TIETO- JA SÄHKÖTEKNIIKAN TIEDEKUNTA

Sara Nikula

ÄLYKKÄÄN LIIKENTEEN VAROITUSVIESTIEN KVANTTITURVALLINEN ALLEKIRJOITTAMINEN

Diplomityö
Tietotekniikan tutkinto-ohjelma
Lokakuu 2021

Nikula S. (2021) Älykkään liikenteen varoitusviestien kvanttiturvallinen allekirjoittaminen. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 61 s.

TIIVISTELMÄ

Älykkäässä liikenteessä ajoneuvot viestivät keskenään parantaen näin liikenteen sujuvuutta ja turvallisuutta. Viestinnän luotettavuus taataan digitaalisilla allekirjoituksilla, jotka varmistavat viestien aitouden ja alkuperän. Nykyisten digitaalisten allekirjoitusten luotettavuus on vaarassa kvanttitietokoneiden kehityksen myötä, koska kvanttitietokoneen avulla niiden taustalla olevien matemaattisten ongelmien ratkaisu helpottuu huomattavasti. On kuitenkin olemassa myös kvanttiturvallisia allekirjoitusalgoritmeja, joilla tuotetut allekirjoitukset eivät ole alttiita kvanttitietokoneella toteutetuille hyökkäyksille.

Tässä diplomityössä liitettiin älykkään liikenteen varoitusviesteihin kvanttiturvallinen allekirjoitus. Toteutuksen pohjana toimivat älykkään liikenteen viestintää määrittävät eurooppalaiset tekniset spesifikaatiot ja kolmen kvanttiturvallisen allekirjoitusalgoritmin valmiit toteutukset. Näitä yhdistelemällä luotiin ohjelma, jossa älykkään liikenteen varoitusviestit allekirjoitetaan kvanttiturvallisesti. Kunkin kvanttiturvallisen allekirjoitusalgoritmin soveltuvuutta liikennekäyttöön arvioitiin vertaamalla sitä nykyisin käytössä oleviin allekirjoitusalgoritmeihin. Vertailussa huomioitiin allekirjoitukseen ja varmennukseen tarvittava aika sekä syntyneiden viestien koko. Tuloksista nähdään, että vertaillut allekirjoitusalgoritmit eroavat toisistaan sekä suoritusnopeuden että syntyneiden viestien koon puolesta. Tulosten perusteella kvanttiturvallisia allekirjoitusalgoritmeja voitaisiin käyttää älykkäässä liikenteessä ilman, että viestien käsittelynopeus kärsisi kohtuuttomasti.

Avainsanat: digitaalinen allekirjoitus, liikenne, kvanttiturvallinen kryptografia, kryptografia

Nikula S. (2021) Quantum-Safe Signing of Notification Messages Sent by Intelligent Transport Systems. University of Oulu, Degree Programme in Computer Science and Engineering, 61 p.

ABSTRACT

Intelligent transport systems improve safety and fluency in traffic by utilizing communication between vehicles. Reliability of this communication is ensured by digital signatures which confirm origin and authenticity of the sent messages. Digital signature algorithms used today are at risk since quantum computers are being developed and could be used to solve mathematical problems underlying these algorithms. However, quantum-safe digital signature algorithms which cannot be broken even by attacks of quantum computers do exist.

In this master's thesis, quantum-safe digital signatures are integrated into notification messages used by intelligent transport systems. The program implemented is based on reference implementations of three quantum-safe digital signature algorithms and European technical specifications regarding communication between intelligent transport systems. By integrating these, a new implementation is constructed, where notification messages are signed using quantum-safe digital signature algorithms. Their suitability for this use is evaluated by measuring their speed and the size of the signed messages, and comparing them with the algorithms used today. The results show that these quantum-safe algorithms perform differently with regard to time required to sign and verify the sent messages as well as the size of these messages. Based on these results, quantum-safe digital signature algorithms could be used by intelligent transport systems with only moderate changes to performance.

Keywords: digital signature, transport, post-quantum cryptography, cryptography

SISÄLLYSLUETTELO

ALKULAUSE

Tämä diplomityö tehtiin Teknologian tutkimuskeskus VTT Oy:lle osana Business Finlandin rahoittamaa Post-Quantum Cryptography -hanketta. Kiitos ohjaajilleni Kimmo Haluselle ja Visa Vallivaaralle työn ohjauksesta ja neuvoista. Kiitos myös kaikille kollegoille, jotka ovat auttaneet työn valmistumista kommentoimalla keskeneräistä työtä ja antamalla vinkkejä hyvistä lähteistä.

Oulussa 4. lokakuuta 2021

Sara Nikula

LYHENTEIDEN JA MERKKIEN SELITYKSET

AT	Authorization Ticket
AVX2	Advanced Vector Extensions 2
CA	Certificate Authority
CAM	Cooperative Awareness Message
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
C-ITS	Cooperative Intelligent Transport Systems
CVP	Closest Vector Problem
DENM	Decentralized Environmental Notification Message
DoS	Denial of Service
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
EN	European Standard
ETSI	European Telecommunications Standards Institute
FPU	Floating Point Unit
HSM	Hardware Security Module
ITS	Intelligent Transport Systems
KEM	Key Encapsulation Mechanism
LDM	Local Dynamic Map
NIST	National Institute of Standards and Technology
OBU	On-Board Unit
PKI	Public Key Infrastructure
PQC	Post-Quantum Cryptography
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
RSA	Rivest–Shamir–Adleman
SIVP	Shortest Independent Vectors Problem
SVP	Shortest Vector Problem
TLS	Transport Layer Security
TR	Technical Report
TS	Technical Specification
V2V	Vehicle-to-vehicle
V2X	Vehicle-to-everything
XOF	extendable-output hash function

1. JOHDANTO

Kuvittele, että ajat autolla moottoritietä pimeänä syysiltana ja voit viestiä edempänä samalla tiellä ajavan auton kuljettajan kanssa. Kuvittele, että tuo kuljettaja havaitsee tien penkalla hirven ja varoittaa sinua tästä. Hidastat tilannenopeuttasi ja muutaman sadan metrin kuluttua ehdit juuri jarruttaa, kun hirvi hypähtää eteesi metsiköstä. Haluaisitko saada tällaisia varoituksia? Kuvittele seuraavaksi, että edelläsi ajavasta autosta alkaa jatkuvasti saapua sinulle viestejä, joissa väitetään, että kilometrin päässä tiellä on lauma poroja. Matka jatkuu läpi taajamien, eikä poroja näy missään, mutta viestien tulo vain jatkuu. Haluaisitko voida erottaa tällaiset turhat huijausviestit oikeista varoitusviesteistä?

Itseajavat autot lienevät monelle tuttu käsite, mutta liikenteen sujuvuutta ja turvallisuutta voidaan kohentaa monilla muillakin tavoilla. Tieverkoston eri paikoissa liikkuvat ajoneuvot muodostavat joukon, jonka kullakin yksittäisellä jäsenellä on hieman erilaista tilannetietoa riippuen siitä, missä maantieteellisessä paikassa ne sattuvat juuri sillä hetkellä sijaitsemaan. Näiden tietojen jakaminen lähistöllä liikkuvien ajoneuvojen kanssa mahdollistaa entistä turvallisemman ja sujuvamman liikenteen.

Automatisaatio ja tekoälyn kehitys sallivat tulevaisuuden korkeasti varusteltujen autojen tehdä itsenäisesti tieturvallisuuteen liittyviä havaintoja esimerkiksi kameroiden ja konenäön avulla. Vaikka kaikkia nykyautoja ei vielä olekaan varustettu konenäöllä, tällaiseen tulevaisuuteen varaudutaan jo nyt valmistelemalla erilaisia standardeja ja spesifikaatioita, jotka tulevaisuudessa ohjaavat ja turvaavat liikenteessä tapahtuvaa ajoneuvojen välistä viestintää. Älykkäässä liikenteessä (ITS, Intelligent Transport Systems) ajoneuvot viestivät langattomasti maantieteellisesti lähellä sijaitsevien tielläliikkujien kanssa. Tavoitteena on tehdä liikenteestä sujuvampaa ja turvallisempaa. Liikenteessä toimitaan usein korkeissa tilannenopeuksissa ja suurten ajoneuvojen kanssa, jolloin luottamus toisiin tielläliikkujiin on tärkeää. Tähän tarpeeseen pyritään vastaamaan digitaalisilla allekirjoituksilla, jotka todentavat viestin lähettäjän identiteetin ja varmistavat sisällön aitouden. Tämän työn pohjana käytetään ETSI:n (European Telecommunications Standards Institute) tuottamia teknisiä spesifikaatioita, jotka määrittelevät älykkäässä liikenteessä tapahtuvan viestinnän digitaalisten allekirjoitusten tyylin ja niiden tuottamiseen käytettävät kryptografiset operaatiot.

Kvanttitietokoneiden kehitys avaa mahdollisuuksia uudenlaiselle laskennalle, mutta samalla nykyisin käytössämme olevat digitaaliset allekirjoitukset joutuvat vaaraan. Ne perustuvat nimittäin matemaattisille operaatioille, jotka ovat lähes mahdottomia ratkaistavia klassisille tietokoneille, mutta jotka kvanttitietokoneet pystyvät ratkaisemaan huomattavasti helpommin. Tarpeeksi tehokas kvanttitietokone romuttaisi myös älykkään liikenteen digitaalisten allekirjoitusten ja siten koko viestinnän luotettavuuden. Viime vuosina on alettu kiinnittää entistä enemmän huomiota kvanttiturvalliseen kryptografiaan (PQC, Post-Quantum Cryptography), jonka murtamista ei voida huomattavasti nopeuttaa kvanttitietokoneella. Kvantitturvallinen kryptografia mahdollistaisi myös kvanttitietokoneen kestävät digitaaliset allekirjoitukset. Tässä diplomityössä yhdistetään älykäs liikenne ja kvanttiturvallinen kryptografia integroimalla kvanttiturvallisia allekirjoitusalgoritmeja älykkään liikenteen viestinnässä käytettäviin teknisiin spesifikaatioihin. Tavoitteena on löytää kvanttiturvallinen allekirjoitusalgoritmi, joka soveltuu liikenteessä

tapahtuvan viestinnän turvaamiseen nyt ja tulevaisuudessa, myös kvanttitietokoneiden aikakaudella.

Tämä tutkielma on jäsennelty seuraavasti. Toisessa luvussa valotetaan työn taustalla olevaa kryptografiaa käymällä läpi digitaalisten allekirjoitusten perusteet, allekirjoitus elliptisillä käyrillä ja kvanttiturvallisen kryptografian perusteet. Lisäksi esitellään kaksi kvanttiturvallisen kryptografian tyyppiä ja tähän työhön valitut algoritmit. Kolmannessa luvussa määritellään älykäs liikenne ja käsitellään sitä koskevia standardeja ja tietoturvauhkia. Neljäs luku kuvailee tätä työtä varten tehdyn ohjelmallisen toteutuksen. Viidennessä luvussa esitellään toteutuksen pohjalta saadut tulokset ja pohditaan niiden merkitystä älykkään liikenteen kannalta. Yhteenveto kertaa lyhyesti kaikki edelliset luvut.

2. DIGITAALINEN ALLEKIRJOITUS JA KVANTTITURVALLINEN KRYPTOGRAFIA

Kryptografia viittaa salaustekniikoiden käyttöön ja tutkimukseen. Salauksen tarkoituksena on muuntaa viesti sellaiseen muotoon, että se on luettavissa tietylle vastaanottajalle, mutta muille ei [1 s. 4]. Kryptografian alaan kuuluvat salatun kommunikaation lisäksi myös viestinnässä käytettävän avaimen vaihto ja viestin aitouden varmistaminen. Nykypäivän tietokoneet suorittavat kryptografisia operaatioita päivittäin esimerkiksi pankkiasioinnissa tai nettisivuja selatessa ilman, että käyttäjän tarvitsee tästä edes tietää [2 s. 3].

Salausjärjestelmät voidaan jakaa kahteen päätyyppiin: symmetrisiin ja epäsymmetrisiin. Symmetrisessä salauksessa samaa avainta käytetään sekä viestin salaamiseen että salatun viestin avaamiseen. Epäsymmetrisessä salauksessa käytössä on kaksi erilaista avainta, joista toinen on julkinen ja toinen yksityinen. Julkista avainta käytetään viestin salaamiseen ja yksityistä avainta salauksen purkamiseen [2 s. 4–5]. Avain tarkoittaa käytännössä tietokoneen muistiin talletettua lukua, jonka suuruusluokka riippuu käytettävästä algoritmista [3].

Digitaalinen allekirjoitus on epäsymmetriseen salaukseen perustuva tapa varmistaa tietyn viestin lähettäjä. Tämä perustuu siihen, että allekirjoittaja laskee allekirjoituksen käyttäen yksityistä avainta, joka vain allekirjoittajalla on hallussaan. Allekirjoituksen paikkansapitävyys tarkistetaan eli varmennetaan yksityisen avaimen kanssa yhteensopivalla julkisella avaimella, joka ei ole salainen. Näin vain yksi taho eli yksityisen avaimen haltija voi allekirjoittaa viestin, mutta kuka tahansa voi tarkistaa allekirjoituksen julkisen avaimen avulla. Allekirjoitus todistaa myös sen, että viesti ei ole muuttunut matkan varrella, eli pätevällä allekirjoituksella varustettu viesti on sama viesti kuin jonka lähettäjä on allekirjoittanut. [2 s. 399–402]

Jotta allekirjoituksesta olisi hyötyä, on tiedettävä, kuka allekirjoituksen loi. **Sertifikaatti** liittää julkisen avaimen ja tietyn identiteetin toisiinsa. Myös itse sertifikaatin paikkansapitävyys täytyy taata, jotta siihen voidaan luottaa. Sertifikaatti liittyy julkisen avaimen infrastruktuuriin (PKI, Public Key Infrastructure), jonka avulla sertifikaatin oikeellisuudesta voidaan varmistua. Sertifikaattiauktoriteetti (CA, Certificate Authority) on luotettavaksi tiedetty taho, jolla on oma julkisen ja yksityisen avaimen parinsa ja joka voi yksityisellä avaimellaan allekirjoittaa muiden infrastruktuurin käyttäjien sertifikaatteja. Sertifikaattiauktoriteetti on siis kolmas osapuoli, johon molemmat viestinnän osapuolet luottavat ja joka todistaa viestinnässä käytettävien sertifikaattien oikeellisuuden.

Käytännön sovelluksissa yksi sertifikaattiauktoriteetti ei välttämättä allekirjoita kaikkia infrastruktuurin käyttäjien sertifikaatteja, vaan se voi valtuuttaa väliasteen auktoriteetteja allekirjoittamaan tavallisten käyttäjien sertifikaatteja puolestaan. Korkein sertifikaattiauktoriteetti allekirjoittaa näiden väliasteen auktoriteettien sertifikaatit, ja tällä tavalla ne voivat todistaa oikeutensa allekirjoittaa tavallisten käyttäjien sertifikaatteja. Tavallisen loppukäyttäjän sertifikaatin oikeellisuus voidaan tarkistaa tarkistamalla väliasteen auktoriteetin allekirjoitus, jonka pätevyys taas voidaan tarkistaa tarkistamalla korkeimman sertifikaattiauktoriteetin allekirjoitus. Tällaista järjestelmää, jossa eri asteiset auktoriteetit allekirjoittavat sertifikaatteja ketjussa, kutsutaan nimellä **sertifikaattiketju**. [4 s. 238–239]

Väärin käytettyjä sertifikaatteja voidaan vetää pois käytöstä ylläpitämällä listaa, johon epäluotettavat sertifikaatit päätyvät. Tästä muut käyttäjät tietävät olla luottamatta kyseisen sertifikaatin esittävään tahoon [2 s. 427]. Esimerkki yleisestä sertifikaattityypistä on X.509-sertifikaatti, joka sisältää tiedot sertifikaatin myöntäjästä, haltijasta, julkisesta avaimesta, voimassaoloajasta ja allekirjoituksessa käytettävästä algoritmista [5].

Allekirjoituksia voidaan luoda useilla epäsymmetriseen salaukseen perustuvilla algoritmeilla. Tähän työhön liittyvässä toteutuksessa käytetään allekirjoituksia, jotka perustuvat joko elliptisiin käyriin, hilaongelmiin tai usean muuttujan polynomeihin. Seuraavissa kappaleissa käsitellään niiden toimintaperiaatteet ja se, millaiset allekirjoitustyypit ovat uhattuna kvanttitietokoneiden aikakaudella.

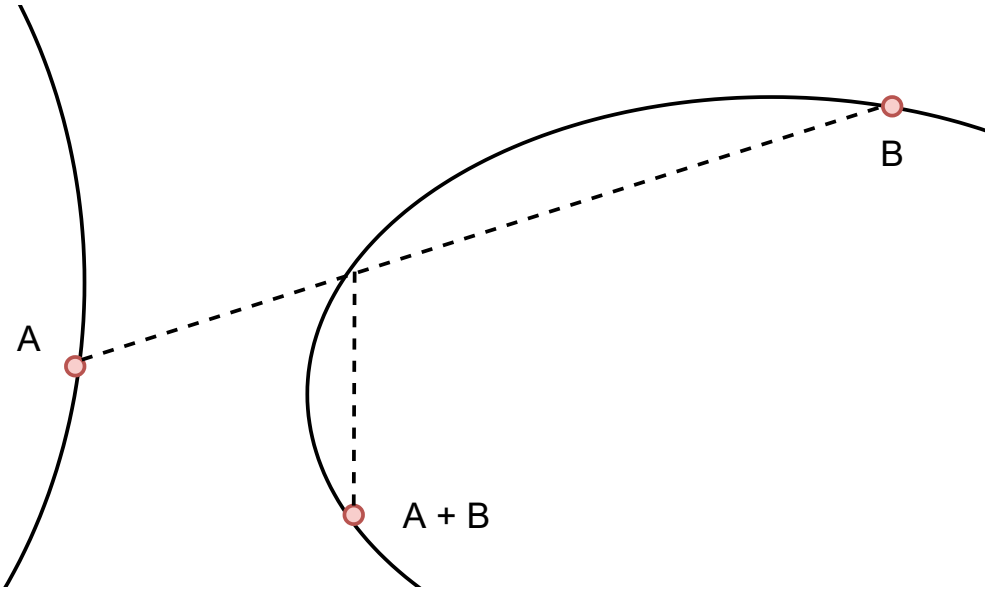
2.1. Allekirjoitus elliptisillä käyrillä

Elliptiset käyrät ovat pinnalla sijaitsevia käyriä, joiden kaikilla pisteillä on x- ja y-koordinaatit. Niitä voidaan käyttää epäsymmetriseen salaukseen, avaintenvaihtoon ja allekirjoitukseen [4 s. 217–218]. Elliptisiä käyriä on useita erilaisia. Tässä tutkielmassa käytettävät elliptiset käyrät ovat muotoa $y^2 = x^3 + Ax + B \bmod p$, ja käytettävät parametrit A, B ja p määrittelevät, mikä käyrä on kyseessä [6, 7].

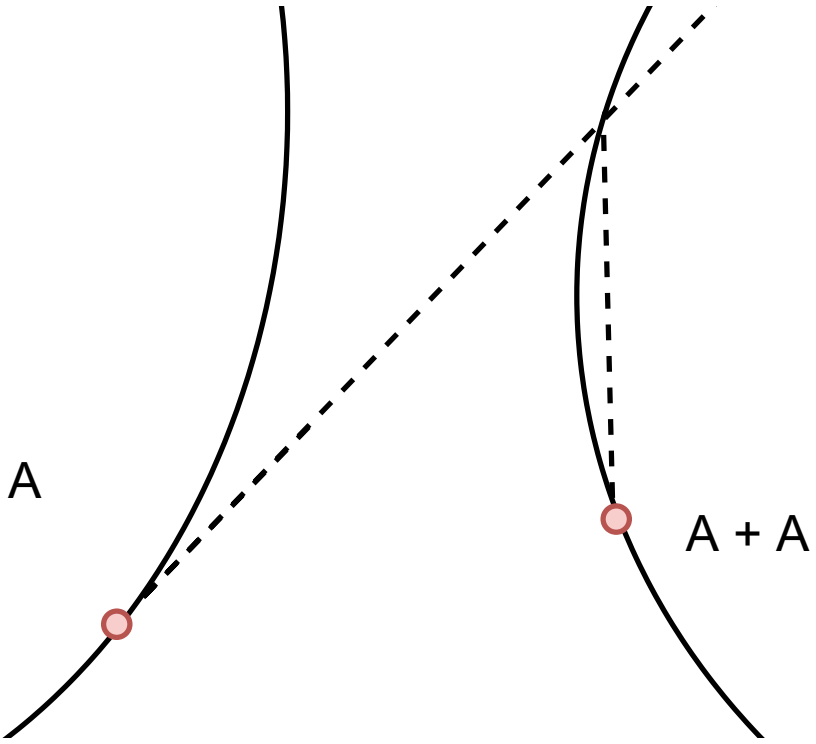
Elliptisten käyrien pisteille on määritelty omat yhteen- ja kertolaskuoperaatiot, jotka ovat erilaisia kuin reaaliluvuilla tehtävät vastaavat operaatiot. Esimerkiksi kahden elliptisen käyrän pisteen yhteenlasku tapahtuu vetämällä viiva pisteiden välille, tarkastelemalla, missä kohtaa tuo viiva leikkaa elliptisen käyrän, ja peilaamalla tämä leikkauskohta x-koordinaatin suhteen. Kuvassa 1 on esimerkki operaatiosta. Pisteen lisääminen itseensä taas tapahtuu siten, että siltä kohdalta, jossa piste sijaitsee, lähdetään seuraamaan elliptisen käyrän mukaista tangenttia niin pitkään, että tangentti leikkaa käyrän seuraavan kerran, ja tämä piste peilataan x-akselin suhteen. Tämä on esitetty kuvassa 2. Pisteen lisääminen itseensä yhden kerran tarkoittaa pisteen kertomista kahdella. Piste voidaan kertoa kolmella lisäämällä se itseensä kaksi kertaa, neljällä lisäämällä se itseensä kolme kertaa ja niin edelleen. Nämä operaatiot mahdollistavat laskutoimitukset elliptisten käyrien pisteillä niin, että lopputulos on myös elliptisen käyrän piste. [4 s. 222–223]

Elliptisillä käyrillä voidaan luoda digitaalisia allekirjoituksia. Allekirjoituksen kohteena ei yleensä ole itse viesti vaan siitä luotu tiiviste. **Tiivistefunktio** on matemaattinen funktio, johon voidaan syöttää allekirjoitettavaksi haluttava viesti, ja funktion ulostulo on kiinteän mittainen merkkijono [2 s. 121]. Tiivisteen käyttäminen alkuperäisen viestin sijaan lyhentää käsiteltävän datan määrää ja siten nopeuttaa prosessia. Allekirjoitusta tarkistaessaan vastaanottaja laskee vastaavan tiivisteen vastaanottamastaan viestistä ja tarkistaa, vastaako mukana oleva allekirjoitus kyseistä tiivistettä.

Elliptisillä käyrillä tehtävä salaus perustuu elliptisten käyrien diskreetin logaritmin ongelmaan (ECDLP, Elliptic Curve Discrete Logarithm Problem). Se tarkoittaa kokonaisluvun k löytämistä, kun tiedetään, että $Q = kP$, missä Q ja P ovat elliptisellä käyrällä sijaitsevia pisteitä. [4 s. 224]



Kuva 1. Esimerkkikuva elliptisen käyrän pisteiden yhteenlaskusta. Pisteet A ja B lasketaan yhteen, ja piste A + B on yhteenlaskun tulos. Kuvassa esiintyy tutkielman kirjoittajan itse keksimä kuvitteellinen käyrä, joka ei liity tässä tutkielmassa käytettäviin elliptisten käyrien allekirjoituksiin.



Kuva 2. Esimerkkikuva elliptisen käyrän pisteen kertomisesta kahdella eli lisäämisestä itseensä. Kuvassa esiintyy tutkielman kirjoittajan itse keksimä kuvitteellinen käyrä, joka ei liity tässä tutkielmassa käytettäviin elliptisten käyrien allekirjoituksiin.

Elliptisten käyrien avainpari koostuu yksityisestä ja julkisesta avaimesta. Yksityinen avain on kokonaisluku d. Julkinen avain on käyrällä sijaitseva piste. Se on saatu kertomalla eräs toinen käyrällä sijaitseva piste, jota kutsutaan generaattoripisteeksi, tuolla yksityisen avaimen kokonaisluvulla. Julkinen avain on siis $P = dG$, missä d on kokonaisluku, G on käyrällä sijaitseva piste ja P näiden kertolaskuna saatu toinen piste. Allekirjoitusvaiheessa viestistä tuotettua tiivistettä käsitellään lukuna, joka on korkeintaan $n - 1$, missä n riippuu valitusta käyrästä. Allekirjoittaja valitsee satunnaisen luvun k, joka myös on korkeintaan $n - 1$. Allekirjoittaja laskee x- ja y-koordinaatit pisteelle kG. G on sama generaattoripiste kuin jota kertomalla julkinen avain P on tuotettu. Se ei ole salainen. Tuotetun pisteen x-koordinaatista lasketaan jakojäännös luvusta n, ja tähän jakojäännökseen viitataan kirjaimella r. Se kerrotaan salaisella luvulla d, lisätään siihen viestin tiivistettä vastaava luku h, jaetaan valitulla luvulla k ja lasketaan vielä jakojäännös luvusta n. Tulos on $(h + rd)/k \pmod n$, ja sitä merkitään kirjaimella s. Allekirjoitus koostuu kahdesta luvusta: s ja r. [4 s. 226]

Allekirjoitus tarkistetaan jakamalla viestistä tuotettua tiivistettä vastaava luku h allekirjoitukseen kuuluvalla s:llä ja tekemällä sama allekirjoituksessa olevalle x-koordinaatille r. Sen jälkeen s:llä jaettu tiiviste kerrotaan käyrän generaattoripisteellä G, ja s:llä jaettu r kerrotaan julkisen avaimen pisteellä P, joka on yhtä kuin piste dG. Allekirjoituksen varmennus tapahtuu laskemalla näin saadut pisteet yhteen ja tarkistamalla, onko pisteen x-koordinaatti sama kuin allekirjoitusvaiheessa tuotettu x-koordinaatti. Matemaattisemmin ilmaistuna lasketaan siis, että

$$u = \frac{h}{s} = \frac{hk}{h + rd} \tag{1}$$

$$v = \frac{r}{s} = \frac{rk}{h + rd} \tag{2}$$

$$uG + vP = uG + vdG \tag{3}$$

$$= \frac{Ghk + Grdk}{h + rd} \tag{4}$$

$$= \frac{Gk(h + rd)}{(h + rd)} \tag{5}$$

$$= Gk \pmod n, \tag{6}$$

missä h on allekirjoitettavasta viestistä tuotettu tiiviste, k on alussa valittu satunnainen luku, P on julkinen avain, r on allekirjoitusvaiheessa tuotettu x-koordinaatti, d on allekirjoittajan yksityiseen avaimeen kuuluva kokonaisluku, G on generaattoripiste ja n riippuu käytettävästä käyrästä.

