Tampereen yliopisto

Eemeli Lottonen

Kryptografia ja CRYSTALS Kyber

Informaatioteknologian ja viestinnän tiedekunta
Pro gradu -tutkielma
Matematiikka
Huhtikuu 2022

Eemeli Lottonen: Kryptografia ja CRYSTALS Kyber
Pro gradu -tutkielma
Tampereen yliopisto
Matematiikan ja tilastotieteen tutkinto-ohjelma
Huhtikuu 2022

TIIVISTELMÄ

Tässä tutkielmassa tarkastellaan CRYSTALS Kyber avainten kapselointimekanismia. Yhdysvaltalainen National Institute of Standards and Technology (NIST) on järjestänyt post quantum cryptography -hankkeen (PQC-hanke), missä pyritään löytämään kvanttilaskentaa kestäviä salausmenetelmiä. Hanke julkaistiin konferenssissa PQCrypto vuonna 2016 ja siihen on lähetetty monia algoritmiehdokkaita, joista yksi on CRYSTALS Kyber. Hanke edistyy kierroksittain ja jokaisella kierroksella huonoimpia ehdokkaita karsitaan pois. CRYSTALS Kyber on päässyt hankkeen kolmannelle kierrokselle.

Ensin tutustutaan todennäköisyyslaskentaan ja erityisesti satunnaismuuttujiin sekä keskitettyyn binomijakaumaan. Tämän jälkeen esitellään algebraa ja tekijärenkaat. Sitten siirrytään kryptografian peruskäsitteisiin symmetriseen ja epäsymmetriseen salaukseen, hajautusfunktioihin sekä avaintenkapselointiin. Perusasioiden esittelyn jälkeen tutkitaan häirityn oppimisen ongelmaa ja sen eri muotoja. Kyberin turvallisuus perustuu häirityn oppimisen ongelman moduliversioon. Ongelma esitellään, mutta turvallisuustodistuksia ei. Tämän jälkeen tarkastellaan itse Kyberiä. Se koostuu epäsymmetrisestä salausskeemasta sekä avainten kapselointiosasta. Epäsymmetrinen salausskeema esitellään ensin, jonka päälle rakennetaan avainten kapselointiskeema. Molemmat sisältävät kolme algoritmia, avainten generoinnin, viestin salauksen sekä viestin avaamisen. Tarkastellaan näitä algoritmeja ja todistetaan niiden oikeellisuus. Tarkastellaan sitten Kyberin parametrivalintoja ja vertaillaan kuinka ja miksi ne ovat muuttuneet ensimmäisestä esityskerrasta vuonna 2017. Lopuksi suoritetaan Kyberin suorituskykymittaus käyttäen kolmannen kierroksen ehdokkaan referenssitoteutusta apuna.

Avainsanat: CRYSTALS Kyber, salausalgoritmit, avaintenkapselointimekanismi, kryptografia, kvanttitietokoneet
Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

Sisällys

1 Johdanto

Viime vuosina kvanttitietokoneihin liittyvää tutkimusta on tehty runsaasti. Kvanttitietokoneet ovat tietokoneita, jotka hyödyntävät kvanttimekaanisia ilmiöitä ratkaistakseen ongelmia. Kvanttitietokoneet ovat usein tavanomaisia tietokoneita tehokkaampia hankalien ongelmien ratkaisemisessa. Ne eroavat tavanomaisista tietokoneista siten, että niiden pienimpänä yksikkönä on kubitit bittien sijaan. Tavanomaisten bittien tila on aina 1 tai 0, kun taas kubittien tila on superpositio näistä tiloista. Kubitin tilaa kuvataan kompleksilukuparina (a, b), missä $|a|^2 + |b|^2 = 1$. Kun kubitin tilaa yritetään mitata saadaan 0 todennäköisyydellä $|a|^2$ ja 1 todennäköisyydellä $|b|^2$ [AB09]. Kubitti voi siis olla kahdessa tilassa samaan aikaan, sillä mittaus tuottaa satunnaisen tuloksen. Tämä tarkoittaa, että kahdella kubitilla voidaan kuvata bittijonot 00, 01, 10 ja 11 samaan aikaan. Kvanttitietokone on yhdessä tilassa, mutta mittaus voi tuottaa minkä tahansa näistä tiloista. Tavanomainen tietokone tarvitsisi 8 bittiä kuvaamaan kaikki bittijonot. Jo pienestä kubittimäärästäkin huomataan kubittien hyöty, tietoa saadaan prosessoitua yhtäaikaa, vaikka kvanttikone on vain yhdessä tilassa.

Ensimmäinen 2-kubitin kvanttititetokone rakennettiin Oxfordin yliopistossa vuonna 1998. Tutkijat olivat Jonathan A. Jones ja Michele Mosca ja he käyttivät tietokonetta Deutschin ongelman ratkaisemiseen [CGK98]. Deuthchin ongelmassa käytetään oraakkelia, joka toteuttaa funktion $f\colon \{0,1\}^n \to \{0,1\}$. Lisäksi on luvattu, että funktio on joko vakiofunktio tai tasapainoitettu funktio. Vakiofunktiolla tarkoitetaan, että $f(x)$ on aina 0 tai 1, jokaisella $x \in \{0,1\}^n$. Tasapainoitetulla funktiolla tarkoitetaan, että lähtöjoukko on jaettu kahteen osaan A ja B, joissa on yhtä monta alkiota ja osalla A pätee $f(x) = 0$ sekä osalla B pätee $f(x) = 1$. Tällöin ongelma on tunnistaa onko f vakio- vai tasapainoitettu funktio käyttämällä oraakkelia [DJ92]. Tämän jälkeen rakennettujen kvanttitietokoneiden kubittimäärä on kasvanut. Vuonna 2020 kiinalaiset tutkijat väittivät rakentaneensa 76 kubitin tietokoneen [Sim20].

Tietoliikenteessä tarvitaan kryptografiaa ja salausta suojaamaan tiedonkulkua. Kryptografialla tarkoitetaan tiedettä, joka tutkii salakirjoitusmenetelmiä. Toisin sanoen halutaan suojata viestejä siten, että niitä ei voi lukea kukaan muu paitsi tarkoitettu vastaanottaja. Tähän tarkoitukseen käytetään usein avaimia, eli jotain informaatiota, joka on vain kahdella kommunikoivalla osapuolella. Viestit salataan käyttäen avainta, jolloin viestin sisällöstä tulee salakirjoitusta, mistä ei voida selvittää alkuperäistä viestiä. Tällöin viesti voidaan lähettää vastaanottajalle, joka osaa avata sen käyttämällä avainta.

Suuren mittakaavan kvanttitietokoneilla olisi mahdollista murtaa monia tänä päivänä käytössä olevia salausjärjestelmiä. Salausjärjestelmien murtaminen vaarantaisi digitaalisen kommunikoinnin luottamuksellisuuden sekä eheyden, ei voitaisi olla varmoja että viesti menee oikealle vastaanottajalle. Arvellaan, että salausalgoritmeja rikkovien kvanttitietokoneiden rakentamiseen menisi noin 20 vuotta. Aika kuulostaa pitkältä, mutta nykyisenkin kryptografiainfrastuktuurin käyttöönotto on kestänyt melkein 20 vuotta, joten kvanttilaskentaa kestävien salausmenetelmien kehittäminen on tärkeää.

Yhdysvaltalainen National Institute of Standards and Technology (NIST) on järjestänyt post quantum cryptography (PQC) hankkeen, missä pyritään löytämään kvanttilaskentaa kestäviä salausmenetelmiä. Hanke julkaistiin konferenssissa PQCrypto vuonna 2016 ja siihen on lähetetty monia algoritmiehdokkaita. Hanke edistyy kierroksittain ja jokaisella kierroksella huonoimpia ehdokkaita karsitaan pois. Hanke on edennyt kolmannelle kierrokselle, joka julkaistiin 22. heinäkuuta 2020.

Tässä tutkielmassa tarkastellaan yhtä PQC-hankkeeseen lähetettyä kolmannen kierroksen algoritmiehdokasta CRYSTALS Kyberiä. Kyber on IND-CCA2-turvallinen avaintenkapseloin-

timekanismi, jonka turvallisuus perustuu häirityn oppimisen ongelman (learning with errors) vaikeuteen. Kyber valittiin ehdokkaiden joukosta matemaattisen mielenkiinnon sekä haastavuuden takia.

Ensimmäisessä luvussa tutustustutaan todennäköisyyslaskentaan. Todennäköisyyslaskentaa tarvitaan Kyberin algoritmien tarkasteluun. Kyberin algoritmit ovat satunnaisia, eli on mahdollista, että esimerkiksi salauksen avaaminen epäonnistuu. Seuraavassa luvussa tutustutaan algebraan. Luvussa Algebra tutkitaan homo- ja isomorfismeja, ideaaleja sekä tekijärenkaita. Tekijärenkaat ovat erityisen tärkeitä, sillä niitä käytetään Kyberin algoritmeissa. Matematiikkaosuuden jälkeen esitellään kryptografian peruskäsitteitä. Tarkastellaan hajautusfunktiota, symmetristä sekä epäsymmetristä salausta ja avaintenkapselointimekanismeja. Luvun Kryptografia jälkeen päästään esittelemään Kyberin turvallisuuden takaava ongelma, häiritty oppiminen. Luvussa aloitetaan helpommasta häirityn oppimisen ongelmasta ja siirrytään sitten Kyberin käyttämään versioon samasta ongelmasta. Sitten esitellään itse Kyber, tarkastellaan ensin sen PKE-algoritmeja ja siirrytään sitten KEM-algoritmeihin. Tarkastelut tehdään matemaattisella tasolla, eikä pseudokoodina. Algoritmeissa ei siis kuvata esimerkiksi optimointiin käytettyä NTT-muunnosta. Lisäksi tutkielmassa keskitytään algoritmien tarkasteluun, eikä todisteta ongelmien vaikeutta. Lopuksi suoritetaan pienimuotoinen suorituskykymittaus käyttäen PQC-hankkeeseen liitettyä referenssitoteutusta, joka on optimoitu prosessoreille, jotka tukevat avx2-käskyjoukkoa.

2 Todennäköisyys

Tässä luvussa tutustutaan todennäköisyyslaskentaan. Todennäköisyyslaskentaa käytetään satunnaisilmiöiden esittämiseen ja matemaattiseen tarkasteluun. Melkein jokainen Kyberin algoritmi on satunnainen. Ensin määritellään todennäköisyysavaruus ja käydään samalla läpi esimerkkejä tyypillisen korttipakkaesimerkin avulla. Seuraavassa osiossa keskitytään satunnaismuuttujiin ja esitellään diskreetin satunnaismuuttujan perusominaisuuksia. Erityisesti kiinnostavaa Kyberin kannalta on keskitetty binomijakauma ja miten sen mukaan voidaan valita arvoja. Satunnaismuuttujien jälkeen tarkastellaan algoritmeja ja kuinka niitä voidaan kuvata eri tavoin. Luvussa oletetaan, että lukija pystyy seuraamaan yksinkertaista pseudokoodia. Pseudokoodissa käytetään ehtolauseita, silmukoita sekä arvon asettamista muuttujaan. Luku perustuu lähteisiin [Tuo96] ja [BDK+18].

2.1 Todennäköisyysavaruus

Todennäköisyyslaskennassa kaikki kuvataan alkeistapauksilla. Alkeistapaukset ovat satunnaisilmiöiden täydellisiä kuvauksia. Alkeistapauksia merkitään symbolilla ω, sekä kaikkien alkeistapausten joukkoa symbolilla Ω.

Tarkastellaan esimerkkinä kortin nostamista korttipakasta. Nostettu kortti voi olla mikä tahansa pakassa olevista korteista olettaen, että se on hyvin sekoitettu. Tällöin alkeistapauksia ovat kaikki kortit, jotka ovat pakassa, sillä jos tiedetään nostettu kortti, niin tiedetään mitä satunnaiskokeessa tapahtui. Esimerkiksi yksi alkeistapaus on, että nostettu kortti on pataässä. Tällöin kaikkien alkeistapausten joukko Ω, sisältää siis jokaisen kortin eli

$$\begin{aligned}\Omega = \{&\text{“nostettu kortti on ruutukakkonen”},\\ &\text{“nostettu kortti on pataässä”},\\ &\vdots\\ &\text{“nostettu kortti on ruutukuningas”}\}.\end{aligned}$$

Alkeistapaukset eivät kuitenkaan yksinään riitä kuvaamaan kaikkia mahdollisia reaalimaailman ilmiöitä. Jos halutaan tarkastella esimerkiksi, onko nostetun kortin arvo alle 4, huomataan, että monet alkeistapaukset täyttävät vaatimuksen. Tällaisia ilmiöitä voidaan kuvata muodostamalla alkeistapauksista joukkoja $A \subseteq \Omega$, joita kutsutaan tapahtumiksi. Tapahtumat auttavat kuvaamaan tilanteita, joissa oikeita vaihtoehtoja on monia. Esimerkiksi voidaan tarkastella tapahtumaa $A =$ “kortin arvo alle 4”. Tällöin tapahtuma A koostuu kaikkien maiden korteista, joiden arvo on 1-3. Tapahtumista voidaan muodostaa edelleen joukko, jota kutsutaan tapahtumaperheeksi ja merkitään symbolilla $\mathcal{F}$. Tapahtumaperhe $\mathcal{F}$ sisältää kaikki mahdolliset tapahtumat.

Esimerkki 2.1. Merkitään korttipakkaa, jossa ei ole jokereita, symbolilla C. Tällöin C on kaikkien korttien joukko ja siinä on yhteensä 52 eri alkiota. Merkitään jokaista korttia Xn, missä X on kortin maa ja n sen arvo. Korttien arvot määritellään seuraavasti: korttien 1–10 arvot ovat 1–10, jätkän arvo on 11, rouvan arvo on 12 ja kuninkaan arvo on 13. Merkitään eri maita symboleilla ♠, ♢, ♣ ja ♡.

Tarkastellaan satunnaisilmiötä, jossa sekoitetusta pakasta valitaan yksi kortti. Tällöin alkeistapausten joukko Ω on kaikkien korttien joukko C eli

$$\Omega = C = \{♠1, ♠2, \ldots, ♡12, ♡13\}.$$

Määritelmä 2.2. Olkoot A joukko ja $\mathcal{A} \subseteq \mathcal{P}(A) = \{B \mid B \subseteq A\}$. Joukkoperhe $\mathcal{A}$ on *σ-algebra*, jos

1. $A \in \mathcal{A}$,
2. jokaisella $B \in \mathcal{A}$ pätee $A \setminus B \in \mathcal{A}$ ja
3. kaikilla $B_i \in \mathcal{A}$, $i \in \{1, 2, \dots\}$, pätee $\bigcup_{i=1}^{\infty} B_i \in \mathcal{A}$.

Tällöin joukkoperhe $\mathcal{A}$ on joukkoon A liitetty *σ-algebra*.

Esimerkki 2.3. Esimerkissä 2.1 esiteltiin kaikkien korttien joukko C. Kiinnitetään seuraavaksi tapahtumaperhe $\mathcal{F}$. Otetaan tapahtumaperheeseen kaikki alkeistapausten joukon C osajoukot. Siis tapahtumaperheeksi $\mathcal{F}$ saadaan $\mathcal{P}(C)$, missä $\mathcal{P}(C)$ on alkeistapausten joukon C potenssijoukko. Tapahtumaperheeseen siis kuuluu esimerkiksi tapahtuma $A =$ "kortin arvo on yli 11". Osajoukkona ilmaistuna tapahtuma

$$A = \{\spadesuit 12, \spadesuit 13, \diamond 12, \diamond 13, \clubsuit 12, \clubsuit 13, \heartsuit 12, \heartsuit 13\} \subseteq C,$$

joten tapahtuma $A \in \mathcal{F}$. Tästä seuraa määritelmän 2.2 nojalla, että $\mathcal{P}(C)$ on joukkoon C liitetty σ-algebra.

Määritelmä 2.4. Olkoot $\mathcal{F}$ joukkoon Ω liitetty σ-algebra ja $\mathbb{P}\colon \mathcal{F} \to \mathbb{R}$ kuvaus. Kuvaus $\mathbb{P}$ on *todennäköisyys*, jos

1. $\mathbb{P}(A) \geq 0$ jokaisella $A \in \mathcal{F}$,
2. $\mathbb{P}(\Omega) = 1$ ja
3. jos tapahtumille $A_i \in \mathcal{F}$, missä $i \in \{1, 2, \dots\}$ pätee $A_i \cap A_j = \emptyset$ jokaisella $j \in \{1, 2, \dots\}$, kun $i \neq j$, eli tapahtumat A_i ovat pareittain erillisiä, niin pätee

$$\mathbb{P}\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mathbb{P}(A_i).$$

Määritelmä 2.5. Kolmikko $(\Omega, \mathcal{F}, \mathbb{P})$ on *todennäköisyysavaruus*, jos

1. alkeistapausten joukolle pätee $\Omega \neq \emptyset$,
2. tapahtumaperhe $\mathcal{F}$ on alkeistapausten joukkoon Ω liitetty σ-algebra ja
3. kuvaus $\mathbb{P}\colon \mathcal{F} \to \mathbb{R}$ on todennäköisyys.

Huomautus. Todennäköisyysavaruutta $(\Omega, \mathcal{F}, \mathbb{P})$, kutsutaan *symmetriseksi todennäköisyysavaruudeksi*, jos jokainen sen alkeistapaus on yhtä todennäköinen.

Esimerkki 2.6. Esimerkissä 2.3 huomattiin, että tapahtumaperhe $\mathcal{P}(C)$ on kaikkien korttien joukkoon C liitetty σ-algebra. Tarkastellaan seuraavaksi todennäköisyyskuvauksia. Valitaan ensin jokin korteista $c \in C$. Valitaan sitten korttipakasta yksi kortti umpimähkään eli jokaisen kortin todennäköisyys on sama. Kysymys on: millä todennäköisyydellä valittu kortti on $k \in C$.

Kysymystä vastaava tapahtuma on joukko $\{k\} \subset \mathcal{P}(C)$. Koska oikeita vaihtoehtoja on tasan 1 ja mahdollisia kortteja 52, niin todennäköisyys on

$$\mathbb{P}(\{k\}) = \frac{1}{52}.$$

Yleisemmin tapahtumalle $A \in \mathcal{P}(C)$ todennäköisyys on

$$\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{|A|}{52},$$

missä $|A|$ on tapahtuman A alkioiden lukumäärä. Huomataan myös, että $\mathbb{P}$ on todella todennäköisyys. Nyt voidaan muodostaa todennäköisyysavaruus $(\Omega, \mathcal{F}, \mathbb{P}) = (C, \mathcal{P}(C), \mathbb{P})$.

