä salakirjoitetaan, allekirjoitus ei enää ole varmennettavissa. (Decryption Transform –spesifikaatio: Imamura ja Maruyama, 2001) XML-allekirjoitusta ja –salakirjoitusta käytettäessä samaan dokumenttiin ja kun dokumenttia vielä muokkaavat eri ihmiset, muodostuu dokumentista helposti monimutkainen algoritmien ja moninkertaisten allekirjoitusten ja salakirjoitusten rypäs. On ongelmallista, jos joidenkin prosesseiden suoritusjärjestys aiheuttaa sen, että dokumentin eheyttä tai autentikointia ei voida enää varmistaa.

Tähän ongelmaan on W3C:ssä kehitetty “decryption transform” eli vapaasti suomennettuna algoritmien purkamisten järjestely. (Imamura ja Maruyama, 2001) Spesifikaatio esittää metodin, jolla voidaan purkaa sellaisten dokumentin osien allekirjoitus, jotka pitää vielä salakirjoittaa. Ideana on, että dokumentin historiaa tarkkaillaan tällaisen virheellisen koodausjärjestyksen estämiseksi taikka paremminkin dokumentti muodostetaan ilman tätä. Dokumentin prosessointijärjestyksen voi määrittää XML-allekirjoituksen <Transform>-elementin avulla. <Transform>-elementin tarkempaa käyttöä ei tässä käydä läpi.

3.3 Esimerkki luottamuksen hallitsemista: Annotea-projekti

Tässä luvussa esitellään esimerkkinä luottamuksen hallitsemisesta Annotea-projektia ja siinä kehiteltyjä menetelmiä. Kahan ja Koivunen (2001) ovat määritelleet Annotean seuraavasti. Annotea on web-pohjainen annotointi- eli huomautusjärjestelmä, joka pohjautuu RDF-metatietomalliin (RDF infrastructure). Annotoinnit (annotations) eli huomautukset ovat dokumentin ulkoista metatietoa ja ne voidaan tallentaa annotointi-palvelimille (annotation servers).

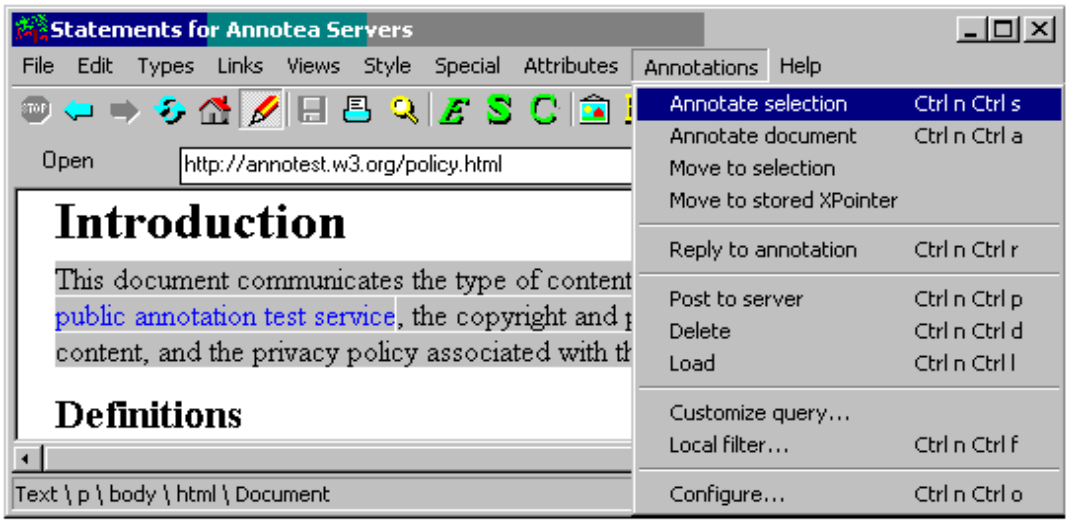
Kielikoneen NetMot-sanakirja antaa seuraavanlaisia suomennoksia englanninkieliselle sanalle ”annotation”.