Laskuissa käytettävä salaista avainta kuvaava kirjain d esiintyy aina kerrottuna x-koordinaatilla r siten, että allekirjoituksen näkijä ei saa alkuperäistä d:n arvoa tietoonsa. Allekirjoituksen laskemisessa käytettyä yksityistä avainta ei siis enää voida päätellä allekirjoituksesta, mutta julkisen avaimen avulla voidaan varmistaa, että

allekirjoitus on tuotettu julkista avainta vastaavalla yksityisellä avaimella. Tämä on mahdollista siksi, että julkinen avain sisältää salaisen luvun d ilman, että d:tä voidaan siitä helposti selvittää. Yksityisen avaimen päätteleminen julkisesta avaimesta vaatisi elliptisten käyrien diskreetin logaritmin ongelman ratkaisemista. [4 s. 224–227]

2.2. Kvanttiturvallinen kryptografia

Nykyisin yleisesti käytössä olevat julkisen avaimen salausjärjestelmät perustuvat ongelmille, jotka on helppo laskea yhteen suuntaan, mutta vaikea toiseen suuntaan, ellei tiettyä auttavaa tietoa ole saatavilla. Kuten edellisessä osiossa mainittiin, elliptisten käyrien kohdalla tämä vaikea tehtävä on diskreetin logaritmin ongelma. Elliptisten käyrien ohella toinen yleinen digitaaliseen allekirjoitukseen käytettävä algoritmi on RSA (Rivest-Shamir-Adleman) [4 s. 181]. Tämän tutkielman ohjelmointiosuudessa ei käytetä RSA:ta, mutta se mainitaan tässä, koska se on hyvin yleisesti käytetty allekirjoitusalgoritmi ja myös altis kvanttitietokonehyökkäyksille. RSA:n turvallisuus perustuu suurten lukujen tekijöihinjakoon [8]. Tekijöihinjako tarkoittaa, että selvitetään, minkä kahden pienemmän luvun tulo jokin suurempi luku on. Tehtävä voidaan pienten lukujen kohdalla laskea helposti, mutta mitä suuremmaksi etsittävä luku käy, sitä vaikeampi tehtävä on [9]. Koska elliptisten käyrien ja RSA:n takana olevien matemaattisten ongelmien ratkaisemiseen tietokoneella menee hyvin kauan [8, 4 s. 171], niihin perustuvia salaus- ja allekirjoitusalgoritmeja pidetään turvallisina. Turvallisuus on kuitenkin uhattuna, jos keksitään tapa nopeuttaa laskentaa, esimerkiksi kvanttitietokoneella.

2.2.1. Kvanttitietokone

Tavallisen tietokoneen pienin informaation yksikkö on **bitti**. Bitillä on kaksi mahdollista tilaa, ykkönen ja nolla, ja se on joka hetki jommassakummassa tilassa. Käytännön tietokoneessa tämä tila määrittyy sähkövarauksen perusteella: sähkövaraus tarkoittaa ykköstä ja sen puute nollaa. Useita bittejä yhdistelemällä saadaan esitettyä monia erilaisia asioita, ja tähän perustuu tietokoneen toiminta. **Tavu** tarkoittaa kahdeksaa bittiä. [10 s. 1–3]

Kvanttitietokone eroaa toimintaperiaatteeltaan klassisesta tietokoneesta, koska sen toiminta perustuu erilaisiin fysiikan lakeihin. Kvanttitietokoneen bittiä kutsutaan **kubitiksi**. Kubitit ovat niin mikroskooppisen pieniä, että niihin pätevät erilaiset fysiikan lait kuin ne, joita kohtaamme arkielämässämme. Näin mikroskooppisen pienet asiat voivat nimittäin olla monessa tilassa yhtä aikaa, siis sekä ykkösiä että nollia samanaikaisesti. Kun kubitin tila mitataan, se näyttäytyy kuitenkin aina joko ykkösenä tai nollana: sen on mittaushetkellä ikään kuin pakko päättää, kummassa tilassa se on. Kubitin pienuuden takia mittaus kuitenkin samalla myös tuhoaa sen tilan, eikä samaa kubittia voida enää käyttää uudelleen laskennassa. Kubittien todennäköisyyttä päätyä eri tiloihin voidaan kuitenkin mitata valmistelemalla useita samanlaisia kubitteja ja mittaamalla niiden tiloja. Näin voidaan päätellä, millä todennäköisyydellä kubitti on ykkönen tai nolla. [10 s. 43–47]

Samoin kuin klassisen tietokoneen kohdalla myös kvanttitietokoneen ohjelmointi perustuu erilaisiin portteihin, joiden läpi informaatio kulkee. Koska kvanttitietokoneen kubitit voivat olla useassa eri tilassa yhtä aikaa, kvanttitietokoneella tapahtuva laskenta on erityisen tehokasta verrattuna normaaliin tietokoneeseen, jonka bitit voivat olla vain yhdessä tilassa kerrallaan [11]. Kvanttitietokone ei kuitenkaan ole yleinen supertietokone, jolla voitaisiin ratkaista helposti mikä tahansa ongelma, vaan sitä on osattava ohjelmoida oikein, jotta siitä on hyötyä. Tarvitaan sopiva algoritmi. Shor [12] julkaisi 1990-luvulla artikkelin, jossa esitellään tapa ratkaista sekä tekijöihinjako että diskreetin logaritmin ongelma polynomiaalisessa ajassa kvanttitietokoneen avulla. Polynomiaalinen suoritusaika on määre, joka kuvaa sitä, kuinka monta operaatiota ongelman ratkaisemiseen vaaditaan. Tällaisessa ajassa tapahtuva ratkaisu olisi huomattava nopeutus verrattuna siihen, kuinka nopeasti kyseisiä ongelmia voidaan nykyisin ratkaista klassisilla tietokoneilla [11 s. 17]. Tämä algoritmi siis nopeuttaa nimenomaan niiden ongelmien ratkaisua, joille sekä elliptiset käyrät että RSA perustuvat.

2.2.2. Kryptografian turvallisuuden määrittely

Kryptografisessa kirjallisuudessa erilaisia salausjärjestelmiä tai avaimia ei yleensä luokitella joko turvallisiksi tai turvattomiksi, vaan niille määritellään erilaisia turvallisuustasoja. Turvallisuustaso määrittää, kuinka paljon aikaa ja työtä salausjärjestelmän murtaminen vaatisi. Jos aikaa ja vaivaa vaaditaan tiettyä raja-arvoa enemmän, voidaan käytännössä katsoa salausjärjestelmän olevan turvallinen: vaikka se onkin teoreettisesti murrettavissa, operaatio kestäisi niin kauan, ettei ole järjellistä ajatella kenenkään pystyvän siihen.

Salausjärjestelmien turvallisuustasot ilmaistaan usein bittimäärinä: n bitin turvallisuustaso tarkoittaa, että noin 2^n laskutoimitusta vaaditaan kyseisen salausjärjestelmän murtamiseen. Esimerkiksi 128-bittinen symmetrisessä salauksessa käytettävä avain tarjoaa 128 bitin turvallisuustason. Avaimen selvittäminen puhtaasti arvaamalla ilman mitään lisäinformaatiota siis tarkoittaisi, että hyökkääjän olisi arvattava oikein kaikki avaimen 128 bittiä. Mahdollisia bittiyhdistelmiä on 2^{128}. Tämä on jo niin suuri määrä, että tehokkaankin nykypäivänä saatavilla olevan tietokoneen avulla kaikkien mahdollisten yhdistelmien läpikäymiseen menisi miljardeja vuosia. Hyökkääjän onnistuminen on tietysti kiinni todennäköisyyksistä: ei ole mahdotonta, että hän sattuu löytämään oikean avaimen jo ensimmäisellä yrityksellä, mutta todennäköisyys tälle on häviävän pieni. [4 s. 40–42]

Elliptisten käyrien käyttäminen on tehokasta, koska elliptisten käyrien erityisominaisuuksien vuoksi haluttu turvallisuustaso saavutetaan tekemällä laskutoimituksia huomattavasti muita vaihtoehtoja pienemmillä parametreilla. Esimerkiksi kappaleessa 2.1 esitetyissä laskuissa 256 bittiä pitkän n-parametrin käyttäminen tuottaisi turvallisuustason, joka vastaa 128 bittiä. Vastaavan turvallisuustason saavuttaminen RSA:lla vaatisi laskutoimituksia tuhansia bittejä pitkillä luvuilla [4 s. 225]. Tämän takia elliptiset käyrät ovat suosittu allekirjoitustapa. Epäsymmetrisessä salauksessa käytettävän avaimen turvallisuus ei siis suoraan määrity sen pituuden perusteella, vaan käytettävän algoritmin tyyppi vaikuttaa siihen, kuinka monta bittiä turvallisuutta minkäkin mittaisella avaimella saavutetaan.

Kvanttitietokoneiden kohdalla turvallisuustasoja on tulkittava eri tavalla, koska kvanttitietokoneet pystyvät tekemään tiettyjä laskutoimituksia vähemmillä operaatioilla kuin perinteiset tietokoneet. Tarpeeksi monia kubitteja ja portteja sisältävän kvanttitietokoneen avulla elliptisiin käyriin perustuvan allekirjoituksen murtaminen kävisi huomattavasti nopeammin kuin mitä perinteisiä tietokoneita silmällä pitäen laaditut turvallisuustasot antavat ymmärtää. Joidenkin arvioiden mukaan elliptisiin käyriin perustuva kryptografia on tehokkaammin murrettavissa kvanttitietokoneella kuin RSA:han perustuva. [13]

Artikkelissaan Shor toteaa kvanttitietokoneen rakentamisen näyttävän periaatteessa mahdolliselta, mutta käytännössä hyvin hankalalta tehtävältä [12]. Vaikka niin suuria kvanttitietokoneita, että ne kykenisivät murtamaan nykypäivänä käytössä olevat RSA- tai elliptisten käyrien salaukset, ei ole kirjoitushetkellä tiedossa, kvanttitietokoneet ovat kuitenkin kehittyneet paljon sitten 1990-luvun. Esimerkiksi loppuvuodesta 2019 esiteltiin 53 kubitilla toimiva Sycamore-kvanttiprosessori [14]. Nykyisin käytössä olevien salausten murtamiseksi kubittien määrää pitäisi onnistua nostamaan vielä merkittävästi [15].

Kvanttitietokoneiden kehitys on joka tapauksessa herättänyt huolta nykyisin käytössä olevan kryptografian turvallisuudesta kvanttitietokoneiden aikakaudella. Tämän takia on herännyt tarve uudenlaiselle julkisen avaimen kryptografialle. Kvanttiturvallinen kryptografia tarjoaa riittävän turvallisuustason myös silloin, kun salausta yritetään murtaa kvanttitietokoneen avulla.

2.2.3. Kvanttiturvallisen kryptografian eri tyypit

Kvanttiturvallinen kryptografia on toteutettavissa tavallisella tietokoneella ja se perustuu sellaisiin ongelmiin, joiden ratkaisua ei tämän hetken tiedon mukaan voida nopeuttaa kvanttitietokoneella. Kvanttiturvallinen kryptografia voidaan jakaa useaan eri luokkaan sen perusteella, millaiseen matematiikkaan se perustuu [16]. Joitakin esimerkkejä kvanttiturvallisen kryptografian takana olevista matemaattisista ongelmista ovat virheitä korjaavat koodit, hilaongelmat, usean muuttujan polynomit ja turvalliset tiivistefunktiot [4 s. 263–267].

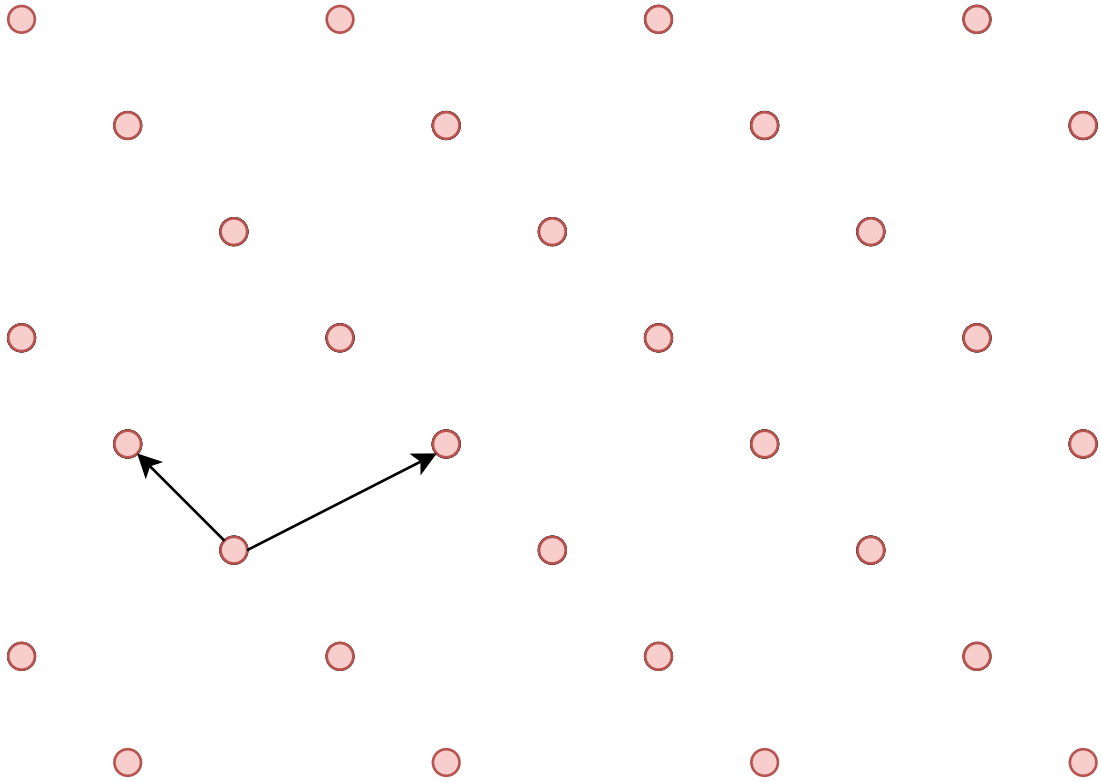
Kvanttiturvallisia salausalgoritmeja voidaan arvioida paitsi suorituskyvyn myös kypsyyden kannalta. Kypsyys tarkoittaa sitä, että algoritmia on tutkittu paljon. Tällaisen algoritmin turvallisuus on luotettavammalla tasolla kuin aivan uuden algoritmin kohdalla, koska voidaan ajatella, että paljon tutkitussa algoritmissa mahdollisesti piilevät turvallisuuspuutteet olisi havaittu tutkimuksen aikana [16]. Uusien algoritmiehdokkaiden kohdalla tutkimusta on luonnollisesti ehditty tehdä vähemmän kuin pitkään tiedossa olleiden algoritmien kohdalla.

Tässä työssä käsitellään hilaongelmiin ja usean muuttujan polynomeihin perustuvia algoritmeja, koska vain näiden tyyppisiä algoritmeja on päässyt NIST:n (National Institute of Standards and Technology) järjestämän kvanttiturvallisen kryptografian kilpailun [17] finaaliin digitaalisten allekirjoitusten sarjassa. Kilpailusta kerrotaan tarkemmin kappaleessa 2.2.4. Hilapohjaisten algoritmien hyvä puoli on se, että niiden turvallisuutta on tutkittu enemmän kuin usean muuttujan polynomeihin perustuvien algoritmien turvallisuutta. Siten niitä voidaan pitää luotettavampana vaihtoehtona. Molempien tyyppien huono puoli tällä hetkellä on se, että ne ovat yhä kehitysvaiheessa,

minkä takia suorituskyky ja käytettävien avainten pituus jättävät vielä toivomisen varaa [16]. Seuraavaksi esitellään lyhyesti molempien allekirjoitustyyppien takana olevan matematiikan perusteet.

Hilaongelmat

Hila on avaruudessa sijaitseva pistejoukko, joka noudattaa tiettyä jaksottaista rakennetta. Hilalla on n kappaletta kantavektoreita, missä n on hilan ulottuvuus. Esimerkiksi kaksiulotteisessa hilassa kantavektoreita on kaksi kappaletta. Hilaan kuuluvat kaikki ne avaruuden pisteet, joihin päädytään, kun näitä kantavektoreita kerrotaan kokonaisluvuilla ja lisätään toisiinsa. Esimerkki kaksiulotteisesta hilasta on kuvassa 3: vaaleanpunaiset täplät ovat hilan pisteitä, ja kuvassa olevat kaksi nuolta ovat hilaan kuuluvia vektoreita. Useampia hilaan kuuluvia vektoreita saadaan virittämällä nuolia minkä tahansa muiden hilan pisteiden välille.



Kuva 3. Esimerkkikuva kaksiulotteisesta hilasta ja kahdesta sen vektorista. Kuvassa esiintyy tutkielman kirjoittajan itse keksimä kuvitteellinen hila, joka ei liity tässä tutkielmassa käytettäviin hila-algoritmeihin.

Hilapohjainen kryptografia perustuu oletukseen hilaongelmien vaikeudesta. Kenties tunnetuin hilaongelma on lyhyimmän vektorin ongelma (SVP, Shortest Vector Problem). Siinä pyritään löytämään hilasta lyhyin mahdollinen vektori, joka kulkee kahden hilaan kuuluvan pisteen välillä. Joissakin versioissa ratkaisuksi voidaan hyväksyä myös vektori, joka ei ole kaikkein lyhyin, mutta kuitenkin tiettyä raja-arvoa lyhyempi. Vaikka kaksiulotteisen hilaesimerkin valossa tehtävä saattaa näyttää helpolta, sen vaikeusaste kasvaa nopeasti, kun hilaan lisätään ulottuvuuksia. Toinen

tunnettu hilaongelma on lähimmän vektorin ongelma (CVP, Closest Vector Problem), jossa hilasta halutaan löytää vektori, joka on lähimpänä tiettyä annettua vektoria. Lyhyimpien riippumattomien vektorien ongelmassa (SIVP, Shortest Independent Vectors Problem) hilasta halutaan löytää useita toisistaan riippumattomia vektoreita niin, että löydetyt vektorit ovat mahdollisimman lyhyitä. [11 s. 147–150]

Hila voidaan esittää matriisimuodossa siten, että kantavektorit kootaan matriisin sarakkeiksi. Hilan eri pisteisiin päästään kertomalla kantavektoreita kokonaislukukertoimilla. Nämä kertoimet voidaan sijoittaa vektoriin, jossa on yhtä monta ulottuvuutta kuin hilassa. Hilan matriisia ja kerroinvektoria keskenään kertomalla päästään mihin tahansa hilan pisteeseen, kun kerroinvektorissa olevia lukuja muutetaan. Matriisilaskun luonteen takia lyhyimmän vektorin ongelman ratkaiseminen on saman tyyppinen laskutoimitus kuin yhtälöryhmän ratkaiseminen. [11 s. 152–154]

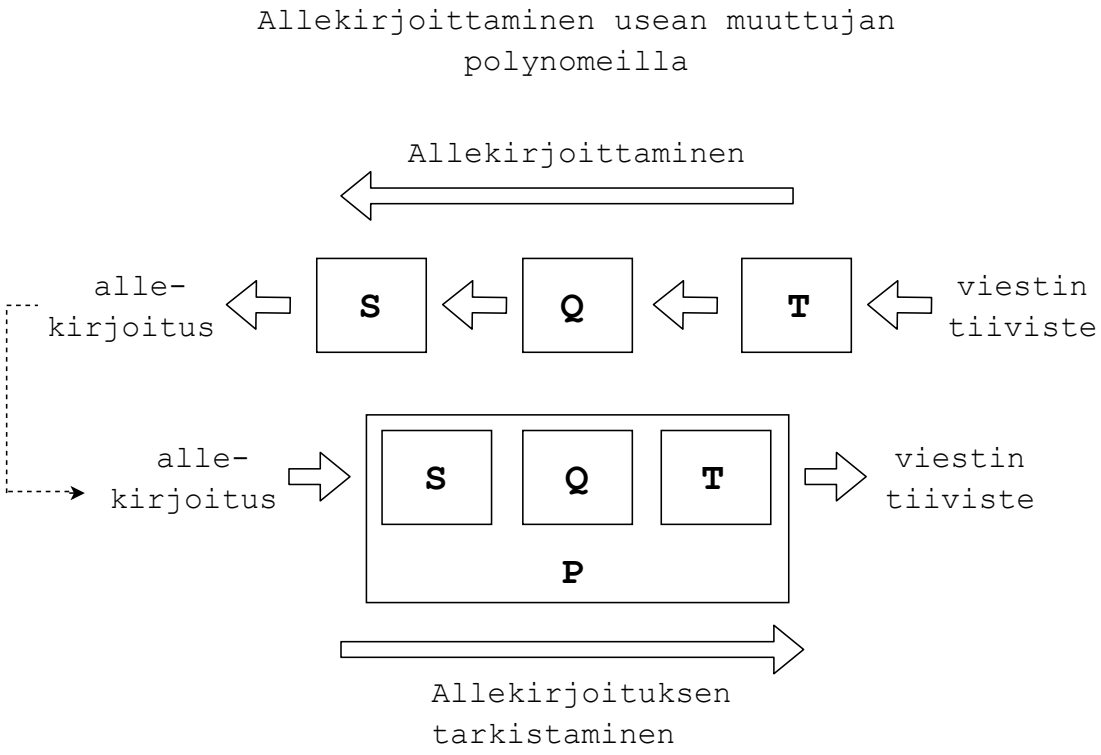
Hilapohjaista kryptografiaa pidetään kvanttiturvallisena siksi, että pitkään jatkuneista yrityksistä huolimatta ei ole vielä tiedossa sellaista kvanttitietokoneella suoritettavaa algoritmia, jolla hilaongelmien ratkaisua pystyttäisiin huomattavasti nopeuttamaan [11 s. 151]. Ei siis ole olemassa todistusta sille, että hilaongelmia ei voitaisi jonain päivänä ratkaista kvanttitietokoneella tehokkaasti, mutta toisaalta tällaista algoritmia ei ole myöskään tähän mennessä löydetty. Siksi hilaongelmia pidetään kvanttitietokoneita vastaan turvallisempina kuin elliptisiä käyriä, joiden tiedetään varmuudella olevan kvanttitietokoneille haavoittuvaisia.

Usean muuttujan polynomit

Usean muuttujan polynomeihin perustuvan allekirjoittamisen taustalla oleva matematiikka perustuu useita muuttujia sisältävien yhtälöiden ratkaisemiseen. Oikea ratkaisu sisältää muuttujat, joilla kaikki yhtälöt ratkeavat [4 s. 265]. Turvallisuus perustuu funktioihin, jotka on helppo laskea yhteen suuntaan, mutta paljon vaikeampi toiseen suuntaan, ellei käytössä ole tiettyä laskentaa helpottavaa tietoa. Käytettävien yhtälöiden on siis oltava tietyn muotoisia, jotta ne sopivat käytettäviksi tällaisessa kryptografiassa [11 s. 193].

Usean muuttujan polynomeihin perustuvan allekirjoituksen salainen avain koostuu matemaattisista muunnoksista, joille on helppo löytää käänteismuunnos eli tehdä sama laskutoimitus takaperin. Julkinen avain saadaan suorittamalla kaikki nämä muunnokset peräjälkeen. Tuloksena on uusi muunnos, josta on vaikea päätellä alkuperäisiä muunnoksia. Sen takia sitä ei myöskään ole niin helppoa tehdä takaperin kuin alkuperäisiä muunnoksia. Allekirjoittaminen toimii viemällä allekirjoitettavan viestin tiiviste yksitellen kaikkien salaisen avaimen muunnosten läpi takaperin. Viimeisen käänteismuunnoksen tulos on allekirjoitus. Allekirjoituksen varmennuksessa allekirjoitus viedään julkisen avaimen läpi etuperin, mikä johtaa samaan lopputulokseen kuin viestin tiiviste oli ennen allekirjoitusta [11 s. 195]. Tämä on havainnollistettu kuvassa 4.

Usean muuttujan polynomeihin perustuvan kryptografian turvallisuus riippuu käytettävien yhtälöiden määrästä ja näissä käytettävien lukujen suuruudesta. Vaikka tämän tyyppinen kryptografia saattaa vaikuttaa yksinkertaisemmalta kuin hilapohjaiset ratkaisut, huonona puolena on julkisen avaimen suuri koko: ollakseen riittävän turvallinen tällainen salausjärjestelmä vaatii muihin epäsymmetrisiin



Kuva 4. Allekirjoittaminen ja allekirjoituksen tarkistaminen usean muuttujan polynomeilla. S, Q ja T ovat matemaattisia muunnoksia ja kuuluvat yksityiseen avaimeen. P on näiden yhdistelmästä saatu julkinen avain.

salausjärjestelmiin verrattuna suuren julkisen avaimen. Tämä ei ole ongelma tavalliselle tietokoneelle, mutta saattaa rajoittaa kryptografian käytettävyyttä muistiltaan rajoittuneilla laitteilla. [11 s. 230]

2.2.4. Kvanttiturvallisen kryptografian standardointikilpailu

Yhä kasvavan kvanttitietokoneiden uhan takia NIST aloitti loppuvuodesta 2016 kilpailun, jonka tavoitteena on löytää uusia kvanttiturvallisia julkiseen avaimeen perustuvia algoritmeja. Kilpailuun toivottiin sekä avaintenvaihtoon että digitaaliseen allekirjoitukseen sopivia algoritmeja, joista parhaat on tarkoitus standardoida. Minimivaatimuksena kilpailuun mukaan pääsylle oli, että tarjotusta algoritmista oli saatavilla C-kielinen toteutus sekä kirjallinen spesifikaatio. Marraskuussa 2017 kilpailuun oli saapunut 82 ehdokasalgoritmia, joista 69 täytti minimivaatimukset. Tästä eteenpäin algoritmeja alettiin arvioida turvallisuuden, suorituskyvyn sekä algoritmin muiden ominaisuuksien perusteella [17]. Toiselle kierrokselle päässeet algoritmit valittiin tammikuussa 2019, ja heinäkuussa 2020 ilmoitettiin kolmannelle kierrokselle eli finaaliin asti päässeet algoritmit. Kilpailun lopputuloksen ja standardoitavien algoritmien on määrä olla selvillä 2022–2024 [18].