Apulause 2.7. Olkoon $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus. Tällöin

$$\mathbb{P}(\emptyset) = 0.$$

Todistus. Olkoon $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus. Koska $\mathcal{F}$ on σ-algebra, niin määritelmän 2.2 kohdan 3 nojalla

$$\mathbb{P}(\emptyset) = \mathbb{P}(\emptyset) + \mathbb{P}(\emptyset) + \dots$$

Yhtälön toteuttaa vain luku 0, siis $\mathbb{P}(\emptyset) = 0$. □

Lause 2.8. *Olkoot $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus ja $A, B \in \mathcal{F}$ tapahtumia, joille pätee $A \subseteq B$. Tällöin tapahtumien A ja B todennäköisyyksille pätee*

$$\mathbb{P}(A) \leq \mathbb{P}(B).$$

Todistus. Olkoot $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus ja $A, B \in \mathcal{F}$ tapahtumia, joille $A \subseteq B$. Muodostetaan tapahtumat $E_1 = A$, $E_2 = B \setminus A$ ja $E_i = \emptyset$, kun $i \geq 3$. Tapahtumien E_i yhdisteelle saadaan $E_1 \cup E_2 \cup \dots = B$. Lisäksi huomataan, että $E_i \cap E_j = \emptyset$, kun $i \neq j$. Nyt tapahtuman B todennäköisyydelle saadaan määritelmän 2.4 kohdan 3 ja lemman 2.7 nojalla

$$\begin{aligned}\mathbb{P}(B) &= \mathbb{P}(A) + \mathbb{P}(B \setminus A) + \sum_{i=3}^{\infty} \mathbb{P}\{\emptyset\} \\ &= \mathbb{P}(A) + \mathbb{P}(B \setminus A) \\ &\geq \mathbb{P}(A).\end{aligned}$$

Täten saadaan väite $\mathbb{P}(A) \leq \mathbb{P}(B)$. □

Määritelmä 2.9. Olkoot $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus ja A, B sen tapahtumia. Tapahtumat A ja B ovat *riippumattomia toisistaan*, jos

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

Määritelmä 2.10. Olkoot $(\Omega, \mathcal{F}, \mathbb{P})$ todennäköisyysavaruus ja $A, B \in \mathcal{F}$ sen tapahtumia, joille pätee $\mathbb{P}(B) > 0$. Osamäärää

$$\frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)},$$

missä $\mathbb{P}(A \cap B)$ on todennäköisyys, millä molemmat tapahtumat A ja B realisoituvat, kutsutaan *ehdolliseksi satunnaisuudeksi* ja merkitään

$$\mathbb{P}(A \mid B).$$

Esimerkki 2.11. Olkoon C kaikkien korttien joukko kuten esimerkissä 2.1. Tarkastellaan tilannetta, jossa pakasta nostetaan kaksi korttia ja tarkastellaan tapahtumia A = "ensimmäisen kortin arvo on 3" ja B = "toisen kortin arvo on 4". Tällöin perusjoukkona Ω on kaikkien korttien järjestetyt parit ja esimerkiksi tapahtuma A sisältää kaikki parit, joissa ensimmäisen alkion tai kortin arvo on 3 ja toinen kortti voi olla mikä tahansa. Tarkemmin ilmaistuna

$$\begin{aligned}\Omega &= \{(a,b) \mid a,b \in C, a \neq b\},\\ A &= \{(a,b) \in \Omega \mid a \in \{\spadesuit 3, \diamond 3, \clubsuit 3, \heartsuit 3\}\} \quad \text{ja}\\ B &= \{(a,b) \in \Omega \mid b \in \{\spadesuit 4, \diamond 4, \clubsuit 4, \heartsuit 4\}\}.\end{aligned}$$

Tarkastellaan tilannetta jossa kortteja ei laiteta takaisin pakkaan. Tällöin tapahtuman A todennäköisyys on

$$\mathbb{P}(A) = \frac{4}{52},$$

sillä arvoa 3 olevia kortteja on 4 kappaletta ja pakassa on yhteensä 52 korttia. Toisen kortin arvo ei vaikuta todennäköisyyteen, sillä se saa olla mikä tahansa. Jos ensimmäisen kortin arvo on 3, niin tapahtuman B todennäköisyydeksi saadaan

$$\mathbb{P}(B \mid A) = \frac{4}{51}.$$

Koska kortteja ei laitettu noston jälkeen takaisin pakkaan, niin pakassa on enää 51 korttia. Jos kortit olisi laitettu takaisin pakkaan, niin koko satunnaiskokeen perusjoukko Ω, olisi $C \times C$, sillä tällöin sama kortti voi tulla uudelleen. Tällöin tapahtuman B todennäköisyys olisi ollut

$$\mathbb{P}(B) = \frac{4}{52}.$$

Lisäksi huomataan, että jos kortteja ei laiteta takaisin pakkaan, niin tapahtumat A ja B ovat riippuvia, sillä

$$\begin{aligned}\mathbb{P}(A \cap B) &= \frac{4}{52} \cdot \frac{4}{51}\\ &\neq \frac{4}{52} \cdot \frac{4}{52}\\ &= \mathbb{P}(A)\mathbb{P}(B).\end{aligned}$$

Jos taas kortit laitetaan takaisin pakkaan, tapahtumat A ja B ovat riippumattomia, sillä

$$\begin{aligned}\mathbb{P}(A \cap B) &= \frac{4}{52} \cdot \frac{4}{52}\\ &= \mathbb{P}(A)\mathbb{P}(B).\end{aligned}$$

2.2 Satunnaismuuttujat

Tässä osiossa tarkastellaan satunnaismuuttujia. Ensin tarkastellaan ominaisuuksia, jotka pätevät yleisesti satunnaismuuttujille ja keskitytään sen jälkeen diskreetteihin satunnaismuuttujiin. Jatkuvia satunnaismuuttujia ei tarkastella. Kaikki satunnaismuuttujien tarkastelut tehdään todennäköisyysavaruudessa $(\Omega, \mathcal{F}, \mathbb{P})$, jollei toisin mainita.

2.2.1 Yleinen satunnaismuuttuja

Määritelmä 2.12. Olkoon $X\colon \Omega \to A$ kuvaus, missä joukolla A on Borelin rakenne eli joukkoon A on liitetty σ-algebra $\mathcal{A}$. Jos lisäksi jokaisella Borelin joukolla $B \in \mathcal{A}$ pätee

$$\{\omega \in \Omega \mid X(\omega) \in B\} \in \mathcal{F},$$

niin kuvausta X sanotaan *satunnaismuuttujaksi*.

Huomautus. Otetaan käyttöön helpottavia merkintöjä. Olkoon $X\colon \Omega \to A$ satunnaismuuttuja. Tällöin voidaan merkitä

$$\{X \in B\} = \{\omega \in \Omega \mid X(\omega) \in B\},$$

missä $B \in \mathcal{A}$ on Borelin joukko. Lisäksi jos $A = \mathbb{R}$, niin voidaan merkitä

$$\{X \leq x\} = \{\omega \in \Omega \mid X(\omega) \leq x\}$$

missä $x \in \mathbb{R}$. Huomataan, että merkintä $\{X \in B\}$ sisältää myös tapaukset $\{X \leq x\}$, valitsemalla oikea osajoukko $B \subseteq A$.

Lisäksi jos $A = \mathbb{R}$, niin satunnaismuuttujaa X kutsutaan reaaliseksi satunnaismuuttujaksi.

Määritelmä 2.13. Olkoon $X\colon \Omega \to \mathbb{R}$ reaalinen satunnaismuuttuja. Funktiota $F\colon \mathbb{R} \to \mathbb{R}$ kutsutaan satunnaismuuttujan X *kertymäfunktioksi*, jos

$$F(x) = \mathbb{P}\{X \leq x\}, \quad \text{jokaisella } x \in \mathbb{R}.$$

Määritelmä 2.14. Satunnaismuuttujat $X\colon \Omega \to A$ ja $Y\colon \Omega \to B$ ovat *riippumattomia*, jos

$$\mathbb{P}\{X \in A', Y \in B'\} = \mathbb{P}\{X \in A'\}\mathbb{P}\{Y \in B'\},$$

eli tapahtumat $\{X \in A'\}$ ja $\{Y \in B'\}$ ovat toisistaan riippumattomia, kun $A' \subseteq A$ ja $B' \subseteq B$ ovat Borelin joukkoja.

Määritelmä 2.15. Olkoot X satunnaismuuttuja, $n \in \mathbb{N}$ ja $X_1, \ldots, X_n$ riippumattomia, samoin jakautuneita satunnaismuuttujia kuin X. Tällöin jonoa

$$X_{(n)} = (X_1, X_2, \ldots, X_n)$$

kutsutaan *otokseksi* satunnaismuttujan X jakaumasta.

2.2.2 Diskreetti satunnaismuuttuja

Diskreeteillä satunnaismuuttujilla tarkoitetaan satunnaismuuttujia, jotka voivat saada vain numeroituvan määrän arvoja.

Määritelmä 2.16. Olkoon $X\colon \Omega \to A$ satunnaismuuttuja. Satunnaismuuttujaa X kutsutaan *diskreetiksi satunnaismuuttujaksi*, jos sen arvojoukko A on numeroituva ja

$$\{X = x\} \in \mathcal{F},$$

jokaisella $x \in A$.

Määritelmä 2.17. Olkoon $X\colon \Omega \to A$ diskreetti satunnaismuuttuja. Satunnaismuuttujan X *pistetodennäköisyysfunktio* on kuvaus $f : A \to \mathbb{R}$, jolle

$$f(x) = \mathbb{P}\{X = x\}, \quad x \in A.$$

Esimerkki 2.18. Tarkastellaan todennäköisyysavaruutta, jossa pakasta nostetaan yksi kortti ja tapahtumaa "kortin maa on pata tai ruutu". Tapahtumaa voidaan kuvata satunnaismuuttujalla $X\colon C \to \{0,1\}$, missä 1 tarkoittaa, että kortti on pata tai ruutu ja 0 tarkoittaa, että ei ollut. Selvästi X on diskreetti satunnaismuuttuja, sillä arvojoukossa on vain kaksi alkiota. Tällöin, jos kortti valitaan satunnaisesti satunnaismuuttujan X pistetodennäköisyysfunktio on

$$f(x) = \frac{1}{2}, \quad x \in \{0,1\}.$$

Määritelmä 2.19. Olkoot $\Omega = A$ joukko ja $|A| = n$ sen alkioiden lukumäärä. Olkoot $(\Omega, \mathcal{F}, \mathbb{P})$ symmetrinen todennäköisyysavaruus ja X sen diskreetti satunnaismuuttuja, jolle $X(\omega) = \omega$. Satunnaismuuttujan X pistetodennäköisyysfunktio on

$$f\colon A \to \mathbb{R}, f(x) = \frac{1}{n}.$$

Tällöin kuvataan tapahtumaa, jossa jokainen joukon A alkio on yhtä todennäköinen. Jos alkio $a \in A$ valitaan pistetodennäköisyysfunktion A mukaisesti sanotaan, että alkio a on valittu *umpimähkään* ja merkitään $a \leftarrow A$.

Diskreetin satunnaismuuttujan X kertymäfunktio F määrätytyy yksikäsitteisesti sen pistetodennäköisyysfunktion f perusteella. Siis

$$F(x) = \sum_{t \le x} f(t), \quad x \in A.$$

Esimerkki 2.20. Esimerkissä 2.6 luotiin todennäköisyysavaruus $(\Omega, \mathcal{F}, \mathbb{P})$. Olkoon $X\colon C \to \mathbb{R}$ diskreetti satunnaismuuttuja, joka kuvaa jokaisen kortin sen arvolle eli ässät kuvautuvat arvolle 1, rouvat arvolle 12 ja niin edelleen. Tarkastellaan tapahtumaa A = "kortin arvo on yli 11". Yhdistämällä edellisten esimerkkien tietoja saadaan tapahtuman A todennäköisyydeksi

$$\mathbb{P}\{A\} = \mathbb{P}\{X > 11\} = \frac{|A|}{52} = \frac{8}{52}.$$

Määritelmä 2.21. Olkoon $X\colon \Omega \to A$ diskreetti reaalinen satunnaismuuttuja, missä A on numeroituva. Tällöin diskreetin satunnaismuuttujan X *odotusarvo* on

$$\mathbb{E}(X) = \sum_{a \in A} a f(a),$$

missä f on X:n pistetodennäköisyysfunktio. Sarjan oletetaan suppenevan itseisesti.

Määritelmä 2.22. Olkoon X diskreetti satunnaismuuttuja. Satunnaismuuttujan X sanotaan olevan *binomijakautunut* parametrein n ja p , jos sen pistetodennäköisyysfunktio on

$$\mathbb{P}\{X = i\} = \binom{n}{i} p^i (1-p)^{n-i},$$

kun $0 \le i \le n$. Merkitään $X \sim \mathrm{Bin}(n,p)$, missä $n \in \mathbb{N}$ ja $p \in [0,1]$.

Binomijakautuneen satunnaismuuttujan X odotusarvo on

$$\mathbb{E}(X) = np.$$

Määritelmä 2.23. Olkoon $\eta \in \mathbb{N}$ ja X diskreetti satunnaismuuttuja, jolle pätee

$$X \sim \mathrm{Bin}(2\eta, \frac{1}{2}).$$

Olkoon $Y = X - \eta$ diskreetti satunnaismuuttuja. Tällöin satunnaismuuttujan Y sanotaan olevan *keskitetysti binomijakautunut* ja merkitään

$$Y \sim \mathbb{B}_\eta.$$

Määritelmässä 2.23 satunnaismuuttujan X pistetodennäköisyysfunktio $\mathbb{P}$ on

$$\begin{aligned}
\mathbb{P}\{X = x\} &= \binom{2\eta}{x} \cdot \left(\frac{1}{2}\right)^x \cdot \left(1 - \frac{1}{2}\right)^{2\eta - x} \\
&= \binom{2\eta}{x} \cdot \left(\frac{1}{2}\right)^{x + (2\eta - x)} \\
&= \binom{2\eta}{x} \cdot \left(\frac{1}{2}\right)^{2\eta} \\
&= \binom{2\eta}{x} \cdot 2^{-2\eta}
\end{aligned}$$

ja sen odotusarvo on

$$\mathbb{E}(X) = 2\eta \cdot \frac{1}{2} = \eta.$$

Satunnaismuuttujan Y odotusarvo on siis $\mathbb{E}(Y) = 0$ ja sen pistetodennäköisyysfunktio $\mathbb{P}$ on

$$\begin{aligned}
\mathbb{P}\{Y = y\} &= \mathbb{P}\{X = y + \eta\} \\
&= \binom{2\eta}{x + \eta} \cdot 2^{-2\eta}.
\end{aligned}$$

Huomautus. Huomataan, että lukuja voidaan valita keskitetyn binomijakauman $\mathbb{B}_\eta$ mukaisesti seuraavasti:

$$\sum_{i=0}^{\eta} a_i - a_i' = Y \in \mathbb{Z},$$

missä $a_i, a_i' \in \{0, 1\}$ ovat riippumattomia ja umpimähkään valittuja bittejä.

2.3 Algoritmit

Algoritmit koostuvat kokoelmasta askeleita. Askeleella tarkoitetaan jotakin yksinkertaista operaatiota, esimerkiksi tuloksen asettamista muuttujaan. Algoritmeille voidaan syöttää arvoja ja ne voivat palauttaa arvoja. Vaikka algoritmit sisältävätkin enemmän tietoa kuin kuvaukset, ne voidaan kuitenkin pelkistää kuvauksiksi. Algoritmeja kuvataan kuvauksena $A\colon X \to Y$, missä X on kaikkien algoritmille annettavien syötteiden joukko ja Y algoritmin tulosten joukko. Jos algoritmi A palauttaa arvon $y \in Y$ argumentilla $x \in X$, merkitään $y := A(x)$.

On tärkeää huomata, että algoritmin kuvaaminen kuvauksena ei kuvaa sen askelien sisältöä, mikä on suuressa osassa tilanteita tarpeen. Algoritmin kuvaaminen kuvauksena yksinkertaistaa sen käsittelyä, mutta samalla piilottaa informaatiota.

Algoritmin askelien kuvaus

Algoritmit kuvataan pseudokoodina. Pseudokoodi on kokoelma algoritmin askeleita, joilla on järjestys. Algoritmin askeleet ovat operaatioita, jotka puolestaan voivat olla ehtolauseita (**if**), silmukoita (**while** ja **for**), arvon asettaminen muuttujaan tai arvon palauttaminen (**return**).

Huomautus. Algoritmien yhteydessä askeleella tarkoitetaan kahta asiaa. Kun puhutaan *algoritmin askeleista* tarkoitetaan algoritmin pseudokoodin rivejä. Jos puhutaan *askeleista*, tarkoitetaan askeleita, joita algoritmi suorittaa. Esimerkiksi for -silmukka on yksi algoritmin askel, mutta se suorittaa monta askelta algoritmin suorituksen yhteydessä.

Esimerkki 2.24. Tarkastellaan seuraavaksi esimerkkiä algoritmin sisällön kuvaamisesta. Kuvataan algoritmi $A\colon \mathbb{Z}_+ \to \mathbb{N} \cup \{-1\}$, joka palauttaa syötteenä saaneen luvun neliön, jos se on alle 100 muuten -1.

Algoritmi 2.1 A: Pieni neliö -algoritmi

Syöte: $n \in \mathbb{Z}_+$

```
1: if n ≤ 10 then
2:     s := n · n
3:     return s
4: else
5:     return −1
6: end if
```

Suoritusaika

Algoritmien suorityskyvyn arviointi on tärkeää ja yksi yleisimmistä mitoista, jota seurataan. Lisäksi saadaan tietoa siitä, onko algoritmin suorittamien mahdollista, jos käytössä on tietty määrä laskentakapasiteettia. Jotkin algoritmit koostuvat vain vakiomäärästä askelia, jolloin suorittaminen on nopeaa. Toisten algoritmien suoritusaika voi riippua annettavista syötteistä, jolloin suoritusaika kasvaa syötteiden mukana. Sanotaan, että algoritmi A on *polynomiaikainen*, jos sen suoritusaika syötteillä, joiden koko on n, on enintään $c \cdot n^k$, joillain $k, c > 0$. Polynomista suoritusaikaa pidetään teoreettisesti nopeana ja usein nykyaikaisilla koneilla riittää laskentakapasiteettia niiden suorittamiseen.

Satunnaisalgoritmit

Algoritmit voivat olla myös satunnaisia. Tällöin on mahdollista, että algoritmi palauttaa eri arvon, vaikka parametrit ovat samat. Satunnaisalgoritmien kuvaamiseen tarvitaan siis erilainen kuvaus. Jos A on satunnaisalgoritmi, se voidaan ajatella kuvauksena $A\colon X \times \Omega \to Y$, missä X on kaikkien algoritmille annettavien parametrien joukko, Ω kaikkien satunnaismalliin liityvien alkeistapausten joukko ja Y algoritmin tulosten joukko. Jos satunnaisalgoritmi A palauttaa arvon $y \in Y$ argumentilla x, niin merkitään $y \leftarrow A(x)$. Lisäksi jos satunnaisalgoritmin A satunnainen komponentti halutaan erotella, niin merkitään $y := A(x; \omega)$, missä ω on realisoitunut alkeistapaus.

Esimerkki 2.25. Tarkastellaan seuraavaksi esimerkkiä satunnaisalgoritmin sisällön kuvaamisesta. Kuvataan algoritmi $B\colon \mathbb{Z}_+ \times \Omega \to \mathbb{Z}_+$, joka palauttaa parametrinaan saaneen luvun neliön tai kuution, kerrottuna vakiolla.