> s, 1. yks, huomautusten teko, selitysten teko, 2. yl mon, huomautus, muistiinpano, (tekstin) selitys, kirjan selostus, kommentaari, merkintä

Sillä siis tarkoitetaan huomautuksia, kommentteja ja muistiinpanoja. Annotea-projektissa sillä tarkoitetaan sellaista huomautusten tekoa, kuvailua ja arvostelua, joita tehdään webissä oleville tietoresursseille. Tässä projektissa huomautuksilla on aivan erityinen luonne ja toteutustapa, joten siksi tässä tutkielmassa käytetään niistä suoraa käännöstä suomenkieleen eli puhutaan annotoimisesta (verbi) ja annotoinnista (substantiivi).

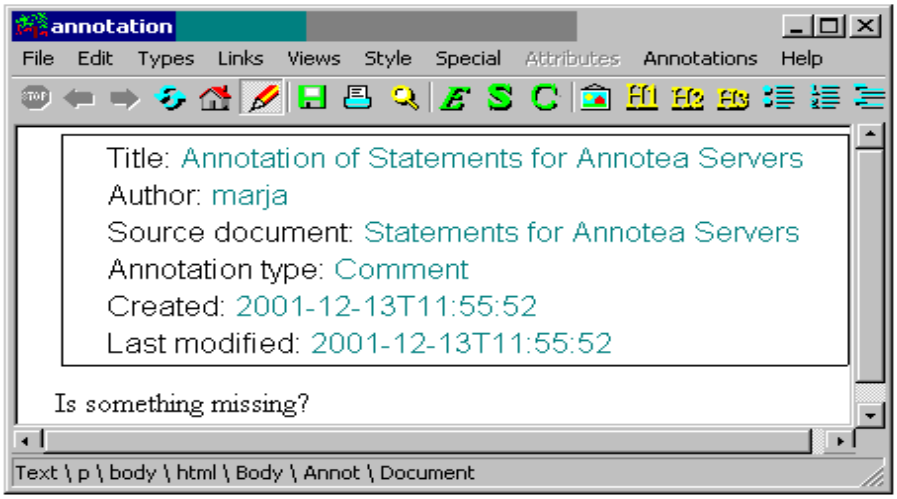
Käytännössä annotointeja voi tehdä Amaya-selaimella. Annotointi liitetään dokumenttiin XPointerin avulla (Kahan ja Koivunen, 2001) ja siksi annotoidut dokumentit täytyvät olla xml-muotoisia eli vähintään XHTML:llä kirjoitettuja. (Daniel, DeRose ja Maler, 2001). Tämä aiheuttaakin sen, että nykyiseen webiin annotoinnit eivät ole aivan käytännöllisiä, koska suurin osa dokumenteista siellä on kirjoitettu HTML:llä ja niihin ei annotointeja voida lisätä. Kuitenkin annotointien keksiminen on suuri kehitysaskel ja luultavasti niitä aletaan enemmän käytää XML:n käytön lisääntyessä.

Seuraavat kuvat ovat suoraan Koivusen tekemästä Annotea Quick Tutorialista (2001) ja niillä on tarkoitus kuvata, kuinka annotointeja käytännössä tehdään. Kuviossa 4 näytetään annotoinnin lisääminen dokumenttiin. Annotointi voidaan siis lisätä pieneen dokumentin osaan ja tämä tapahtuu mustaamalla haluttu osa, jonka jälkeen annotointi lisätään.



KUVIO 4. Annotoinnin lisääminen dokumenttiin tai mustattuun dokumentin osaan.

Kun haluttu dokumentin osa on valittu, seuraavaksi luodaan annotoinnille sisältö. Kuviossa 5 on Amaya-selaimen annotointi-ikkuna, johon voi lisätä esimerkiksi annotoinnin otsikon ja kirjoittajan. Varsinainen annotointi kirjoitetaan loppuun (tässä: ”Is something missing?”). Annotoinnin kirjoittamisen jälkeen se tallennetaan annotointi-palvelimelle. Kuviossa 6 näytetään, kuinka annotointi esitetään webissä. Annotointi näkyy kommentoidussa dokumentissa kynän kuvana, josta annotoinnin saa kokonaan näkyviin. Annotoinnit toimivat lisätietoina, jotka saattavat olla hyödyksi kenelle tahansa dokumentin lukijoista.