NIST:n toiveena oli, että algoritmeja olisi mahdollista käyttää useissa eri yhteyksissä ja useilla eri laitteilla. Mukana haluttiin pitää useanlaisia algoritmeja siltä varalta, että jonkinlainen uusi hyökkäystyyppi jotain tiettyä kvanttiturvallisena pidettyä

algoritmia vastaan keksittäisiin. Tällöin varavaihtoehtona olisi vielä toisenlaisiin ongelmiin perustuvia algoritmeja [17]. Kilpailussa algoritmeja arvioidaan erilaisilla etukäteen määritellyillä turvatasoilla. Turvataso 1 vastaa 128 bitin turvallisuustasoa. Turvataso 5 vastaa 256 bitin turvallisuustasoa, ja loput tasot ovat jotain tältä väliltä [19]. Algoritmien kehittäjiä pyydettiin keskittymään pääasiallisesti matalamman pään turvatasoihin 1–3. Tätä korkeammat turvatasot ovat mukana siltä varalta, että jokin erityinen kryptoanalyyttinen läpimurto yllättäen heikentäisi algoritmien turvallisuutta [17].

Nykyisten finalistien joukossa on neljä avaintenvaihtoon käytettävää KEM-algoritmia (Key Encapsulation Mechanism) ja kolme allekirjoitusalgoritmia. Lisäksi valittiin kahdeksan vaihtoehtoista algoritmia: viisi KEM-algoritmia ja kolme allekirjoitusalgoritmia. Nämä algoritmit haluttiin pitää mukana varavaihtoehtoina, vaikka niiden standardoimista NIST pitikin epätodennäköisenä [20]. Finaaliin asti päässeet allekirjoitusalgoritmit ovat nimeltään CRYSTALS-Dilithium, FALCON ja Rainbow [21].

Kuten elliptisten käyrien allekirjoituksessa, CRYSTALS-Dilithium, FALCON ja Rainbow perustuvat siihen, että käytössä on kaksi avainta: salainen avain, jolla allekirjoitus luodaan, ja julkinen avain, jolla allekirjoitus voidaan varmentaa. CRYSTALS-Dilithium ja FALCON perustuvat hilaongelmiin ja Rainbow usean muuttujan polynomeihin. Seuraavissa kappaleissa nämä finalistialgoritmit esitellään tarkemmin.

CRYSTALS-Dilithium

CRYSTALS-Dilithiumin toiminta perustuu lyhyiden vektorien löytämiseen hilasta ja turvallisuus tämän ongelman vaikeuteen. Avaintenluomisvaiheessa luodaan kaksi salaista vektoria ja polynomeja sisältävä matriisi. Lisäksi luodaan kolmas vektori, joka saadaan kertomalla matriisi ensimmäisen salaisen vektorin kanssa ja lisäämällä siihen toinen salainen vektori. Salaiset vektorit vastaavat yksityistä avainta, ja matriisi sekä matriisin ja salaisten vektorien yhdistelmästä saatu kolmas vektori muodostavat julkisen avaimen.

CRYSTALS-Dilithiumim allekirjoitusvaihe toimii *Fiat-Shamir with Aborts* -tekniikalla ja perustuu siihen, että allekirjoitettavan viestin tiivisteestä luotu vektori pyritään muuttamaan hilaan kuuluvaksi lyhyeksi vektoriksi. Tämä lyhyt vektori toimii allekirjoituksena. Tekniikan nimi viittaa siihen, että allekirjoituksen luomista saatetaan joutua yrittämään useita kertoja, mikäli vektorista tulee liian pitkä. Allekirjoituksen varmentaminen on mahdollista siksi, että allekirjoitusvaiheessa käytettävät lyhyet vektorit ovat osa julkiseen avaimeen kuuluvaa vektoria. Sen takia tekijöihin jaolla ja yksinkertaisilla plus- ja miinuslaskuilla allekirjoituksen varmentaja pääsee julkisen avaimen kautta lopulta suurin piirtein samaan vektoriin kuin jonka allekirjoittaja on luonut. Vektorista tarkastetaan vain suurimmat kertoimet, minkä takia lähestulkoon sama vektori riittää, kunhan se on tarpeeksi lyhyt. Vastaanottaja voi nyt todeta, että allekirjoittajalla on hallussaan salaista tietoa, jonka avulla julkinen avain on luotu. Salaista vektoria ei kuitenkaan voida helposti päätellä julkisesta avaimesta. [22]

FALCON

FALCON (Fast-Fourier Lattice-based Compact Signatures over NTRU) perustuu *Gentry-Peikert-Vaikuntanathan*-teoriaan [23] ja NTRU-hiloihin [24]. NTRU-hilalla tarkoitetaan tietyn rakenteista hilaa [11 s. 168]. FALCONin avaintenluomisvaiheessa luodaan neljä polynomia, jotka muodostavat salaisen avaimen. Julkinen avain on myöskin polynomi. Se saadaan salaisen avaimen polynomeista jakolaskulla niin, että salaista avainta on hyvin vaikea päätellä julkisesta avaimesta.

Allekirjoitusvaiheessa allekirjoitettavan viestin tiivisteestä luotu vektori pyritään salaiseen avaimeen kuuluvien polynomien avulla muokkaamaan kahdeksi hilaan kuuluvaksi lyhyeksi vektoriksi. Ensimmäinen vektori voidaan laskea toisesta, joten allekirjoitukseksi riittää toinen vektori. Vastaanottaja varmentaa allekirjoituksen luomalla samat vektorit uudelleen allekirjoituksen ja julkisen avaimen avulla. Tämä on mahdollista, koska julkinen avain koostuu samoista polynomeista kuin salainen avain. Lopuksi vielä tarkistetaan, että syntyneet vektorit ovat tarpeeksi lyhyitä, ennen kuin allekirjoitus hyväksytään.

FALCONin kehitystyössä on pyritty minimoimaan julkisen avaimen ja allekirjoituksen yhteenlaskettu koko, koska algoritmin kehittäjien mukaan kvanttiturvallisen kryptografian ongelmana ei yleensä ole laskennan nopeus vaan suuret avaimet ja allekirjoitukset [24]. CRYSTALS-Dilithiumin kehityksessä on ollut sama tavoite [22]. Koska CRYSTALS-Dilithium ja FALCON perustuvat molemmat hilaongelmiin, niistä tullaan standardoimaan NIST:n kilpailussa korkeintaan toinen [20].

Rainbow

Usean muuttujan polynomeihin perustuvana allekirjoitusalgoritmina Rainbow'n taustalla oleva matematiikka eroaa CRYSTALS-Dilithiumista ja FALCONista ja toimii kuten on kerrottu kappaleessa 2.2.3. Rainbow'n yksityinen avain sisältää takaperin käännettävissä olevia matemaattisia muunnoksia, ja julkinen avain on näistä saatu yhdistelmä. Allekirjoittaminen toimii syöttämällä viestistä luotu tiiviste yksitellen kaikkien yksityisen avaimen muunnosten läpi takaperin. Viestiä varmennettaessa sama tehdään toiseen suuntaan syöttämällä allekirjoitus julkisen avaimen läpi, jolloin päästään takaisin viestin tiivisteeseen. Vastaanottaja voi varmentaa allekirjoituksen yksinkertaisesti tarkistamalla, onko lopputulos sama kuin viestin tiiviste. [25]

Rainbow eteni NIST:n kilpailun finaaliin asti ainoana usean muuttujan polynomeihin perustuvana algoritmina. Kilpailun kolmannen kierroksen aikana tehdyt tutkimukset [26] paljastivat yllättäviä puutteita Rainbow'n turvallisuudessa, minkä takia NIST avasi kesällä 2021 uudelleen mahdollisuuden lähettää kilpailuun uusia allekirjoitusalgoritmeja [27]. Tämän diplomityön ohjelmointiosuutta toteutettaessa tämä ei vielä ollut tiedossa.

Finalistialgoritmeja koskevaa tutkimusta

Kvanttiturvallisten allekirjoitusalgoritmien integrointia erilaisiin tosielämän käyttötarkoituksiin on sivuttu aiemmissa tutkimuksissa. Eräässä tutkimuksessa testattiin niiden sopivuutta TLS-protokollan (Transport Layer Security) yhteyteen, jota käytetään yleisesti internetissä tapahtuvan kommunikaation turvaamiseen.

Tutkimuksessa NIST:n kilpailuun lähetettyjä allekirjoitusalgoritmeja käytettiin osana TLS 1.3 -protokollaa. Mukana vertailussa olivat myös CRYSTALS-Dilithium, FALCON ja Rainbow, ja CRYSTALS-Dilithiumin ja FALCONin todettiin olevan parhaat vaihtoehdot tähän tarkoitukseen. [28]

NIST:n kilpailuun toimitettujen allekirjoitusalgoritmien toimivuutta on testattu myös teollisuudessa ja sulautetuilla laitteilla. Sulautettuja laitteita koskevassa tutkimuksessa allekirjoitusalgoritmien todettiin sopivan turvaamaan laitteen toimintaa siten, että ennen koodin suorittamista sen aitous tarkistettiin allekirjoituksesta. Tämä tutkimus tehtiin NIST:n kilpailun aiemmassa vaiheessa, jolloin nykyisiä finalisteja ei ollut vielä valittu [29]. Teollisuuslaitteita koskevassa tutkimuksessa NIST:n kilpailussa mukana olevia hilapohjaisia allekirjoitus- ja avaintenvaihtoalgoritmeja käytettiin avaintenvaihdossa yhdessä X.509-sertifikaatin kanssa [30]. NIST:n kilpailuun toimitetut kvanttiturvalliset allekirjoitusalgoritmit ovat siis todistetusti käytettävissä varsin erilaisissa konteksteissa. Niitä ei kuitenkaan liene sovellettu tämän tutkielman tapaan älykkään liikenteen viestintään, vaikka esimerkiksi [31] ehdottaakin CRYSTALS-Dilithiumin käyttämistä auton pysäköimistä helpottavissa sovelluksissa.

3. ÄLYKÄS LIIKENNE

Älykäs liikenne tarkoittaa sitä, että liikenteen turvallisuutta ja sujuvuutta pyritään parantamaan tielläliikkujien välisen kommunikaation ja automaattisen tietojen keräilyn avulla [32]. Termi C-ITS (Cooperative Intelligent Transport Systems) korostaa kommunikaation merkitystä. Kyse voi olla esimerkiksi siitä, että vaaratilanteen havainnut ajoneuvo tiedottaa tapahtumasta muille lähistöllä liikkuville autoille [33, 34], tai siitä, että lähekkäin ajavat ajoneuvot viestivät toisilleen omasta nopeudestaan ja kulkusuunnastaan, jolloin yksittäinen ajoneuvo voi muodostaa paremman kokonaiskuvan tilanteesta [35]. Myös reittisuunnittelussa kommunikaatiosta on hyötyä. Kartat eivät aina ole ajan tasalla, ja paikkansapitävistäkin kartoista on mahdotonta saada tietoa esimerkiksi yllättäen tielle syntyvistä ruuhkatilanteista. Tällöin muilta ajoneuvoilta saadut tiedot voivat auttaa kuljettajaa valitsemaan toisen reitin ja näin vähentää ruuhkaisen tieosuuden liikennemääriä [36]. Ajoneuvot voivat myös pitää yllä dynaamista karttaa (LDM, Local Dynamic Map), johon kerätään liikenteen turvallisuuden ja sujuvuuden kannalta tärkeää tietoa. Kartta voi kertoa lähellä olevista liikkuvista objekteista, kuten autoista, tai liikkumattomista objekteista, kuten liikennemerkeistä [37]. Liikenteessä tapahtuvan kommunikoinnin erityispiirre on lokaalius: esimerkiksi ajoneuvojen lähettämät vaaratiedotteet ovat merkityksellisiä ainoastaan samalla alueella liikkuville ajoneuvoille [38].

Jotta älykäs ajoneuvo voisi viestiä muille tielläliikkujille kohtaamistaan tilanteista, se tarvitsee tietoa ulkomaailmasta. Älykkäät ajoneuvot on varustettu suurella määrällä erilaisia sensoreita, jotka tarkkailevat esimerkiksi auton sijaintia ja nopeutta. Ympäristön tapahtumia voidaan analysoida kameroiden avulla. Sensorit ja kamerat keräävät suuria määriä dataa, jonka analysointi pohjautuu usein tekoälyyn. Sen takia ympäristöstä kerätyn tiedon käsittely vaatii paljon laskentakapasiteettia. Tämän takia autonomisissa ja osittain autonomisissa ajoneuvoissa on yleensä useita erilaisia laskentaelementtejä, jotta kaikki sensorien keräämä data pystyttäisiin käsittelemään kyllin nopeasti. [35]

Viestien lähetys tapahtuu autojen välillä langattomasti. Autoissa on erityisesti kommunikointiin tarkoitetut laitteet, joita kutsutaan nimellä On-Board Unit (OBU). Ne voivat viestiä keskenään esimerkiksi IEEE 802.11p-protokollaa käyttäen ollessaan toisistaan signaalin kantomatkan päässä [39]. Älykkäässä liikenteessä viestivänä osapuolena voi olla myös tien läheisyyteen asetettuja kiinteitä rakennelmia tai muita osapuolia, jotka eivät ole autoja. Tämän takia liikenteen viestintäverkossa mukana olevista asemista ei käytetä nimitystä auto vaan yleisesti ITS-asema (ITS-S, Intelligent Transport System Station) [40]. Alan kirjallisuudessa kommunikaatio saatetaan eritellä sen mukaan, tapahtuuko se autolta autolle (V2V, vehicle-to-vehicle) vai autolta mille tahansa lähellä olevalle vastaanottajalle (V2X, vehicle-to-everything) [35, 41].

3.1. Älykkään liikenteen standardit

Standardointi on prosessi, jossa luodaan tiettyyn toimintaan liittyviä määritelmiä ja ohjeita, joita alan toimijat yhdessä noudattavat. Standardien tavoitteena on mahdollistaa kyseiseen toimintaan liittyvien tuotteiden tekninen yhteensopivuus,

siirrettävyys ja globaali kauppa. Standardointityössä tulee ottaa huomioon monien erilaisten ryhmien toiveita, ja sen takia standardointi on usein pitkä prosessi [41]. Euroopassa on kolme virallisesti tunnustettua standardoimisjärjestöä: CEN (European Committee for Standardization), CENELEC (European Committee for Electrotechnical Standardization) ja ETSI (European Telecommunications Standards Institute) [42]. Vain näiden muotoilemat standardit voidaan hyväksyä eurooppalaisiksi standardeiksi (EN, European Standard) [43]. ETSI julkaisee lisäksi teknisiä vaatimuksia sisältäviä teknisiä spesifikaatioita (TS, Technical Specification) sekä niiden kohteena olevia aiheita laajemmin selittäviä teknisiä raportteja (TR, Technical Report) [44].

Euroopan komissio valmistelee älykästä liikennettä koskevaa lainsäädäntöä ja on pyytänyt myös eurooppalaisia standardoimisjärjestöjä valmistelemaan älykästä liikennettä koskevia standardeja [45]. Valmistelutyötä tehdään tällä hetkellä CEN:n teknisessä komiteassa numero 278 [46]. Useat erilaiset tekniset spesifikaatiot ja standardit liittyvät älykkääseen liikenteeseen, joka vaatii monimutkaisen kokoelman erilaisia säännöksiä ja protokollia toimiakseen. Siksi kehitystyössä on mukana useita toimijoita. Älykkään liikenteen kommunikaation turvallisuus ja luotettavuus on tarkoitus taata ETSI:n kehittämillä teknisillä spesifikaatioilla [32], jotka määrittelevät mm. lähetetyissä viesteissä käytettävät salaus- ja allekirjoitustyypit [47].

3.1.1. Älykkään liikenteen sertifikaattijärjestelmä

ITS-asemien välisessä viestinnässä halutaan taata sekä viestinnän luotettavuus että anonymiteetti, joka tarkoittaa, että yksittäisen ITS-aseman liikkeitä ei voida seurata sen lähettämien viestien perusteella. Näiden takaamiseksi ITS-asemien välillä lähetettävät viestit muotoillaan tietyllä teknisissä spesifikaatioissa määritellyllä tavalla. Keskeisiä spesifikaatioita ovat ETSI TS 102 941 [48] ja ETSI TS 103 097 [47].

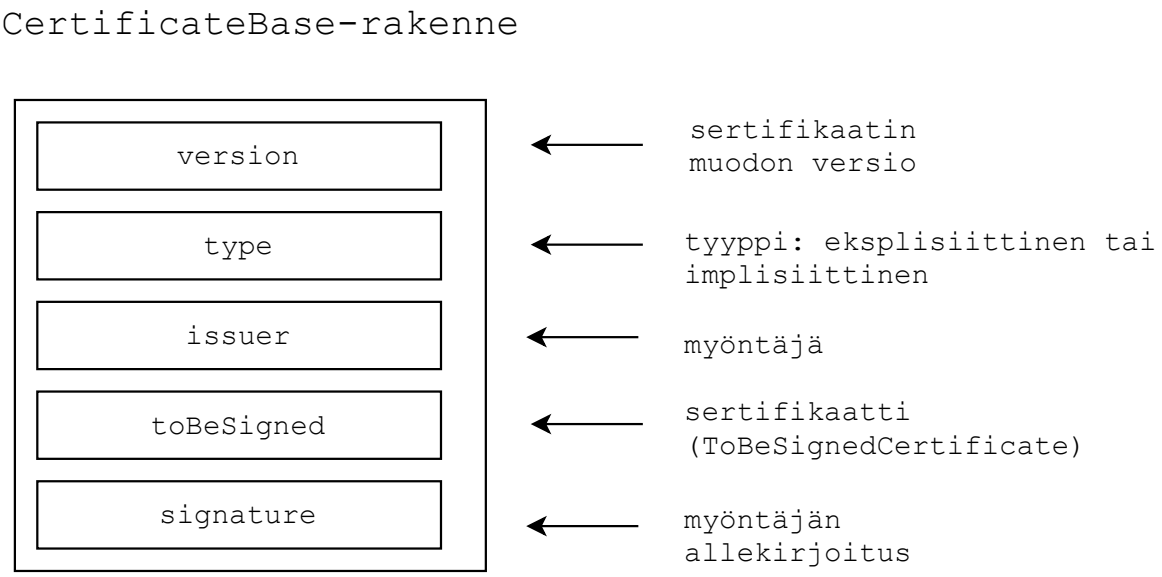
Julkisen avaimen infrastruktuuri on määritelty spesifikaatiossa ETSI TS 102 940. Se perustuu toisen luvun alussa mainitun sertifikaattiketjun käyttämiseen: ajoneuvolle myönnettävässä sertifikaatissa on erillinen kenttä sertifikaatin myöntäneen tahon tiedoille ja allekirjoitukselle. Dokumentissa määritellään julkisen avaimen infrastruktuuri, joka perustuu kaikkien ITS-asemien tiedossa olevaan luotettavaan tahoon, joka allekirjoittamalla todentaa muiden ITS-asemien sertifikaattien oikeellisuuden. Tämän kaikkein korkeimman luotettavan tahon ja tavallisten ITS-asemien välillä on vielä väliasteikon auktoriteetti. Se saa toimintavaltuutensa ylimmältä auktoriteetilta ja saa allekirjoittaa tavallisten ITS-asemien välisessä viestinnässä käytettäviä sertifikaatteja. [49]

ETSI TS 102 941 määrittelee sertifikaattien käytön periaatteet. Valmistusvaiheessa jokaiselle ITS-asemalle luodaan oma julkisen ja yksityisen avaimen sisältävä avainpari viestien allekirjoittamista ja salaamista varten. Lisäksi sille annetaan yhteystiedot sertifikaatin pyytämistä varten. Kun tämä liikenneväline otetaan käyttöön, se pyytää itselleen sertifikaatin, joka liittää valmistusvaiheessa annetun julkisen avaimen kyseiseen ITS-asemaan. Sertifikaatista käy ilmi myös, onko kyseisellä ITS-asemalla joitakin erityisoikeuksia vai onko se tavallinen ajoneuvo.

Anonymiteetin säilyttämisen vuoksi ITS-asema ei kuitenkaan käytä tätä identifioivaa sertifikaattia muille ITS-asemille lähetettävissä viesteissä vaan

ainoastaan infrastruktuurin ylläpitäjien kanssa viestiessään. Viestintäinfrastruktuuri on suunniteltu siten, että kun ajoneuvo on rekisteröitynyt, se ei jatkossa viesti muiden ITS-asemien kanssa omalla identiteetillään vaan pseudonyymiin liitetyillä väliaikaisilla sertifikaateilla, joita järjestelmässä kutsutaan nimellä Authorization Ticket (AT). Ajoneuvo voi pyytää niitä infrastruktuurin ylläpitäjiltä allekirjoittamalla viestin omalla identiteetillään. Jos ajoneuvoa ei ole asetettu epäluotettavan käytöksen takia kieltolistalle, sille myönnetään kerrallaan tietty määrä näitä väliaikaisia sertifikaatteja, jotka infrastruktuurin luotettavaksi tiedetty taho on allekirjoittanut. Ajoneuvo käyttää niitä muiden ITS-asemien kanssa viestiessään. Väliaikaisesta sertifikaatista ei käy ilmi ajoneuvon todellinen identiteetti vaan ainoastaan se, että kyseinen sertifikaatti on luotettavan tahon myöntämä. Näin säilytetään ajoneuvojen anonymiteetti kuitenkin varmistaen samalla, että käytössä olevat sertifikaatit ovat luotettavia. [48]

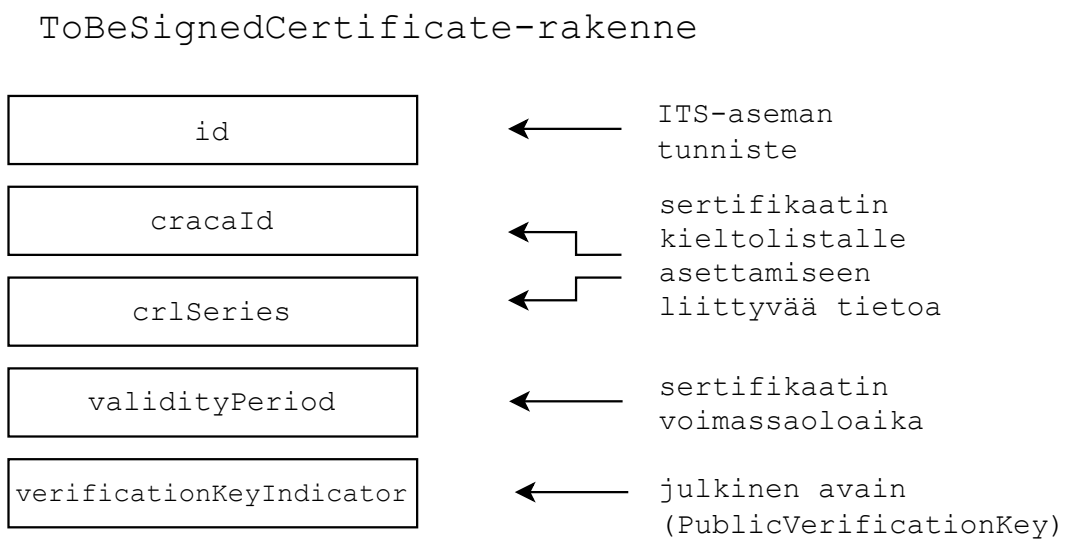
ETSI TS 103 097 [47] määrittelee ITS-asemien välisessä viestinnässä käytettävän sertifikaatin rakenteen. Sertifikaatti sisällytetään laajempaan rakenteeseen, joka sisältää itse ITS-aseman sertifikaatin, sertifikaatin version ja tyypin sekä sertifikaatin myöntäneen auktoriteetin tunnistetiedot ja allekirjoituksen. Tyypiksi voidaan valita joko eksplisiittinen tai implisiittinen, joista eksplisiittinen tarkoittaa, että sertifikaatti on viestissä mukana. Implisiittinen tyyppi tarkoittaa, että sertifikaatti on lähetetty aiemmin, mutta sitä ei ole mukana juuri kyseisessä viestissä. Tämä rakenne (teknisessä spesifikaatiossa CertificateBase) on esitetty kuvassa 5. Varsinainen sertifikaatti paketoidaan osaksi laajempaa rakennetta siksi, että se on osa laajempaa julkisen avaimen infrastruktuuria, minkä takia sen mukana halutaan välittää luotettavaksi tiedetyn tahon allekirjoitus ja myöntäjän tiedot.



Kuva 5. Spesifikaation mukainen CertificateBase-rakenne, joka liittää ITS-aseman sertifikaatin osaksi laajempaa julkisen avaimen infrastruktuuria. ToBeSignedCertificate viittaa toiseen teknisissä spesifikaatioissa määriteltyyn rakenteeseen.

Varsinaiseen sertifikaattiin kuuluvat seuraavat tiedot: sertifikaatin tunnistenumero; tiedot siitä, miten sertifikaatti voidaan asettaa kieltolistalle; tieto sertifikaatin

voimassaoloajasta; sertifikaattiin liitetyn julkisen avaimen tyyppi ja itse avain. Tämän spesifikaation mukaisesti luotuja sertifikaatteja ei voida asettaa erilliselle kieltolistalle, vaan kaikki jo myönnetyt sertifikaatit ovat käyttökelpoisia [47]. Sertifikaatin rakenne (teknisessä spesifikaatiossa ToBeSignedCertificate) on esitelty kuvassa 6.



Kuva 6. Spesifikaation mukainen ToBeSignedCertificate-rakenne, joka vastaa ITS-aseman sertifikaattia. PublicVerificationKey viittaa toiseen teknisissä spesifikaatioissa määriteltyyn rakenteeseen.