Algoritmi 2.2 A: Neliö tai kuutio -algoritmi

Syöte: $n \in \mathbb{Z}_+$

1: $r \leftarrow \{2, 3\}$ ▷ Valitaan r umpimähkään.
2: $k \leftarrow \{1, 2, 3\}$
3: $s := k \cdot n^r$
4: **return** s

Algoritmissa käytetään kahta satunnaismuuttujaa r ja k, joiden perusjoukot ovat $\Omega_r = \{2, 3\}$ ja $\Omega_k = \{1, 2, 3\}$. Satunnaismuuttujat r ja k ovat riippumattomia toisistaan, joten satunnaismuuttuja, joka tekee algoritmista satunnaisen on $X \colon \Omega_r \times \Omega_k \to A_r \times A_k$. Tällöin siis

$$B \colon \mathbb{Z}_+ \times (\underbrace{\Omega_r \times \Omega_k}_{= \Omega}) \to \mathbb{Z}_+$$

ja merkitään $B \leftarrow B(n)$ tai $b := B(n; X)$, jokaisella $n \in \mathbb{Z}_+$.

3 Algebra

Kyberin operaatiot tapahtuvat pääasiassa muotoa $\mathbb{Z}_q^n/\langle I\rangle$ olevissa renkaissa. Tässä luvussa selvitetään, mitä nämä renkaat ovat ja kuinka ne on muodostettu. Tarkastelu aloitetaan renkaista sekä tietyistä syklotomisista polynomeista, jotka ovat Kyberin kannalta tärkeitä. Tämän jälkeen tutustutaan ideaaleihin ja muodostetaan tekijärenkaiden konstruktio. Lopuksi vielä tarkastellaan homomorfismeja ja isomorfismeja, joita tarvitaan häirityn oppimisen ongelman yhteydessä.

Luvussa oletetaan, että lukija on tutustunut ekvivalenssirelaation, Abelin ryhmän, kokonaisalueen, kunnan ja polynomirenkaan käsitteisiin. Lähteinä on käytetty [Mol11] ja [BDK+18].

Määritelmä 3.1. Olkoot a, b reaalilukuja ja n kokonaisluku, jolle $n > 1$. Jos pätee $a - b = kn$, jollakin $k \in \mathbb{Z}$ niin sanotaan, että a on *kongruentti* b modulo n ja merkitään $a \equiv b \pmod{n}$.

Määritelmä 3.2. Olkoon n positiivinen kokonaisluku. Jos kokonaisluvulle r pätee $r' \equiv r \pmod{n}$, niin tällöin $r' = r \bmod^{\pm} n$ on yksikäsitteinen alkio, jolle pätee

$$-\frac{n}{2} < r' \leq \frac{n}{2} \quad .$$

Lisäksi kokonaisluku $r'' = r \bmod^{+} n$ on yksikäsitteinen alkio, jolle pätee $r'' \equiv r \pmod{n}$ sekä $0 \leq r'' < n$.

Esimerkki 3.3. Tarkastellaan esimerkkiä kongruenssista. Esimerkiksi luvuille 7 ja -1 pätee

$$\begin{aligned} 7 &\equiv -1 \pmod{4}, \\ 7 \bmod^{\pm} 4 &= -1 \quad \text{ja} \\ 7 \bmod^{+} 4 &= 3. \end{aligned}$$

Määritelmä 3.4. Kolmikko $(R, +, \cdot)$ on *rengas*, jos

1. $(R, +)$ on Abelin ryhmä,
2. kertolaskulle pätee
   - 2.1. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ jokaisella $x, y, z \in R$
   - 2.2. $x \cdot e = e \cdot x$ jokaisella $x \in R$,
3. laskutoimituksille pätee

$$\begin{aligned} x(y + z) &= xy + xz \quad \text{ja} \\ (x + y)z &= xz + yz \end{aligned}$$

   jokaisella $x, y, z \in R$ ja
4. on olemassa yksikköalkio $1 \in R$, jolla $1 \cdot x = x \cdot 1 = x$ jokaisella $x \in R$.

Määritelmä 3.5. Olkoon n positiivinen kokonaisluku. Tällöin 2^n:s *syklotominen polynomi* on

$$\Phi_{2^n} = x^{2^{n-1}} + 1.$$

Huomautus. Syklotomiset polynomit voidaan määritellä myös yleisemmin, mutta Kyberin kontekstissa tarkastellaan erityisesti syklotomisia polynomeja Φ_{2^n}.

Määritelmä 3.6. Olkoon f polynomi. Polynomin f *aste* on sen nollastapoikkeavan termin suurin x:n eksponentti. Vakioiden $c \in \mathbb{R}$ aste on 0 ja nollan aste on $-\infty$. Polynomin f astetta merkitään $\deg(f)$.

Esimerkki 3.7. Polynomin $f = x^2 + \frac{1}{2} \in \mathbb{Q}[x]$ aste on

$$\deg(f) = \deg(x^2 + \frac{1}{2}) = 2 \quad \text{ja}$$

polynomin $g = x^5 + 4x^4 + 5 \in \mathbb{Z}[x]$ aste on

$$\deg(g) = \deg(x^5 + 4x^4 + 5) = 5.$$

Määritelmä 3.8. Olkoot $n, m \in \mathbb{Z}_+$ ja $\mathbf{A} \in \mathbb{R}^{n \times m}$ matriisi. Matriisin $\mathbf{A}$ *transpoosi* on matriisi $\mathbf{A}^{\mathrm{T}}$, jolla $\mathbf{A}[i, j] = \mathbf{A}^{\mathrm{T}}[j, i]$, missä $0 \leq i < m$ ja $0 \leq j < n$.

Esimerkki 3.9. Matriisin

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \quad \text{ja vektorin} \quad \mathbf{b} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

transpoosit ovat

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \quad \text{ja} \quad \mathbf{b}^{\mathrm{T}} = \begin{bmatrix} x & y & z \end{bmatrix}.$$

3.1 Ideaalit

Määritelmä 3.10. Olkoot $\mathbf{R} = (R, +, \cdot)$ rengas ja $I \subseteq R$ sen osajoukko, jolla $I \neq \emptyset$. Osajoukkoa I kutsutaan renkaan $\mathbf{R}$ *ideaaliksi*, jos

1. $I \neq \emptyset$,
2. $a - b \in I$, jokaisella $a, b \in I$ ja
3. $ra, ar \in I$, jokaisella $a \in I$, $r \in R$.

Huomataan, että määritelmästä 3.10 seuraa, että jos $a_1, a_2, \ldots, a_n \in I$, missä $n \in \mathbb{N}$, niin $r_1 a_1 + r_2 a_2 + \cdots + r_n a_n \in I$ jokaisella $r_1, r_2, \ldots, r_n \in R$. Lisäksi jos $\mathbf{R}$ on vaihdannainen rengas ja $a_1, a_2, \ldots, a_n \in R$, niin joukko

$$\langle a_1, a_2, \ldots, a_n \rangle = \left\{ \sum_{j=1}^{n} r_j a_j \mid r_j \in R \right\},$$

on renkaan $\mathbf{R}$ ideaali. Käytetään jatkossa myös merkintää

$$a + I = \{a + i \mid i \in I\},$$

kun I on renkaan $\mathbf{R}$ ideaali.

Esimerkki 3.11. Olkoon $\mathbf{R}$ rengas. Tällöin R on itsensä ideaali.

Esimerkki 3.12. Tunnetusti $\mathbb{Z}$ on rengas. Tällöin $n\mathbb{Z}$ on renkaan $\mathbb{Z}$ ideaali jokaisella $n \in \mathbb{N}$.

Esimerkki 3.13. Olkoon K kunta. Tiedetään, että yksi polynomirenkaan $(K[x], +, \cdot)$ ideaaleista on $\langle x \rangle$.

Esimerkki 3.14. Olkoot $R = (\mathbb{Z}[x], +, \cdot)$ polynomirengas ja $f = x^{256} + 1 \in \mathbb{Z}[x]$ polynomi. Tälloin f on syklotominen polynomi, ja saadaan

$$I = \langle f \rangle = \{r \cdot f \mid r \in \mathbb{Z}[x]\},$$

joka on polynomirenkaan $\mathbb{Z}[x]$ ideaali. Esimerkiksi

$$x^{257} + x = x \cdot (x^{256} + 1) = x \cdot f \in I \quad \text{ja}$$
$$6x^{361} - 4x^{256} + 6x^{105} - 4 = (6x^{105} - 4) \cdot (x^{256} + 1) = (6x^{105} - 4) \cdot f \in I.$$

Tätä polynomirengasta tarvitaan jatkossa.

3.1.1 Homomorfismit

Määritelmä 3.15. Olkoot **A** ja **B** renkaita ja $f\colon A \to B$ kuvaus. Kuvaus f on *rengas homomorfismi*, jos pätee

1. $f(a + b) = f(a) + f(b)$ jokaisella $a \in A$ ja $b \in B$,
2. $f(ab) = f(a)f(b)$ jokaisella $a \in A$ ja $b \in B$ ja
3. $f(1_A) = 1_B$.

Määritelmä 3.16. Olkoot **A**, **B** renkaita ja $f\colon A \to B$ kuvaus. Tällöin kuvauksen f *ydin* on

$$\text{Ker}\,(f) = \{x \in A \mid f(x) = 0\}.$$

Kuvauksen f *kuva* on

$$\text{Im}\,(f) = \{f(x) \mid x \in A\}.$$

Esimerkki 3.17. Olkoot $\mathbb{Z}$ ja $\mathbb{Z}_4 = (\mathbb{Z}_4, +, \cdot)$ renkaita. Tällöin kuvaus

$$f\colon \mathbb{Z} \to \mathbb{Z}_4, f(x) = x + 4\mathbb{Z}$$

on homomorfismi.

3.1.2 Isomorfismit

Määritelmä 3.18. Olkoot **A**, **B** renkaita ja $f\colon A \to B$ rengashomomorfismi. Jos rengashomomorfismi f on bijektio, niin kuvausta f sanotaan *isomorfismiksi*. Tällöin merkitään $\mathbf{A} \cong \mathbf{B}$ ja sanotaan, että rengas **A** ja **B** ovat *isomorfisia*.

Esimerkki 3.19. Olkoon **R** rengas. Tällöin identiteettikuvaus $\text{id}_R\colon R \to R, \text{id}_R(x) = x$ on isomorfismi.

Lause 3.20. *Rengas $\mathbb{Z}[x]/\langle x + 1 \rangle$ on isomorfinen renkaan $\mathbb{Z}$ kanssa.*

Todistus. Merkitään $I = \langle x + 1 \rangle$. Olkoon $h\colon \mathbb{Z}[x]/I \to \mathbb{Z}$ kuvaus, jolla $h(f + I) = f(-1)$. Osoitetaan sitten, että h on homomorfismi. Ensinnäkin jos $a + I, b + I \in \mathbb{Z}[x]/I$, niin

$$\begin{aligned} h(a + I) + h(b + I) &= a(-1) + b(-1) \\ &= \sum_{i=0}^{k} a_i(-1)^i + \sum_{j=0}^{m} b_j(-1)^j \\ &= \sum_{i=0}^{\max(k,m)} (a_i + b_i)(-1)^i \\ &= h((a + b) + I). \end{aligned}$$

Vastaavasti saadaan tulolle

$$\begin{aligned} h(a + I)h(b + I) &= a(-1)b(-1) \\ &= \left(\sum_{i=0}^{n} a_i(-1)^i\right)\left(\sum_{j=0}^{m} b_j(-1)^j\right) \\ &= \sum_{k=0}^{n+m} \left(\sum_{i=0}^{k} a_i b_{k-i}\right)(-1)^i \\ &= h(ab + I). \end{aligned}$$

Selvästi ykkösalkion kuva on

$$h(1 + I) = 1_{\mathbb{Z}}.$$

Täten h on homomorfismi. Osoitetaan sitten, että h on injektio ja surjektio. Homomorfismin h ydin on

$$\operatorname{Ker} h = \{f + I \in \mathbb{Z}[x]/I \mid f(-1) = 0\} = 0 + I,$$

joten h on injektio. Olkoon $a \in \mathbb{Z}$. Tällöin myös $a \in \mathbb{Z}[x]$, joten $h(a + I) = a(-1) = a$. Täten h on surjektio ja siis isomorfismi, joten $\mathbb{Z}[x]/I \cong \mathbb{Z}$. □

3.2 Tekijärenkaat

Määritelmä 3.21. Joukon R ekvivalenssirelaatio $\sim$ on renkaan $\mathbf{R}$ *kongruenssi*, jos jokaisella $x, x', y, y' \in R$, joille $x \sim x'$ ja $y \sim y'$, pätee $x + y \sim x' + y'$ ja $xy \sim x'y'$. Kongurenssia vastaava *tekijärengas* $\mathbf{R}/\!\sim$ on rengas, missä

$$R/\!\sim \; = \{[x]_\sim \mid x \in R\} \quad \text{ja} \quad [x]_\sim = \{y \in R \mid x \sim y\}.$$

Laskutoimitukset ovat

$$\begin{aligned} [x]_\sim + [y]_\sim &= [x + y]_\sim \quad \text{ja} \\ [x]_\sim \cdot [y]_\sim &= [x \cdot y]_\sim. \end{aligned}$$

Lause 3.22. *Olkoon* $\mathbf{R}$ *rengas ja* $\sim$ *sen kongruenssi. Tällöin* $I = [0]_\sim$ *on renkaan* $\mathbf{R}$ *ideaali ja jokaisella* $x, y \in R$ *pätee*

$$x \sim y \iff x + I = y + I,$$

missä $x + I = \{x + a \mid a \in I\}$.

Todistus. Olkoon $\sim$ renkaan $\mathbf{R}$ kongruenssi ja $I = [0]_\sim$. Selvästi $0 \in I$, joten $I \neq \emptyset$. Olkoot $x, y \in I$ ja $r \in R$. Tällöin $x \sim 0$ ja $y \sim 0$. Lisäksi huomataan, että $-y \sim -y$. Koska $\sim$ on kongruenssi, niin $y + (-y) \sim 0 + (-y)$ siis $0 \sim -y$. Edelleen koska $x \sim 0$ ja $-y \sim 0$, niin $x + (-y) \sim 0 + 0$, joten $x - y \in I$. Lisäksi koska $r \sim r$, jokaisella $r \in R$, niin $x \cdot r \sim 0 \cdot r = 0$ ja $r \cdot x \sim r \cdot 0 = 0$. Siis I on renkaan $\mathbf{R}$ ideaali.

Olkoot $x, y \in R$. Tällöin

$$\begin{aligned} x \sim y &\iff (-y) + x \sim (-y) + y \\ &\iff (-y) + x \sim 0 \\ &\iff x - y = -y + x \in I \\ &\iff x + I = y + (-y) + x + I = y + I. \end{aligned}$$

□

Merkitään renkaan $\mathbf{R} = (R, +, \cdot)$ tekijärengasta $\mathbf{R}/\sim_I$ lyhyemmin $\mathbf{R}/I$, missä I on renkaan $\mathbf{R}$ ideaali ja $\sim_I$ sen kongruenssi. Merkitään myös perusjoukkoa $R/I = R/\sim_I$, missä $\sim_I$ on $\mathbf{R}$:n ideaalia I vastaava kongruenssi.

Esimerkki 3.23. Tarkastellaan polynomirenkaan $\mathbb{Z}[x]$ tekijärengasta $\mathbb{Z}[x]/\langle x^2 \rangle$. Tällöin

$$\begin{aligned} x^3 + \langle x^2 \rangle &= x^2 \cdot x + \langle x^2 \rangle \\ &= 0 + \langle x^2 \rangle \in \mathbb{Z}[x]/\langle x^2 \rangle. \end{aligned}$$

Määritelmä 3.24. Olkoon $\mathbb{Z}[x]$ polynomirengas ja $a + I \in \mathbb{Z}[x]/I$, missä I on polynomirenkaan $\mathbb{Z}[x]$ ideaali. Tällöin polynomia $a \in \mathbb{Z}[x]$ kutsutaan ekvivalenssiluokan $a + I$ *edustajaksi*.

Esimerkki 3.25. Olkoot $\mathbb{Z}[x]$ polynomirengas ja $f = x^{256} + 1$ polynomi. Tarkastellaan tekijärengasta $\mathbb{Z}[x]/I$, missä $I = \langle f \rangle$. Tällöin saadaan esimerkiksi

$$\begin{aligned} x^{256} + 1 + I = 0 + I \in \mathbb{Z}[x]/I \quad \text{ja} \\ 5x^{261} + 5x^5 + I = 5x^5 + I \in \mathbb{Z}[x]/I. \end{aligned}$$

Huomataan, että ekvivalenssiluokilla on monta eri edustajaa.

Lause 3.26. *Olkoot* $\mathbf{K}$ *kunta,* $\mathbf{K}[x]$ *sen polynomirengas ja* $f, g \in K[x]$ *polynomeja, joille pätee* $g \neq 0$. *Tällöin on olemassa yksikäsitteiset polynomit* $q, r \in K[x]$, *joille pätee* $f = gq + r$ *ja* $r = 0$ *tai* $\deg(r) < \deg(g)$.

Todistus. Jos $f = 0$, niin väite pätee arvoilla $q, r = 0$.

Jos $\deg(f) < \deg(g)$, niin väite pätee arvoilla $q = 0$ ja $r = f$.

Kun $\deg(f) \geq \deg(g)$, niin todistetaan, että väitteen mukaiset polynomit ovat olemassa kaikille polynomeille g induktiolla polynomin f asteen suhteen. Jos $\deg(f) = 0$, niin myös $\deg(g) = 0$. Tällöin $f, g \in K$. Koska $\mathbf{K}$ on kunta ja $g \neq 0$, niin on olemassa g^{-1}, jolle pätee $g \cdot g^{-1} = 1$. Valitaan $q = g^{-1} f$ ja $r = 0$. Tällöin saadaan

$$f = g(g^{-1} f) + 0 = gq + r,$$

joten väite pätee. Tehdään induktio-oletus, että väite pätee polynomille f, jolla $\deg(f) < n$. Täytyy osoittaa, että väite pätee polynomille f, jonka aste on n. Merkitään polynomeja muodossa

$$\begin{aligned} f = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \quad \text{ja} \\ g = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_1 x + b_0, \end{aligned}$$

missä $m \leq n$ ja $a_n, a_m \neq 0$. Koska $\mathbf{K}$ on kunta ja $b_m \neq 0$, niin on olemassa kerroin $b_m^{-1} \in K$, jolle pätee $b_m \cdot b_m^{-1} = 1$. Tällöin kertomalla polynomia g saadaan

$$\begin{aligned}(a_n b_m^{-1} x^{n-m})g &= (a_n b_m^{-1} x^{n-m})(b_m x^m + \cdots + b_1 x + b_0) \\ &= a_n x_n + a_n b_m^{-1} b_{m-1} x^{m-1} + a_n b_m^{-1} b_0 x^{n-m}.\end{aligned}$$

Huomataan, että tämän polynomin ensimmäinen kerroin on sama kuin polynomin f. Tällöin polynomin asteelle pätee $\deg(f - (a_n b_m^{-1} x^{n-m})g)) < n$. Koska polynomin aste on alle n, voidaan soveltaa induktio-oletusta. Induktio-oletuksen nojalla on olemassa polynomit q_1, r, joille pätee

$$f - (a_n b_m^{-1} x^{n-m})g = gq_1 + r,$$

missä $r = 0$ tai $\deg(r) < \deg(g)$. Täten saadaan edelleen

$$f = (a_n b_m^{-1} x^{n-m} + q_1)g + r,$$

missä $r = 0$ tai $\deg(r) < \deg(g)$. Täten on osoitettu, että väite pätee kun $\deg(f) = n$.