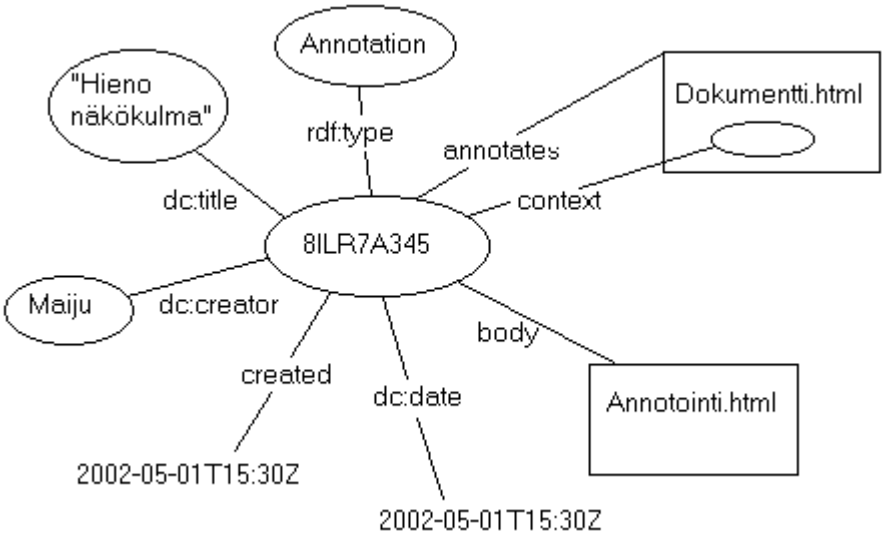


KUVIO 5. Annotoinnin sisällön luominen.

KUVIO 6. Annotointien esittäminen webissä.

Annotea-projektissa annotoinnit on kirjoitettu RDF/XML:llä. Ne tallennetaan annotointi-palvelimelle käyttäen HTTP-protokollaa. Annotointi-malli käyttää monenlaisia RDF-skeemoja, kuten Dublin Core (dc:), kuvaamaan annotoinnin perusominaisuudet. Nämä ominaisuudet on kuvattuna RDF-mallin mukaisessa kuvassa kuviossa 7 mukaillen Koivusta ja Swickia. Kuviossa 7 oleva annotates-ominaisuus tarkoittaa annotoitua dokumenttia. Käytännössä siis annotointi-palvelimelle tallennetaan annotates-ominaisuuteen sen dokumentin URI, jota on annotointu. Context-ominaisuus viittaa annotoinnin todelliseen sijaintiin dokumentissa eli määrää tarkalleen sen dokumentin kohdan ja osan, johon annotointi on tarkoitettu. Body-ominaisuus sisältää varsinaisen annotoinnin sisällön eli kommentin dokumentista ja dc:title ja muut ominaisuudet kuvaavat annotointia. (Koivunen, Swick 2001) Nämä ominanaisuudet siis sisältyvät jokaiseen annotointiin.

Koivusen ja Swickin mukaan (2001) tätä skeemaa voi joustavasti laajentaa. Annotointiin voi myös vastata toisella annotoinnilla. Tällöin annotointi esitetään kommentoidun annotoinnin jäljessä. Tällöin kuviossa 7 olevaan annotointi-malliin täytyy lisätä ainakin reply-to -ominaisuus, joka määrittää mihin annotointiin vastattiin. Kun taas on kyse jaetuista kirjanmerkeistä (shared bookmarks) tarvitaan lisäksi kategorisointi-ominaisuus. Missään näistä tapauksista ei tarvitse tehdä muutoksia annotointi-palvelimelle.



KUVIO 7. Annotointi RDF-mallin mukaisena kuvana Koivusen ja Swickin (2001) kuvaa mukaillen.