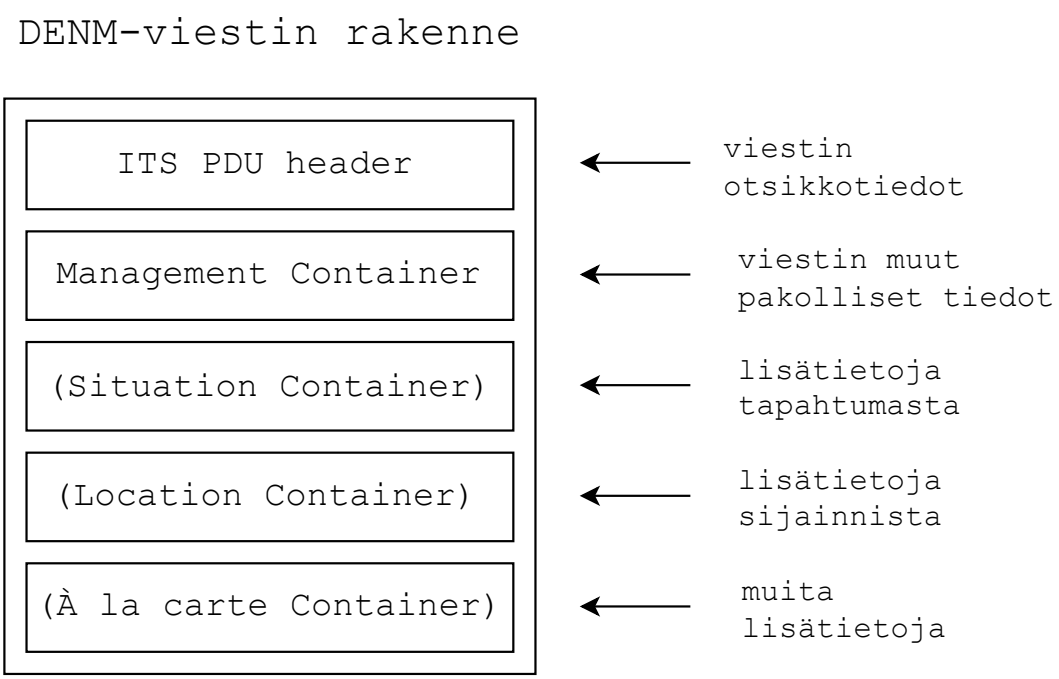
3.1.2. CAM ja DENM

CAM (Cooperative Awareness Message) ja DENM (Decentralized Environmental Notification Message) ovat standardeissa ETSI EN 302 637-2 [33] ja ETSI EN 302 637-3 [34] määriteltyjä viestityyppejä. Niiden tarkoitus on auttaa tiellä liikkuvia ajoneuvoja pysymään tietoisena sellaisista ympäristön tapahtumista, joilla on merkitystä liikenteen sujuvuuden kannalta. Ne ovat tärkeitä tiedon lähteitä dynaamisten karttojen [37] ylläpitämisessä. CAM-viestin tarkoituksena on välittää tielläliikkujien välillä viestejä, jotka mahdollistavat turvallisen yhteistoiminnan. Viesti voi sisältää tietoa esimerkiksi liikennevälineen sijainnista, koosta ja suunnasta. Tämä mahdollistaa esimerkiksi autojen välisen törmäysriskin arvioinnin ja kuljettajan varoittamisen tarvittaessa [33].

DENM-viestit liittyvät epänormaaleihin tai vaarallisiin liikennetilanteisiin, ja ne välitetään kaikille tietyllä maantieteellisellä alueella sijaitseville ITS-asemille. Viestin vastaanottanut ajoneuvo punnitsee viestin merkittävyyden ja voi tarpeen mukaan esimerkiksi varoittaa kuljettajaa tapahtumasta. Viestin lopullinen käyttötarkoitus riippuu ajoneuvosta ja tilanteesta.

ETSI EN 302 637-3 määrittelee viestin pakolliset kentät ja niiden sisällön. DENM-viestin rakenne on esitetty kuvassa 7. Viestin pakollisia osia ovat otsikkokenttä (ITS PDU Header) sekä viestin metatietoja sisältävä lisätietokenttä (Management Container). Otsikkokentästä käyvät ilmi käytettävän protokollan versio, lähettävän

ITS-aseman tunniste sekä viestin tunnistenumero. Standardi määrittelee, kuinka otsikkokenttä tulee täyttää. Esimerkiksi protokollan versio riippuu käytettävän standardin versiosta. Lisätietokenttään sisältyvät ainakin seuraavat tiedot: viestin järjestysnumero; aika, jolloin DENM-viestin aiheena oleva tapahtuma havaittiin; aika, jolloin viesti luotiin; paikka, jossa tapahtuma havaittiin; aikamääre sille, kuinka pitkään viesti on relevantti, ja lähettävän ITS-aseman tyyppi [34]. ITS-aseman tyypiksi voidaan valita jalankulkija, pyöräilijä, mopo, moottoripyörä, henkilöauto, bussi, kevyyt tai painava kuorma-auto, perävaunu, raitiovaunu tai erikoisajoneuvo [50].



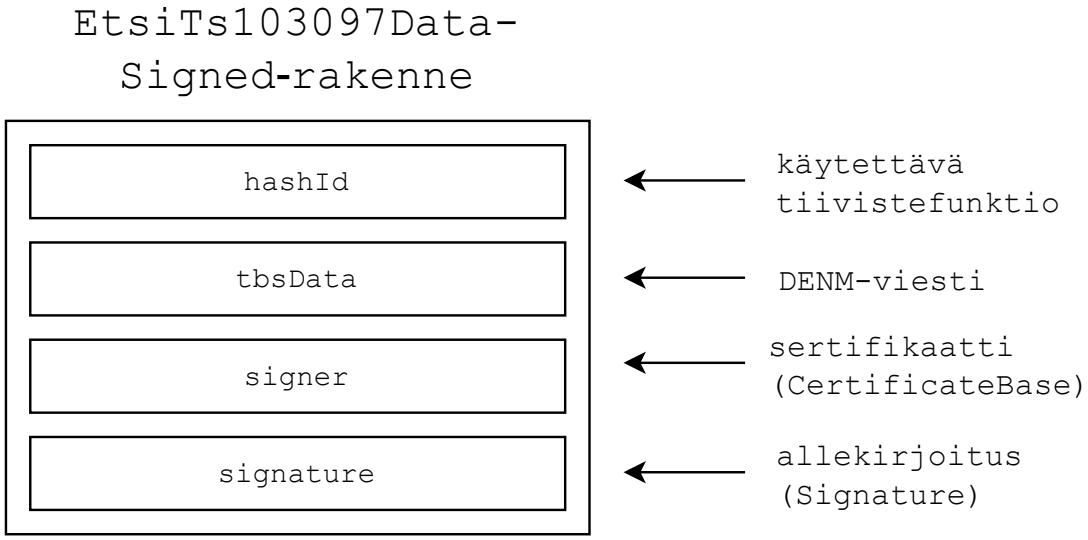
Kuva 7. Standardin mukaisen DENM-viestipaketin rakenne. Suluissa olevat kentät eivät ole pakollisia.

DENM-viestin valinnaisia kenttiä ovat havaittua tapahtumaa kuvaileva kenttä (Situation Container), tarkempia paikkatietoja sisältävä kenttä (Location Container) ja ylimääräisiä lisätietoja sisältävä kenttä (À la Carte Container). Havaittua tapahtumaa kuvaava kenttä sisältää tiedot havaitun tapahtuman luonteesta ja välitettävän tiedon laadusta. Tieto havaitun tapahtuman luonteesta ilmaistaan numerokoodilla. Eri numerokoodien selitykset on lueteltu DENM-viestejä koskevassa standardissa, ja kukin numerokoodi tarkoittaa tiettyä tapahtumaa. Omat koodinsa on määritelty esimerkiksi onnettomuudelle, tietyölle, huonoille sääolosuhteille, tiellä liikkuvalle ihmiselle, väärään suuntaan ajavalle ajoneuvolle, hitaasti ajavalle ajoneuvolle, tiellä liikkuville eläimille, rikkoutuneelle ajoneuvolle ja lähestyvälle hälytysajoneuvolle. Välitettävän tiedon laatu tarkoittaa sitä, kuinka varma havainto tapahtumasta on saatu, toisin sanottuna miten suurella todennäköisyydellä havainto todella pitää paikkansa. Tiedot saadaan muista osista liikennevälineen järjestelmiä, esimerkiksi sensoreilta. [34]

3.1.3. Viestien muotoilu

Tekninen spesifikaatio ETSI TS 103 097 [47] määrittelee ITS-asemien välisessä viestinnässä käytettävien viestien muodon. Muoto riippuu lähetettävän viestin laadusta. Viestien muotoilun pohjana ovat IEEE-standardit IEEE Std 1609.2-2016 [51], IEEE Std 1609.2a-2017 [52] ja IEEE Std 1609.2b-2019 [53], joiden tarjoamia rakenteita viestinnässä käytetään. ETSI TS 103 097 määrittelee, mitä rakennetta kulloinkin käytetään ja kuinka ITS-asemien väliseen kommunikaatioon liittyvät viestit tulee niihin sisällyttää [47].

Tämän spesifikaation mukaan CAM- ja DENM-viestit tulee lähettää käyttäen allekirjoitettua viestirakennetta (EtsiTs103097Data-Signed). Kuva 8 esittää allekirjoitetun viestirakenteen, jonka sisällä on DENM-viesti. CAM- tai DENM-viesti sisällytetään allekirjoitettavalle viestille varattuun kenttään. Allekirjoituskentässä on viestin tiivisteen perusteella laskettu allekirjoitus. Allekirjoitettuun viestirakenteeseen sisällytetään tieto siitä, mitä tiivistefunktiota allekirjoituksen yhteydessä on käytetty. Lisäksi vaaditaan lähettäjän sertifikaatti, josta käy ilmi allekirjoituksen varmennukseen käytettävä julkinen avain [47]. Nykyisen teknisen spesifikaation mukaisesti allekirjoitukseen käytetään elliptisiä käyriä, joista on valittavana kolme eri tyyppiä: NIST P-256, brainpoolP256r1 ja brainpoolP384r1 [52]. Sertifikaatin ja allekirjoituksen avulla viestin alkuperä voidaan varmentaa.



Kuva 8. Spesifikaation mukaisen allekirjoitetun viestipaketin rakenne, kun sitä käyttäen lähetetään DENM-viesti. CertificateBase ja Signature viittaavat toisiin teknisissä spesifikaatioissa määriteltyihin rakenteisiin.

3.2. Älykkään liikenteen standardit käytännössä

Eräs älykkäässä liikenteessä tapahtuvan kommunikaation piirteistä on, että sen tulee toimia eri tyyppisten laitteiden välillä. Tämä on välttämätöntä siksi, että käytännössä saatavilla olevat autojenväliseen kommunikointiin käytettävät laitteet ovat usein

yksityisten valmistajien tuotteita, ja niiden arkkitehtuurit voivat poiketa toisistaan [38]. Kommunikoinnin sujuvuuden varmistamiseksi käytettävien protokollien on siis sovelluttava useiden eri valmistajien laitteille. Saatavilla on useita kaupallisten valmistajien OBU-laiteratkaisuja [54, 55, 56, 57].

Suosittuja OBU:issa käytettäviä prosessoreita ovat erilaiset ARM-arkkitehtuuriin perustuvat prosessorit [54, 58, 59, 60]. Ne ovat RISC-pohjaisia (Reduced Instruction Set Computer) laitteita [61], mikä tarkoittaa, että prosessori osaa suorittaa rajoitetun määrän kiinteän mittaisia käskyjä. Nämä prosessorit soveltuvat käyttötarkoituksiin, joissa vaaditaan energiatehokkuutta ja reaaliaikaista toimintaa [62]. Useissa OBU:issa on mukana myös erillinen turvallisuusmoduuli (HSM, Hardware Security Module), joka suorittaa tietoturvan piiriin kuuluvia operaatioita, kuten viestien allekirjoittamisen ja varmennuksen [54, 58, 59].

Kvanttiturvalliset allekirjoitusalgoritmit vaativat usein paljon muistia ja energiaa, ja siksi niiden integroiminen laskentateholtaan ja muistiltaan rajoitetuille pienille laitteille saattaa olla haasteellista. Esimerkiksi pieni RAM-muisti (Random Access Memory) saattaa johtaa siihen, että joitakin algoritmeja ei voida käyttää laitteella ollenkaan [16]. Tässä tutkielmassa kuitenkin oletetaan, että autoissa käytettävät OBU-laitteet eivät ole muistiltaan kaikkein rajoittuneimmasta päästä, sillä niiden on toimittava luotettavasti datan nopeassa käsittelyssä.

Useita tämänhetkisen ETSI:n C-ITS-protokollapinon toteuttavia avoimen lähdekoodin toteutuksia on saatavilla käyttöön. Esimerkki tästä on C++-kielellä toteutettu Vanetza [63], joka sopii käytettäväksi esimerkiksi MK5 OBU -prosessorin kanssa [55, 64].

3.3. Älykkään liikenteen tietoturvauhat

Liikennevälineiden välisestä viestinnästä on hyötyä vain, jos ne voivat luottaa toistensa lähettämiin viesteihin ja viestien toimitus tapahtuu kyllin nopeasti. Älykkäässä liikenteessä tapahtuvaan viestintään liittyvät pitkälti samat tietoturvauhat kuin mihin tahansa kommunikaatioon, sillä erotuksella, että liikenteessä viestinnän nopeus on erityisen tärkeää. Liian suuri viive viestien lähettämisessä tai prosessoinnissa voi aiheuttaa sen, että tilanne menee jo ohi, ennen kuin ITS-asema on edes tietoinen siitä. Viestien toimitusongelmia voidaan aiheuttaa esimerkiksi DoS-hyökkäyksellä (Denial of Service), jossa generoidaan suuri määrä valheellisia viestejä, joiden prosessointi estää ITS-asemaa käsittelemästä oikeiden lähettäjien viestejä. [65]

Toisten ajoneuvojen lähettämiä viestejä tallentamalla ja uudelleenlähettämällä voidaan aiheuttaa hämmennystä esimerkiksi siitä, missä ajoneuvot milloinkin sijaitsevat [66]. Ilkivaltaa voidaan tehdä myös väärennetyillä viesteillä: jos esimerkiksi henkilöauto esiintyy toisen ITS-aseman nimissä ja väittää olevansa hälytysajossa oleva ambulanssi, voi tästä aiheutua liikenteeseen hämmennystä ja kaaosta [65]. Tällaista huijaamista pyritään estämään kryptografisesti allekirjoitetuilla sertifikaateilla, joista ilmenevät tiedot ajoneuvon tyypistä ja sille mahdollisesti myönnetyistä erityisoikeuksista. Sertifikaatin ja lähetettävän viestin tietoja vertailemalla voidaan varmistaa, että ITS-asema lähettää vain sen asemaan nähden soveliaita viestejä [67]. Liikennetilanteissa tapahtuva viestintä esimerkiksi ajoneuvojen nopeuksista voi olla

myös juridisesti merkittävää informaatiota. Tällöin allekirjoituksia voidaan käyttää todistamaan viestin alkuperä [65].

Jos jotain ITS-asemaa ei voida enää pitää luotettavana esimerkiksi tahallisesti väärennettyjen viestien vuoksi, voivat viestintäinfrastruktuurin ylläpitäjät lisätä tästä tiedon erityiselle epäluotettujen lähettäjien listalle. Silloin ITS-asemalle ei enää myönnetä enempää viestinnässä käytettäviä sertifikaatteja. On kuitenkin huomattava, että järjestelmässä ei ole mahdollisuutta välittömästi poistaa ajoneuvon sertifikaatteja toiminnasta, vaan sille voidaan ainoastaan lakata myöntämästä lisää väliaikaisia sertifikaatteja. [67]

ETSI:n teknisissä spesifikaatioissa [47] viestien väärentämistä ja valheellisten viestien lähettämistä älykkäässä liikenteessä pyritään estämään elliptisten käyrien avulla tehtävillä digitaalisilla allekirjoituksilla. Nämä ovat varsin tehokas keino estää viestien väärentäminen tavallisella tietokoneella. Kvanttitietokoneiden kehitys uhkaa kuitenkin älykkään liikenteen kommunikaatiota siinä missä mitä tahansa muutakin viestintää. ETSI:n alaisuudessa toimii kvanttiturvalliseen kryptografiaan keskittynyt tekninen komitea CYBER QSC (Quantum Safe Cryptography), jonka tarkoituksena on tutkia kvanttiturvallisen kryptografian soveltamista. Vuonna 2021 CYBER QSC aikoo julkaista teknisen raportin kvanttiturvallisista allekirjoitusalgoritmeista sekä toisen teknisen raportin, joka koskee älykkään liikenteen viestinnän muuttamista kvanttiturvalliseksi. Ryhmä myös seuraa NIST:n kvanttiturvallisten algoritmien standardointikilpailua ja aikoo julkaista teknisen raportin kilpailun finaaliin asti päässeistä KEM-algoritmeista [68]. ETSI tekee siis jo nyt työtä älykkään liikenteen spesifikaatioiden päivittämiseksi kvanttiturvalliseen muotoon.

Tämän tutkielman ohjelmointiosuuden toteutushetkellä näitä CYBER QSC:n teknisiä raportteja kvanttiturvalliseen allekirjoitukseen ja ITS-viestintään liittyen ei vielä ole saatavilla. Vertailtavat allekirjoitusalgoritmit ja testit on toteutettu näistä raporteista riippumatta. Seuraavassa kappaleessa käydään tarkemmin läpi tutkielmaa varten tehty ohjelmallinen toteutus.

4. OHJELMALLINEN TOTEUTUS JA TESTIASETELMA

Älykkäässä liikenteessä tapahtuvan turvallisen viestinnän tehokkuutta voidaan mitata ainakin kolmella eri aspektilla: kuinka monta lisätavua lähetettävään viestiin pitää lisätä turvallisuustoimien vuoksi, kuinka pitkän ajan lähettäjä käyttää tällaisen viestin valmisteluun ja kuinka pitkän ajan vastaanottaja käyttää viestin purkamiseen [66]. Tässä tutkielmassa viestin valmistelulla ja purkamisella viitataan digitaalisen allekirjoituksen luomiseen ja varmentamiseen. Lisätavuilla viitataan siihen, kuinka monta tavua pakettiin joudutaan lisäämään viestin allekirjoitusta ja allekirjoittajan julkisen avaimen sisältävää sertifikaattia varten verrattuna siihen, että viestiä ei allekirjoitettaisi. Käytettävien allekirjoitusalgoritmien turvallisuutta ei tässä tutkielmassa analysoida matemaattisesti, vaan niiden oletetaan täyttävän 128 bitin turvallisuustason ja keskitytään niiden käytännöllisiin ominaisuuksiin.

Tämän työn ohjelmoinnillisen toteutuksen tavoitteena oli integroida kolme NIST:n standardointikilpailun finaaliin valittua kvanttiturvallista allekirjoitusalgoritmia älykkään liikenteen kommunikaatiossa käytettäviin rakenteisiin, tutkia kyseisten algoritmien käytettävyyttä tässä kontekstissa sekä vertailla niiden suorituskykyä nykyisin käytössä oleviin elliptisiin käyriin. Esimerkkitapauksena käytettiin allekirjoitetun DENM-viestin luomista. Ohjelmointiosuudessa toteutettiin ETSI-spesifikaation mukainen DENM-viestin lähetys allekirjoitetussa muodossa ja siihen liittyvä sertifikaatti niin, että allekirjoitukseen käytettiin joko brainpoolP256r1- tai NIST P-256 -käyrää tai yhtä kolmesta NIST:n kilpailun finaaliin päässeestä allekirjoitusalgoritmista. Koska CAM- ja DENM-viestit lähetetään spesifikaation mukaisesti samanlaiseen tietorakenteeseen sisällytettyinä [47], voidaan tulokset yleistää koskemaan myös CAM-viestejä. Suorituskykyä mitattiin allekirjoituksen ja allekirjoituksen varmennuksen nopeudella. Lisäksi tutkittiin, kuinka suuri allekirjoitetusta viestistä tulee, kun siihen käytetään eri allekirjoitusalgoritmeja.

Koska OBU:issa käytettävien prosessorien ominaisuudet, kuten kellotaajuus ja käytettävissä olevan muistin määrä, vaihtelevat keskenään, allekirjoitusalgoritmeille ei voida ennustaa yleispäteviä suoritusaikoja. Tähän diplomityöhön sisältyvät suoritusaikamittaukset on toteutettu kannettavalla tietokoneella, minkä takia tulokset eivät vastaa sellaisenaan mitään tiettyä autoissa käytettävää prosessoria. Ne antavat kuitenkin tietoa allekirjoitusalgoritmien suhteellisista ominaisuuksista toisiinsa verrattuna.

4.1. ETSI:n teknisten spesifikaatioiden toteutus

ETSI tarjoaa tässä työssä tarvittavat tietorakenteet teknisten spesifikaatioidensa [47] [34] liitteissä ASN.1-muodossa. ASN.1 on tapa esittää datan rakenne standardoidulla syntaksilla. Siinä määritellään myös datan koodaus, jonka avulla data saadaan muokattua yksiselitteisesti bittijonoksi. Erilaisten työkalujen avulla ASN.1-notaatiot voidaan kääntää useille eri ohjelmointikielille [69]. Tässä tutkielmassa tekninen spesifikaatio ja siihen liittyvä ohjelma toteutettiin C-ohjelmointikielellä. Tämä valinta tehtiin siksi, että kvanttiturvallisten allekirjoitusalgoritmien toteutukset olivat saatavilla C-kielisinä. Siksi ne oli helpointa yhdistää C-kieliseen ohjelmaan. Ensin käännettiin allekirjoitettua datarakennetta kuvaavat ASN.1-koodit C-kielisiksi asn1c-

kääntäjällä [70] käyttäen komentoa

```
asn1c Ieee1609Dot2BaseTypes.asn Ieee1609Dot2.asn
EtsiTs103097ExtensionModule.asn EtsiTs103097Module.asn
-pdu=all -fincludes-quoted
```

missä .asn-päätteiset sanat ovat ETSI:n teknisen spesifikaation TS 103 097 [47] liitteissä tarjottujen ASN.1-kielisten tiedostojen nimiä. DENM-viestiä varten tarvittavat rakenteet kopioitiin DENM-standardista [34] siltä osin kuin niitä käytettiin tässä tutkielmassa. Asn1c-kääntäjällä tuotettuja rakenteita karsittiin niin, että jäljelle jäivät vain ne, joita tarvittiin allekirjoitetun DENM-viestin ja siihen liittyvän sertifikaatin luomiseen.

4.2. Käytetyt allekirjoitusalgoritmit

Nykyisessä muodossaan ETSI:n tekninen spesifikaatio määrittelee, että ITS-asemien välisten viestien allekirjoittamisessa käytetään elliptisiä käyriä, joista valittavana ovat käyrätyypit NIST P-256, brainpoolP256r1 ja brainpoolP384r1 [47, 52]. Näistä NIST P-256 ja brainpoolP256r1 käyttävät 256-bittisiä elliptisten käyrien koordinaatteja ja brainpoolP384r1 384-bittisiä [6, 7]. Koska n bitin mittaisia koordinaatteja käyttävä elliptinen käyrä tarjoaa turvallisuustason, joka on luokkaa $n/2$ [2 s. 225], 256-bittisiä koordinaatteja käyttävä elliptinen käyrä tarjoaa 128 bitin turvallisuustason ja 384-bittisiä koordinaatteja käytettäessä turvallisuustaso on 192 bittiä.

Tämän työn tarkoitus on löytää tehokas ja nopea kvanttiturvallinen allekirjoitusalgoritmi älykkään liikenteen käyttöön. Suurempien parametriensa vuoksi käyrä brainpoolP384r1 tarjoaa korkeamman turvallisuustason klassisia tietokoneita vastaan, mutta samalla laskenta on hitaampaa kuin 256-bittisillä vaihtoehdoilla. Toisaalta tämä käyrä ei kuitenkaan tarjoa suojaa kvanttitietokoneita vastaan, joten kvanttitietokoneen kestävän kryptografian näkökulmasta se on lähes yhtä turvaton kuin 256-bittiset elliptiset käyrät. Sen takia tässä työssä vertaillaan kvanttitietokoneen kestäviä allekirjoitusalgoritmeja 256-bittisiin elliptisiin käyriin, jotka tällä hetkellä ovat nopein teknisten spesifikaatioiden tarjoama allekirjoitustapa.

4.2.1. Elliptisten käyrien toteutus

BrainpoolP256r1- ja NIST P-256 -käyrien ero on se, että ne käyttävät laskennassa erilaisia alkulukuja [65]. NIST P-256-käyrän käyttämät alkuluvut ovat tietyn muotoisia, mikä nopeuttaa laskentaa, mutta toisaalta asettaa käyrän alttiiksi useammille sivukanavahyökkäyksille [71]. Sivukanavahyökkäys tarkoittaa, että salauksen murtamisessa ei hyödynnetä salausjärjestelmän matemaattisia heikkouksia vaan muunlaista laskennasta saatavaa tietoa, esimerkiksi prosessorin laskentaan käyttämää aikaa [2 s. 175]. BrainpoolP256r1-käyrän käyttämät alkuluvut ovat satunnaisempia, minkä takia se on turvallisempi tiettyjä sivukanavahyökkäyksiä vastaan, mutta toisaalta tämä ominaisuus tekee matemaattisten operaatioiden suorittamisesta hitaampaa [71]. Tämän johdosta brainpoolP256r1-käyrä toimii viestien

allekirjoituksessa ja varmennuksessa hitaammin kuin NIST P-256. Koska molemmat käyrätyypit ovat hyväksyttyjä vaihtoehtoja ETSI:n teknisissä spesifikaatioissa, ne molemmat toteutetaan ja vertaillaan tässä tutkielmassa ja niiden oletetaan olevan yhtä turvallisia.