Osoitetaan vielä polynomien q ja r yksikäsitteisyys. Olkoot q' ja r' polynomeja, joille pätee $f = q'g + r'$ ja $r' = 0$ tai $\deg(r') < \deg(g)$. Tällöin olisi

$$\begin{aligned} & gq + r = gq' + r' \\ \iff\ & g(q - q') = r - r'\end{aligned}$$

joillakin $q, r \in K[x]$. Jos olisi $q - q' \neq 0$, niin tällöin $\deg(g(q - q')) \geq \deg(g)$. Koska $r = 0$, $r' = 0$, $\deg(r) < \deg(g)$ tai $\deg(r') < \deg(g)$, niin $\deg(r - r') < \deg(g)$. Täten on oltava niin, että $\deg(q - q') = 0$ eli $q = q'$. Tällöin saadaan $\deg(g(q - q')) = 0$, joten myös $\deg(r - r') = 0$. Tästä seuraa, että $r = r'$. Täten polynomit q, r ovat yksikäsitteisiä. □

Olkoon $\mathbf{R}[x]/\langle h \rangle$ tekijärengas, missä $h \in R[x]$ ja $h \neq 0$. Merkitään $I = \langle h \rangle$. Edellisen lauseen perusteella tekijärakenteiden alkioille $a + I \in R[x]/I$ voidaan valita yksikäsitteiset edustajat r, joille pätee $a = gh + r$ ja $r = 0$ tai $\deg(r) < \deg(h)$. Olkoon $a + I \in R[x]/I$. Tällöin $a \in R[x]$ ja lauseen 3.26 nojalla on olemassa $q, r \in R[x]$, joilla $a = hq + r$ ja $\deg(r) = 0$ tai $\deg(r) < \deg(h)$. Tällöin $a + I = hq + r + I = r + I$, missä $\deg(r) < \deg(h)$, joten r on se yksikäsitteinen polynomi, jolla $a + I = r + I$. Tällöin siis r on saman ekvivalenssiluokan edustaja kuin a, mutta sen aste on pienempi kuin polynomin h. Lisäksi jos $\deg(r) = 0$, niin $a = 0$.

Todistettiin, että jos $a + I \in R/I$, niin on olemassa yksikäsitteinen $b + I \in R/I$, jolla $\deg(b) < \deg(h)$ ja $a + I = b + I$ eli $a \sim b$. Tällöin alkioita voidaan käsitellä myös pelkkien edustajiensa avulla, sillä ne ovat yksikäsitteiset. Voidaan siis merkitä $a, b \in R/I$. Tällöin operaatiot toimivat samalla tavalla kuin renkaassa $R[x]$ varustettuna yhteen ja kertolaskulla modulo h.

Esimerkki 3.27. Olkoon $I = \langle x^{256} + 1 \rangle$. Tällöin renkaalle $\mathbb{Z}_{3329}[x]/I$ pätee

$$\begin{aligned} x^{256} + 2 = 1 \in \mathbb{Z}_{3329}[x]/I \quad &\text{ja} \\ 5x^5 + 9 \in \mathbb{Z}_{3329}[x]/I.& \end{aligned}$$

Rengas $\mathbb{Z}_{3329}[x]/I$ sisältää siis polynomeja, joiden aste on alle 256 ja kertoimet ovat alle 3329.

4 Kryptografia

Kryptografia on tiede, joka tutkii salakirjoitusmenetelmiä. Niitä käytetään esimerkiksi viestinnän luottamuksellisuuden varmistamiseen, käyttäjien todentamiseen sekä viestien eheyden tarkastamiseen. Salakirjoitusmenetelmillä pyritään siis turvaamaan kommunikaatiota. Tässä luvussa tutustutaan kryptografiaan ja kryptografiassa tarvittaviin peruskäsitteisiin. Tarkastellaan ensin kryptografisten menetelmien käyttötarkoituksia ja sen jälkeen tutustutaan eri salausmenetelmiin. Ensimmäisenä menetelmänä tarkastellaan kryptografisia hajautusfunktioita, minkä jälkeen siirrytään symmetrisen ja epäsymmetrisen salauksen käsitteisiin. Tämän jälkeen esitellään symmetristä ja epäsymmetristä salausta yhdistelevä avainten kapselointimekanismi. Jokaisesta kryptografian salauskeinosta pyritään selvittämään hyvät sekä huonot puolet ja pyritään ratkaisemaan ongelmat, joita niissä esiintyy. Luku perustuu lähteisiin [Kes16], [Dan09], [oST15] ja [LCR+09].

Kryptografiaa tarvitaan epäluotettavan kanavan kautta kommunikoidessa. Epäluotettavalla kanavalla tarkoitetaan kanavaa, jossa muut entiteetit kuin kommunikoijat voivat päästä käsiksi välitettävään informaatioon. Entiteetillä tarkoitetaan, jotakin tunnistettavaa erillistä osaa. Entiteettejä voivat olla esimekiksi henkilöt, yritykset sekä tietokoneet.

Esimerkiksi internet on epäluotettava kanava ja internetin kautta kommunikoinnissa tarvitaan kryptografiaa, sillä kommunikointi tapahtuu muiden tietokoneiden kautta. Kryptografiaa käytetään moniin eri tarkoituksiin:

1. Todentaminen: Todennetaan, että viestin lähettäjä tai vastaanottaja on oikea entiteetti.
2. Luottamuksellisuus: Varmistetaan, että vain viestissä osoitettu vastaanottaja voi lukea viestin.
3. Eheys: Varmistetaan, että viestiä ei ole muokattu lähettämisen jälkeen.

Määritellään seuraavaksi sanastoa. *Viesti* on mikä tahansa bittijono. Viesteillä kuvataan lähetettävää informaatiota. *Avaimeksi* kutsutaan mitä tahansa asiaa, jonka avulla viestejä voidaan salata, lukea tai molempia. Avaimiksi voidaan luokitella esimerkiksi informaatio, jonka avulla tietokone osaa salata viestejä ja julkisesti tunnetut algoritmit, joiden avulla viesti on salattu. Merkitään viestiä v, joka on salattu avaimella a, symbolilla v_a. Olkoot $\mathcal{K}_E$ kaikkien salaukseen käytettävien avainten joukko, $\mathcal{K}_D$ kaikkien salauksen avaamiseen käytettävien avainten joukko, C kaikkien mahdollisten salattujen viestien joukko ja $V = \{0,1\}^*$ kaikkien viestien joukko. Esimerkiksi UTF-8 koodauksen avulla biteistä voidaan muodostaa sanoja. Tällöin viestin salaaminen on kuvaus

$$\mathrm{Enc}\colon \mathcal{K}_E \times V \to C.$$

Salauksen avaaminen on kuvaus

$$\mathrm{Dec}\colon \mathcal{K}_D \times C \to V.$$

Viestejä voidaan myös allekirjoittaa ja verifioida. Allekirjoituksen idea on sama kuin salauksen, mutta sitä käytetään lähettäjän verifiointiin, eli varmistamaan että lähettäjä on oikea entiteetti. Keskitytään kuitenkin vain viestien salaamiseen ja avaamiseen.

Kryptografia jaetaan usein kolmeen osaan:

1. Kryptografiset hajautusfunktiot: Viestistä muodostetaan äärellismittainen tiiviste ilman avainten käyttöä. Hajautusfunktioita käytetään esimerkiksi digitaalisessa allekirjoituksessa.

2. Symmetrinen salaus: Viestin salaaminen ja avaaminen tehdään samalla avaimella.

3. Epäsymmetrinen salaus: Viestin salaaminen ja avaaminen tehdään eri avaimilla.

4.1 Kryptografiset hajautusfunktiot

Hajautusfunktiot ovat kuvauksia, jotka kuvaavat äärellisen pituisia bittijonoja n-mittaisiksi bittijonoiksi. Niiden tarkoituksena on kuvata pienetkin muutokset argumentissa suurina muutoksina sen kuva-alkiossa.

Määritelmä 4.1. Olkoon n positiivinen kokonaisluku. Tällöin kuvaus $f\colon \{0,1\}^* \to \{0,1\}^n$ on *hajautusfunktio*, missä $\{0,1\}^*$ on kaikkien äärellismittaisten bittijonojen joukko. Hajautusfunktion kuva-alkiota $f(v) = w$ kutsutaan *tiivisteeksi*.

Määritelmä 4.2. Olkoon f hajautusfunktio. Tällöin hajautusfunktiota f sanotaan *kryptografiseksi hajautusfunktioksi*, sille pätee alkukuvakestävyys (preimage resistance): Olkoon $a \in \{0,1\}^n$ tiiviste. Ei ole olemassa polynomiaikaista algoritmia B, jolle pätee $x := B(a)$, missä B voi käyttää hajautusfunktiota f oraakkelina ja $f(x) = a$. Hajautusfunktion f käyttämisellä oraakkelina tarkoitetaan, että algoritmi A voi käyttää hajautusfunktiota f toteutuksessaan. Toisin sanoen ei ole olemassa polynomiaikaista algoritmia, jolla löydetään tiivisteen a alkukuva x.

Huomautus. Hajautusfunktioilta vaaditaan myös lisäehto törmäyskestävyys, jota ei voida määrittää matemaattisesti. Törmäyskestävyys (collision resistance): On laskennallisesti mahdotonta löytää paria (x, x'), jolle pätee $x \neq x'$ ja $f(x) = f(x')$. Laskennallisesti mahdotonta tarkoittaa sitä, että parin löytäminen vie tämänhetkisiltä tietokoneilta niin paljon aikaa (universumin elinikä), ettei niitä voida käyttää. Toisin sanoen on laskennallisesti mahdotonta löytää eri alkiot, joiden tiivisteet ovat samat.

Huomautus. Lisäksi huomataan, että hajautusfunktion ehdosta 1 seuraa toisen alkukuvan kestävyys (second preimage resistance). Toisen alkukuvan kestävyys on määreitelty seuraavasti: Jos $f(x) = a$, niin on laskennallisesti mahdotonta löytää alkiota x', jolle $x' \neq x$ ja $f(x) = f(x')$. Toisin sanoen on vaikeaa löytää eri alkio, jonka tiiviste on sama kuin tunnetun alkion tiiviste.

Tunnettuja kryptografisia hajautusfunktioita ovat esimerkiksi SHA3-128 sekä SHAKE-256. Molemmat algoritmit kuuluvat SHA-3 (secure hash algorithm 3) algoritmi perheeseen ja ne ovat kuvauksia

$$\text{SHA3-128}\colon \{0,1\}^* \to \{0,1\}^{128} \quad \text{ja}$$
$$\text{SHAKE-256}\colon \{0,1\}^* \to \{0,1\}^{256}.$$

Esimerkki 4.3. Tarkastellaan kryptografista hajautusfunktiota SHA3_{128} ja kuinka argumentin muuttaminen vaikuttaa tiivisteeseen.

Argumentti	Argumentti bitteinä	Kuva-alkio $\text{SHA3}_{128}(x)$
b	01100010	641f6330d9945eeba58bdd147ac90079
c	01100011	6a030ed9881caea749d114bc89aa9997

Huomataan, että yhdenkin bitin muuttaminen muuttaa tiivisteen aivan toisenlaiseksi. Taulukon argumentit ovat UTF-8 koodattua tekstiä, jotka on muutettu biteiksi seuraavassa sarakkeessa. Tämän jälkeen bittijonoista voidaan muodostaa SHA3 tiiviste.

4.1.1 Viestin todentamiskoodi

Viestin todentamiskoodit (message authentication code, MAC) ovat kryptografisten hajautusfunktioiden tapaisia algoritmeja. Niillä on samat turvallisuusvaatimukset kuin kryptografisilla hajautusfunktioilla. Viestin todentamiskoodit esitetään kuvauksina

$$f\colon \{0,1\}^* \times \mathcal{K} \to \{0,1\}^n,$$

missä $n \in \mathbb{N}$, $\{0,1\}^*$ on kaikkien viestien joukko ja $\mathcal{K}$ on kaikkien avainten joukko. Viestin todentamiskoodit tekevät *tunnisteen* viestin ja avaimen avulla. Tällöin, jos Matti lähettää Liisalle viestin, niin Matti lisää viestiin sen tunnisteen. Kun Liisa vastaanottaa viestin, tunniste erotetaan viestistä ja Liisa laskee itse tunnisteen viestiosasta. Jos Matin ja Liisan laskemat tunnisteet ovat samat, tiedetään, ettei viestiä ole muokattu.

4.2 Symmetrinen salaus

Symmetrisessä salauksessa (symmetric cryptography, secret key cryptography) käytetään samaa avainta viestien salaamiseen sekä avaamiseen. Symmetrisessä salauksessa käytettävää avainta kutsutaan myös *symmetriseksi avaimeksi*.

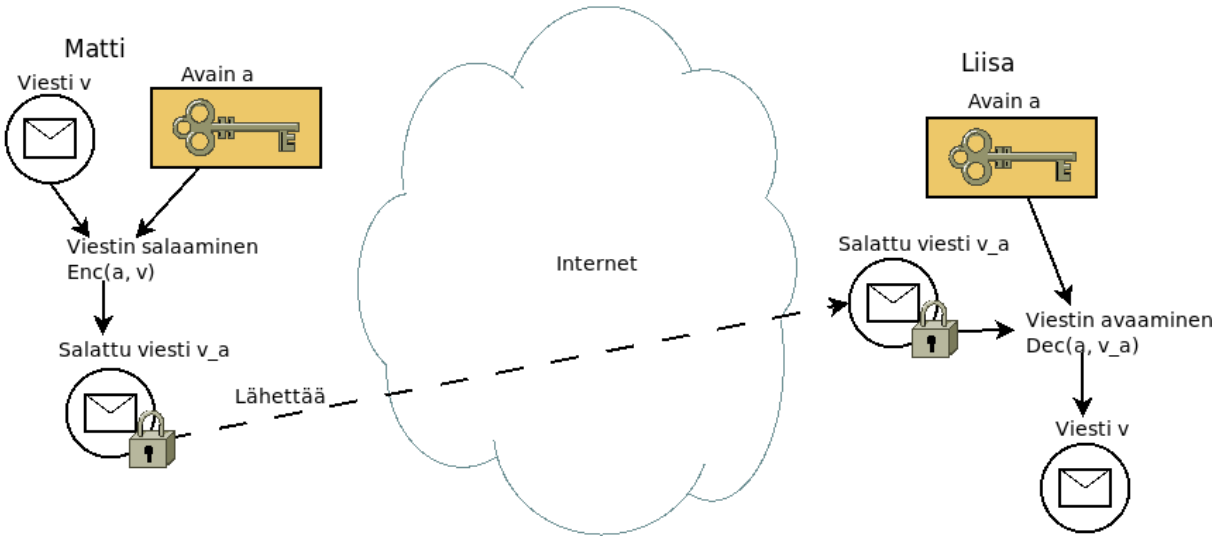
Salausta kutsutaan *symmetriseksi salaukseksi*, jos sen algoritmeille Enc ja Dec pätee

$$v = \mathrm{Dec}(a, \mathrm{Enc}(a, v)),$$

missä a on symmetrinen avain ja v viesti.

Tunnetut symmetriset salausskeemat jaetaan usein virta- ja lohkosalauksiin. Virtasalauksessa käsitellään yksittäistä bittiä tai tavua kerrallaan ja avaimesta muodostetaan pseudosatunnainen virta, joka on siis eri jokaisen salattavan bitin tai tavun salauksen yhteydessä. Lohkosalauksessa puolestaan salataan määrätynmittainen lohko kerrallaan. Lohkosalauksessa käytössä on sama avain, jolla jokainen lohko salataan. Tämä tarkoittaa, että joillakin salausmoodeilla sama bittijono salautuu aina samaksi salatuksi bittijonoksi. Ongelma voidaan kuitenkin välttää käyttämällä lohkojen ketjutusmoodia, eli sisällytetään salaukseen pala edellistä lohkoa. Salausmoodilla tarkoitetaan lohkosalauksen operaatiomuotoa, joka kertoo, kuinka lohkosalauksella voidaan salata tietoa, joka on pidempi kuin yksi lohko. Virtasalauksessa sama bitti tai tavu ei salaudu samalla tavalla, koska avain on eri jokaisen salattavan bitin tai tavun kohdalla.

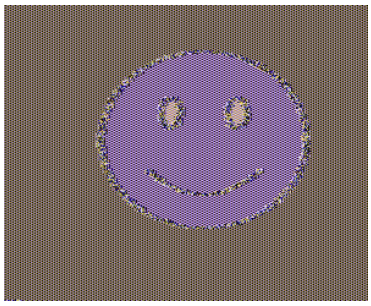
Kuva 4.1 esittää kuinka symmetristä avainta voidaan käyttää turvalliseen viestin välitykseen.



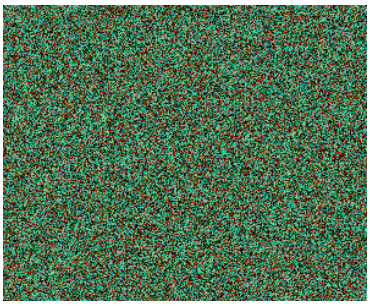
Kuva 4.1: Symmetrinen salaus

Matti ja Liisa generoivat ensin yhteisen symmetrisen avaimen a. Tällöin viestin voidaan välittää epäluotettavan kanavan yli, sillä salatusta viestistä ei saa koko viestin informaatiosisältöä selville ilman avainta a. Salatusta viestistä voidaan kuitenkin saada jotain selville salausmoodista riippuen. Esimerkiksi viestin pituus ja bitmap-kuvat eivät kaikilla algoritmien moodeilla salaudu tarpeeksi hyvin.

1: Salaamaton kuva.



2: Kuva salattu ECB moodilla



3: Kuva salattu CBC moodilla

Kuva 4.2: Kuvan salaaminen eri AES moodeilla

Lisäksi on tärkeää huomata, että vaikka viestiä voidaan muokata välitysvaiheessa, muokkaus voidaan huomata käyttämällä viestin todentamiskoodia.