Käyttökohteita annotoinneille on useita. Koivunen ja Swick esittelevät artikkelissaan (2001) metatietomallin hyötyjä yhteistyötä tukevissa sovelluksissa. He pohjustavat aihetta sillä, että kun yhteistyötä tehdään webin kautta, pelkkä dokumenttien julkaiseminen siellä ei riitä. Tällöin myös palaute ja vuorovaikutus ovat ehdottoman tärkeitä. Annotointien periaate on se, että web-dokumentin lukija voi lisätä kommentteja ja kysymyksiä dokumentin sisällöstä, eikä hänellä kuitenkaan tarvitse olla kirjoitusoikeutta dokumenttiin.

Koivunen ja Swick esittävät kolme konkreettista käyttöesimerkkiä annotoinneille. Ensimmäinen koskee yhteistyötä. Opiskelijat tekevät yhteistä raporttia ja annotoinnit helpottavat keskustelemaan aiheesta, kun voidaan viitata täsmällisesti dokumentteihin. Kun he kokoavat listaa lähteistä, jokainen arvioi lähteen kiinnostavuuden ja määrittelee kategorian, jonne lähde kuuluu. He lisäävät tarpeen mukaan lähteisiin kommentteja, kysymyksiä ja avainsanoja. Samaan tapaan toisessa esimerkissä lähteiden etsimiseen käytetään jaettuja kirjanmerkkejä (shared bookmarks). Kirjanmerkeissä on tärkeää valita niille kategoria, joka kuuluu erityiseen ontologiaan eli yhteisesti määriteltyyn sanastoon. Ryhmän jäsenillä on yhteinen annotointi-palvelin, jonne metatieto tallennetaan.

Nämä ovat erityisen kiinnostavia esimerkkejä tämän tutkielman ja luotettavuuden kannalta. Jaetut annotoinnit kuvaavat luottamusta tietyn yhteisön välillä. Yhteisöllä on yleensä samantapainen tausta ja samanlaiset tavoitteet. Niinpä yhteisön jäsen voi hyötyä toisten jäsenien erilaisille lähteille valitsemista kategorioista. Kategoriaesimerkkejä voisivat opiskelijaesimerkkiin liittyen olla esimerkiksi: Relevantit lähteet/primäärilähteet, epärelevantit lähteet, sekundaarilähteet/taustatiedot. Yhteisön jäsenet voivat luottaa toistensa merkitsemiin kommentteihin ja lähteiden luokitteluun, joten jokaisen yhteisön jäsenen ei tarvitse tarkistaa jokaista lähdettä erikseen, varsinkaan epärelevantteja lähteitä.

Metatieto tallennetaan Annoteassa dokumentin ulkopuoliselle annotointi-palvelimelle. Edes annotointilinkin tekeminen ei muuta alkuperäistä dokumenttia, koska linkki on ainoastaan annotointi-palvelimella ja näkyy dokumentissa selaimen ominaisuutena. Eduiksi ulkoiselle metatiedolle Marja-Riitta Koivunen ja Ralph Swick (2001)

mainitsevat sen, että vältytään päällekkäisiltä kirjoitustapahtumilta samaan dokumenttiin ja tietojen menetyksiltä konfliktoivissa päivityksissä. Muutenkin tekijänoikeuslait kieltävät toisen tekemän dokumentin muokkaamisen. Tällöin olisi vaarana, että raja alkuperäisen dokumentin ja annotointien välillä hämärtyisi.

4 YHTEENVETO

Tämän tutkielman tavoitteena oli kartoittaa XML-pohjaisia menetelmiä ja tekniikoita luottamuksen varmistamiseksi. Luottamukseen esitettiin kaksi perusnäkökulmaa: Luottamuksellisten tietojen salassa pysyminen ja se, että miten voimme tietää, mihin verkossa olevaan palveluun tai tietoon voimme yleensäkään luottaa. Pohdittiin myös luottamuksen syntymistä ihmismielessä Bickmoren ja Casselin mallissa, jossa luottamus syntyy vähitellen vaikkapa kevyen jutustelun myötä. Neljää keskeistä tekniikkaa eli P3P, XML-allekirjoitus ja XML-salakirjoitus sekä Annotea-projekti esiteltiin. Vaikka tiedot tarkistettiinkin aina W3C:n spesifikaatioista, ei raportissa käsitelty teknisiä yksityiskohtia tarkasti. Kartoitettiin olemassa olevia menetelmiä ja niiden käyttökohteita luottamuksen hallitsemiseen. Ongelmia ja puutteita luottamuksen hallinnassa esiintyy vielä useita, mutta näitä ongelmakohtia ei olla ryhdytty analysoimaan. Esiteltiin lähinnä jokaisen tekniikan perusominaisuuksia ja mahdollisuuksia, mitä sillä voidaan toteuttaa. Tekniikoita tulkittiin perusnäkökulmien puitteissa ja pohdittiin kuinka tekniikat tukevat ja ratkaisevat näiden näkökulmien ilmaisemia ongelmakohtia.