BrainpoolP256r1- ja NIST P-256 -käyrien turvallisuustaso vastaa NIST:n kilpailussa määriteltyä turvatasoa 1 [19] klassisia tietokoneita vastaan. NIST:n kilpailuun toimitettujen allekirjoitusalgoritmien referenssitoteutukset turvallisuustasolla 1 tarjoavat samanlaisen turvallisuustason myös kvanttitietokoneiden aikakaudella. Tämän takia tässä tutkielmassa vertaillaan 256-bittisiä elliptisiä käyriä NIST:n kilpailuun lähetettyjen allekirjoitusalgoritmien turvatasoon 1 tai sitä lähimpään saatavilla olevaan turvatasoon. Näin allekirjoitusten turvallisuustaso pysyy suurin piirtein samana sillä erotuksella, että kolme NIST:n kilpailun finalistialgoritmia tarjoavat turvallisuutta myös kvanttitietokoneella tehtävää hyökkäystä vastaan.

ASN.1-kielellä määritellyt rakenteet eivät sisällä elliptisten käyrien allekirjoitusten implementaatioita, joten nämä tuotiin OpenSSL:stä. OpenSSL on TLS- ja SSL-protokollien toteuttamiseen sopiva kirjasto, joka sisältää myös yleiskäyttöisen kryptografiakirjaston. OpenSSL:n kryptografiakirjasto sisältää elliptisten käyrien allekirjoituksia, ja valittavana on useita eri elliptisiä käyriä, mukana myös ETSI:n spesifikaatiossa määritellyt käyrät NIST P-256 ja brainpoolP256r1 [72]. Toteutuksessa käytettiin OpenSSL:n versiota 1.1.1. Elliptisten käyrien allekirjoitukset toteutettiin OpenSSL:llä siksi, että se on laajalti tunnettu, hyvin dokumentoitu ja vapaasti käytettävissä oleva kirjasto.

4.2.2. Kvanttiturvallisten allekirjoitusalgoritmien toteutus

Kaikista käytetyistä NIST:n kilpailun finalisteista on saatavilla C-kielinen referenssitoteutus, johon sisältyy NIST-rajapinta [73, 74, 75]. Rajapinnassa määritellään kolme eri funktiota: avainparin luominen, viestin allekirjoittaminen ja allekirjoitetun viestin varmentaminen. Avainparin luovaan funktioon syötetään sisään kaksi muistialuetta, joista toinen on varattu yksityistä ja toinen julkista avainta varten. Funktio täyttää nämä muistialueet kyseisillä avaimilla. Allekirjoitusfunktioon syötetään sisälle allekirjoitettava viesti, viestin pituus, yksityinen avain, muistialue allekirjoituksen säilömistä varten ja muistialue allekirjoituksen pituuden säilömistä varten. Funktio kirjoittaa allekirjoituksen ja allekirjoituksen pituuden osoitetuille muistipaikoille ja palauttaa tiedon siitä, onnistuiko allekirjoituksen luominen. Varmennusfunktioon syötetään allekirjoitus, allekirjoituksen pituus, julkinen avain, alkuperäinen viesti ja alkuperäisen viestin pituus. Funktio palauttaa tiedon siitä, onnistuiko viestin varmennus vai ei.

Kaikista kolmesta kvanttiturvallisesta allekirjoitusalgoritmista on finaaliin mennessä toimitettu NIST:n kilpailuun useita eri turvatasoja vastaavia toteutuksia. FALCONista on saatavilla kaksi eri versiota, NIST-turvatasoa 1 vastaava FALCON-512 ja turvatasoa 5 vastaava FALCON-1024 [76]. Tässä työssä käytetty toteutus on FALCON-512. CRYSTALS-Dilithiumista on toimitettu kolme eri versiota, jotka vastaavat turvatasoja 2, 3 ja 5 [77]. Koska turvatason 1 toteutusta ei ollut saatavilla, tässä työssä käytettiin lähintä saatavilla olevaa turvatasoa eli tasoa 2. Rainbow'sta on saatavilla kolme eri turvatasoa, jotka ovat tasot 1, 3 ja 5. Lisäksi siitä on toimitettu kolme erilaista versiota:

Standard Rainbow, Cyclic Rainbow ja Compressed Rainbow [78]. Nämä vaihtoehdot käyttävät keskenään hieman erilaisia avaimia. Standard Rainbow oli alkuperäinen esitelty algoritmi, ja muut vaihtoehdot tuotiin mukaan kilpailun toisella kierroksella [79]. Tässä toteutuksessa käytettiin Standard Rainbow'n turvatasoa 1.

Kaikista kolmesta finalistista on saatavilla myös optimoitu toteutus, joka hyödyntää AVX2-käskyjä [73, 74, 75]. AVX2 (Advanced Vector Extensions 2) on teknologiayhtiö Intelin tarjoama laajennus, joka mahdollistaa suoritettavan ohjelman optimoinnin käyttämällä erityisiä rinnakkaisprosessoinnin mahdollistavia käskyjä. Useat prosessorityypit tukevat AVX2-käskyjä, mutta eivät kaikki [80]. Autoihin sisällytettävien OBU-laitteiden prosessoriarkkitehtuurit vaihtelevat, joten ei voitane luottaa siihen, että kaikki saatavilla olevat laitteet tukisivat AVX2-käskyjä. Esimerkiksi OBU:issa yleiset ARM-prosessorit perustuvat erilaiseen arkkitehtuuriin kuin Intelin prosessorit [61] eivätkä siten hyödy AVX2-kiihdytyksestä. Tämän takia tässä diplomityössä ei käytetty allekirjoitusalgoritmien AVX2-optimoituja versioita. C-kielisten referenssitoteutusten vertailu tarjonnee laajimmin käyttökelpoisia tuloksia, vaikka ne eivät ehkä annakaan täysin totuudenmukaista kuvaa algoritmien suorituskyvystä.

Kaikista kvanttiturvallisista allekirjoitusalgoritmeista käytettiin rajapinnan tarjoamia funktioita nimeltä *crypto_sign* ja *crypto_sign_open*. Nämä funktiot sisälsivät tämän ohjelman toimintaan nähden ylimääräisiä toiminnallisuuksia. Funktiossa *crypto_sign* allekirjoitettava viesti kopioitiin varsinaisen allekirjoituksen jatkoksi allekirjoitukselle varattuun muistitilaan, ja viestiä varmennettaessa funktiossa *crypto_sign_open* allekirjoituksen mukana ollut alkuperäinen viesti erotettiin omaan muistitilaansa. Tämä oli tämän työn kontekstissa tarpeetonta, sillä käytetyt tekniset spesifikaatiot olettivat allekirjoitettavan viestin olevan eri paikassa allekirjoitettua viestirakennetta. CRYSTALS-Dilithium olisi tarjonnut myös kaksi yksinkertaistettua funktiota allekirjoitusta ja varmennusta varten, *crypto_sign_signature* ja *crypto_sign_verify*. Nämä funktiot suorittivat ainoastaan viestin allekirjoittamisen ja varmennuksen ilman alkuperäisen viestin kopiointia. Rainbow ja FALCON eivät tarjonneet näitä yksinkertaisempia funktioita. Tasapuolisuuden vuoksi myös CRYSTALS-Dilithiumin kohdalla käytettiin samoja funktioita kuin FALCONin ja Rainbow'n kohdalla. Allekirjoitettavan viestin kopioiminen suorituksen aikana saattoi aiheuttaa hieman lisäviivettä kvanttiturvallisten vaihtoehtojen suoritusaikoihin, mutta toisaalta keskinäisessä vertailussa ne kaikki toimivat samalla tavalla.

4.3. Integrointi

Integrointivaiheessa kvanttiturvallisten allekirjoitusalgoritmien C-kieliset toteutukset liitettiin ASN.1-kieleltä käännettyihin tietorakenteisiin. Tämä tarkoitti sitä, että allekirjoitusta ja julkista avainta kuvaavia rakenteita muutettiin niin, että niissä saatettiin käyttää myös jotain näistä kvanttiturvallisista allekirjoitusalgoritmeista.

Teknisen spesifikaation mukaan julkista avainta kuvaava rakenne (PublicVerificationKey) sisältää tiedon siitä, onko käyräksi valittu NIST P-256, brainpoolP256r1 vai brainpoolP384r1. Mitä tahansa näistä käytettäessä varsinainen avain ilmaistaan elliptisen käyrän pisteen x- ja y-koordinaateilla. Spesifikaation mukaan koordinaateista tallennetaan muistialue, jonne koordinaatti on tallennettu,

sekä koordinaatin pituus tavuina [47]. Tässä työssä julkista avainta kuvaavaan kenttään lisättiin mahdollisuus valita käytettäväksi avaintyypiksi CRYSTALS-Dilithium, FALCON tai Rainbow, ja mahdollisuus valita brainpoolP384r1 poistettiin. Rakenteen muutos on esitetty kuvassa 9. Tämä rakenne sisältyy kuvissa 14 ja 15 SignedData-rakenteen kenttään "Sertifikaatti". Kenttien nimissä esiintyvä ECDSA (Elliptic Curve Digital Signature Algorithm) viittaa elliptisten käyrien allekirjoitukseen.

Kuva 9. Rakenteen muokkaus: sertifikaattiin sisällytettävän julkisen avaimen rakenne alkuperäisessä spesifikaatiossa ja muokattuna tätä diplomityötä varten.

Koska sekä CRYSTALS-Dilithiumin, FALCONin että Rainbow'n NIST-rajapinnoissa avainten luomisessa käytettävä funktio ottaa sisääntuloksi kaksi muistialuetta, joihin yksityinen ja julkinen avain tallennetaan, toteutettiin näiden julkisia avaimia kuvaavaan rakenteeseen muistialue sekä muistialueen pituus. Pituudeksi asetettiin se pituus, joka kussakin rajapinnassa oli määritelty julkisen avaimen vaatimaksi tilaksi tavuina.

Allekirjoitus on määritelty spesifikaatioissa saman tyyppisesti kuin julkinen avain. Rakenne sisältää tiedon siitä, onko allekirjoitus tehty NIST P-256 -, brainpoolP256r1- vai brainpoolP384r1-käyrällä. Kuten mainittu kappaleessa 2.1, elliptisten käyrien allekirjoitukseen sisältyy kaksi lukua, joista toinen on elliptisen käyrän pisteen x-koordinaatti ja toinen varsinainen allekirjoitus. Spesifikaation mukaisessa rakenteessa x-koordinaatista tallennetaan muistialue, jonne koordinaatti on tallennettu, sekä koordinaatin pituus. Myös varsinaisesta allekirjoituksesta tallennetaan samat tiedot eli muistialue ja allekirjoituksen pituus [47]. Tässä työssä allekirjoitusrakenteelle tehtiin samat muokkaukset kuin julkisen avaimen rakenteelle: siitä poistettiin mahdollisuus käyttää brainpoolP384r1-allekirjoitusta ja lisättiin mahdollisuus valita allekirjoitustyypiksi joko CRYSTALS-Dilithium, FALCON tai Rainbow. Rakenteen

muutos on esitetty kuvassa 10. Tämä rakenne sisältyy kuvissa 14 ja 15 SignedData-rakenteen kenttään "Allekirjoitus".

Kuva 10. Rakenteen muokkaus: allekirjoitettuun viestiin sisällytettävän allekirjoituksen rakenne alkuperäisessä spesifikaatiossa ja muokattuna tätä diplomityötä varten.

NIST-rajapinnassa määritellyt allekirjoitusfunktiot ottavat sisääntuloksi yhden muistialueen, jonne allekirjoitus kirjoitetaan, ja palauttavat syntyneen allekirjoituksen pituuden. Siksi kaikkia näitä allekirjoituksia vastaavaan rakenteeseen toteutettiin kaksi kenttää: yksi kenttä muistialueelle ja toinen kenttä muistialueen pituudelle.

4.4. Tiivistefunktiot

NIST:n standardointikilpailuun toimitettujen referenssitoteutusten ja elliptisten käyrien OpenSSL-toteutuksen toiminnallisuudet eroavat hieman toisistaan. OpenSSL:n tarjoama elliptisten käyrien allekirjoitusfunktio olettaa, että allekirjoitettavana on viestistä tiivistefunktion avulla saatu tiiviste [81]. Tämän takia OpenSSL:n elliptisten käyrien allekirjoitusfunktion yhteydessä on käytettävä myös tiivistefunktiota. OpenSSL tarjoaa ETSI:n spesifikaation hyväksymät SHA256- ja SHA384-tiivistefunktiot [82], joista tässä työssä käytettiin SHA256-vaihtoehtoa. Kvanttiturvallisten allekirjoitusalgoritmien NIST-rajapinnan allekirjoitusfunktiot taas olettavat sisääntulon olevan kokonainen allekirjoitettava viesti, ja tiivistefunktion käyttö tapahtuu allekirjoitusfunktion sisällä. Kvanttiturvallisten algoritmien referenssitoteutukset mahdollistivat siis sen, että viestin allekirjoittamisessa voitiin jättää yhden kerran tiivistefunktion käyttö väliin. Siksi OpenSSL:llä toteutettujen elliptisten käyrien allekirjoitusten suoritusaikoihin laskettiin suoritusaikamittauksissa mukaan myös tiivistefunktion käyttöön kulunut aika.

Monimuuttujamenetelmäisessä Rainbow'ssa tiivistefunktion käyttö on saman tyyppistä kuin elliptisten käyrien kohdalla, eli allekirjoitettavasta viestistä saadaan tiiviste OpenSSL:n SHA256-tiivistefunktiota käyttäen [75]. Hilapohjaisissa CRYSTALS-Dilithiumissa ja FALCONissa tiivistefunktion rooli on erilainen, ja sitä käytetään erilaisten matemaattisten operaatioiden yhteydessä. Näihin operaatioihin soveltuvan tiivistefunktion tulee olla ominaisuuksiltaan tietynlainen XOF-tyyppinen (extendable-output hash function) funktio, joka kykenee tuottamaan vaihtelevanmittaisia ulostuloja. Sekä CRYSTALS-Dilithiumissa että FALCONissa tähän tarkoitukseen on valittu SHAKE256-niminen tiivistefunktio [22, 24].

4.5. Ohjelmointiympäristö

Mittauksiin käytetyn tietokoneen käyttöjärjestelmä oli Ubuntu 16.04 LTS ja prosessori Intel Core i7-8665U CPU, jonka peruskellotaajuus on 1.90 GHz. C-kieliset ohjelmat käännettiin GCC:n versiolla 5.4.0. Ohjelmasta luotiin viisi erillistä versiota, kullekin allekirjoitusalgoritmille omansa. GCC tarjoaa erilaisia vaihtoehtoja käännettävän ohjelman optimointiin [83]. Ohjelmat käännettiin käyttäen sellaista GCC-optimointia, jota kyseisen algoritmin sisältäneessä ohjelmistopaketissa oli oletuksena käytetty tiedostojen kääntämiseen. Elliptisten käyrien, Rainbow'n ja CRYSTALS-Dilithiumin kohdalla tämä tarkoitti optimointia -O3 ja FALCONin kohdalla optimointia -O2. Koska kaikkien allekirjoitusalgoritmien toteutukset käyttivät OpenSSL-kirjastoa ainakin tiivisteen luomiseen, käytettiin käännettäessä tähän liittyviä komentoja. Kvanttiturvallisten allekirjoitusalgoritmien toteutukset sisälsivät useita lisätiedostoja, joita ilman ohjelmat eivät toimineet. Nämä sisällytettiin mukaan ohjelmia käännettäessä. Lopullinen käsky, jolla ohjelmat käännettiin, oli esimerkiksi

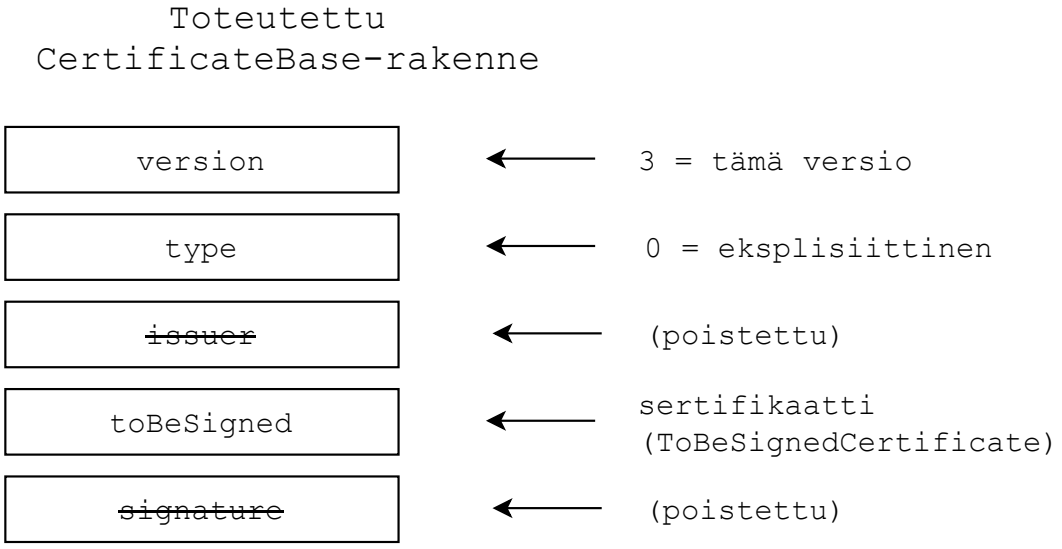
```
gcc -o signVerify etsi_nistp256.c -lssl -lcrypto -O3
```

jos haluttiin tuottaa ohjelma nimeltä signVerify etsi_nistp256.c-nimisestä tiedostosta käyttäen optimointia -O3.

4.6. Ohjelmallinen toteutus

Jotta viestien allekirjoittamisen ja varmennuksen tehokkuutta voitiin testata, luotiin ohjelma, joka vuorotellen loi, allekirjoitti ja varmensi DENM-viestejä. Ensimmäiseksi ohjelmassa luotiin valitun allekirjoitusalgoritmin mukainen avainpari joko NIST-rajapinnassa määritellyllä tai OpenSSL:stä tuodulla funktiolla. Muistista varattiin kunkin allekirjoitusalgoritmin avainten pituuksien mukaiset tilat yksityiselle ja julkiselle avaimelle, ja algoritmin toteutuksessa tarjotulla funktiolla nämä kentät täytettiin avaimilla. Elliptisten käyrien kyseessä ollessa avainten luomiseen käytettiin OpenSSL:n funktiota *EC_KEY_generate_key* ja kvanttiturvallisten algoritmien kohdalla NIST-rajapinnassa tarjottua funktiota *crypto_sign_keypair*. Luodun avainparin tyyppi ratkaisi sen, mitä kenttiä myöhemmin täytettiin sertifikaattia ja allekirjoitettua datarakennetta luotaessa.

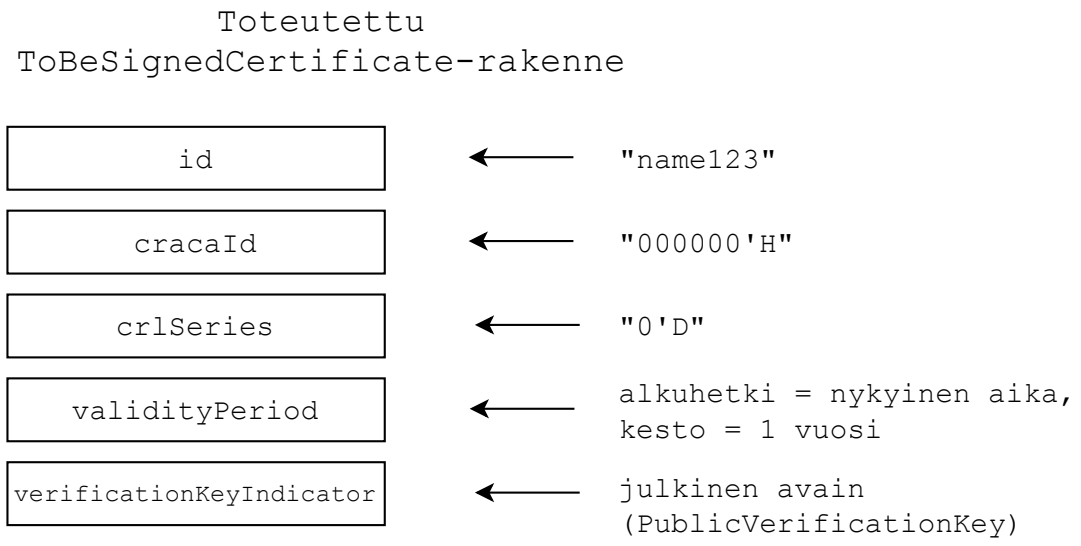
Kun avaimet olivat valmiit, ohjelmassa luotiin yksinkertaistettu versio ETSI:n spesifikaatiossa [47] määritellystä sertifikaatin sisältävästä rakenteesta (CertificateBase). Spesifikaation mukaisesti ITS-asemien sertifikaattien mukana toimitetaan luotettavaksi tiedetyn tahon allekirjoitus, josta voidaan varmistaa sertifikaatin aitous. Tämän tutkielman aihepiiriin ei kuitenkaan kuulunut kokonaisen sertifikaattiketjun toteuttaminen, joten sertifikaatin myöntäjälle ja allekirjoitukselle varatut kentät poistettiin. Muut kentät täytettiin kuten spesifikaatiossa määritellään: sertifikaatin versioksi täytettiin 3 ja tyypiksi 0 eli eksplisiittinen. Toteutetun rakenteen sisältö on esitetty kuvassa 11.



Kuva 11. Tätä työtä varten toteutettu rakenne, johon ITS-aseman sertifikaatti sisällytettiin, sisältöineen. ToBeSignedCertificate viittaa toiseen teknisissä spesifikaatioissa määriteltyyn rakenteeseen.

Itse sertifikaatin rakennetta ei muokattu, vaan se toteutetiin samalla tavalla kuin se on määritelty spesifikaatiossa [47] ja esitelty kappaleessa 3.1.1. Koska kyseessä oli kuvitteellinen ITS-asema, sertifikaatille keksittiin kuvitteellinen voimassaoloaika ja ITS-aseman nimi: voimassaoloajaksi asetettiin yksi vuosi, voimassaolon alkamisajaksi sertifikaatin luomishetki ja ITS-aseman nimeksi merkkijono "name123". Kieltolistalle asettamiseen liittyvät tiedot täytettiin kuten spesifikaatiossa oli määritelty: crlSeries-kenttään kirjoitettiin merkkijono "0’D"ja cracaId-kenttään "000000’H". Ohjelman toiminnallisuuden kannalta tärkeä osa sertifikaattia oli kenttä, joka sisälsi tiedon allekirjoituksen varmentamiseen käytettävän julkisen avaimen tyypistä ja itse avaimen. Rakenteen muokkauksen jälkeen valittavana oli NIST P-256 -, brainpoolP256r1-, CRYSTALS-Dilithium-, FALCON- tai Rainbow-tyyppinen julkinen avain. Kenttään kopioitiin julkinen osa luodusta avainparista. Toteutetun rakenteen sisältö on esitetty kuvassa 12.

Sertifikaatin jälkeen ohjelmassa luotiin DENM-standardin [34] mukainen DENM-viesti. Allekirjoitettavan viestin yksityiskohdat eivät vaikuta allekirjoitusalgoritmien suorituskykyyn, sillä allekirjoitettavana on aina viestistä luotu tiiviste eikä viesti sellaisenaan. Asetelman realistisuuden vuoksi viestin kentät kuitenkin täytettiin. Tässä tutkielmassa DENM-viestiin sisällytettiin pakollisten otsikko- ja metatietokenttien



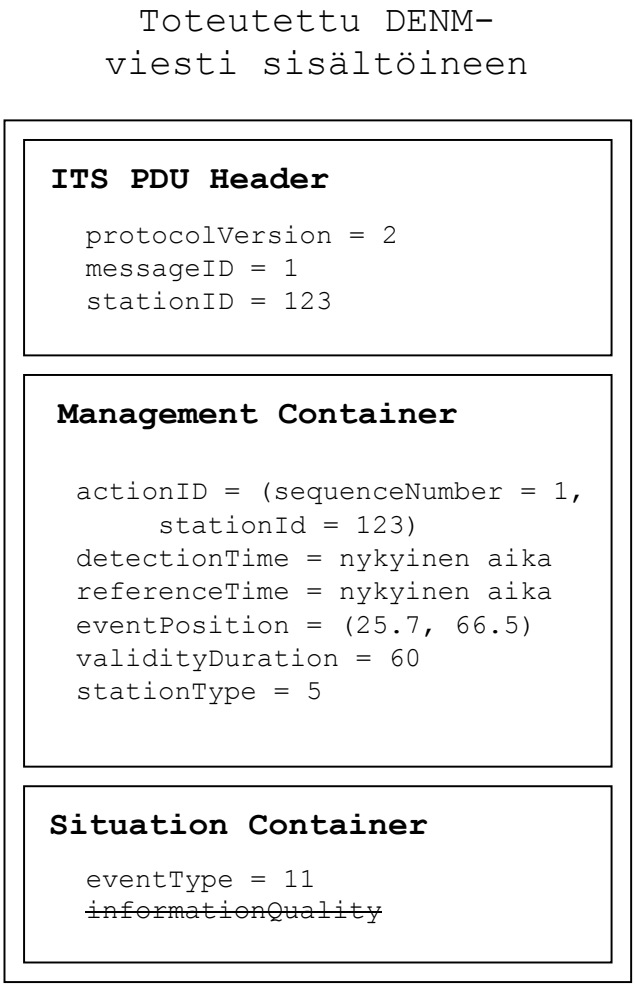
Kuva 12. Tätä työtä varten toteutettu ITS-aseman sertifikaattia kuvaava rakenne sisältöineen. PublicVerificationKey viittaa toiseen teknisissä spesifikaatioissa määriteltyyn rakenteeseen.

lisäksi valinnainen kenttä Situation Container, joka sisältää tiedot tapahtuman luonteesta sekä havainnon varmuudesta. Koska tutkielmassa toteutettiin ainoastaan kommunikointiin liittyvä osuus eikä lainkaan esimerkiksi ympäristöä havainnoivia sensoreita, yksinkertaisuuden vuoksi havainnon varmuutta kuvaava kenttä jätettiin toteutuksesta pois ja kenttään sisällytettiin ainoastaan tapahtumaa kuvaava syykoodi.