4.2.1 Hyvät ja huonot puolet

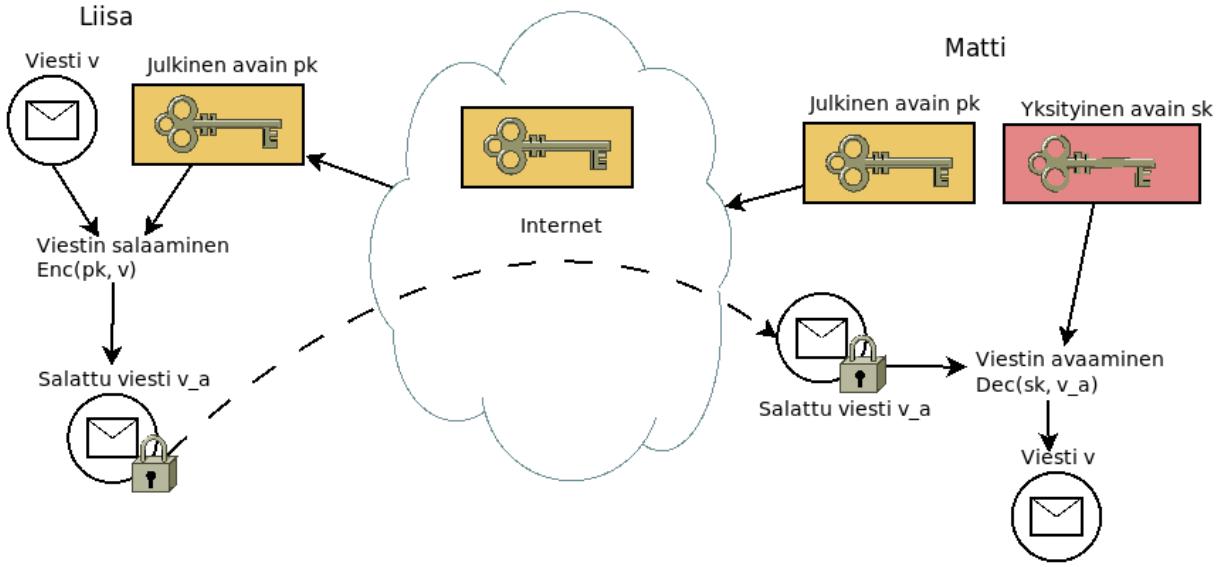
Algoritmit ovat yleisesti nopeita, minkä takia ne soveltuvat hyvin suurten tietomäärien salaamiseen. Salausta voidaan edelleen nopeuttaa erityisillä kryptografiaa nopeuttavilla laitteistoilla (hardware accelerated cryptography).

Symmetrisen salauksen suurin ongelma on, että avain täytyy olla molemmilla osapuolilla, jotta viestit pystytään salaamaan ja avaamaan. Avaimen lähettäminen turvallisesti on hankalaa, sillä jos avain joutuu ylimääräisten entiteettien haltuun, sitä voidaan käyttää viestien avaamiseen ja salaamiseen.

4.3 Epäsymmetrinen salaus

Epäsymmetrisessä salauksessa (public key cryptography, asymmetric cryptography) käytetään avainpareja. Avainparit koostuvat julkisesta avaimesta ja yksityisestä avaimesta. Yksityiset avaimet on tarkoitus nimensä mukaisesti pitää salassa ja vain avaimen tekijän hallussa. Julkiset avaimet on sen sijaan tarkoitettu yleisesti jaettaviksi. Epäsymmetristä salausta kutsutaan myös julkisen avaimen salaukseksi, mutta käytetään jatkossa termiä epäsymmetrinen salaus.

Tarkastellaan seuraavaksi esimerkkiä epäsymmetrisestä salauksesta. Olkoon (pk, sk) Matin generoima avainpari, missä pk on julkinen ja sk on yksityinen avain. Liisalla on viesti v, jonka hän haluaa lähettää Matille. Tällöin Liisa salaa viestin $v \in \mathcal{M}$ käyttämällä Matin julkista avainta pk ja lähtettää salatun viestin $\mathrm{Enc}(\mathrm{pk}, v) = v_{\mathrm{pk}}$ Matille epäturvallista kanavaa pitkin. Nyt Matti käyttää yksityistä avainta sk salauksen avaamiseen ja saa alkuperäisen viestin $\mathrm{Dec}(\mathrm{sk}, v_{\mathrm{pk}}) = v$ kuvan 4.3 mukaisesti.



Kuva 4.3: Salaaminen julkisella avaimella

Julkisen avaimen kryptografia perustuu yksisuuntaisiin funktioihin. Yksisuuntaisella funktiolla tarkoitetaan funktiota, jonka arvon tietokone on nopea laskemaan, mutta hidas laskemaan sen käänteisfunktion arvon. Yksisuuntaisiin funktioihin sisältyy yleensä avain, jolla alkuperäinen arvo on nopea laskea.

Esimerkki 4.4. Olkoot P kaikkien alkulukujen joukko ja

$$X = \{(a, b) \in P \times P \mid a \leq b\}.$$

Olkoon

$$f \colon X \to \mathbb{N},\ f(x, y) = xy.$$

Olkoot $a = 2503$ ja $b = 7901$. Tällöin $a, b \in P$ ja $a \leq b$. Tällöin $f(a, b) = ab = 16220753$ on nopea laskea. Kuitenkaan $f^{-1}(19776203)$ ei ole läheskään yhtä helppo ja nopea laskea kuin $f(a, b)$. Jos kuitenkin tiedetään, että $a = 2503$, saadaan helposti selville, että

$$b = \frac{19776203}{2503} = 7901.$$

4.3.1 Hyvät ja huonot puolet

Epäsymmetrisessä salauksessa yksityistä avainta ei lähetetä missään tilanteessa, joten vältytään symmetrisen salauksen avaimen lähetysongelmalta. Lisäksi jokaisella käyttäjällä on oma yksityinen avain, joka mahdollistaa myös käyttäjän tunnistautumisen.

Algoritmit ovat merkittävästi hitaampia kuin symmetrisen salauksen algoritmit, joten epäsymmetrinen salaus ei sovellu suurien tietomäärien lähetykseen. Lisäksi julkisen avaimen lähettäminen toiselle osapuolelle on ongelmallista. Oletetaan, että Liisa kysyy Matilta hänen julkista avaintaan, mutta viestin kaappaa Pekka. Sitten Pekka lähettää oman julkisen avaimensa Liisalle tekeytyen Matiksi ja Liisa lähettää oman julkisen avaimensa Pekalle, mutta luulee Pekan olevankin Matti. Sitten Pekka tekee normaalin julkisten avainten vaihdon Matin kanssa. Tällöin Pekka pystyy lukemaan viestit, jotka Liisa luulee lähettävänsä Matille, vaikka todellisuudessa viestit menevätkin Pekalle, joka vain uudelleen lähettää ne Matille. Tätä kutsutaan väliintulohyökkäykseksi. Tämän takia tarvitaan jokin entiteetti, johon molemmat osapuolet luottavat, joka pystyy todentamaan, että julkinen avain on todellakin Matin eikä Pekan. Keskittymättä tämän ongelman ratkaisuun enempää sanottakoon, että julkisen avaimen infrastruktuuri on luotu ratkaisemaan tämän ongelman ja se on laajasti käytössä.

4.4 Avainten kapselointimekanismi

Molemmissa edellä esitellyissä menetelmissä on parannettavaa. Epäsymmetrisessä salauksessa viestin pituus voi olla rajoitettu ja algoritmit ovat hitaita, jos salattavaa tietoa on paljon. Symmetrisessä salauksessa avainten vaihto on ongelmallista. Avainten kapselointimekanismi (key encapsulation mechanism, KEM) on kehitetty yhdistämään molemmat menetelmät. Symmetrinen avain voidaan jakaa käyttäen epäsymmetristä salausta. Kun molemmilla osapuolilla on symmetrinen avain, salaus on nopeampaa. Avainten kapseloinnin avulla voidaan myös käyttää eri avainta joka viestille. Tällöin jos hyökkääjä murtaa avaimen, hyökkääjä saa tietoonsa vain sillä avaimella salatun viestin sisällön. Avaimen kapselointi on kuvaus

$$\text{Encaps}\colon \mathcal{K}_P \to C \times \mathcal{K}_W,$$

missä $\mathcal{K}_P$ on kaikkien julkisten avainten joukko, C on salattu viesti ja $\mathcal{K}_W$ kaikkien symmetristen avainten joukko. Salattu viesti $c \in C$ on valittu siten, että kun se avataan, siitä voidaan päätellä samalla generoitu symmetrinen avain $w \in \mathcal{K}_W$. Huomataan että on mahdollista, että c on vain epäsymmetrisellä avaimella salattu symmetrinen avain w_{pk}, mutta sen ei tarvitse olla. Avainten kapseloinnin avaaminen on kuvaus

$$\text{Decaps}\colon \mathcal{K}_S \times C \to \mathcal{K}_W,$$

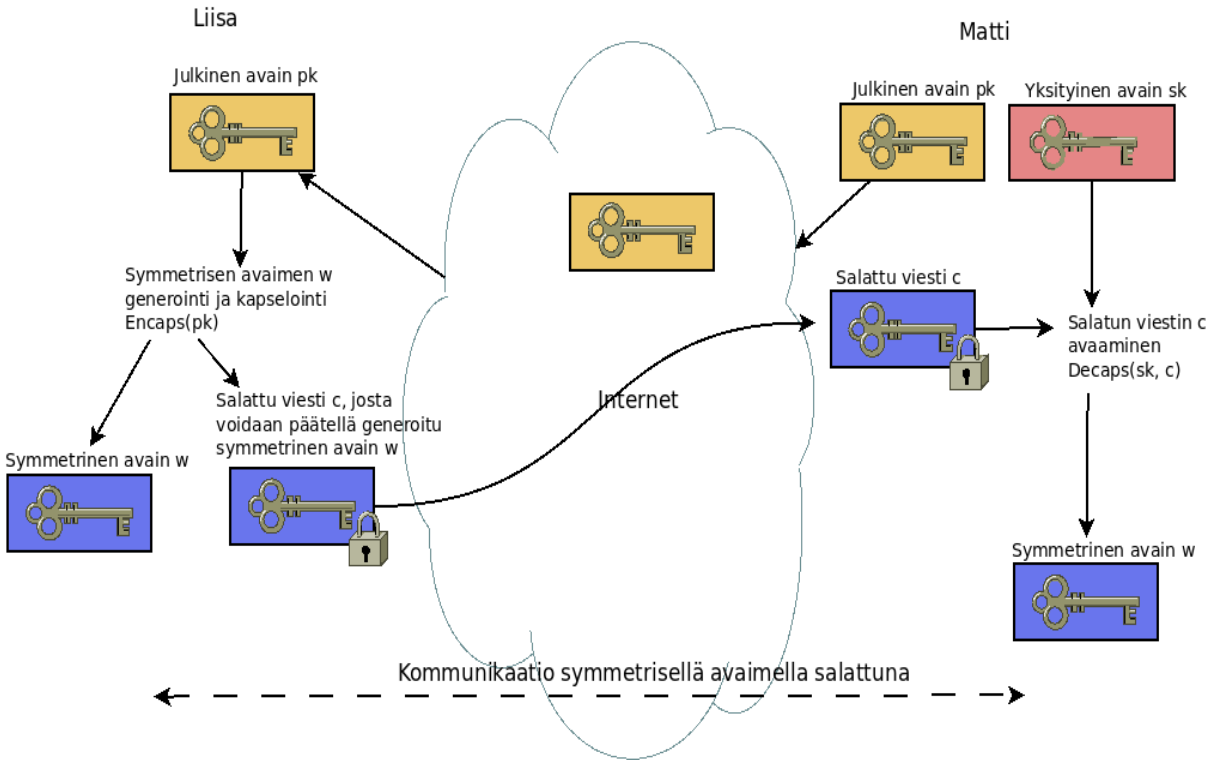
missä $\mathcal{K}_S$ on kaikkien viestien avaamiseen käytettävien avainten joukko.

Tarkastellaan seuraavaksi esimerkkiä avainten kapseloinnista. Olkoon (pk, sk) Matin generoima epäsymmetrinen avainpari. Sitten Liisa generoi symmetrisen avaimen w seuraavasti:

$$\text{Encaps}(\text{pk}) = (c, w),$$

missä salatusta tekstistä c voidaan johtaa symmetrinen avain w. Tällöin Matti saa symmetrisen avaimen w avaamalla salatun viestin c

$$\text{Decaps}(\text{sk}, c) = w.$$



Kuva 4.4: Avainten kapselointimekanismi

Huomataan, että avainten kapseloinnilla ratkaistaan symmetristen avainten välityksen ongelma. Lisäksi epäsymmetristen algoritmien hitaus ei enää haittaa, sillä epäsymmetrisiä avaimia käytetään vain (lyhyen) symmetrisen avaimen salaamiseen ja avaamiseen. Tällöin symmetristä avainta voidaan käyttää viestin v lähettämiseen. Huomataan vielä, että vaikka kuvassa 4.4 salataan avain w, voidaan avaimen w tilalla käyttää mitä tahansa informaatiota, josta avain w on pääteltävissä.

4.4.1 Hyvät ja huonot puolet

Avainten kapselointi käyttää hyväkseen symmetrisen salauksen algoritmien nopeutta. Epäsymmetristä salausta käytetään vain symmetristen avainten lähetykseen, jolloin salauksen hitaus ei haittaa. Tämän jälkeen kommunikaatiota voidaan jatkaa nopeilla symmetrisillä avaimilla. Avainten kapselointi siis ratkaisee symmetristen avainten lähetysongelman käyttämällä epäsymmetristä salausta.

Avainten kapselointi on myös haavoittuvainen väliintulohyökkäykselle. Väliintulohyökkäys toimii myös avainten kapselointiin, sillä ensimmäinen vaihe samanlainen kuin epäsymmetrisessä salauksessa.

5 Häiritty oppiminen

Tässä luvussa tutustutaan häirityn oppimisen ongelmaan (learning with errors). Häirityn oppimisen ongelma takaa Kyberin turvallisuuden kryptografiakäytössä ja sen uskotaan kestävän kvanttilaskentaa [Reg09]. Tutustutaan ensin tavanomaiseen häirityn oppimisen ongelmaan ja häirityn oppimisen rengasversioon. Tämän jälkeen muodostetaan yleisempi molemmat ongelmat sisältävä häiritty oppiminen moduuleissa ja todistetaan, että se todella sisältää edelliset häirityn oppimisen ongelmat. Luvun lähteinä on käytetty artikkeleita [Reg09], [LPR10] ja [BVG12].

5.1 Esittely

Häirityn oppimisen ongelma voidaan korkealla tasolla kuvata seuraavasti. Valitaan salaisuus s, jonka kanssa kerrotaan satunnaisia alkioita a. Kertolaskulla tarkoitetaan pistetuloa tässä yhteydessä. Kertolaskun tulokseen lisätään satunnainen pieni virhe e, jotta tulokset eivät ole aina oikein. Ongelman ratkaisijan tehtävä on löytää salaisuus s pyytämällä satunnaisia alkioita $(a, a \cdot s + e)$ äärellisen monta kappaletta.

Olkoot $q, n \in \mathbb{Z}$ parametreja, missä parametrille q pätee $q \geq 2$. Olkoot $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ mielivaltaisesti valittu salaisuus ja $\chi \colon \mathbb{Z} \to \mathbb{R}$ pistetodennäköisyysfunktio, jonka mukaan voidaan valita satunnaisia virheitä. Muodostetaan seuraavaksi yksittäinen satunnaiskoe, josta saadaan satunnaisia alkioita joukosta $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Valitaan $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ umpimähkään ja $e \in \mathbb{Z}$ pistetodennäköisyysfunktion χ mukaisesti. Tällöin $\mathbf{a}$ ja e voidaan ajatella satunnaismuuttujiksi ja voidaan muodostaa yhdistetty satunnaismuuttuja $A_{\mathbf{s},\chi} \colon \mathbb{Z}_q^n \times \mathbb{Z} \to \mathbb{Z}_q^n \times \mathbb{Z}_q$, jolle

$$A_{\mathbf{s},\chi}(\mathbf{a}, e) = (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle +_q e),$$

missä $+_q$ on yhteenlasku modulo q ja

$$\langle \mathbf{a}, \mathbf{s} \rangle = \sum_{i=1}^{n} a_i \cdot s_i.$$

Nyt satunnaismuuttuja $A_{\mathbf{s},\chi}$ kuvaa yksittäistä satunnaiskoetta.

Alun korkeantason kuvauksessa ongelman ratkaisija saa kuitenkin pyytää äärellisen monta alkiota. Tämä tarkoittaa, että koetta toistetaan äärellisen monta kertaa. Muodostetaan otos riippumattomista satunnaismuuttujista $A_{\mathbf{s},\chi,i}$, jotka ovat jakautuneet kuten $A_{\mathbf{s},\chi}$. Olkoon otos

$$A_{\mathbf{s},\chi,(k)} = (A_{\mathbf{s},\chi,1}, \ldots, A_{\mathbf{s},\chi,k}),$$
$$A_{\mathbf{s},\chi,(k)}((\mathbf{a}_1, e_1), \ldots, (\mathbf{a}_k, e_k)) = (A_{\mathbf{s},\chi}(\mathbf{a}_1, e_1), \ldots, A_{\mathbf{s},\chi}(\mathbf{a}_k, e_k)),$$

Tällöin ongelman ratkaisijan täytyy päätellä salaisuus $\mathbf{s}$ äärellisestä otoksesta $A_{\mathbf{s},\chi}$. Tätä ongelmaa kutsutaan *häirityksi oppimiseksi* ja merkitään $\mathrm{LWE}_{n,q,\chi}$. Erityisesti tätä esitysmuotoa kutsutaan häirityn oppimisen hakuversioksi. Ongelma voidaan muotoilla myös päätösversioksi. Päätösversiossa ongelma on päätellä, ovatko otoksen alkiot virheellisiä sisätuloja vai virheettömiä umpimähkään valittujen alkioiden sisätuloja.

Ilman ongelman hankaluuden todistamista voidaan todeta, että se vaikuttaa vaikealta [Reg09]. Ratkaisijan tulisi tunnistaa salaisuus $\mathbf{s}$ satunnaisten arvojen sisätuloista, mutta tulokset eivät ole välttämättä oikein. On kuitenkin tärkeää huomata pistetodennäköisyysfunktion χ rooli ongelmassa. Jos χ on vakio, niin tulokset ovat häiriöttömiä tai jos χ valitsee arvon umpimähkään niin ongelman ratkaisu vaikuttaa mahdottomalta vaikka käytössä olisi rajaton laskentakapasiteetti.

Yleensä kuitenkin ongelma voidaan ratkaista, jos käytössä on rajaton laskentakapasiteetti, sillä käsiteltävät renkaat ovat äärellisiä. Ongelma on kuitenkin osoittautunut riittävän vaikeaksi ettei laskentakapasiteetti riitä ongelman ratkaisuun siedettävässä ajassa. Kryptografiassa vaikeus perustellaan usein toisella hyvin tutkitulla ongelmalla, jonka tiedetään olevan vaikea ratkaista, tai kukaan ei ainakaan ole julkaissut ratkaisua. Tällaiset ongelmat voidaan sitten muokata uuteen muotoon ja todeta, että uusi ongelma on myös vaikea, sillä se on muodostettu jo tiedetystä vaikeasta ongelmasta. Esimerkiksi häiritty oppiminen perustuu pariteetin oppimisen ongelmaan (learning with parity).