Luottamuksen hallinnassa on useita ongelmia, koska webin käyttäjä- ja tietomäärä kasvaa koko ajan niin huimaa vauhtia. Tämän suuren informaatiotulvan hallitsemiseksi on syytä kehittää uusia menetelmiä. Web-dokumentteihin pitäisi pystyä liittämään niiden sisältöä kuvaavaa ja niitä arvioivaa metatietoa. XML on joustavuutensa vuoksi tulossa keskeiseksi tekniikaksi webissä. Tiedon salausmenetelmiä on kehitettävä myös XML-muotoiselle tiedolle. Erityisesti elektronisen liiketoiminnan yleistymisen myötä webissä liikkuu paljon arkaluontoista ja luottamuksellista tietoa. On ensiarvoisen tärkeää, että nämä tiedot eivät vuoda ulkopuolisten käsiin.

Tutkielman keskeisenä tuloksena on, että luottamuksen ilmaisemiseen löytyy monia potentiaalisia ja käyttökelpoisia tekniikoita. Luotettavan tiedon tunnistamiseen liittyy Annotea-projekti, jonka tekniikoiden avulla metatietoa ja kommentteja webin

resursseista voidaan joustavasti ja helposti esittää ja hallita ja jakaa näin luottamusta yhteisöjen välillä.

Luottamuksellisten tietojen salassa pitämiseen esiteltiin kolme tekniikkaa. Tietoa voidaan salata XML-salakirjoituksen ja –allekirjoituksen avulla. XML-allekirjoitus käyttää hyväkseen julkisen avaimen järjestelmää ja varmistaa osapuolten tunnistuksen, tiedon eheyden ja kiistattomuuden. XML-salakirjoitusta ja –allekirjoitusta voidaan käyttää myös yhdessä, mutta silloin täytyy olla varovainen, että allekirjoitus ei menetä kelpoisuuttaan. Näin käy, jos tieto allekirjoituksen jälkeen vielä salakirjoitetaan. Niinpä järjestys salakirjoituksessa ja allekirjoituksessa täytyy olla aina tämä.

Luottamuksellisien tietojen salassa pitämistä voi hallita myös P3P-yhteyskäytännön (The Platform for Privacy Preferences) avulla. Sen periaatteiden mukaisesti palveluntarjoaja määrittelee tietosuojakäytännöt, joita käyttäjän selaimen on mahdollista tulkita. Tämä helpottaa käyttäjän päätöksentekoa ja käyttäjä voi P3P-tietosuojamäärittelyjen perusteella varmistua siitä, että luottamuksellisia tietoja, kuten henkilötietoja, ei käytetä väärin.

Muutaman viime vuoden aikana on otettu suuria kehitysaskelia niin luottamukseen liittyvien tekniikoiden kuin muidenkin webin tekniikoiden kehittämisessä. Kuitenkin ongelmia ja kehitysalueita löytyy vielä paljon ja tutkimus on monessa kohdin vielä kesken. Osa tekniikoista ei ole vielä edes suositusasteella, joten niitä voidaan vielä muokata paljon, yhdistellä tai mahdollisesti haudata kokonaan. Aika näyttää mitkä tekniikat todella otetaan vastaan.

5 LÄHTEET

Bickmore T., Cassell J. 2001. Relational Agents: A Model and Implementation of Building User Trust. ACM Digital Library 1(3), 396-403.