Otsikkokentän protokollaversio ja viestin tunniste täytettiin kuten DENM-standardissa määritellään: protokollaversioksi täytettiin 2 ja viestin tunnisteeksi 1. Viestin loput kentät täytettiin kuvitteellisilla tapahtumaa kuvailevilla tiedoilla. Viestin järjestysnumeroksi täytettiin 1. Lähettävän ITS-aseman tunnisteeksi täytettiin kuvitteellinen ITS-aseman tunniste 123. Tapahtuman havaitsemisajaksi ja viestin luomisajaksi asetettiin sen hetkinen aika, joka saatiin tietokoneen kellon avulla. Viestin voimassaoloajaksi asetettiin 60 sekuntia. ITS-aseman tyypiksi asetettiin 5 eli tavallinen henkilöauto [50]. Maantieteellinen sijainti oli 25,7 astetta pituutta ja 66,5 astetta leveyttä. Syykoodiksi täytettiin 11, joka tarkoittaa, että tiellä on havaittu eläimiä [34]. Luodun viestin rakenne ja sisältö on esitetty kuvassa 13.

Kun DENM-viesti ja sertifikaatti olivat valmiit, luotiin niiden pohjalta allekirjoitetun datan sisältävä rakenne (EtsiTs103097Data-Signed). Se sisälsi ETSI-spesifikaation [47] mukaisesti tiedon käytettävästä tiivistefunktiosta, allekirjoitettavasta datasta, allekirjoittajan tiedot ja allekirjoituksen. Allekirjoittajan tietoihin sisällytettiin aiemmin luotu sertifikaatti. Allekirjoitettavan datan kohdalle kopioitiin luotu DENM-viesti. Tiivistefunktioksi valittiin SHA256, ja allekirjoitettavasta viestistä luotiin tiivistefunktion avulla 256-bittinen tiiviste. IEEE-standardin [52] mukaisesti tähän merkkijonoon liitettiin vielä toinen, lähettäjän tietoja kuvaavasta merkkijonosta luotu 256-bittinen tiiviste.

Koska kaikkien kolmen kvanttiturvallisen allekirjoitusalgoritmin referenssitoteutukset sisälsivät tiivistefunktion, voitiin niillä allekirjoittaa suoraan tämä 512-bittinen yhdistetty merkkijono käyttäen funktiota *crypto_sign*. Elliptisten käyrien

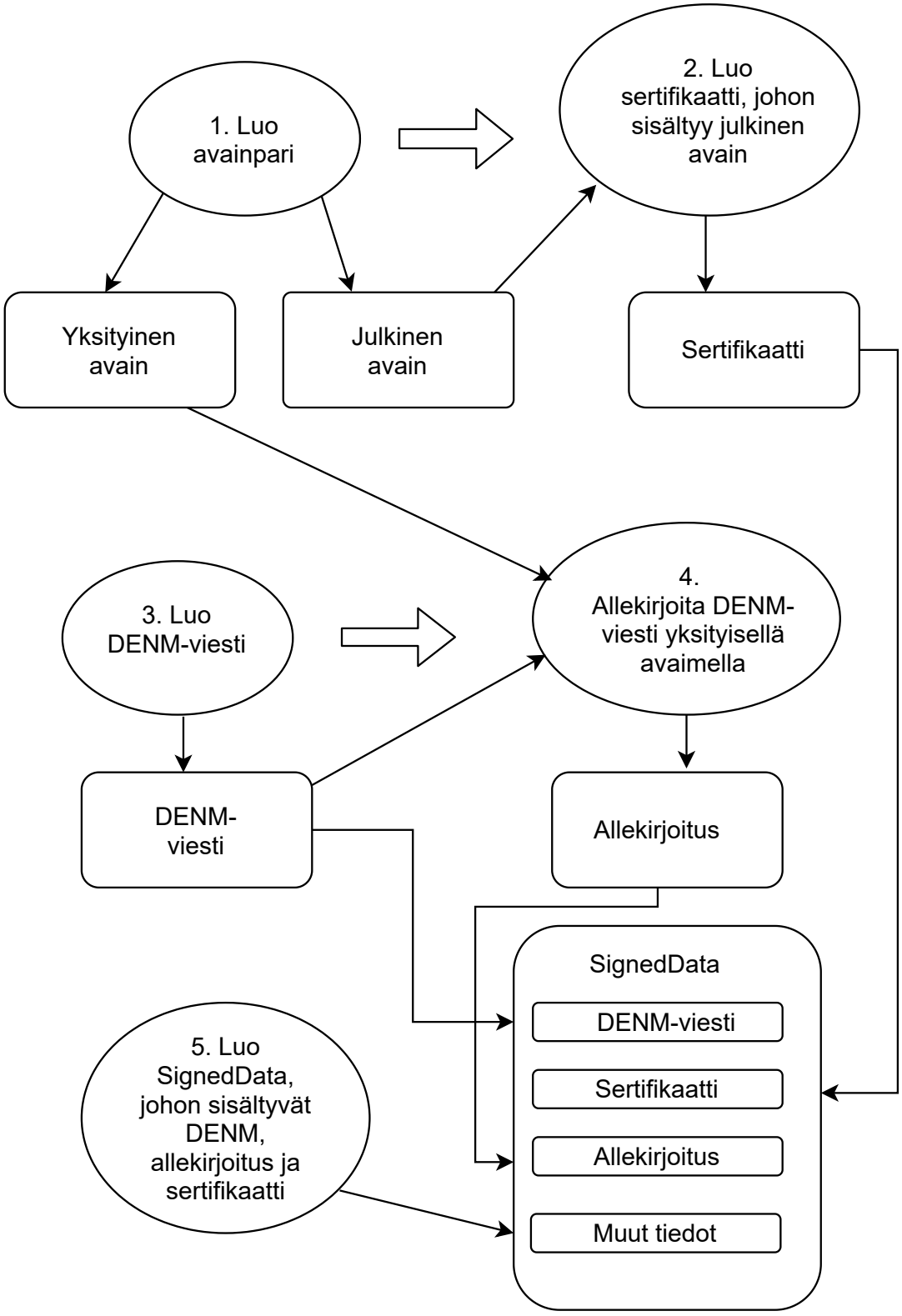


Kuva 13. Tätä tutkielmaa varten tehdyssä toteutuksessa luotu DENM-viesti sisältöineen. Yliviivattu kenttä on poistettu.

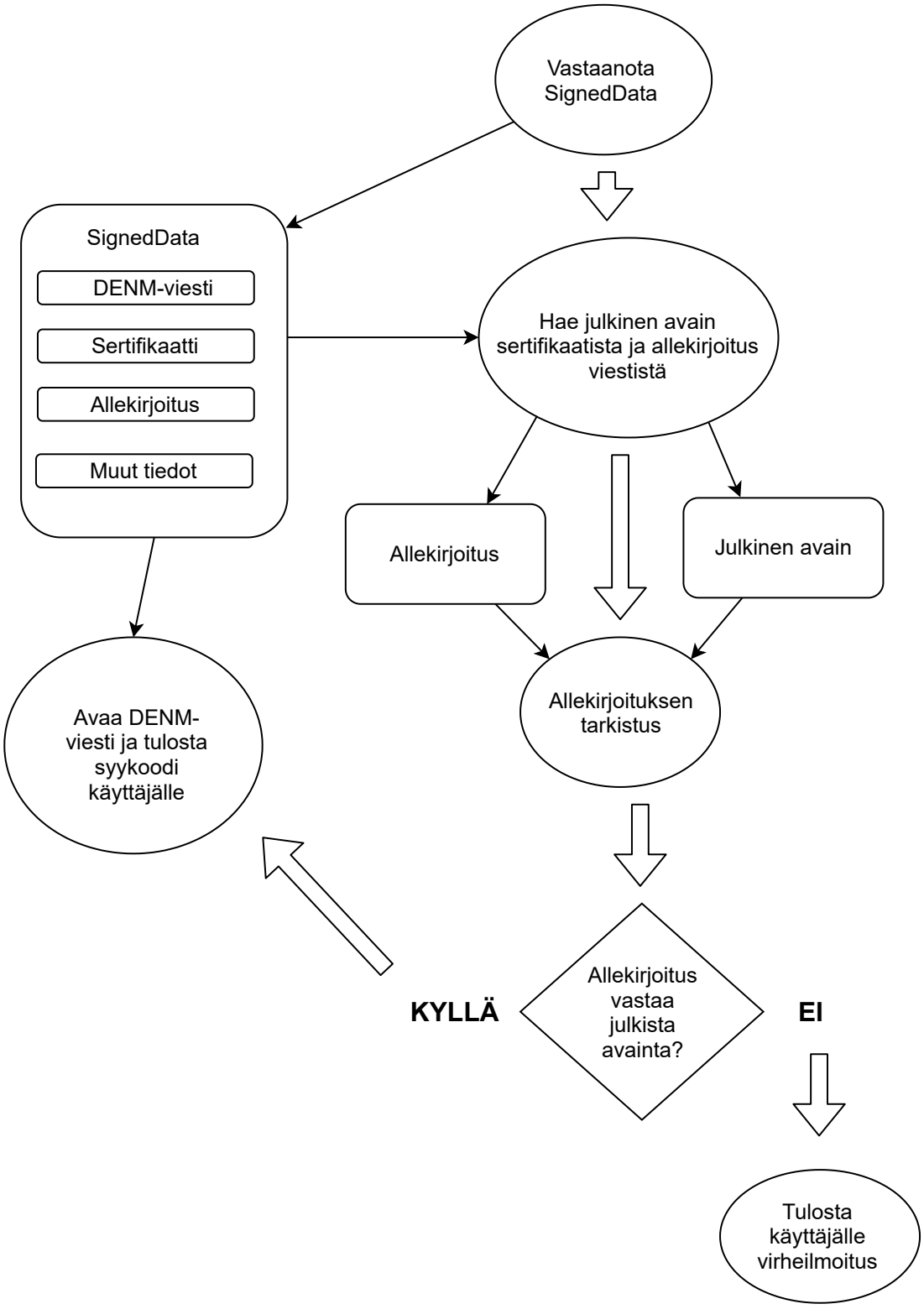
tapauksessa allekirjoitettava viesti oli tästä merkkijonosta vielä kertaalleen luotu 256-bittinen tiiviste, joka allekirjoitettiin käyttäen OpenSSL-funktiota *ECDSA_sign*.

Allekirjoituksessa käytettiin ohjelman alussa luotua yksityistä avainta. Yksityistä avainta ei sisällytetty lähetettävään viestiin, vaan viestin varmennus tapahtui sertifikaatissa mainitulla julkisella avaimella. Ohjelman toiminta tähän saakka on kuvattu kuvassa 14.

Kun allekirjoitetun datan sisältävä rakenne oli valmis, siirryttiin ohjelman toiseen osioon eli viestin varmennukseen. Allekirjoitetusta rakenteesta etsittiin allekirjoitus ja allekirjoitettava viesti sekä allekirjoittajan sertifikaatista tämän julkinen avain. Allekirjoitus varmennettiin elliptisten käyrien tapauksessa OpenSSL:n funktiolla *ECDSA_verify* ja kvanttiturvallisten allekirjoitusalgoritmien yhteydessä funktiolla *crypto_sign_open*. Funktiot palauttivat tiedon siitä, oliko allekirjoituksen todentaminen onnistunut vai epäonnistunut. Jos varmennus ei onnistunut, näytölle tulostettiin tieto epäonnistumisesta eikä DENM-viestiä avattu. Jos varmennus onnistui, DENM-viestistä haettiin viestin syykoodi ja tämä tulostettiin näytölle. Ohjelman toisen osion toiminta on kuvattu kuvassa 15.



Kuva 14. Kaavio ohjelman toiminnasta: ohjelman ensimmäinen osa eli allekirjoitetun viestin luominen.



Kuva 15. Kaavio ohjelman toiminnasta: ohjelman jälkimmäinen osa eli viestin avaaminen, allekirjoituksen varmennus ja varmennuksen lopputuloksen ilmoittaminen käyttäjälle.

Kuvissa 16 ja 17 näkyvät kuvakaappaukset ohjelman toiminnasta varmennuksen onnistuessa ja epäonnistuessa. Suoritusaikamittausten yhteydessä ei kuitenkaan tulostettu mitään, jottei tämä olisi vaikuttanut suoritusaikamittauksiin.

```
Dilithium-avainpari valmis
Sertifikaatti valmis
Luodaan DENM-viestiä numero 1
Lähetettävä syykoodi: 11        ELÄIMIÄ TIELLÄ!
DENM-viesti valmis
Allekirjoitus valmis
SignedData-rakenne luotu ja allekirjoitettu salaisella Dilithium-avaimella
Haetaan julkista avainta sertifikaatista...
Tarkistetaan allekirjoitusta... todennettu julkisella Dilithium-avaimella
Syykoodi: 11    ELÄIMIÄ TIELLÄ!
```

Kuva 16. Kuvakaappaus ohjelman toiminnasta, kun allekirjoituksen varmennus onnistuu. Käytettävä allekirjoitusalgoritmi on tässä CRYSTALS-Dilithium.

```
Dilithium-avainpari valmis
Sertifikaatti valmis
Luodaan DENM-viestiä numero 1
Lähetettävä syykoodi: 11        ELÄIMIÄ TIELLÄ!
DENM-viesti valmis
Allekirjoitus valmis
SignedData-rakenne luotu ja allekirjoitettu salaisella Dilithium-avaimella
Haetaan julkista avainta sertifikaatista...
Tarkistetaan allekirjoitusta... ei voitu todentaa julkisella Dilithium-avaimella
```

Kuva 17. Kuvakaappaus ohjelman toiminnasta, kun allekirjoituksen varmennus ei onnistu. Käytettävä allekirjoitusalgoritmi on tässä CRYSTALS-Dilithium.

4.7. Suoritusaikojen ja viestikokojen mittaaminen

Allekirjoitusalgoritmien suoritusaikaa vertailtiin sekä viestiä allekirjoitettaessa että allekirjoitusta varmennettaessa. Ohjelmaan kuului myös avainparin luominen, mutta tämän suoritusaikaa ei mitattu. ETSI:n spesifikaatiossa [48] määritellyssä viestintäympäristössä avaimia ei luoda reaaliaikaisesti digitaalisen allekirjoituksen yhteydessä, vaan allekirjoituksessa ja varmennuksessa käytetään avaimia, jotka on luotu jo aiemmin väliaikaisten sertifikaattien hankkimisen yhteydessä. Tämän takia avainten luomiseen kuluva aika ei suoranaisesti vaikuta reaaliaikaisen viestinnän tehokkuuteen.

Kaikki suoritusajat mitattiin 5000 suorituskerran keskiarvosta käyttäen C-kielen komentoa *clock*, joka mittaa kulunutta aikaa mikrosekunteina. Tiedot kerättiin suorittamalla allekirjoitetun viestin luomista ja varmennusta 5000 kertaa peräjälkeen. Aluksi luotiin ohjelmallisesti avainpari. Sen jälkeen jokaisella suorituskerralla luotiin uusi DENM-viesti ja sertifikaatti. DENM-viesti allekirjoitettiin, ja allekirjoitettu viesti, sertifikaatti ja allekirjoitus kerättiin allekirjoitettuun viestirakenteeseen. Allekirjoitukseen kuluvan ajan mittaus aloitettiin kvanttiturvallisten allekirjoitusalgoritmien kohdalla juuri ennen allekirjoituksen aloittamista ja lopetettiin heti, kun allekirjoitusfunktio oli palauttanut tiedon siitä,

oliko allekirjoitus onnistunut. Elliptisten käyrien kohdalla ajanotto aloitettiin ennen viimeistä tiivistefunktiota, joka suoritettiin juuri ennen allekirjoitusta, ja lopetettiin heti allekirjoitusfunktion palautettua tiedon allekirjoituksen onnistumisesta. Kaikkiin 5000 allekirjoitukseen kuluneet ajat laskettiin yhteen ja lopuksi laskettiin näiden keskiarvo jakamalla summa 5000:lla. Varmennuksiin kuluvan ajan mittaaminen aloitettiin joka kierroksella juuri ennen varmennusta ja lopetettiin heti, kun varmennusfunktio oli palauttanut tiedon siitä, oliko varmennus onnistunut. Kaikkiin 5000 varmennukseen kuluneet ajat laskettiin yhteen ja keskimääräinen suoritusaika laskettiin jakamalla summa 5000:lla.

Suoritusaikojen lisäksi kiinnostavaa oli eri allekirjoitusalgoritmeja käytettäessä syntyneen allekirjoitetun viestirakenteen koko. Tämä mitattiin tekemällä kutakin algoritmia käyttävästä toteutuksesta versio, joka ohjelman ensimmäisen osion lopuksi mittasi ja tulosti tiedon siitä, kuinka monta tavua kukin viestin osio vei muistista. Allekirjoitusten koot saatiin suoraan allekirjoitusfunktioista, koska nämä palauttivat tiedon syntyneen allekirjoituksen pituudesta. Julkisen avaimen koko mitattiin elliptisten käyrien osalta OpenSSL:n käskyllä *BN_num_bytes*, koska koordinaatteja käsiteltiin muodossa *BIGNUM* [84]. Kvanttiturvallisten algoritmien toteutukset sisälsivät etukäteen muuttujan *CRYPTO_PUBLICKEYBYTES*, joka kertoi julkisen avaimen vaatiman tavumäärän. Muun datan ja koko sertifikaatin koko mitattiin laskemalla sen osien koot C-kielen käskyllä *sizeof* ja summaamalla nämä.

5. TULOKSET JA POHDINTA

Tässä kappaleessa käydään läpi tulokset, jotka saatiin edellisessä osiossa selitetyn ohjelmallisen toteutuksen pohjalta. Huomiota kiinnitetään suoritusaikoihin, syntyneiden viestien kokoihin sekä siihen, kuinka helppoa allekirjoitusalgoritmien C-kielisten toteutusten sovittaminen älykkään liikenteen teknisiin spesifikaatioihin oli. Saaduista tuloksista voidaan myös päätellä jotain algoritmien soveltuvuudesta liikennekäyttöön.

Kvanttiturvallisten allekirjoitusalgoritmien tarjolla olevat C-kieliset toteutukset sisälsivät rajapinnan, joka kattoi avainten luomiseen, viestin allekirjoittamiseen ja allekirjoituksen tarkistamiseen tarvittavat funktiot. Näissä funktioissa sekä allekirjoitus että julkinen avain ilmaistiin varattuna muistitilana ja sen pituutena. Tämä esitys vastasi lähes sellaisenaan ASN.1-kielisissä tiedostoissa valmiiksi määriteltyä rakennetta, johon elliptisten käyrien allekirjoituksen ja julkisen avaimen arvot säilöttiin. Sen takia uusien algoritmien avainten ja allekirjoitusten integroiminen ETSI:n teknisessä spesifikaatiossa määriteltyihin rakenteisiin oli kohtalaisen yksinkertaista.

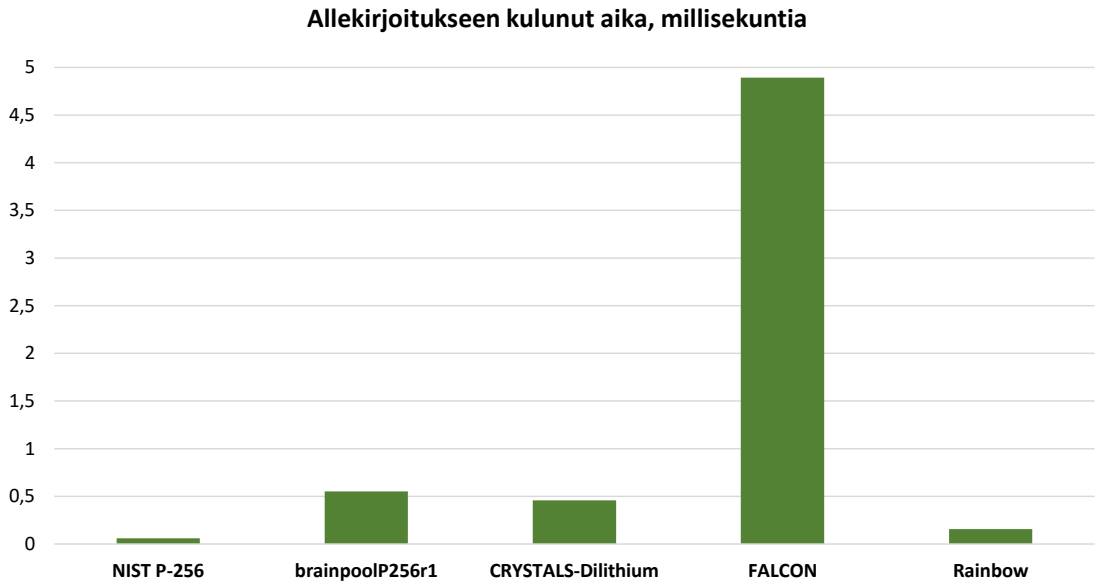
Koska nykyinen tekninen spesifikaatio olettaa allekirjoitukseen käytettävän ainoastaan elliptisiä käyriä, tuli kvanttiturvallisten algoritmien kohdalla julkisesta avaimesta poistaa yksi rakenne, joka määritteli, onko muistialueeseen säilötty elliptisen käyrän pelkkä x-koordinaatti vai sekä x- että y-koordinaatit. Lisäksi siinä, missä elliptisten käyrien allekirjoitus ilmaistiin kahdella eri arvolla, oli kaikkien kvanttiturvallisten allekirjoitusalgoritmien toteutukset suunniteltu niin, että koko allekirjoitus tallennettiin yhteen muuttujaan. Kyseessä lienee puhtaasti ohjelmoinnillinen ratkaisu; esimerkiksi FALCONin allekirjoitus koostuu tosiasiassa kahdesta arvosta [24], mutta C-kielisessä referenssitoteutuksessa ne kirjoitettiin peräjälkeen samaan muuttujaan. Tämän valinnan takia kvanttiturvallisten algoritmien allekirjoitusta ja julkista avainta varten toteutetuista rakenteista tuli yksinkertaisempia kuin elliptisten käyrien vastaavista. Niiden säilömiseen vaadittu muistitila oli kuitenkin huomattavasti suurempi, kuten tuloksista nähdään.

Tekniseen spesifikaatioon liitetty elliptisten käyrien allekirjoitus ja julkinen avain säilöttiin ja sitä käsiteltiin samalla tavalla riippumatta siitä, oliko valittu brainpoolP256r1- vai NIST P-256 -käyrä. Samoin kaikki kvanttiturvalliset algoritmit toteuttivat samanlaisen rajapinnan, joten ne voitiin säilöä samanlaisiin rakenteisiin vaihdellen vain varatun muistialueen pituutta ja joissakin tapauksissa muuttujan tietotyyppiä. Yhden kvanttiturvallisen allekirjoitusalgoritmin integroimisen jälkeen kahden muun lisääminen onnistui siis kohtuullisen helposti.

Kvanttiturvallisten allekirjoitusalgoritmien C-kieliset referenssitoteutukset sisälsivät joitakin älykkään liikenteen teknisiin spesifikaatioihin nähden ylimääräisiä ominaisuuksia, kuten allekirjoitettavan viestin toistamisen allekirjoituksen yhteydessä. Tämä johtuu siitä, että näitä referenssitoteutuksia ei ole alun perin suunniteltu tätä loppukäyttötarkoitusta varten. Koska niiden liittäminen teknisiin spesifikaatioihin oli tästä huolimatta kohtalaisen helppoa, voitaneen sanoa, että kaikki kolme referenssitoteutusta olivat selkeydessään kiitettävällä tasolla.

5.1. Suoritusaikavertailujen tulokset

DENM-viestin allekirjoittamiseen ja allekirjoituksen varmennukseen tarvittujen suoritusaikojen keskiarvot mitattuna 5000 kierroksen keskiarvosta on esitelty kuvissa 18 ja 19. Käytettävä yksikkö on millisekunti.

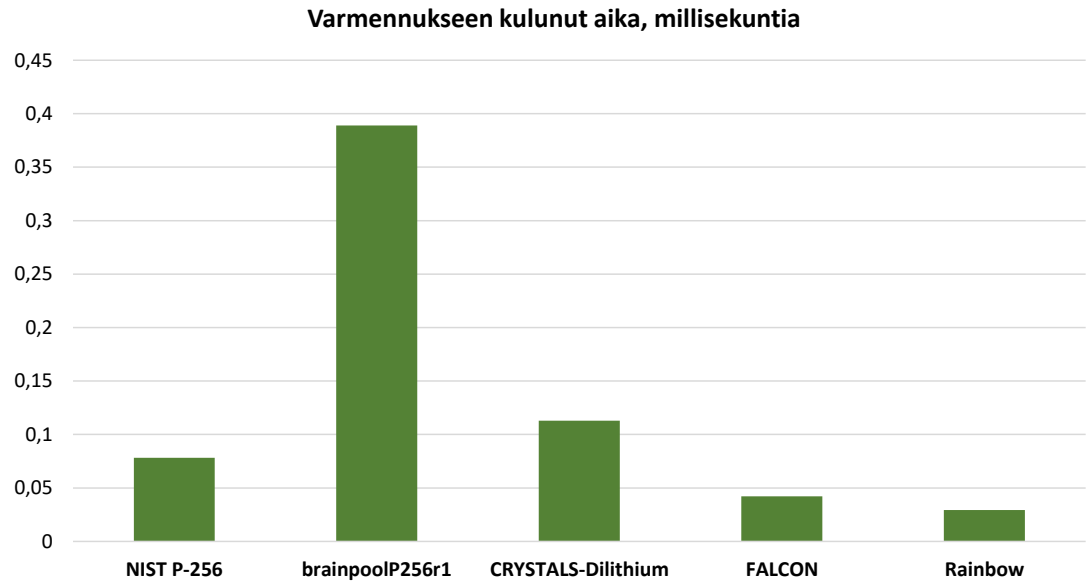


Kuva 18. Viestin allekirjoitukseen kulunut keskimääräinen aika eri allekirjoitusalgoritmeja käytettäessä.

Allekirjoitusvaiheessa elliptisten käyrien allekirjoitus NIST P-256 -käyrää käyttäen oli selvästi kaikkein nopein vaihtoehto 0,06 millisekunnin suoritusajallaan. Toiseksi nopein vaihtoehto oli Rainbow, jonka suoritusaika oli 0,16 millisekuntia. CRYSTALS-Dilithium-allekirjoitus oli selvästi Rainbow'ta hitaampi, 0,46 millisekuntia. Se oli kuitenkin nopeampi kuin brainpoolP256r1-käyrä, jonka suoritusaika oli 0,55 millisekuntia ja siten venyi lähes kymmenkertaiseksi NIST P-256 -käyrään verrattuna. Kaikkein hitain allekirjoittaja oli FALCON, jonka suoritusaika oli 4,89 millisekuntia ollen moninkertaisesti muita vaihtoehtoja hitaampi.