Ongelma muistuttaa hieman epäsymmetrisen salauksen piirteitä, sillä vektorin $\mathbf{s}$ voidaan kuvitella symboloivan yksityistä avainta ja otoksen $A_{\mathbf{s},\chi,(k)}$ realisaation julkista avainta. Tällöin ongelman ratkaisija yrittää itse asiassa saada selville yksityisen avaimen käyttämällä vain julkista avainta. Ongelma ei kuitenkaan ota kantaa siihen, miten yksityisellä avaimella salataan tietoa.

Esimerkki 5.1. Tarkastellaan esimerkkiä häiritystä oppimisesta $\mathrm{LWE}_{n,q,\chi}$, valitaan parametrit $n = 3$ ja $q = 2$. Valitaan sitten salaisuus

$$(1, 0, 1) = \mathbf{s} \in \mathbb{Z}_2^3.$$

Pistetodennäköisyysfunktio χ on kuvaus $\chi\colon \mathbb{Z} \to \mathbb{R}$ ja yksittäistä satunnaiskoetta kuvaa satunnaismuuttuja

$$\begin{aligned} A_{\mathbf{s},\chi}\colon (\mathbb{Z}_2^3 \times \mathbb{Z}) &\to (\mathbb{Z}_2^3 \times \mathbb{Z}_2), \\ A_{\mathbf{s},\chi}(\mathbf{a}) &= (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle +_q e). \end{aligned}$$

Oletetaan, että ongelman ratkaisija tarvitsee vain 4 satunnaiskoetta. Valitaan siis 4 otoksen kooksi. Tällöin otos on

$$A_{\mathbf{s},\chi,(4)} = (A_{\mathbf{s},\chi,1}, A_{\mathbf{s},\chi,2}, A_{\mathbf{s},\chi,3}, A_{\mathbf{s},\chi,4}).$$

Otetaan esimerkki yhdestä otoksen realisaatiosta. Oletataan, että umpimähkään valitut vektorit $\mathbf{a}_i$ ja niitä vastaavat virheet e_i ovat

$$\begin{aligned} (\mathbf{a}_1, e_1) &= ((1, 0, 0), 1) \\ (\mathbf{a}_2, e_2) &= ((1, 1, 1), 0) \\ (\mathbf{a}_3, e_3) &= ((0, 0, 0), 1) \\ (\mathbf{a}_4, e_4) &= ((0, 1, 0), 0). \end{aligned}$$

Tällöin ongelman ratkaisijan tulisi päätellä salaisuus $\mathbf{s}$ otoksen realisaatiosta

$$\begin{aligned} A_{\mathbf{s},\chi,(4)}((\mathbf{a}_1, e_1), \ldots, (\mathbf{a}_4, e_4)) = (&((1, 0, 0), 0), \\ &((1, 1, 1), 0), \\ &((0, 0, 0), 1), \\ &((0, 1, 0), 0)). \end{aligned}$$

Määritelmä 5.2. Olkoon $\mathrm{LWE}_{n,q,\chi}$ häirityn oppimisen ongelma ja $A_{s,\chi,(k)}$ sen otos. Sanotaan, että algoritmi

$$B\colon \bigcup_{k \in \mathbb{Z}_+} (\mathbb{Z}_q^n \times \mathbb{Z}_q)^k \to \mathbb{Z}_q^n$$

ratkaisee ongelman $\mathrm{LWE}_{n,q,\chi}$, jos jokaisella $k \in \mathbb{Z}_+$ ja $\mathbf{s} \in \mathbb{Z}_q^n$ se palauttaa vektorin $\mathbf{s}$ todennäköisyydellä, joka on eksponentiaalisen lähellä yhtä k:n suhteen eli $B(A_{\mathbf{s},\chi,(k)}) = \mathbf{s}$ vähintään todennäköisyydellä $1 - b^k$, missä $0 \leq b < 1$.

5.2 Häiritty oppiminen renkaissa

Olkoot $f = x^d + 1$ polynomi, missä $d = 2^t$ jollain $t \in \mathbb{Z}_+$. Olkoon $q \geq 2$ kokonaisluku. Olkoot $R = \mathbb{Z}[x]/\langle f \rangle$ ja $R_q = \mathbb{Z}_q[x]/\langle f \rangle$ renkaita. Olkoon $\chi \colon R \to \mathbb{R}$ pistetodennäköisyysfunktio. Olkoot sitten $s \leftarrow R_q$ mielivaltaisesti valittu salaisuus ja $A_{s,\chi} \colon R_q \times R \to R_q \times R_q$ yksittäistä satunnaiskoetta kuvaava satunnaismuuttuja, jolle

$$A_{s,\chi}(a, e) = (a, a \cdot s + e),$$

missä $a \leftarrow R_q$ on valittu umpimähkään ja virhe $e \in R$ pistetodennäköisyysfunktion χ mukaisesti. Kuvataan satunnaiskokeen toistoa k kertaa otoksella. Olkoot $A_{s,\chi,1}, \ldots, A_{s,\chi,k}$ satunnaismuuttujia, jotka ovat jakautuneet kuten $A_{s,\chi}$. Muodostetaan otos riippumattomista satunnaismuuttujista $A_{x,\chi,i}$, jotka ovat jakautuneet kuten $A_{s,\chi}$. Otos on

$$A_{s,\chi,(k)} = (A_{s,\chi,1}, \ldots, A_{s,\chi,k}),$$
$$A_{s,\chi,(k)}(a_1, e_1), \ldots, (a_k, e_k)) = (A_{s,\chi}(a_1, e_1), \ldots, A_{s,\chi}(a_k, e_k)),$$

missä $k \in \mathbb{N}$. Ongelmana on päätellä polynomi s otoksesta $A_{s,\chi,(k)}$. Tätä ongelmaa kutsutaan *häirityksi oppimiseksi renkaissa* ja merkitään R-LWE$_{d,q,\chi}$.

Esimerkki 5.3. Tarkastellaan ongelmaa R-LWE$_{d,q,\chi}$. Olkoot $k = 3$, $q = 2$ ja $d = 8 = 2^3$ parametreja. Tällöin polynomiksi f saadaan $x^8 + 1$ ja renkaat R ja R_q ovat

$$R = \mathbb{Z}[x]/\langle x^8 + 1 \rangle \quad \text{ja}$$
$$R_q = \mathbb{Z}_2[x]/\langle x^8 + 1 \rangle.$$

Tällöin pistetodennäköisyysfunktio χ on $\chi \colon R \to \mathbb{R}$, ja satunnaismuuttuja $A_{\mathbf{s},\chi}$ on $A_{\mathbf{s},\chi} \colon R_q \times R \to R_q \times R_q$. Oletetaan, että valittu salainen polynomi on

$$s = x^3 + x + 1 \in R_q.$$

Valitaan sitten satunnaiskokeen toistojen määräksi 3. Tällöin saadaan otos

$$A_{s,\chi,(3)} = (A_{s,\chi,1}, A_{s,\chi,2}, A_{s,\chi,3}).$$

Oletetaan, että arvotut polynomit ja niitä vastaavat virheet ovat

$$(a_1, e_1) = (x^4, x^2 + 1),$$
$$(a_2, e_2) = (x^5 + x^2, 0) \qquad \text{ja}$$
$$(a_3, e_3) = (x^3 + 1, x^4 + x^2).$$

Tällöin otoksen $A_{s,\chi,(3)}$ realisaatio on

$$\begin{aligned}
A_{s,\chi,(3)}((a_1, e_1), \ldots, (a_3, e_3)) &= ((a_1, a_1 \cdot s + e_1), \\
&\quad (a_2, a_2 \cdot s + e_2), \\
&\quad (a_3, a_3 \cdot s + e_3)) \\
&= ((a_1, (x^7 + x^5 + x^4) + (x^2 + 1)), \\
&\quad (a_2, (x^8 + x^6 + 2x^5 + x^3 + x^2 + \underbrace{1 + 1}_{=0}) + 0), \\
&\quad (a_3, (x^6 + x^4 + x + 1) + (x^4 + x^2))) \\
&= ((a_1, x^7 + x^5 + x^4 + x^2 + 1), \\
&\quad (a_2, x^6 + x^3 + x^2 + 1), \\
&\quad (a_3, x^6 + x^2 + x + 1))
\end{aligned}$$

Ongelma on siis päätellä tästä otoksen realisaatiosta polynomi s.

5.3 Häiritty oppiminen moduleissa

Häirityn oppimisen ongelma ja sen rengasversio ovat hyvin lähellä toisiaan. Ongelmissa käytetään ainoastaan eri renkaita, häirityssä oppimisessa rengasta $\mathbb{Z}$ ja rengasversiossa polynomirengasta $\mathbb{Z}_q[x]$. Toinen poikkeava asia on dimensio, häirityssä oppimisessa dimensio on n, kun taas rengasversiossa vakio 1. Samankaltaisuuksien vuoksi ongelma voidaan määritellä yleisemmin ongelmaksi, joka sisältää molemmat tapaukset. Tätä yleistettyä ongelmaa kutsutaan häirityksi oppimiseksi moduleissa.

Olkoot $n \in \mathbb{Z}_+$ dimensio ja $f = x^d + 1$ polynomi, missä jollain $t \in \mathbb{N}$ pätee $d = 2^t$. Olkoot $q \in \mathbb{Z}$, jolle $q \geq 2$, $R = \mathbb{Z}[x]/\langle f \rangle$ ja $R_q = \mathbb{Z}_q[x]/\langle f \rangle$ renkaita. Olkoon $\chi \colon R \to \mathbb{R}$ pistetodennäköisyysfunktio. Olkoon $\mathbf{s} \leftarrow R_q^n$ mielivaltaisesti valittu salaisuus. Kuvataan yksittäistä satunnaiskoetta satunnaismuuttujalla $A_{\mathbf{s},\chi} \colon R_q^n \times R \to R_q^n \times R_q$, jolle

$$A_{\mathbf{s},\chi}(\mathbf{a}, e) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + e),$$

missä $\mathbf{a} \leftarrow R_q^n$ on valittu umpimähkään ja virhe $e \in R$ pistetodennäköisyysfunktion χ mukaisesti. Muodostetaan otos riippumattomista satunnaismuuttujista $A_{\mathbf{s},\chi,i}$, jotka ovat jakautuneet kuten $A_{\mathbf{s},\chi}$. Olkoon otos

$$\begin{aligned} A_{\mathbf{s},\chi,(k)} &= (A_{\mathbf{s},\chi,1}, \ldots, A_{\mathbf{s},\chi,k}), \\ A_{\mathbf{s},\chi,(k)}((\mathbf{a}_1, e_1), \ldots, (\mathbf{a}_k, e_k)) &= (A_{\mathbf{s},\chi}(\mathbf{a}_1, e_1), \ldots, A_{\mathbf{s},\chi}(\mathbf{a}_k, e_k)), \end{aligned}$$

missä $k \in \mathbb{N}$. Ongelmana on päätellä vektori $\mathbf{s}$ otoksesta $A_{\mathbf{s},\chi,(k)}$. Tätä ongelmaa kutsutaan *häirityksi oppimiseksi moduleissa* ja merkitään M-LWE$_{n,f,q,\chi}$.

Osoitetaan seuraavaksi, että häiritty oppiminen moduuleissa todella sisältää ongelmat LWE$_{n,q,\chi}$ ja R-LWE$_{d,q,\chi}$.

Lause 5.4. M-LWE$_{n,f,q,\chi}$ *on* LWE$_{n,q,\chi}$*, jos valitaan* $d = 1$.

Todistus. Olkoon $d = 1$ ja tarkastellaan ongelmaa M-LWE$_{n,f,q,\chi}$. Tällöin saadaan polynomiksi $f = x^d + 1 = x + 1$. Edelleen saadaan $R = \mathbb{Z}[x]/\langle f \rangle = \mathbb{Z}[x]/\langle x + 1 \rangle$. Tällöin lauseen 3.20 nojalla $R \cong \mathbb{Z}$, joten ongelma on vastaava. □

Lause 5.5. M-LWE$_{n,f,q,\chi}$ *on* R-LWE$_{d,q,\chi}$*, jos valitaan* $n = 1$.

Todistus. Olkoon $n = 1$. Tällöin ongelmat M-LWE$_{n,f,q,\chi}$ ja R-LWE$_{d,q,\chi}$ ovat täysin identtiset, sillä $R_q^n = R_q^1 = R_q$. Täten M-LWE$_{n,f,q,\chi}$ on R-LWE$_{d,q,\chi}$, jos $n = 1$. □

Täten on todistettu, että M-LWE$_{n,f,q,\chi}$ todellakin sisältää molemmat ongelmat LWE$_{n,q,\chi}$ ja R-LWE$_{d,q,\chi}$.

Esimerkki 5.6. Tarkastellaan vielä esimerkkiä ongelmasta M-LWE$_{n,f,q,\chi}$. Olkoot $n = 2$, $f = x^4 + 1$ ja $q = 2$ parametreja. Tällöin tarkasteltavaksi renkaiksi saadaan

$$\begin{aligned} R &= \mathbb{Z}[x]/\langle x^4 + 1 \rangle \quad \text{ja} \\ R_q &= \mathbb{Z}_2[x]/\langle x^4 + 1 \rangle. \end{aligned}$$

Tälllöin pistetodennäköisyysfunktio χ on $\chi \colon R \to \mathbb{R}$, ja yhtä satunnaiskoetta kuvaava satunnaismuuttuja $A_{\mathbf{s},\chi}$ on

$$A_{\mathbf{s},\chi} \colon R_q^n \times R \to R_q^n \times R_q.$$

Olkoon umpimähkään valittu salaisuus $s = (x^3 + x, 0) \in R_q^n$. Oletetaan, että ratkaisija tarvitsee vain 2 satunnaiskoetta. Oletetaan, että umpimähkään valitut vektorit ja niitä vastaavat virheet ovat

$$(\mathbf{a}_1, e_1) = \Big((x+1, x^2), x\Big) \quad \text{ja}$$
$$(\mathbf{a}_2, e_2) = \Big((1, x^3 + x), 0\Big).$$

Tällöin saadaan 2 toiston satunnaiskokeesta otos $A_{\mathbf{s},\chi,(2)} = (A_{\mathbf{s},\chi,1}, A_{\mathbf{s},\chi,2})$, jonka realisaatio edellä määritetyillä arvoilla on

$$\begin{aligned} A_{\mathbf{s},\chi,(2)} = (&((x+1, x^2), (x+1, x^2) \cdot (x^3 + x, 0) + x^2) \\ &((1, x^3 + x), (1, x^3 + x) \cdot (x^3 + x, 0) + 0)) \\ =&(((x+1, x^2), x^3 + x + 1) \\ &((1, x^3 + x), x^3 + x)). \end{aligned}$$

Tästä otoksesta ratkaisijan tulisi päätellä vektori $\mathbf{s}$.

Häirityn oppimisen moduliversiossa salaisuus $\mathbf{s}$ valittiin renkaasta R_q^n. Salaisuus $\mathbf{s}$ voidaan valita myös pistetodennäköisyysfunktion χ mukaisesti eli samasta pistetodennäköisyysfunktiosta kuin virheet e valittiin. Tällöin $s \leftarrow \chi$ eli $\mathbf{s} \in R^n$. Valinnan muuttaminen ei vaikuta ongelman vaikeuteen [ACPS09].

6 CRYSTALS Kyber

CRYSTALS Kyber on avainten kapselointimekanismi (KEM, key encapsulation mechanism). Lyhenne CRYSTALS tulee sanoista "cryptographic suite for algebraic lattices". Kyberin turvallisuus perustuu häirityn oppimisen moduliversion ratkaisemisen vaikeuteen, jonka uskotaan kestävän kvanttilaskentaa.

Vuonna 2016 NIST (National Institute of Standards and Technology) käynnisti hankkeen PQC (post quantum cryptography), jonka tarkoituksena on etsiä kvanttilaskentaa kestäviä salausalgoritmeja. Kyber on yksi hankkeen ehdokkaista ja se on edennyt kolmannen kierroksen ehdokkaaksi.

Kyber jakautuu kahteen osaan. Ensimmäinen osa on epäsymmetrinen salausskeema, jossa määritellään avainten generointi, salaus ja salauksen avaamiseen käytettävät algoritmit. Toinen osa on avainten kapselointimekanismiosa, jossa määritellään epäsymmetrisen salausskeeman avulla avainten generointi, avaimen kapselointi sekä avaimen kapseloinnin avaamisalgoritmit. Tarkastellaan aluksi epäsymmetrisen salausskeeman algoritmeja ja siirrytään sitten avainten kapselointialgoritmeihin.

Tässä luvussa tutustutaan alkuperäiseen Kyberin toimintaan, joka julkaistiin vuonna 2017 [BDK+18]. Tarkastellaan sitten kuinka Kyber on kehittynyt alkuperäisestä muodostaan verrattuna viimeisimpään julkaisuun PQC-hankkeen kolmannen kierroksen ehdokkaana [ABD+21]. Erityisesti tarkastellaan kehityksen vaikutusta Kyberin parametrisointiin. Luvun lähteinä on käytetty artikkeleita [BDK+18], [ABD+21], [HHK17], [Wu15] ja [CETU20].

6.1 Merkinnät

Tässä osiossa tutustutaan myöhemmin tarkastelua helpottaviin merkintöihin sekä määritelmiin.

6.1.1 Polynomit ja vektorit

Olkoot positiivinen kokonaisluku n ja alkuluku q Kyberin parametreja. Parametrien konkreettiset arvot esitetään ja perustellaan vasta alaluvussa Parametrit. Merkitään käytössä olevia renkaita seuraavasti:

$$R = \mathbb{Z}[x]/\langle x^n + 1\rangle \quad \text{ja}$$
$$R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle.$$

Vektoreita, joiden kertoimet ovat edellämainituissa renkaissa merkitään lihavoiduin pienin kirjaimin ja kaikkien vektorien oletetaan olevan sarakevektoreita, jollei toisin mainita.

6.1.2 Todennäköisyys

Käytetään jo esiteltyä merkintää umpimähkään valitsemiselle. Jos A on joukko niin alkio $a \leftarrow A$ on valittu joukosta A umpimähkään. Lisäksi jos X on diskreetti satunnaismuuttuja ja f on sen pistetodennäköisyysfunktio, niin $b \leftarrow f$ tarkoittaa, että alkio b on valittu pistetodennäköisyysfunktion f mukaisesti.

6.1.3 Jatkofunktio

Jatkofunktiot ovat kuvauksia, joiden avulla bittijonoja voidaan kuvata äärellisen mittaisiksi bittijonoiksi. Jatkofunktiot saavat syötteenä siemenen, josta ne muodostavat äärellisen pitkän bittijonon. On tärkeää huomata, että jatkofunktio ei jatka siementä, vaan tuottaa kokonaan uuden bittijonon. Äärellisen mittaisista bittijonoista voidaan eri säännöstöjen mukaan luoda eri joukon alkioita. Esimerkiksi UTF-8 tekstikoodaus on säännöstö miten bittejä tulee tulkita kirjaimina.

Määritelmä 6.1. Kuvaus $\mathrm{XOF}\colon \{0,1\}^* \times \mathbb{Z}_+ \to \{0,1\}^*$ on *jatkofunktio* (extandable output function), jos sille pätevät seuraavat ehdot.