Bigham J., Borrell J., Robles S., Tokarchuk L., Cuthbert L. 2001. Design of a trust model for a secure multi-agent marketplace. ACM Digital Library 1(5), 77-78.

Boyer J. 2001. Canonical XML Version 1.0. W3C Recommendation [online]. [viitattu 25.4.2002]. Saatavilla www-muodossa < http://www.w3.org/TR/xml-c14n >.

Cranor L., Langheinrich M., Marchiori M, Presler-Marshall M. & Reagle J. 2002. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3-C Recommendation [online]. [viitattu 5.4.2002]. Saatavilla www-muodossa < http://www.w3.org/TR/P3P/>.

Cranor L., Langheinreich M. & Marchiori M. 2002. A P3P Preference Exchange Language 1.0 (APPEL1.0). W3C Working Draft [online], [viitattu 5.4.2002]. Saatavilla www-muodossa <http://www.w3.org/TR/P3P-preferences/>.

Daniel Jr. R., DeRose S. & Maler E. 2001. XML Pointer Language (XPointer) Version 1.0. W3C Candidate Recommendation [online]. [viitattu 2.5.2002]. Saatavilla www-muodossa < http://www.w3.org/TR/xptr/>.

Dillaway B., Imamura T. & Simon E. 2002. XML Encryption Syntax and Processing. W3C Candidate Recommendation [online]. [viitattu 22.4.2002]. Saatavilla www-muodossa < http://www.w3.org/TR/xmlenc-core/ >.

Fox B., Bartel M., Boyer J., LaMaccia B. & Simon E. 2001. XML Signature Syntax and Processing. W3C Candidate Recommendation [online]. [viitattu 22.4.2002]. Saatavilla www-muodossa < http://www.w3.org/TR/xmldsig-core/ >.

Grimm R. & Rossnagel A. 2000. Can P3P Help to Protect Privacy Worldwide? ACM Digital Library 11(1), 157-160.

Hyvönen E. 2001. Semantic Web – kohti uutta merkitysten Internetiä [online], [viitattu 15.3.2002]. Saatavilla www-muodossa < http://www.cs.helsinki.fi/u/eahyvone/stes/semanticweb/SemanticWebVisio.PDF >.

Imamura T. & Maruyama H. 2001. Decryption Transform for XML Signature. WG Working Draft [online]. [viitattu 25.4.2002]. Saatavilla www-muodossa <http://www.w3.org/TR/2001/WD-xmlenc-decrypt-20010626 >.

Koivunen M., Swick R. 2001. Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond [online], [viitattu 13.3.2002]. Saatavilla www-muodossa <http://www.w3.org/2001/Annotea/Papers/KCAP01/annotea.html>.

Koivunen M. 2001. Annotea Quick Tutorial [online], [viitattu 13.3.2002]. Saatavilla www-muodossa
< http://www.w3.org/2001/Annotea/User/Tutorial/quicktutorial.html>.

Kahan J., Koivunen M., Prud'Hommeaux E., Swick R. 2001. Annotea: An Open RDF Infrastructure for Shared Web Annotations [online], [viitattu 13.3.2002]. Saatavilla www-muodossa < http://www.w3.org/2001/Annotea/Papers/www10/annotea-www10.html >.

Mactaggart M. 2001. An Introduction to XML encryption and XML signature [online], [viitattu 15.4.2002]. Saatavilla www-muodossa < http://www-106.ibm.com/developerworks/xml/library/s-xmlsec.html/index.html >.

Simon E., Madsen P. & Adams C. 2001. An Introduction to XML Digital Signatures [online], [viitattu 15.4.2002]. Saatavilla www-muodossa <http://www.xml.com/pub/a/2001/08/08/xmldsig.html >.

ATK-sanakirja [online]. Tietotekniikan liitto ry [viitattu 22.4.2002]. Saatavilla www-muodossa < http://www.ttlry.fi/sana/sisallys.htm >.

Tietoturva: FAQ [online]. Viestintävirasto [viitattu 22.4.2002]. Saatavilla www-muodossa < http://www.ficora.fi/suomi/tietoturva/faq.htm >.