Tuloksista huomataan, että elliptisten käyrien vaihtaminen kvanttiturvalliseen vaihtoehtoon muuttaa allekirjoitukseen vaadittavaa aikaa eri tavoin riippuen siitä, onko alkuperäinen käytössä ollut käyrä NIST P-256 vai brainpoolP256r1. NIST P-256 -käyrä suoriutuu allekirjoituksesta niin nopeasti, että vaihto mihin tahansa muuhun allekirjoitustyyppiin hidastaa ohjelman toimintaa. Sen sijaan brainpoolP256r1 asettuu suoritusajaltaan kvanttiturvallisten allekirjoitusalgoritmien väliin, jolloin allekirjoittaminen jopa nopeutuu kvanttiturvalliseen vaihtoehtoon siirryttäessä, ellei valita FALCONia.

Allekirjoituksen varmennuksessa nopein vaihtoehto oli Rainbow 0,029 millisekunnin suoritusajallaan. Toiseksi nopein varmentaja oli FALCON 0,042 millisekunnin suoritusajallaan ja kolmanneksi nopein NIST P-256 -käyrä, jonka varmennus kesti keskimäärin 0,078 millisekuntia. CRYSTALS-Dilithiumin



Kuva 19. Viestin allekirjoituksen varmennukseen kulunut keskimääräinen aika eri allekirjoitusalgoritmeja käytettäessä.

varmennuksessa kesti 0,113 millisekuntia. Selvästi hitain varmentaja oli brainpoolP256r1-käyrä, jonka varmennus kesti 0,39 millisekuntia.

Varmennukseen tarvituissa suoritusajoissa ei nähdä yhtä suuren mittaluokan vaihteluita kuin allekirjoitusvaiheessa: ero nopeimman ja hitaimman varmentajan välillä oli noin 13-kertainen (Rainbow ja brainpoolP256r1), kun ero nopeimman ja hitaimman allekirjoittajan välillä oli noin 81-kertainen (NIST P-256 ja FALCON). Elliptisiin käyriin vertailtaessa huomataan, että mikä tahansa kvanttiturvallinen allekirjoitustyyppi on varmennuksessa nopeampi kuin brainpoolP256r1. NIST P-256 -käyrään verrattunakin varmennus hidastuu vain hieman, jos käyttöön valitaan CRYSTALS-Dilithium, ja muutoin se nopeutuu. Tämä viittaa siihen, että älykkään liikenteen siirtyminen kvanttiturvallisiin allekirjoitusalgoritmeihin ei tule tuottamaan ongelmia ainakaan varmennuksen nopeuden suhteen.

Allekirjoittamisen ja varmennuksen suoritusaikoja vertailemalla nähdään, että nopein allekirjoittaja ei välttämättä ole nopein varmentaja, vaan näiden välinen suhde voi vaihdella paljonkin. Esimerkiksi FALCONin allekirjoitus vaati huomattavan paljon aikaa verrattuna muihin allekirjoitustyyppeihin, mutta varmennusvaiheessa se oli toiseksi nopein vaihtoehto. Parhaan kvanttiturvallisen kompromissin allekirjoitus- ja varmennusaikojen välillä näyttää tarjoavan Rainbow, joka on kummassakin luokassa nopeimmasta päästä. CRYSTALS-Dilithium sijoittuu keskikastiin sekä allekirjoituksessa että varmennuksessa; se ei ole kummassakaan nopein eikä hitain.

BrainpoolP256r1- ja NIST P-256 -käyrien ero allekirjoittamiseen ja varmennukseen vaadittavan ajan suhteen on huomattava. Tämä johtuu erilaisten alkulukujen käyttämisestä, kuten kappaleessa 4.2.1 todettiin. Koska nykyisessä spesifikaatiossa molemmat ovat hyväksyttyjä allekirjoitustyyppejä, voitaneen päätellä, että tämän mittaluokan vaihtelut allekirjoitettujen viestien käsittelyssä eivät ole liikenteessä kohtalokkaita.

5.2. Ohjelmointiympäristön vaikutus

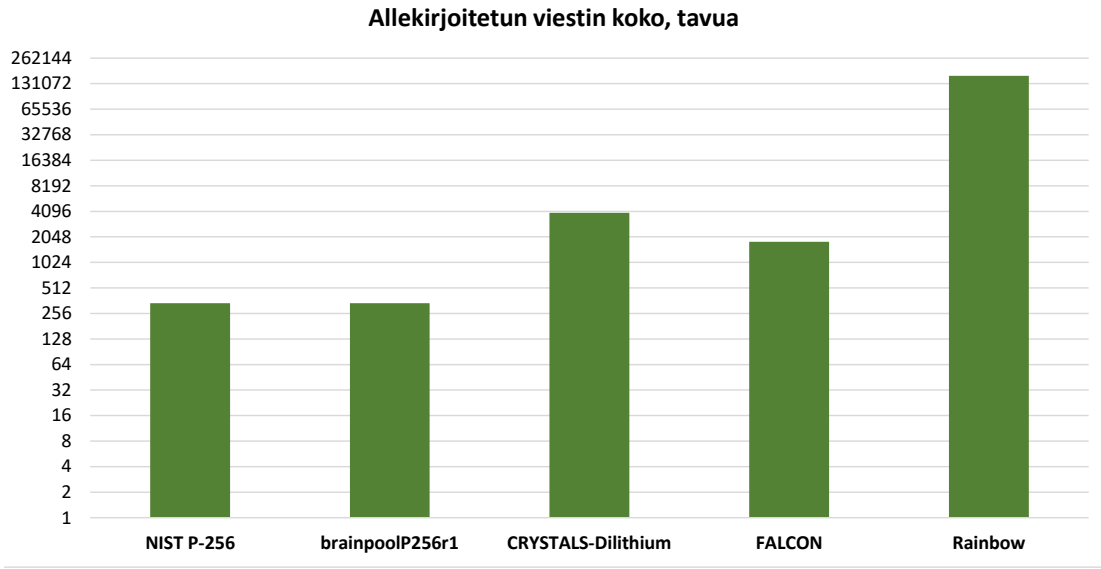
FALCON ja CRYSTALS-Dilithium tarjoavat kotisivuillaan [76, 77] viitesuoritusajat allekirjoitukselle ja varmennukselle. CRYSTALS-Dilithiumin osalta tässä tutkimuksessa saadut suoritusajat eivät eroa huomattavasti sen kotisivuilla ilmoitetuista ajoista optimoimattomalle C-versiolle. FALCONin kohdalla mittauksissa saadut suoritusajat taas ovat huomattavasti pidemmät kuin sen kotisivujen etusivulla mainitut viitesuoritusajat. FALCONia esittelevän spesifikaation [24] mukaan nämä suoritusajat on mitattu AVX2-optimointia käyttäen. Tämä selittää, miksi tässä mittauksessa ilman AVX2-optimointia saadut suoritusajat ovat pidemmät. CRYSTALS-Dilithiumin sivuilla esitellyt suoritusaikamittaukset siis tuovat FALCONia selkeämmin ilmi, mitä toteutusta annetut suoritusaikamittaukset koskevat eivätkä siten anna ylioptimistista kuvaa algoritmin nopeudesta. Rainbow'n kotisivuilla [78] selkeitä viitesuoritusaikoja allekirjoitukselle ja varmennukselle ei kirjoitushetkellä ole saatavilla, joten tässä saatuja tuloksia ei voida suhteuttaa niihin.

Toinen FALCONin toimintaan vaikuttava tekijä, jonka vaikutus ei tullut tässä tutkielmassa ilmi, on käytettävän prosessorin arkkitehtuuri. FALCONin allekirjoitusvaiheessa käytetään kompleksilukuja, jotka esitetään tietokoneelle liukulukuina. Jos prosessorista puuttuu erityinen liukulukujen laskentaan erikoistunut yksikkö (FPU, Floating Point Unit), liukuluvut pitää emuloida. FALCON on suunniteltu toimimaan tässäkin tapauksessa [24]. Tällöin suoritusaika kuitenkin saattaa kasvaa moninkertaiseksi verrattuna tilanteeseen, jossa prosessorissa olisi erityinen liukulukulaskentayksikkö [30]. Liukulukulaskentayksiköllä viitataan tässä yhteydessä IEEE-754-standardin toteuttavaan erityiseen laitteistoon [30, 85]. Myös ARM:n valmistamat liukulukulaskentayksiköt ovat tämän standardin mukaisia [86], mutta kaikissa ARM:n prosessoreissa kyseistä yksikköä ei ole automaattisesti mukana. Esimerkiksi Ettifoksen OBU:ssa [60] käytetyssä ARM Cortex-M4:ssä erillinen liukulukulaskentayksikkö ei ole pakollinen lisävaruste, joskin se voidaan lisätä siihen [87]. Tässä tutkielmassa mittauksissa käytetyssä prosessorissa liukulukulaskentayksikkö on mukana. Ilman sitä FALCONin suoritusajat olisivat mitä todennäköisimmin vielä huomattavasti pidemmät.

Hilapohjaisten vaihtoehtojen keskinäinen vertailu näyttää tuottavan erilaisia tuloksia riippuen paljolti käyttötarkoituksesta ja prosessorista. Yksinkertaista prosessoria simuloivalla Raspberry Pi:llä suoritetuissa kokeissa todettiin CRYSTALS-Dilithiumin olevan FALCONia nopeampi sekä allekirjoituksessa että varmennuksessa [16]. Samansuuntaisia tuloksia saatiin toisessakin vertailussa [88], jossa käytettiin AMD Ryzen 9 3950X -prosessoria; CRYSTALS-Dilithium suoriutui FALCONia nopeammin sekä allekirjoituksesta että varmennuksesta. Kvanttiturvallisten allekirjoitusalgoritmien TLS-käyttöä tutkineessa paperissa [28] käytettiin Intelin i5-8350U- ja Xeon-prosessoreita. Tässä tapauksessa päädyttiin samaan lopputulokseen kuin tässä tutkielmassa, eli FALCON oli CRYSTALS-Dilithiumia huomattavasti hitaampi allekirjoittaja mutta nopeampi varmentaja.

5.3. Viestin tavumääräinen koko eri allekirjoitusalgoritmeilla

Eri allekirjoitusalgoritmeilla tuotettujen allekirjoitettujen viestirakenteiden kokoja vertailtiin tavumääräisesti. Allekirjoitettava DENM-viesti ja muut teknisen spesifikaation vaatimat kentät pysyivät kaikkien allekirjoitustyyppien kohdalla saman kokoisina, ja nämä viestin osat olivat aina kooltaan yhteensä 160 tavua. Koko viestin yhteenlaskettu koko riippui tuotetusta allekirjoituksesta, joka sisällytettiin allekirjoitettuun viestiin, sekä julkisesta avaimesta, joka sisältyi viestin mukana tulevaan sertifikaattiin. Kuva 20 esittää allekirjoitettujen datarakenteiden koot tavuina sen mukaan, mitä allekirjoitusalgoritmia on käytetty.



Kuva 20. Pakettien koot tavuina. Huomaa logaritminen asteikko, eli paketin koko kaksinkertaistuu harmaiden vaakaviivojen välillä.

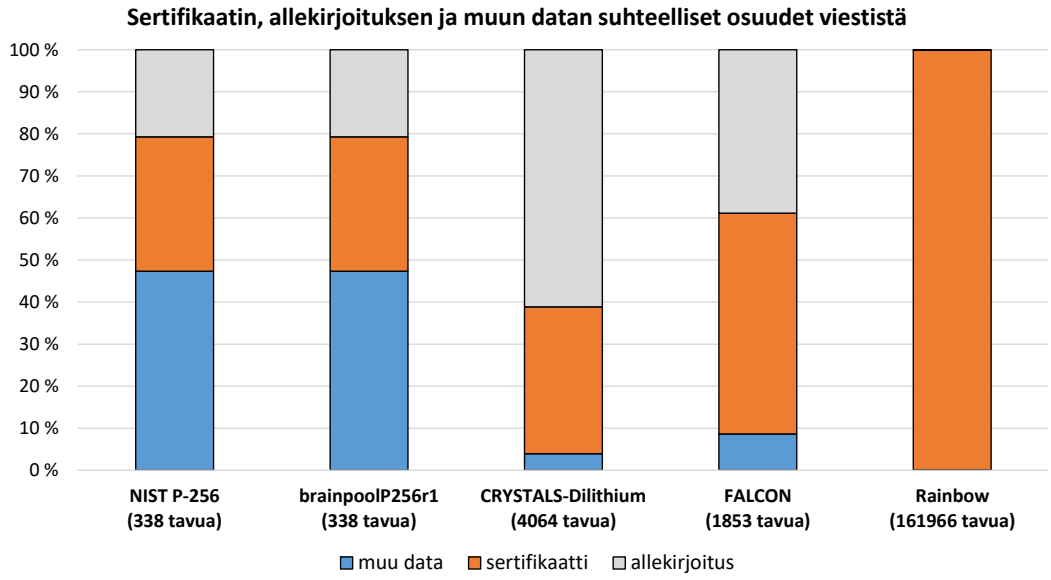
BrainpoolP256r1- ja NIST P-256 -käyrät tuottavat saman pituiset allekirjoitukset ja julkiset avaimet, jolloin myös koko paketin pituus oli niiden kesken sama, 338 tavua. Kvanttiturvallisten allekirjoitusalgoritmien tuottamat paketit olivat selkeästi suurempia. Niistä lyhyin oli FALCONin tuottama paketti, 1853 tavua, ja sen jälkeen CRYSTALS-Dilithiumilla tuotettu paketti, 4064 tavua. Rainbow'lla allekirjoitettu paketti kasvoi 161966 tavun mittaiseksi.

Allekirjoituksia ja tuotettujen pakettien kokoja tarkasteltaessa on otettava huomioon kvanttiturvallisten vaihtoehtojen C-kielisissä referenssitoteutuksissa tehdyt valinnat. Kuten kappaleessa 4.2.2 todettiin, kvanttiturvallisten allekirjoitusalgoritmien tarjolla olevat C-kieliset funktiot oli toteutettu niin, että allekirjoitusta varten varattuun kenttään kopioitiin myös allekirjoitettava viesti. Käytetyt älykkään liikenteen tekniset spesifikaatiot olettivat alkuperäisen viestin olevan eri kohdassa rakennetta, joten tästä seurasi, että kvanttiturvallisten allekirjoitusalgoritmien kohdalla allekirjoitettava osuus sisällytettiin viestirakenteeseen yhden ylimääräisen kerran. Kyseinen osuus koostui tässä tapauksessa kahdesta yhteen liitetystä SHA256-tiivistefunktion tuloksesta eli 64 tavusta. Kvanttiturvallisten vaihtoehtojen kohdalla tuotettujen pakettikokojen ja allekirjoitusten voidaan siis katsoa todellisuudessa olevan 64 tavua lyhyempiä.

Allekirjoitetun viestirakenteen koostumusta tarkasteltiin mittaamalla, kuinka suuren osan kunkin allekirjoitusalgoritmin julkisen avaimen sisältävä sertifikaatti ja allekirjoitus vievät rakenteesta. Lyhyimmät allekirjoitukset tuotettiin elliptisillä käyrillä, joista kumpikin käyrävaihtoehto tuotti 70 tavun mittaisen allekirjoituksen. Seuraavaksi lyhyin allekirjoitus oli Rainbow'lla, 130 tavua. Ottaen huomioon yllä esitetty huomio allekirjoitusten rakenteesta Rainbow tuotti itse asiassa lyhyemmän allekirjoituksen kuin elliptiset käyrät, sillä pelkkä allekirjoitus vei 66 tavua. FALCONin tuottama allekirjoitus vaati 720 tavua (ilman allekirjoitettavan viestin toistoa 656 tavua) ja CRYSTALS-Dilithiumin 2484 tavua (ilman allekirjoitettavan viestin toistoa 2420 tavua).

Julkisten avainten pituuksissa oli enemmän vaihtelua kuin allekirjoituksissa. BrainpoolP256r1- ja NIST P-256 -käyrien julkiset avaimet ovat saman pituisia, joten nämä sisältävistä sertifikaateista tuli molemmista 108 tavun mittaisia. FALCONin julkisen avaimen sisältävä sertifikaatti oli 973 tavua pitkä, ja CRYSTALS-Dilithiumin kohdalla pituus oli 1420 tavua. Rainbow'n julkisen avaimen sisältävästä sertifikaatista tuli kaikkein suurin, 161676 tavua.

Kuva 21 esittää allekirjoituksen, sertifikaatin ja muun datan vaatimat suhteelliset osuudet koko allekirjoitetusta viestistä. Osuudet ovat prosenttimääräisiä, koska viestien absoluuttinen pituus vaihtelee, kuten on esitetty kuvassa 20. Suhteelliset osuudet näyttävä kuva on mukana sen valottamiseksi, kuinka paljon tilaa allekirjoitus ja sertifikaatti vievät muuhun dataan verrattuna ja onko allekirjoitus vai sertifikaatti suurempi.



Kuva 21. Allekirjoituksen, sertifikaatin ja muun datan suhteelliset osuudet koko viestin koosta eri allekirjoitusalgoritmeilla. Koska osuudet on esitetty prosenttimääräisinä, viestin *muu data* -osuus näyttää sitä pienemmältä, mitä suurempi viesti kokonaisuudessaan on.

Molempien elliptisten käyrien, NIST P-256:n ja brainpoolP256r1:n, julkiset avaimet ja allekirjoitukset ovat saman kokoiset, joten allekirjoituksen ja sertifikaatin suhteelliset osuudet ovat samat ja kattavat yhteensä hieman yli puolet koko viestistä.

CRYSTALS-Dilithiumin allekirjoitus on sekä suhteellisesti että absoluuttisesti pisin ja kattaa viestistä suuremman osan kuin millään muulla vaihtoehdolla. Rainbow'n kohdalla huomataan, että julkisen avaimen suuren koon takia sertifikaatti valtaa lähes kaiken alan allekirjoituksen ja muun viestin ollessa siihen verrattuna lähes mitättömän kokoiset.

5.4. Avainten luominen ja tiivistäminen

CRYSTALS-Dilithium tarjoaa mahdollisuuden pienentää julkisen avaimen vaatimaa muistitilaa käyttämällä kompressointia. Tämä tarkoittaa, että koko julkista avainta ei säilötä laitteen muistiin vaan ainoastaan siemenarvo, jonka avulla julkinen avain voidaan laskea yksiselitteisesti. Tällöin julkinen avain pitäisi laskea siemenarvosta aina ennen allekirjoituksen varmentamista [22]. FALCON tarjoaa vastaavaa mahdollisuutta yksityiselle avaimelle, jonka koko voitaisiin tällä tavalla pudottaa murto-osaan nykyisestä [76]. Tässä tapauksessa avain pitäisi laskea siemenarvosta joka kerta, kun yksityistä avainta halutaan käyttää allekirjoittamiseen. Älykkään liikenteen kaltaisissa reaaliaikasovelluksissa tällainen kompressointi saattaa kääntyä itseään vastaan, koska se vaatii viestintätilanteessa tapahtuvaa ylimääräistä laskentaa verrattuna siihen, että avaimet säilöttäisiin muistiin sellaisenaan. Liikenteessä tällaiseen ylimääräiseen laskentaan ei välttämättä kaikissa tilanteissa ole aikaa.

Rainbow'sta olisi ollut tarjolla kaksi muutakin versiota, Cyclic Rainbow ja Compressed Rainbow. Ne lisättiin Standard Rainbow'n vaihtoehdoiksi NIST:n kilpailun toisella kierroksella. Rainbow'n kehittäjät tosin totesivat Compressed-version pidentävän sekä allekirjoitukseen että varmennukseen vaadittavaa aikaa [79]. Rainbown' kotisivujen [78] mukaan Cyclic-versio olisi lyhentänyt julkisen avaimen pituutta alle puoleen tässä tutkielmassa käytettyyn versioon verrattuna, joskin tällöinkin Rainbow'lla tuotetun allekirjoitetun viestin pituus olisi yhä ollut 15-kertainen verrattuna toiseksi pisimpään vaihtoehtoon CRYSTALS-Dilithiumiin. Mikäli Rainbow osoittautuisi sopivaksi vaihtoehdoksi liikennekäyttöön kaikilta muilta osin kuin julkisen avaimen pituuden osalta, voisi Cyclic Rainbow -version käyttäminen ehdottomasti parantaa sen asemia vertailussa.

CRYSTALS-Dilithium ja FALCON ovat ilmoittaneet myös avainten luomiseen kuluvan ajan kotisivuillaan [76, 77]. Nämä tiedot ovat tarpeellisia sovelluksissa, joissa avaimia täytyy luoda usein. Yksittäisen ITS-aseman ei kuitenkaan tarvitse luoda avaimia viestiessään liikenteessä muiden ITS-asemien kanssa, vaan se käyttää etukäteen hankkimiinsa sertifikaatteihin sisältyviä avaimia. Sen takia tässä työssä ei ole vertailtu avainten luomiseen kuluvaa aikaa eikä sitä pidetä arviointiperusteena allekirjoitusalgoritmien hyvyydelle.

5.5. Vertailtujen allekirjoitusalgoritmien soveltuvuus osaksi älykästä liikennettä

Tässä työssä toteutettujen mittausten perusteella tarjolla olevissa kvanttiturvallisissa allekirjoitusalgoritmeissa on eroavaisuuksia niin suoritusajan kuin julkisen avaimen ja tuotetun allekirjoituksen pituuden suhteen. Hilapohjaisista algoritmeista FALCON tuotti lyhyemmän viestin, kun viestiin sisällytettiin sekä julkisen avaimen sisältävä

sertifikaatti että tuotettu allekirjoitus. Allekirjoittamisessa CRYSTALS-Dilithium oli selvästi FALCONia nopeampi, mutta hävisi sille varmennuksen nopeudessa. Se, kumpaa näistä vaihtoehdoista lopulta pidetään parempana, riippuu siitä, halutaanko painottaa tuotettujen viestien pituutta, allekirjoittamiseen kuluvaa aikaa vai varmennukseen kuluvaa aikaa.

Yksittäisen ITS-aseman voitaisiin ajatella tekevän enemmän allekirjoitusten varmennusta kuin allekirjoittamista, koska se voi vastaanottaa viestejä kaikilta ympärillään liikkuvilta ITS-asemilta. Toisaalta ETSI:n määrittelemät spesifikaatiot sisältävät myös laajemman julkisen avaimen infrastruktuurin [49]. Jos koko sertifikaattiketjun halutaan käyttävän kvanttiturvallista allekirjoitusta, korostuu myös allekirjoitukseen kuluvan ajan merkitys, sillä väliasteikon sertifikaattiauktoriteetit allekirjoittavat paljon loppukäyttäjien sertifikaatteja. Kuten TLS-protokollaan keskittyneessä tutkimuksessa todetaan, yksi mahdollinen kompromissi on toteuttaa sertifikaattiketjun eri osat eri allekirjoitusalgoritmeilla [28].

Ottaen huomioon älykkäiden autojen vaihtelevat arkkitehtuurit on eduksi, jos algoritmi on siirrettävissä useille erilaisille alustoille. Aiemmin tehtyihin tutkimuksiin viitaten [29, 30, 85] vaikuttaa siltä, että FALCONin suoritusnopeus on CRYSTALS-Dilithiumia voimakkaammin riippuvaista käytettävän prosessorin arkkitehtuurista. Tämän perusteella CRYSTALS-Dilithium voi olla varmempi valinta liikennestandardeihin, joiden tulee mukautua useanlaisten yksityisten valmistajien laitteiden vaatimuksiin. Tällöin joudutaan ehkä tyytymään hieman hitaampaan viestien varmennukseen ja suurempiin viestikokoihin. Todelliset suoritusajat riippuvat luonnollisesti liikennevälineeseen integroidun prosessorin ominaisuuksista. Esimerkiksi Commsignian ja Ecnoliten OBU:t toimivat huomattavasti tässä testissä käytettyä prosessoria (1.9 GHz) matalammilla kellotaajuuksilla (800 MHz), joskin niiden arkkitehtuuri on muutenkin erilainen [54, 58].

Usean muuttujan polynomeihin pohjautuva Rainbow suoriutui allekirjoituksesta ja varmennuksesta hilapohjaisia kilpakumppaneitaan nopeammin. Reaaliaikaviestinnän asettamia aikavaatimuksia ajatellen Rainbow olisikin ihanteellinen valinta liikennekäyttöön. Myös Rainbow'n tuottamat allekirjoitukset ovat huomattavasti lyhyempiä kuin hilapohjaisten vaihtoehtojen. Sen vahva puoli ovatkin erittäin kompaktit allekirjoitukset, joiden pituus on lähellä elliptisillä käyrillä tuotettuja allekirjoituksia. Pelkästä lyhyestä allekirjoituksesta ei kuitenkaan ole tämän tutkielman kontekstissa juuri hyötyä, koska allekirjoituksen varmentamiseen vaaditaan julkisen avaimen sisältävä sertifikaatti. Jos lyhyen allekirjoituksen vastapainoksi sertifikaatti kasvaa todella suureksi, saavutettu hyöty jää vähäiseksi. Nopeista suorituksista ja lyhyistä allekirjoituksista huolimatta Rainbow'n julkisen avaimen suuri koko ja vastikään paljastuneet turvallisuuspuutteet [26] tekevät sen käytön älykkäässä liikenteessä ainakin nykyisessä muodossaan kyseenalaiseksi. Tämä tutkielma osoitti kuitenkin sen, että hyvien rajapintojen ollessa saatavilla usean muuttujan polynomeihin perustuvan allekirjoitusalgoritmin yhdistäminen älykkään liikenteen teknisiin spesifikaatioihin on ohjelmoinnillisesti täysin mahdollista.