1. Kuva-alkio $a = \mathrm{XOF}(x, n)$ on n bittiä pitkä jokaisella $n \in \mathbb{Z}_+$.
2. Kuva-alkioiden $a = \mathrm{XOF}(x, n)$ ja $b = \mathrm{XOF}(x, m)$ biteille pätee $a_i = b_i$ jokaisella $i \in \{k \in \mathbb{Z}_+ \mid k \leq \min(n, m)\}$.
3. Olkoon $a = \mathrm{XOF}(x, n)$ bittijono. Ei ole olemassa polynomiaikaista algoritmia A, jolle pätee $x := A(a)$, missä A voi käyttää jatkofunktiota XOF oraakkelina.

Lisäksi bittijonoa x kutsutaan jatkofunktion XOF *siemeneksi*.

Huomautus. Jatkofunktioilta vaaditaan myös hajautusfunktioiden tapaan törmäyskestävyys sekä toisenalkukuvan kestävyys. Kertauksena törmäyskestävyys edellyttää, että täytyy olla vaikeaa löytää alkiot $x \neq x'$, joille pätee $\mathrm{XOF}(x, n) = \mathrm{XOF}(x', n)$, kaikilla $n \in \mathbb{Z}_+$. Toisenalkukuvan kestävyys edellyttää, jos $a = \mathrm{XOF}(x, n)$, niin täytyy olla laskennallisesti vaikeaa löytää alkio $x \neq x'$, jolla pätee $a = \mathrm{XOF}(x', n)$.

Huomataan, että jos $n \in \mathbb{Z}_+$ ja siemen $x \in \{0,1\}^*$ on umpimähkään valittu, niin myös $a = \mathrm{XOF}(x, n)$ on satunnainen. Tällöin bittijonosta a voidaan muodostaa umpimähkään valittuja arvoja eri joukoista. Jos valitaan arvo $z \in S$ satunnaisesti, merkitään $z \sim S := \mathrm{XOF}(x, n)$. Lisäksi, koska haluttujen bittien määrä n riippuu vain joukosta, jonka alkio halutaan muodostaa, niin voidaan merkitä lyhyemmin $z \sim S := \mathrm{XOF}(x)$. Palautetaan mieleen, että keskitetyn binomijakauman $\mathbb{B}_\eta$ mukaisesti jakautuneita arvoja voidaan valita käyttämällä kahta η pituista umpimähkään valittua bittijonoa. Jos jatkofunktiosta muodostetaan keskitetyn binomijakauman mukainen alkio, niin merkitään $z \sim \mathbb{B}_\eta \leftsquigarrow \mathrm{XOF}(x)$.

6.1.4 Alkioiden pituudet

Olkoot $q \in \mathbb{N}$ ja $t \in \mathbb{Z}_q$. Tällöin merkitään

$$\|t\| = |t \bmod^{\pm} q|.$$

Olkoon $v = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1} \in R$. Tällöin merkitään

$$\|v\|_\infty = \max_{i=0}^{n-1} \|v_i\| \quad \text{ja}$$
$$\|v\| = \sqrt{\|v_0\|^2 + \cdots + \|v_{n-1}\|^2}.$$

Olkoot $k \in \mathbb{Z}_+$ ja $\mathbf{v} = (v_1, v_2, \ldots, v_k) \in R^k$. Tällöin vastaavasti merkitään

$$\|\mathbf{v}\|_\infty = \max_{i=1}^{k} \|v_i\|_\infty \quad \text{ja}$$
$$\|\mathbf{v}\| = \sqrt{\|v_1\|^2 + \cdots + \|v_k\|^2}.$$

6.1.5 Pyöristäminen

Olkoon $x \in \mathbb{R}$. Tällöin merkintä $\lceil x \rfloor$, tarkoittaa pyöristystä lähimpään kokonaislukuun. Jos kaksi kokonaislukua on yhtä lähellä lukua x, $\lceil x \rfloor$ on niistä isompi. Käytetään myös tavanomaista kattomerkintää $\lceil x \rceil$, joka pyöristää luvun seuraavaan kokonaislukuun.

6.1.6 Pakkaus ja purku

Pakkaamisen avulla voidaan pienentää julkisen avaimen ja salatun tekstin kokoa, eikä sillä ole suurta vaikutusta todennäköisyyteen, jolla avaaminen epäonnistuu.

Määritellään seuraavaksi kuvaukset Compress_q ja Decompress. Kuvaus Compress_q saa argumenttina luvun $x \in \mathbb{Z}_q$ ja palauttaa luvun $a \in \{0, \ldots, 2^d - 1\}$, missä $d < \lceil \log_2(q) \rceil$. Kuvaus Decompress_q saa argumenttina Compress_q -kuvauksen arvon

$$\text{Decompress}_q\left(\text{Compress}_q(x, d), d\right) = x',$$

missä arvolle x' pätee

$$|x' - x \bmod^{\pm} q| \leq \left\lceil \frac{q}{2^{d+1}} \right\rfloor.$$

Seuraavat funktiot täyttävät nämä vaatimukset:

$$\text{Compress}_q(x, d) = \left\lceil \frac{2^d}{q} \cdot x \right\rfloor \bmod^{+} 2^d \quad \text{ja}$$
$$\text{Decompress}_q(x, d) = \left\lceil \frac{q}{2^d} \cdot x \right\rfloor.$$

Pakkaus- ja purkukuvauksia voidaan myös soveltaa alkioihin $x \in R_q$ ja $\mathbf{y} \in R_q^k$, mutta tällöin kuvausta käytetään jokaiseen kertoimeen ja koordinaattiin erikseen, tätä merkitään $\text{Compress}_q(x, d)$ tai $\text{Compress}_q(\mathbf{y}, d)$.

6.2 Epäsymmetrinen salausskeema

Määritelmä 6.2. *Epäsymmetrinen salausskeema*

$$\text{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec}),$$

on kolmikko satunnaisalgoritmeja viestiavaruuden $\mathcal{M}$ kera. Algoritmi KeyGen palauttaa avainparin (pk, sk). Salausalgoritmi Enc saa parametreina julkisen avaimen pk ja viestin $m \in \mathcal{M}$ ja palauttaa salatun viestin m_{pk}. Salauksen avaus -algoritmi Dec on deterministinen ja saa parametreina yksityisen avaimen sk ja salatun viestin m_{pk} ja palauttaa viestin $m' \in \mathcal{M}$. On tärkeää huomata, että on mahdollista, että viestit m ja m' eroavat toisistaan. Tällöin sanotaan, että salatun viestin avaaminen on epäonnistunut.

Määritelmä 6.3. Olkoon PKE epäsymmetrinen salausskeema. Kolmikon PKE sanotaan olevan $(1 - \delta)$-*oikein*, jos odotusarvo

$$\mathbb{E}\left(\max_{m \in \mathcal{M}} \mathbb{P}\left\{\text{Dec}\big(\text{sk}, \text{Enc}\,(\text{pk}, m)\big) = m\right\}\right) \geq 1 - \delta,$$

missä satunnaismuuttujapari (pk, sk) on algoritmin KeyGen tuottama avainpari. Toisin sanoen halutaan, että viestin salaaminen ja sen jälkeen avaaminen tuottaa saman viestin todennäköisyydellä $1 - \delta$ kullakin viestillä.

6.2.1 Valitun viestin hyökkäys

Valitun viestin hyökkäys (chosen plaintext attack, CPA) on hyökkäys, jolla pyritään saamaan informaatiota, joka heikentää epäsymmetristä salauskeemaa. Hyökkäyksessä oletetaan, että hyökkääjä saa julkisen avaimen pk ja valitsee saman pituiset viestit m_0 ja m_1. Lisäksi hyökkääjä voi käyttää salausalgoritmia Enc oraakkelina. Hyökkääjä saa salatun viestin $\text{Enc}(\text{pk}, m)$, missä $m \leftarrow \{m_0, m_1\}$ on valittu umpimähkään. Hyökkääjän tarkoitus on tunnistaa onko salattu viesti muodostettu salaamalla m_0 vai m_1. Tällöin hyökkääjä pystyy tunnistamaan tietyllä avaimella salatut viestit.

Olkoot PKE = (KeyGen, Enc, Dec) epäsymmetrinen salausskeema, pk julkinen avain ja m_0, m_1 saman pituisia salaamattomia viestejä. Olkoon $b \leftarrow \{0, 1\}$ umpimähkään valittu bitti ja $c = \text{Enc}(\text{pk}, m_b)$ bittiä vastaava salattu viesti. Epäsymmetrinen salausskeema PKE on CPA-*turvallinen*, jos ei ole olemassa polynomiaikaista algoritmia A, jolle pätee $b' := A(m_0, m_1, \text{pk}, c)$, missä $b = b'$ ja algoritmi A voi käyttää salausalgoritmia Enc oraakkelina.

Toisin sanoen jos PKE on CPA-turvallinen, niin kaikki salatut viestit näyttävät samalta sivustakatselijalle. On tärkeää huomata, että salausalgoritmin Enc täytyy olla satunnainen. Sillä muuten algoritmi A pystyisi salaamaan molemmat viestit m_0 ja m_1 ja vertaamaan kumpi salatuista viesteistä on c. Satunnaisuuden avulla saman viestin salaaminen useaan kertaan tuottaa eri salatun viestin.

6.2.2 Kyber PKE-algoritmit

Kyber koostuu kahdesta eri osasata. Ensimmäinen on epäsymmetrinen salausskeema, joka salaa viestejä. Epäsymmetrinen salausskeema Kyber.PKE on kolmikko algoritmeja (KeyGen, Enc, Dec). Algoritmi KeyGen vastaa epäsymmetrisen avainparin generoinnista, Enc 32-tavuisten viestien salaamisesta ja Dec salattujen viestin avaamisesta. Koska Kyberin algoritmit ovat satunnaisia, niin on mahdollista, että salatun viestin avaaminen ei aina tuota oikeaa lopputulosta.

Määritellään seuraavaksi Kyberin epäsymmetrinen salausskeema. Epäsymmetrinen salausskeema on parametrisoitu positiivisilla kokonaisluvuilla n, k, q, η, d_t, d_u ja d_v.

Tällöin epäsymmetrinen salausskeema Kyber.CPA = (KeyGen, Enc, Dec), missä algoritmit ovat määritelty kuten 6.1, 6.2 ja 6.3. Todetaan vielä, että Kyber.CPA on CPA-turvallinen, mutta turvallisuustodistusta ei tässä esitetä [BDK+18].

Avainten generoiminen

Tarkastellaan ensin avaintengenerointialgoritmia. Sen tarkoituksena on tuottaa avainpari (sk, pk), missä sk on yksityinen avain ja pk on julkinen avain.

Algoritmi 6.1 Kyber.CPA.KeyGen: Avainten generoiminen

1: $\rho, \sigma \leftarrow \{0, 1\}^{256}$
2: $\mathbf{A} \sim R_q^{k \times k} := \text{XOF}(\rho)$
3: $(\mathbf{s}, \mathbf{e}) \sim \mathbb{B}_\eta^k \times \mathbb{B}_\eta^k \leftsquigarrow \text{XOF}(\sigma)$
4: $\mathbf{t} = \text{Compress}_q(\mathbf{As} + \mathbf{e}, d_t)$
5: **return** $(\text{pk} = (\mathbf{t}, \rho), \text{sk} = \mathbf{s})$

Vaiheessa 1 muuttujiin ρ ja σ sijoitetaan umpimähkään valittuja bittejä. Vaiheessa 2 umpimähkään valittua bittijonoa jatketaan jatkofunktiolla XOF ja saatavasta bittijonosta muodostetaan joukon $R_q^{k \times k}$ alkio. Matriisin $\mathbf{A}$ alkioina on polynomeja. Vaiheessa 3 jatketaan umpimähkään

valittua bittijonoa jatkofunktiolla XOF ja mudostetaan kaksi polynomivektoria $\mathbb{B}_n^k$ mukaisesti. Vaiheessa 4 pakataan tulos $\mathbf{As} + \mathbf{e}$.

Viestin salaaminen

Tarkastellaan sitten viestien salausalgoritmia. Viestin salaamiseen tarvitaan salattava viesti $m \in \mathcal{M}$ ja julkinen avain pk.

Algoritmi 6.2 Kyber.CPA.Enc: Viestin salaaminen

Syöte: pk $= (\mathbf{t}, \rho), m \in \mathcal{M}$

1: $r \leftarrow \{0,1\}^{256}$
2: $\mathbf{t} = \text{Decompress}_q(\mathbf{t}, d_t)$
3: $\mathbf{A} \sim R_q^{k \times k} := \text{XOF}(\rho)$
4: $(\mathbf{r}, \mathbf{e_1}, e_2) \sim \mathbb{B}_\eta^k \times \mathbb{B}_\eta^k \times \mathbb{B}_\eta \leftsquigarrow \text{XOF}(r)$
5: $\mathbf{u} = \text{Compress}_q(\mathbf{A}^\text{T}\mathbf{r} + \mathbf{e_1}, d_u)$
6: $v = \text{Compress}_q(\mathbf{t}^\text{T}\mathbf{r} + e_2 + \lceil \frac{q}{2} \rfloor \cdot m, d_v)$
7: **return** $c = (\mathbf{u}, v)$

Vaiheessa 1 sijoitetaan muuttujaan r umpimähkään valittuja bittejä. Sitten puretaan julkisen avaimen pakattu arvo $\mathbf{t}$. Tämän jälkeen generoidaan matriisi $\mathbf{A}$ samoin kuin algoritmissa 6.1. Vaiheessa 4 muodostetaan polynomivektorit r, e_1 ja polynomi e_2, jakauman $\mathbb{B}_\eta^k$ mukaisesti. Tämän jälkeen pakataan tulokset ja palautetaan salattu viesti c.

Viestin avaaminen

Viestin avaamiseen tarvitaan yksityinen avain sk, sekä salattu viesti c. Avaaminen tehdään purkamalla pakattu salattu viesti c ja sitten käyttämällä yksityistä avainta sen avaamiseen. On tärkeä muistaa, että avaus-algoritmi ei välttämättä tuota alkuperäistä viestiä, joka salattiin.

Algoritmi 6.3 Kyber.CPA.Dec: Salatun viestin avaaminen

Syöte: sk $= \mathbf{s}, c = (\mathbf{u}, v)$

1: $\mathbf{u} = \text{Decompress}_q(\mathbf{u}, d_u)$
2: $v = \text{Decompress}_q(v, d_v)$
3: **return** $\text{Compress}_q(v - \mathbf{s}^\text{T}\mathbf{u}, 1)$

6.2.3 Oikeellisuus

Todistetaan seuraavaksi algoritmien 6.1, 6.2 ja 6.3 oikeellisuus. Oikeellisuudella tarkoitetaan sitä, että salauksen avaaminen onnistuu riittävällä todennäköisyydellä. Oikeellisuutta säädetään Kyberin parametrien avulla ja se onkin yksi tärkeimmistä syistä niiden valitsemiselle.

Lause 6.4. *Olkoon k positiivinen kokonaisluku. Olkoot $\mathbf{s}, \mathbf{e}, \mathbf{r}, \mathbf{e}_1, e_2$ kuten algoritmeissa 6.2 ja 6.1. Olkoot*

$$\begin{aligned} \mathbf{c}_t &\leftarrow \psi_{d_t}^k, \\ \mathbf{c}_u &\leftarrow \psi_{d_u}^k \quad \text{ja} \\ \mathbf{c}_v &\leftarrow \psi_{d_v} \end{aligned}$$

satunnaismuuttujia, missä ψ_d^k on seuraavan algoritmin palauttaman satunnaismuuttujan jakauma:

1: $\mathbf{y} \leftarrow R^k$
2: **return** $(\mathbf{y} - \text{Decompress}_q(\text{Compress}_q(\mathbf{y}, d), d)) \bmod^{\pm} q$.

Tällöin Kyber.CPA *on* $(1-\delta)$*-oikein, missä*

$$\delta = \mathbb{P}\left\{\left\|\mathbf{e}^{\mathrm{T}}\mathbf{r} + e_2 + \mathbf{c}_v - \mathbf{s}^{\mathrm{T}}\mathbf{e}_1 + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{c}_u\right\|_\infty \geq \left\lceil \frac{q}{4} \right\rfloor\right\}.$$

Todistus. Täytyy siis todistaa, että

$$\begin{aligned}
&\mathbb{E}\left(\max_{m\in\mathcal{M}} \mathbb{P}\Big\{\text{Dec}\big(\text{sk}, \text{Enc}\,(\text{pk}, m)\big) = m\Big\}\right) \\
&\geq 1 - \mathbb{P}\left\{\left\|\mathbf{e}^{\mathrm{T}}\mathbf{r} + e_2 + \mathbf{c}_v - \mathbf{s}^{\mathrm{T}}\mathbf{e}_1 + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{c}_u\right\|_\infty \geq \left\lceil \frac{q}{4} \right\rfloor\right\}.
\end{aligned}$$

Tarkastellaan ensin algoritmeja ja valitaan niistä tarvittavia arvoja. Algoritmissa 6.2 rivillä 6 muuttujan $\mathbf{t}$ arvo on

$$\begin{aligned}
\mathbf{t} &= \text{Decompress}_q(\text{Compress}_q(\mathbf{As} + \mathbf{e}, d_t), d_t) \\
&= \mathbf{As} + \mathbf{e} + \mathbf{c}_t,
\end{aligned}$$

jollain $\mathbf{c}_t \in R^k$. Algoritmissa 6.3 muuttujien $\mathbf{u}$ ja v arvot ovat

$$\begin{aligned}
\mathbf{u} &= \text{Decompress}_q(\text{Compress}_q(\mathbf{A}^{\mathrm{T}}\mathbf{r} + \mathbf{e}_1, d_u), d_u) \\
&= \mathbf{A}^{\mathrm{T}}\mathbf{r} + \mathbf{e}_1 + \mathbf{c}_u,
\end{aligned}$$

jollain $\mathbf{c}_u \in R^k$ ja

$$\begin{aligned}
v &= \text{Decompress}_q(\text{Compress}_q(\mathbf{t}^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m, d_v), d_v) \\
&= \mathbf{t}^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v \\
&= (\mathbf{As} + \mathbf{e} + \mathbf{c}_t)^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v \\
&= (\mathbf{As} + \mathbf{e})^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v + \mathbf{c}_t^{\mathrm{T}}\mathbf{r},
\end{aligned}$$

jollain $c_v \in R$. Huomataan, että jokainen algoritmeissa valittu alkio $\mathbf{c}_t$, $\mathbf{c}_u$ ja c_v on muotoa

$$(\mathbf{y} - \text{Decompress}_q(\text{Compress}_q(\mathbf{y}, d), d)) \bmod^{\pm} q,$$

missä $\mathbf{y} \leftarrow R^k$. Tällöin jokainen $\mathbf{c}_t$, $\mathbf{c}_u$ ja c_v on jakautunut jakauman ψ, ja

$$\begin{aligned}
v - \mathbf{s}^{\mathrm{T}}\mathbf{u} &= (\mathbf{As} + \mathbf{e})^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\mathbf{r} + \mathbf{e}_1 + \mathbf{c}_u) \\
&= \mathbf{A}^{\mathrm{T}}\mathbf{s}^{\mathrm{T}}\mathbf{r} + \mathbf{e}^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{e}_1 - \mathbf{s}^{\mathrm{T}}\mathbf{c}_u \\
&= \mathbf{e}^{\mathrm{T}}\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m + c_v + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{e}_1 - \mathbf{s}^{\mathrm{T}}\mathbf{c}_u.
\end{aligned}$$