Kaikki kvanttiturvalliset vaihtoehdot toteuttivat saman tyyppisen rajapinnan, mikä teki niiden integroimisesta teknisiin spesifikaatioihin melko vaivatonta. Siirtymävaiheessa älykkään liikenteen teknisiin spesifikaatioihin voitaisiinkin ehkä hyvissä ajoin sisällyttää kaksi erilaista rakennetta allekirjoitukselle ja julkiselle avaimelle: yksi elliptisiä käyriä ja toinen mitä tahansa kvanttiturvallista

allekirjoitusalgoritmia varten. Varsinainen algoritmin valinta voitaisiin vahvistaa myöhemmin ilman, että spesifikaatiota tarvitsisi enää muuttaa. On kuitenkin pidettävä mielessä, että tässä työssä hyödynnetyt C-kieliset referenssitoteutukset on tehty NIST:n kilpailua varten algoritmin esittelemiseksi ja toiminnan todistamiseksi. Siksi niiden tarjoamiin toiminnallisuuksiin ei kannata takertua liiaksi. Standardoitavaksi valittavasta allekirjoitusalgoritmista lienee ajan myötä odotettavissa useita erilaisiin käyttötarkoituksiin ja eri alustoille optimoituja versioita.

Huomionarvoista on myös se, että NIST:n kilpailuun lähetettyjen allekirjoitusalgoritmien C-kieliset referenssitoteutukset hyödyntävät ulkoisia kirjastoja, kuten OpenSSL:ää. Tämän takia ne eivät sellaisenaan olisi muutenkaan siirrettävissä kaikkiin ympäristöihin, koska OpenSSL-kirjaston siirtäminen joillekin alustoille saattaa olla ongelmallista [16]. Muutoksia toteutuksiin jouduttaisiin siis joka tapauksessa tekemään tiettyjä käyttötarkoituksia varten.

Tilanne, jossa viesti vastaanotetaan, vaikuttaa luonnollisesti siihen, millaisia allekirjoitusalgoritmeja voidaan harkita käytettäväksi. Jos kyse on hyvissä ajoin tulevasta tiedotteesta koskien esimerkiksi kilometrien päässä sijaitsevaa liikenneruuhkaa, viestin luominen ja varmentaminen ehditään tehdä hitaammallakin allekirjoitusalgoritmilla. Jos taas kyseessä on akuutti yhteentörmäyksen riski, viive laskennassa on kriittisempää. Toisaalta trendinä on varustella älyautoja yhä moninaisemmilla kameroilla ja sensoreilla [35], joten toivottavaa on, että kolarien välttäminen ei jää kiinni viestien käsittelynopeudesta. Ainakin lähivuosina lopullinen vastuu jäänee joka tapauksessa ajoneuvon kuljettajalle.

5.6. Jatkotutkimusmahdollisuuksia

NIST:n on määrä valita standardoitava kvanttiturvallinen allekirjoitusalgoritmi lähivuosina. Viimeistään tämän jälkeen lienee syytä testata sitä markkinoilla olevilla OBU-laitteilla ja mahdollisesti kehittää siitä juuri tätä tarkoitusta varten optimoitu versio. Nykyiset OBU-laitteet sisältävät usein erityisen tietoturvaoperaatioihin tarkoitetun laskentayksikön. Standardoitavan allekirjoitusalgoritmin toteuttaminen tällaisella yksiköllä tullee kyseeseen valinnan varmistuessa.

ETSI:n teknisissä spesifikaatioissa määritellyt kryptografiset operaatiot eivät rajoitu digitaalisiin allekirjoituksiin, vaan tiettyjen viestityyppien yhteydessä käytetään myös viestien symmetristä salausta ja siihen käytettävien avainten johtamista elliptisten käyrien avulla [47]. Olisi mielenkiintoista tutkia, kuinka tämä toimenpide onnistuu käyttäen elliptisten käyrien tilalla NIST:n kilpailuun lähetettyjä kvanttiturvallisia KEM-algoritmeja. Tässä työssä toteutettua ohjelmoinnillista osuutta voitaisiin myös laajentaa toteuttamalla kokonainen sertifikaattiketju, joka sisältää myös korkeimman sertifikaattiauktoriteetin ja väliasteen auktoriteettien allekirjoitukset. Tällöin voitaisiin arvioida, mikä allekirjoitusalgoritmi tai algoritmien yhdistelmä parhaiten sopisi julkisen avaimen infrastruktuurin toteuttamiseen.

6. YHTEENVETO

Tämän diplomityön tavoitteena oli arvioida kvanttiturvallisten allekirjoitusalgoritmien soveltuvuutta älykkään liikenteen viestinnässä käytettäväksi. Työn lähtökohtana käytettiin ETSI:n julkaisemia teknisiä spesifikaatioita, jotka määrittelevät älykkäässä liikenteessä tapahtuvan ajoneuvojen välisen viestinnän muodon. Nämä spesifikaatiot määrittelevät viestien digitaalisen allekirjoittamisen, joka on tärkeää viestien aitouden ja luotettavuuden varmistamiseksi. Nykyisessä spesifikaatiossa viestit voidaan allekirjoittaa kolmella eri elliptisellä käyrällä. Elliptisten käyrien ongelmana on kuitenkin se, että niihin perustuvia allekirjoituksia voitaisiin väärentää kvanttitietokoneella. Tätä tarkoitusta varten tarpeeksi suurta kvanttitietokonetta ei vielä ole, mutta niitä kehitetään koko ajan.

Tässä työssä älykkään liikenteen teknisiin spesifikaatioihin liitettiin kolme kvanttiturvallisena pidettyä allekirjoitusalgoritmia: CRYSTALS-Dilithium, FALCON ja Rainbow. Nämä algoritmit ovat edenneet finaaliin asti NIST:n kvanttiturvallisen kryptografian standardointikilpailussa. Kaksi niistä perustuu hilaongelmiin ja yksi usean muuttujan polynomeihin. Kaikista on saatavilla valmis C-kielinen toteutus. Elliptisistä käyristä toteutettiin kaksi teknisen spesifikaation sallimaa vaihtoehtoa OpenSSL-kirjaston avulla.

Allekirjoitusalgoritmien toiminnan arvioimiseksi luotiin ohjelma, joka vuorotellen loi epätavallisista tieolosuhteista varoittavia DENM-viestejä, allekirjoitti niitä ja varmensi niiden allekirjoituksia. Ohjelmasta luotiin viisi eri versiota, joista kukin käytti allekirjoitukseen eri algoritmia: joko elliptisiä käyriä tai jotain kolmesta kvanttiturvallisesta allekirjoitusalgoritmista. Jokaisen vaihtoehdon kohdalla viestien allekirjoittamiseen ja allekirjoituksen varmennukseen kulunut aika mitattiin tietokoneen kellolla. Lisäksi mitattiin, kuinka pitkä DENM-viestin sisältävästä allekirjoitetusta viestipaketista tuli.

Viestien pituuksien vertailu paljasti, että kaikki kvanttiturvalliset allekirjoitusalgoritmiehdokkaat tuottavat pidempiä allekirjoitettuja viestejä kuin nykyisin käytössä olevat elliptiset käyrät. Kvanttiturvallisten vaihtoehtojen välillä oli kuitenkin moninkertaisia eroja. Suoritusaikavertailusta kävi ilmi, että toinen testatuista elliptisistä käyristä oli huomattavasti toista nopeampi. Sen takia kvanttiturvalliseen vaihtoehtoon siirtyminen vaikuttaa suoritusaikoihin eri tavoin riippuen siitä, kumpi elliptinen käyrä alun perin oli käytössä. Kvanttiturvalliset allekirjoitusalgoritmit osoittautuivat kuitenkin varsin kilpailukykyisiksi erityisesti viestien varmennuksessa. Allekirjoitusten käsittelyyn käytettävän prosessorin arkkitehtuuri voi vaikuttaa suoritusaikoihin huomattavasti.

Tulosten perusteella voidaan sanoa, että jompikumpi hilapohjaisista vaihtoehdoista, CRYSTALS-Dilithium tai FALCON, voisi olla potentiaalinen vaihtoehto älykkään liikenteen käyttöön. Usean muuttujan polynomeihin perustuva Rainbow tuotti erityisen suuria allekirjoitettuja viestejä. Lisäksi siitä on vastikään löytynyt turvallisuuspuutteita, minkä vuoksi se ei liene yhtä luotettava vaihtoehto tähän tarkoitukseen. Mielenkiintoisia jatkotutkimusmahdollisuuksia ovat kokonaisen sertifikaattiketjun ja muiden teknisissä spesifikaatioissa määriteltyjen kryptografisten operaatioiden toteuttaminen kvanttiturvallisesti.

7. VIITTEET

[1] Ferguson N., Schneier B. & Kohno T. (2010) Cryptography Engineering: Design Principles and Practical Applications. Wiley, Chichester.

[2] Katz J. & Lindell Y. (2007) Introduction to Modern Cryptography. CRC Press, Boca Raton, London, New York, Washington, D. C.

[3] Barker E. (2020) NIST Special Publication 800-57 Part 1 Revision 5. Recommendation for Key Management: Part 1 – General. DOI: https://doi.org/10.6028/NIST.SP.800-57pt1r5.

[4] Aumasson J.P. (2018) Serious Cryptography: A Practical Introduction to Modern Cryptography. No Starch Press, San Fransisco.

[5] Barker E. & Dang Q. (2015) NIST Special Publication 800-57 Part 3 Revision 1. Recommendation for Key Management: Part 3: Application-Specific Key Management Guidance. DOI: http://dx.doi.org/10.6028/NIST.SP.800-57pt3r1.

[6] Adalier M. & Teknik A. (2015) Efficient and Secure Elliptic Curve Cryptography Implementation of Curve P-256. NIST Workshop on Elliptic Curve Cryptography Standards, June 2015.

[7] Lochter M. & Merkle J. (2010), Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639. URL: https://datatracker.ietf.org/doc/html/rfc5639. DOI: 10.17487/RFC5639. Vierailtu 29.7.2021.

[8] Vidakovic D., Parezanovic D., Nikolic O. & Kaljevic J. (2013) RSA Signature: Behind the Scenes. Advanced Computing: An International Journal 4(2), ss. 27–40. DOI: http://dx.doi.org/10.5121/acij.2013.4203.

[9] Lenstra A.K. (2000) Integer Factoring. Designs, Codes, and Cryptography 19(2), ss. 101–128.

[10] Meglicki Z. (2019) Quantum Computing Without Magic - Devices. The MIT Press.

[11] Bernstein D.J., Buchmann J. & Dahmen E. (toim.) (2009) Post-Quantum Cryptography. Springer-Verlag Berlin Heidelberg.

[12] Shor P.W. (1997) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing 26(5), ss. 1484–1509. DOI: http://dx.doi.org/10.1137/S0097539795293172.

[13] Roetteler M., Naehrig M., Svore K.M. & Lauter K. (2017) Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. Teoksessa: T. Takagi & T. Peyrin (toim.) Advances in Cryptology – ASIACRYPT 2017, Springer International Publishing, Cham, ss. 241–270.

[14] Arute F., Arya K., Babbush R., Bacon D., Bardin J.C., Barends R., Biswas R., Boixo S., Brandao F.G.S.L., Buell D.A., Burkett B., Chen Y., Chen Z., Chiaro B., Collins R., Courtney W., Dunsworth A., Farhi E., Foxen B., Fowler A., Gidney C., Giustina M., Graff R., Guerin K., Habegger S., Harrigan M.P., Hartmann M.J., Ho A., Hoffmann M., Huang T., Humble T.S., Isakov S.V., Jeffrey E., Jiang Z., Kafri D., Kechedzhi K., Kelly J., Klimov P., Knysh S., Korotkov A., Kostritsa F., Landhuis D., Lindmark M., Lucero E., Lyakh D., Mandrà S., McClean J.R., McEwen M., Megrant A., Mi X., Michielsen K., Mohseni M., Mutus J., Naaman O., Neeley M., Neill C., Niu M.Y., Ostby E., Petukhov A., Platt J.C., Quintana C., Rieffel E.G., Roushan P., Rubin N.C., Sank D., Satzinger K.J., Smelyanskiy V., Sung K.J., Trevithick M.D., Vainsencher A., Villalonga B., White T., Yao Z.J., Yeh P., Zalcman A., Neven H. & Martinis J.M. (2019) Quantum supremacy using a programmable superconducting processor. Nature 574, ss. 505–510.

[15] Gidney C. & Ekerå M. (2021) How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum 5, s. 433. DOI: https://doi.org/10.22331/q-2021-04-15-433.

[16] Malina L., Ricci S., Dzurenda P., Smekal D., Hajny J. & Gerlich T. (2020) Towards Practical Deployment of Post-quantum Cryptography on Constrained Platforms and Hardware-Accelerated Platforms. Teoksessa: Innovative Security Solutions for Information Technology and Communications - 12th International Conference, SecITC 2019, Bucharest, Romania, November 14–15, 2019, Revised Selected Papers, Springer International Publishing, Cham, ss. 109–124.

[17] Alagic G., Alperin-Sheriff J., Apon D., Cooper D., Dang Q., Miller C., Moody D., Peralta R., Perlner R., Robinson A., Smith-Tone D. & Liu Y.K. (2019) Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD.

[18] Workshops and Timeline. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/workshops-and-timeline. Vierailtu 28.7.2021.

[19] Security (Evaluation Criteria). URL: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria). Vierailtu 28.7.2021.

[20] PQC Standardization Process: Third Round Candidate Announcement. URL: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement. Vierailtu 28.7.2021.

[21] Round 3 Submissions. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions. Vierailtu 28.7.2021.

[22] Bai S., Ducas L., Kiltz E., Lepoint T., Lyubashevsky V., Schwabe P., Seiler G. & Stehlé D. (2021), CRYSTALS-Dilithium Algorithm

Specifications and Supporting Documentation (Version 3.1). URL: https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf. Vierailtu 30.7.2021.

[23] Gentry C., Peikert C. & Vaikuntanathan V. (2008) How to use a short basis: Trapdoors for hard lattices and new cryptographic constructions. Electronic Colloquium on Computational Complexity (ECCC) 14.

[24] Fouque P.A., Hoffstein J., Kirchner P., Lyubashevsky V., Pornin T., Prest T., Ricosset T., Seiler G., Whyte W. & Zhang Z. (2020), FALCON: Fast-Fourier Lattice-based Compact Signatures over NTRU - Specification v1.2 — 01/10/2020. URL: https://falcon-sign.info/falcon.pdf. Vierailtu 19.7.2021.

[25] Petzoldt A. (2020) Efficient Key Generation for Rainbow. Teoksessa: J. Ding & J.P. Tillich (toim.) Post-Quantum Cryptography, 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Springer International Publishing, Cham, ss. 92–107. DOI: http://dx.doi.org/10.1007/978-3-030-44223-1_6.

[26] Beullens W. (2021) Improved Cryptanalysis of UOV and Rainbow. Teoksessa: A. Canteaut & F.X. Standaert (toim.) Advances in Cryptology – EUROCRYPT 2021, Springer International Publishing, Cham, ss. 348–373.

[27] Status Update on the 3rd Round. URL: https://csrc.nist.gov/Presentations/2021/status-update-on-the-3rd-round. Vierailtu 10.8.2021.

[28] Sikeridis D., Kampanakis P. & Devetsikiotis M. (2020) Post-Quantum Authentication in TLS 1.3: A Performance Study. International Association for Cryptologic Research (IACR) Cryptology ePrint Archive 2020.

[29] Marzougui S. & Krämer J. (2019) Post-Quantum Cryptography in Embedded Systems. Teoksessa: ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and Security, Association for Computing Machinery, ss. 1–7.

[30] Paul S. & Scheible P. (2020) Towards Post-Quantum Security for Cyber-Physical Systems: Integrating PQC into Industrial M2M Communication. Teoksessa: L. Chen, N. Li, K. Liang & S. Schneider (toim.) Computer Security – ESORICS 2020, Springer International Publishing, Cham, ss. 295–316.

[31] Malina L., Dzurenda P., Ricci S., Hajny J., Srivastava G., Matulevičius R., Affia A.A., Laurent M., Sultan N. & Tang Q. (2021) Post-Quantum Era Privacy Protection for Intelligent Infrastructures. IEEE Access 9, ss. 36038–36077.

[32] C-ITS Secure Communications. URL: https://www.itsstandards.eu/highlighted-projects/c-its-secure-communications/. Vierailtu 28.7.2021.

[33] European Telecommunications Standards Institute (2019), ETSI EN 302 637-2 V1.4.1 (2019-04). URL: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf.

[34] European Telecommunications Standards Institute (2014), ETSI EN 302 637-3 V1.2.2 (2014-11). URL: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.02_60/en_30263703v010202p.pdf.

[35] Hussain R. & Zeadally S. (2019) Autonomous cars: Research results, issues, and future challenges. IEEE Communications Surveys Tutorials 21(2), ss. 1275–1313.

[36] Anderson J.M., Nidhi K., Stanley K.D., Oluwatola O.A., Samaras C. & Sorensen P. (2014) Autonomous Vehicle Technology: A Guide for Policymakers. RAND Corporation, Santa Monica.

[37] European Telecommunications Standards Institute (2014), ETSI EN 302 895 V1.1.1 (2014-09). URL: https://www.etsi.org/deliver/etsi_en/302800_302899/302895/01.01.01_60/en_302895v010101p.pdf.

[38] Kousaridas A., Schimpe A., Euler S., Vilajosana X., Fallgren M., Landi G., Moscatelli F., Barmpounakis S., Vázquez-Gallego F., Sedar R., Silva R., Dizambourg L., Wendt S., Muehleisen M., Eckert K., Härri J. & Alonso-Zarate J. (2020) 5G Cross-Border Operation for Connected and Automated Mobility: Challenges and Solutions. Future Internet 12(1). DOI: https://doi.org/10.3390/fi12010005.

[39] Sedar R., Vázquez-Gallego F., Casellas R., Vilalta R., Silva R., Dizambourg L., Barciela A., Vilajosana X., Datta S.K., Härri J. & Alonso-Zarate J. (2021) Standards-Compliant Multi-Protocol On-Board Unit for the Evaluation of Connected and Automated Mobility Services in Multi-Vendor Environments. Sensors 21(6).

[40] European Telecommunications Standards Institute (2009), ETSI TR 102 638 V1.1.1 (2009-06). URL: https://www.etsi.org/deliver/etsi_tr/102600_102699/102638/01.01.01_60/tr_102638v010101p.pdf.

[41] (2020), Cooperative intelligent transport systems (C-ITS) Guidelines on the usage of standards. URL: https://www.itsstandards.eu/app/uploads/sites/14/2020/10/C-ITS-Brochure-2020-FINAL.pdf. Vierailtu 27.7.2021.

[42] About us. URL: https://www.cencenelec.eu/aboutus/Pages/default.aspx. Vierailtu 28.7.2021.

[43] About ETSI. URL: https://www.etsi.org/about. Vierailtu 28.7.2021.

[44] Standards, Specifications and Reports. URL: https://www.etsi.org/standards/types-of-standards. Vierailtu 28.7.2021.

[45] Intelligent Transport Systems. URL: https://www.itsstandards.eu/. Vierailtu 28.7.2021.

[46] Technical Bodies. URL: https://standards.cen.eu/dyn/www/f?p=CENWEB:6:::NO:::. Vierailtu 28.7.2021.

[47] European Telecommunications Standards Institute (2020), ETSI TS 103 097 V1.4.1 (2020-10). URL: https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/01.04.01_60/ts_103097v010401p.pdf.

[48] European Telecommunications Standards Institute (2021), ETSI TS 102 941 V1.4.1 (2021-01). URL: https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/01.04.01_60/ts_102941v010401p.pdf.

[49] European Telecommunications Standards Institute (2018), ETSI TS 102 940 V1.3.1 (2018-04). URL: https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.03.01_60/ts_102940v010301p.pdf.

[50] European Telecommunications Standards Institute (2018), ETSI TS 102 894-2 V1.3.1 (2018-08). URL: https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.03.01_60/ts_10289402v010301p.pdf.

[51] IEEE Vehicular Technology Society (2016), IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages. DOI: https://doi.org/10.1109/IEEESTD.2016.7426684.

[52] IEEE Vehicular Technology Society (2017), IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages: Amendment 1. DOI: https://doi.org/10.1109/IEEESTD.2017.8065169.

[53] IEEE Vehicular Technology Society (2019), IEEE Standard for Wireless Access in Vehicular Environments — Security Services for Applications and Management Messages: Amendment 2: PDU Functional Types and Encryption Key Management. DOI: https://doi.org/10.1109/IEEESTD.2019.8734860.

[54] Powerful V2X Onboard Unit. URL: https://www.commsignia.com/products/obu/. Vierailtu 18.8.2021.

[55] MK5 OBU. URL: http://www.cohdawireless.com/solutions/hardware/mk5-obu/. Vierailtu 18.8.2021.

[56] Communication modules: V2X. URL: https://www.ficosa.com/products/advanced-communication-systems/connected-vehicle-v2x/. Vierailtu 18.8.2021.

[57] RoadLINK® SAF5400 Single Chip Modem for V2X. URL: https://www.nxp.com/products/wireless/dsrc-safety-modem/roadlink-saf5400-single-chip-modem-for-v2x:SAF5400. Vierailtu 18.8.2021.

[58] MobiWAVE On-Board-Unit (OBU) Datasheet. URL: http://www.econolite.com/wp-content/uploads/sites/9/2019/01/OBU-Econolite.pdf. Vierailtu 20.8.2021.

[59] OBU-301E Specification. URL: https://www.unex.com.tw/sheet/OBU-301E.pdf. Vierailtu 20.8.2021.

[60] Ettifos On-Board Unit (OBU). URL: https://www.ettifos.com/platforms. Vierailtu 20.8.2021.

[61] Arm Architecture: A Foundation for Computing Everywhere. URL: https://www.arm.com/why-arm/architecture/cpu. Vierailtu 19.8.2021.

[62] Elahi A. (2017) Computer Systems: Digital Design, Fundamentals of Computer Architecture and Assembly Language. Springer International Publishing AG, Cham.

[63] Vanetza in a nutshell. URL: https://www.vanetza.org/. Vierailtu 18.8.2021.

[64] Building Vanetza for Cohda MK5 using Cohda SDK. URL: https://www.vanetza.org/recipes/cohda-sdk-build/. Vierailtu 18.8.2021.

[65] Fernandes B., Rufino J., Alam M. & Ferreira J. (2018) Implementation and Analysis of IEEE and ETSI Security Standards for Vehicular Communications. Mobile Networks and Applications 23(3), ss. 469–478.

[66] Hamida E.B., Noura H.N. & Znaidi W. (2015) Security of Cooperative Intelligent Transport Systems: Standards, Threats Analysis and Cryptographic Countermeasures. Electronics 4, ss. 380–423.

[67] Lonc B., Haidar F. & Filatov D. (2020) Cooperative ITS Security Standards: Implementation, assessment and next challenges. Teoksessa: Virtual ITS European Congress, Nov 2020, Lisbonne (virtual), Portugal.

[68] Technical Committee (TC) CYBER (Cybersecurity) Activity Report 2020. URL: https://www.etsi.org/committee-activity/activity-report-cyber. Vierailtu 28.7.2021.

[69] Abstract Syntax Notation One (ASN.1). URL: https://portal.etsi.org/Services/Centre-for-Testing-Interoperability/ETSI-Approach/Specification-Languages/ASN1. Vierailtu 27.7.2021.

[70] asn1c - ASN.1 Compiler. URL: http://manpages.ubuntu.com/manpages/trusty/man1/asn1c.1.html. Vierailtu 27.7.2021.

[71] Costello C., Longa P. & Naehrig M. (2015) A brief discussion on selecting new elliptic curves. NIST Workshop on Elliptic Curve Cryptography Standards, June 2015 .

[72] OpenSSL. URL: https://www.openssl.org/. Vierailtu 27.7.2021.

[73] Falcon source files (reference implementation). URL: https://falcon-sign.info/impl/README.txt.html. Vierailtu 30.7.2021.

[74] Dilithium. URL: https://github.com/pq-crystals/dilithium. Vierailtu 30.7.2021.

[75] GitHub - fast-crypto-lab/rainbow-submission-round2: Rainbow signature system for Round THREE submission. URL: https://github.com/fast-crypto-lab/rainbow-submission-round2. Vierailtu 30.7.2021.

[76] FALCON - Fast-Fourier Lattice-based Compact Signatures over NTRU. URL: https://falcon-sign.info/. Vierailtu 19.7.2021.

[77] Dilithium. URL: https://pq-crystals.org/dilithium/index.shtml. Vierailtu 19.7.2021.

[78] Rainbow Signature. URL: https://www.pqcrainbow.org/. Vierailtu 19.7.2021.

[79] Rainbow round 2 presentation. URL: https://csrc.nist.gov/CSRC/media/Presentations/rainbow-round-2-presentation/images-media/rainbow-ding.pdf. Vierailtu 3.9.2021.

[80] Overview: Intrinsics for Intel® Advanced Vector Extensions 2 (Intel® AVX2) Instructions. URL: https://software.intel.com/content/www/us/en/develop/documentation/cpp-compiler-developer-guide-and-reference/top/compiler-reference/intrinsics/intrinsics-for-intel-advanced-vector-extensions-2/overview-intrinsics-for-intel-advanced-vector-extensions-2-intel-avx2-instructions.html. Vierailtu 28.7.2021.

[81] ECDSA_SIG_new. URL: https://www.openssl.org/docs/man1.1.1/man3/ECDSA_SIG_get0_r.html. Vierailtu 6.8.2021.

[82] SHA256_Init. URL: https://www.openssl.org/docs/man1.1.1/man3/SHA1.html. Vierailtu 2.8.2021.

[83] 3.11 - Options that control Optimization. URL: https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html. Vierailtu 11.8.2021.

[84] bn. URL: https://www.openssl.org/docs/man1.0.2/man3/bn.html. Vierailtu 3.9.2021.

[85] Pornin T. (2019), New Efficient, Constant-Time Implementations of Falcon. Cryptology ePrint Archive, Report 2019/893. URL: https://ia.cr/2019/893.

[86] ARM Cortex-R Series Programmer's Guide. URL: https://developer.arm.com/documentation/den0042/a/Floating-Point/Floating-point-basics-and-the-IEEE-754-standard. Vierailtu 23.8.2021.

[87] Cortex-M4 Technical Reference Manual. URL: https://developer.arm.com/documentation/ddi0439/b/Floating-Point-Unit. Vierailtu 23.8.2021.

[88] Raavi M., Wuthier S., Chandramouli P., Balytskyi Y., Zhou X. & Chang S. (2021) Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms. Teoksessa: Applied Cryptography and Network Security 19th International Conference, ACNS 2021, Kamakura, Japan, June 21–24, 2021, Proceedings, Part II, Springer International Publishing, Cham, ss. 424–447.