Jos

$$\left\|\mathbf{e}^{\mathrm{T}}\mathbf{r} + e_2 + c_v + \mathbf{c}_t^{\mathrm{T}}\mathbf{r} - \mathbf{s}^{\mathrm{T}}\mathbf{e}_1 - \mathbf{s}^{\mathrm{T}}\mathbf{c}_u\right\|_\infty < \left\lceil \frac{q}{4} \right\rfloor,$$

niin $v - \mathbf{s}^{\mathrm{T}}\mathbf{u} = w + \lceil \frac{q}{2} \rfloor \cdot m$, missä $\|w\|_\infty < \lceil \frac{q}{4} \rfloor$. Olkoon $m' = \mathrm{Compress}_q(v - \mathbf{s}^{\mathrm{T}}\mathbf{u}, 1)$. Tällöin

$$\begin{aligned}\left\lceil \frac{q}{4} \right\rfloor &\geq \left\| v - \mathbf{s}^{\mathrm{T}}\mathbf{u} - \left\lceil \frac{q}{2} \right\rfloor \cdot m' \right\|_\infty \\ &= \left\| w + \left\lceil \frac{q}{2} \right\rfloor \cdot m - \left\lceil \frac{q}{2} \right\rfloor \cdot m' \right\|_\infty \\ &= \left\| w + \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') \right\|_\infty .\end{aligned}$$

Tällöin kolmioepäyhtälön muodosta

$$\|a\| - \|b\| \leq \|a + b\| ,$$

ja tuloksista

$$\begin{aligned}\|w\|_\infty &< \left\lceil \frac{q}{4} \right\rfloor \quad \text{ja} \\ \left\| w + \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') \right\|_\infty &\leq \left\lceil \frac{q}{4} \right\rfloor\end{aligned}$$

saadaan

$$\begin{aligned}&\left\| \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') \right\| - \|w\|_\infty \leq \left\| \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') + w \right\|_\infty \\ \iff &\left\| \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') \right\| \\ \leq \quad &\left\| w + \left\lceil \frac{q}{2} \right\rfloor \cdot (m - m') \right\|_\infty + \|w\|_\infty \leq \left\lceil \frac{q}{4} \right\rfloor + \|w\|_\infty < \left\lceil \frac{q}{4} \right\rfloor + \left\lceil \frac{q}{4} \right\rfloor < 2 \cdot \left\lceil \frac{q}{4} \right\rfloor .\end{aligned}$$

Tästä seuraa, että jokaisella parittomalla parametrilla q pätee $m = m'$. Koska parametri q on alkuluku, niin se on myös pariton. Täten Kyber.CPA on $(1 - \delta)$- oikein. □

6.3 Avainten kapselointiskeema

Määritelmä 6.5. *Avainten kapselointiskeema*

$$\mathrm{KEM} = (\mathrm{KeyGen}, \mathrm{Encaps}, \mathrm{Decaps})$$

on kolmikko todennäköisyysalgoritmeja, avainavaruuden $\mathcal{K}$ kera. Algoritmi KeyGen palauttaa avainparin (pk, sk). Kapselointialgoritmi Encaps saa argumenttina julkisen avaimen pk ja tuottaa avaimen $k \in \mathcal{K}$ ja salatun viestin c. Kapseloinnin avaaminen on algoritmi Decaps ja saa argumenttina yksityisen avaimen sk ja salatun viestin c ja tuottaa joko avaimen $k \in \mathcal{K}$ tai symbolin $\perp$, joka merkitsee salauksen avauksen epäonnistumista.

Määritelmä 6.6. Olkoon KEM avainten kapselointiskeema. Kolmikon KEM sanotaan olevan $(1 - \delta)$-*oikein*, jos odotusarvo

$$\mathbb{P}\left\{ \mathrm{Decaps}(\mathrm{sk}, c) = k \;\middle|\; \mathrm{Encaps}(\mathrm{pk}) = (c, k) \;\middle|\; (\mathrm{pk}, \mathrm{sk}) \right\} \geq 1 - \delta,$$

missä satunnaismuuttujapari (pk, sk) on algoritmin KeyGen tuottama avainpari.

6.3.1 Kyber KEM-algoritmit

Tarkastellaan sitten Kyberin avainten kapselointiskeemaa Kyber ja siihen liittyviä algoritmeja. Olkoot $G\colon \{0,1\}^* \to \{0,1\}^{256} \times \{0,1\}^{256}$ ja $H\colon \{0,1\}^* \to \{0,1\}^{256}$ kryptografisia hajautusfunktioita, k, d_t, d_u, d_v positiivisia kokonaislukuja ja $\mathcal{M} = \{0,1\}^n$ viestiavaruus. Tällöin avainten kapselointiskeema Kyber on kolmikko (KeyGen, Encaps, Decaps), missä KeyGen on algoritmi 6.4, Encaps on algoritmi 6.5 ja Decaps on algoritmi 6.6.

Avainten kapselointiskeeman algoritmit perustuvat Fujisaki-Okamoto transformaatioon. Fujisaki-Okamoto transformaatio muuntaa minkätahansa epäsymmetrisen salauskeeman avainten kapselointiskeemaksi [HHK17]. Kyberin epäsymmetrisestä salausskeema muunnetaan lähes suoraan Fujisaki-Okamoto transformaation mukaan. Ainoana erona on hajatusfunktion H käyttö myös salattuun viestiin c. Ei perehdytä Fujisaki-Okamoto transformaatioon tämän enempää.

Avainten generointi

Tarkastellaan Kyberin avainten generointia. Algoritmi Kyber.KeyGen muistuttaa selvästi PKE avainten generointia. Itseasiassa se on identtinen, mutta yksityiseen avaimeen sk on lisätty julkinen avain pk sekä umpimähkään valittu 256:n bitin pituinen bittijono z.

Algoritmi 6.4 Kyber.KeyGen: Avainten generointi

1: $\rho, \sigma \leftarrow \{0,1\}^{256}$
2: $\mathbf{A} \sim R_q^{k \times k} := \mathrm{XOF}(\rho)$
3: $(\mathbf{s}, \mathbf{e}) \sim \mathbb{B}_\eta^k \times \mathbb{B}_\eta^k \leftsquigarrow \mathrm{XOF}(\sigma)$
4: $\mathbf{t} = \mathrm{Compress}_q(\mathbf{As} + \mathbf{e}, d_t)$
5: $z \leftarrow \{0,1\}^{256}$
6: **return** $(\mathrm{pk} = (\mathbf{t}, \rho), \mathrm{sk} = (\mathbf{s}, z, \mathbf{t}, \rho))$

Avaimen kapselointi

Algoritmi 6.5 kuvaa avaimen kapseloinnin. Algoritmi palauttaa symmetrisen avaimen K sekä salatun viestin c, josta avain K on mahdollista päätellä. Olkoot G ja H kryptografisia hajautusfunktioita.

Algoritmi 6.5 Kyber.Encaps: Avaimen kapselointi

Syöte: $\mathrm{pk} = (\mathbf{t}, \rho)$
1: $m \leftarrow \{0,1\}^{256}$
2: $(\hat{K}, r) := G(H(pk), m)$
3: $(\mathbf{u}, v) := \mathrm{Kyber.CPA.Enc}((\mathbf{t}, \rho), m; r)$
4: $c := (\mathbf{u}, v)$
5: $K := H(\hat{K}, H(c))$
6: **return** (c, K)

Algoritmin parametrina on julkinen avain pk ja se palauttaa salatun viestin c ja avaimen K. Ensin asetetaan umpimähkään valittu bittijono muuttujaan m. Sitten käytetään hajautusfunktioita G ja H, minkä jälkeen salataan m Kyber.CPA.Enc algoritmilla, jossa satunnaisuus otetaan muuttujasta r. Salattu m on salattu viesti, josta yhteinen avain K voidaan päätellä.

Avaimen kapseloinnin avaaminen

Algoritmi 6.6 kuvaa avaimen kapseloinnin avaamista, jonka tarkoituksena on palauttaa symmetrinen avain K, joka avainten kapseloinnissa kapseloitiin.

Algoritmi 6.6 Kyber.Dec: Avaimen kapseloinnin avaaminen

Syöte: sk = $(\mathbf{s}, z, \mathbf{t}, \rho), c = (\mathbf{u}, v)$

1: $m' = \text{Kyber.CPA.Dec}(s, (\mathbf{u}, v))$
2: $(\hat{K}', r') := G(H(\text{sk}), m')$
3: $(\mathbf{u}', v') := \text{Kyber.CPA.Enc}((\mathbf{t}, \rho), m'; r')$
4: **if** $(\mathbf{u}', v') = (\mathbf{u}, v)$ **then**
5: **return** $K := H(\hat{K}', H(c))$
6: **else**
7: **return** $K := H(z, H(c))$
8: **end if**

Alussa yritetään saada selville kapseloinnissa luotu m avaamalla salatun viestin c sisältö. Sitten salataan oletettu viesti uudelleen julkisella avaimella, kuten avainten kapseloinnissakin ja verrataan tuloksia. Jos saatu salattu viesti on sama kuin parametrina saatu, saadaan selville avain K, muuten palautetaan pseudosatunnainen avain K.

On tärkeää huomata, että Decaps ei voi epäonnistua missään tilanteessa. Jos algoritmin rivillä 4 huomataan, että salatun viestin avaaminen on epäonnistunut, niin se palauttaa pseudosatunnaisen avaimen $K := H(z, c)$, missä z on avainten generoinnissa yksityiseen avaimeen sisällytetty umpimähkään valittu bittijono. Tämä pseudosatunnainen avain vaikuttaa oikeanlaiselta arvolta, mutta ei sitä tietenkään ole.

6.3.2 Oikeellisuus

Artikkelissa [HHK17] todettiin että, Fujisaki-Okamoto -transformaation käyttäminen Kyberin kaltaiseen $(1 - \delta)$-oikeaan salauskeemaan tuottaa myös $(1 - \delta)$-oikean avaintenkapselointiskeeman. Todistukseen ei perehdytä tämän enempää. Täten, jos Kyber.CPA on $(1 - \delta)$-oikein ja hajautusfunktio G on satunnainen oraakkeli, niin myös avaintenkapselointimekanismi Kyber on $(1 - \delta)$-oikein.

6.4 Parametrit

Tässä osiossa tarkastellaan Kyberin parametrien konkreettisten arvojen valintoja ja valintojen syitä. Kyber on kuitenkin kehittynyt sen alkuperäisestä versiosta, joten tutustutaan ensin alkuperäisiin parametreihin. Kun on perusteltu alkuperäisten parametrien syyt, tarkastellaan, kuinka Kyber on kehittynyt ja kehityksen vaikutuksia parametreihin.

6.4.1 Alkuperäiset parametrit

Kaikkien parametrien valinnat ovat koottuna taulukossa 6.1.

	n	k	q	η	(d_u, d_v, d_t)	δ
Kyber	256	3	7681	4	$(11, 3, 11)$	2^{-142}

Taulukko 6.1: Alkuperäisen Kyberin parametrit

Ensin valittiin $n = 256$, koska halutaan kapseloida 256 bittiä entropiaa, jotka voidaan sitten muuntaa polynomien kertoimiksi. Entropialla tarkoitetaan epäjärjestyksen määrää, jota tietokone kerää esimerkiksi hiiren liikkeistä. Seuraavaksi valittiin $q = 7681$, joka on pienin alkuluku, jolle pätee

$$q \equiv 1 \pmod{2n}.$$

Arvon q valinnan ansiosta toteutuksessa voidaan käyttää NTT-muutosta (negacyclic number-theoretic transform) polynomien kertolaskun nopeuttamiseen renkaassa R_q. Seuraavaksi valittiin $k = 3$. Parametrilla k kontrolloidaan matriisin kokoa ja samalla algoritmin turvallisuutta.

Loput parametrit η, d_u, d_v ja d_t valittiin tasapainottelemalla turvallisuuden, viestin avauksen epäonnistumisen δ, julkisen avaimen koon sekä salatun viestin koon välillä.

Kaikki käytettävät jatko- ja hajautusfunktiot on rakennettu Keccak-algoritmin päälle, joka standardisoitiin vuonna 2015 [Fib15]. Matriisin $\mathbf{A}$ generointiin käytettävä jatkofunktio käyttää SHAKE-128 algoritmia, virhepolynomien generointiin käytettävä jatkofunktio käyttää SHAKE-256 algoritmia, ja hajautusfunktiot H ja G on toteutettu SHA3-256 ja SHA3-512 algoritmeina.

6.4.2 Kyberin parametrit

Viimeisin Kyberin julkaisu julkaistiin PQC-hankkeen kolmannen kierroksen ehdokkaana elokuussa 2021 [ABD+21]. Alkuperäisiin parameterihin verrattuna muutoksia on tullut reilusti. Taulukossa 6.2 on esitelty Kyberin parametrit.

	n	k	q	η_1	η_2	(d_u, d_v)	δ
Kyber-512	256	2	3329	3	2	$(10, 4)$	2^{-139}
Kyber-768	256	3	3329	2	2	$(10, 4)$	2^{-164}
Kyber-1024	256	4	3329	2	2	$(11, 5)$	2^{-174}

Taulukko 6.2: Kyberin parametrit

Ensiksi pienennettiin parametria q alkuperäisestä $q = 7681$ arvoon $q = 3329$. Tällä muutoksella saadaan pienennettyä julkisen avaimen sekä salatun viestin kokoa, mikä nopeuttaa niiden siirtämistä. Sitten julkisen avaimen $\text{pk} = (\mathbf{t}, \rho)$ arvon $\mathbf{t}$ pakkaus poistettiin, sillä Kyberin turvallisuustodistukset perustuvat itseasiassa muotoiluun, jossa julkista avainta ei pakata. Täten turvallisuustodistukset eivät välttämättä päteneet alkuperäiselle versiolle [AASA+15]. Lisäksi pakkauksen poisto tietenkin suurentaa julkista avainta. Keskitetyn binomijakauman parametria η, josta häirityn oppimisen virheet valitaan, pienennettiin, jotta alkioiden valitseminen sen mukaisesti olisi nopeampaa. Lisäksi monia alkioita lähetetäänkin NTT-muodossa, jotta vältytään turhilta muunnoksilta ja polynomien kertolasku on vieläkin nopeampaa.

Kokonaisuudessaan, koska julkisen avaimen pakkaus poistettiin, voidaan hävittää parametri d_t. Keskitetylle binomijakaumalle saadaan nyt kaksi eri parametria η_1 ja η_2, sillä vain virheen parametria muutettiin.

7 CRYSTALS Kyberin suorituskyky

Salauksessa suorituskyky on erityisen tärkeää. Toteutusten täytyy olla tarpeeksi suorituskykyisiä tai käyttäjille se näkyy viiveenä. Tässä luvussa tarkastellaan Kyberin suorityskykyä. Tarkastellaan kaikkien sekä Kyber.PKE:issä, että Kyber.KEM:issä käytettävien algoritmien suorituskykyä. Suorituskykymittauksen tarkoituksena on todentaa, että Kyberin suorituskyky on tarpeeksi hyvä, jotta sitä on realistista käyttää. Kyberin toteutuksena on käytetty NIST projektiin liitettyä referenssitoteutusta [ref21]. Referenssitoteutuksessa käytetään viimeisimpiä parametreja.

7.1 Asettelu

Prosessorin sykli vastaa pienintä mahdollista operaatiota, jonka prosessori suorittaa. Vauhtia, jolla prosessori suorittaa syklejä kutsutaan kellotaajuudeksi. Nykyaikaisten tietokoneiden kellotaajuudet ovat useimmiten 2 - 4GHz välillä eli ne suorittavat 2-4 miljardia sykliä sekunnissa.

Prosessorille annettavia operaatiota kutsutaan käskyiksi (instruction). Käskyt ovat esimerkiksi “tallenna luku muistiin“ tai “lisää kaksi lukua yhteen“. Käskyjä on kuitenkin suuri määrä, sillä prosessorin käyttö tapahtuu näitä käskyjä käyttämällä. Yhden käskyn suorittaminen voi viedä yhden tai satoja syklejä riippuen käskyn vaativuudesta.

Toteutuksena käytetään Kyberin referenssitoteutusta, joka on optiomoitu prosessoreille, jotka tukevat avx2-käskyjoukkoa. Suorituskykymitataan käyttämällä x86-prosessorien Time Stamp Counteria (TSC). Time Stamp Counter pitää tarkan luvun jokaisesta prosessorin tekemästä syklistä. On tärkeää huomata, että TSC pitää lukua vain tehdyistä sykleistä, eikä ajasta. Kulutettu aika riippuu prosessorin kellotaajuudesta. Esimerkiksi prosessori, jonka kellotaajuus on 300MHz, suorittaa 300 miljoonaa sykliä sekunnissa, mutta jos prosessorin kellotaajuus onkin 600MHz saman syklimäärän suorittaminen vie vain puoli sekuntia.

7.2 Tulokset

Mittauksessa käytettiin Intel(R) Core(TM) i5-7200U prosessoria, jonka kellotaajuus on 3.1Ghz. Openssl:stä käytettiin versiota 1.1.1K. Suorituskykytestit ovat osana referenssitoteutusta ja jokainen suoritettiin 1000 kertaa.

Algoritmi	Kyber-512	Kyber-768	Kyber-1024
Kyber.PKE.KeyGen	13511	27809	35657
Kyber.PKE.Enc	14034	25725	35507
Kyber.PKE.Dec	1072	1405	1856
Kyber.KEM.KeyGen	25302	36289	50299
Kyber.KEM.Encaps	35454	49033	68323
Kyber.KEM.Decaps	26985	38256	54867

Taulukko 7.1: Algoritmien suorituskyky sykleissä.

Muutetaan taulukon 7.1 syklit mikrosekunneiksi (μs). Prosessorin kellotaajuus on 3.1GHz eli se suorittaa 3 100 000 000 sykliä sekunnissa. Yksi sekunti on 1 000 000 mikrosekuntia, joten yksi sykli vie

$$\frac{1}{3\,100\,000\,000\,1/\mathrm{s}} \approx 3.2258 \cdot 10^{-10}\,\mathrm{s} = 0.00032258\mu\mathrm{s}$$

Algoritmi	Kyber-512	Kyber-768	Kyber-1024
Kyber.PKE.KeyGen	4.358	8.971	11.502
Kyber.PKE.Enc	4.527	8.298	11.454
Kyber.PKE.Dec	0.346	0.453	0.599
Kyber.KEM.KeyGen	8.162	11.706	16.225
Kyber.KEM.Encaps	11.437	15.817	22.040
Kyber.KEM.Decaps	8.705	12.341	17.699

Taulukko 7.2: Algoritmien suorituskyky mikrosekunteina 3.1GHz prosessorilla.

Huomataan, että kun matriisin koko k kasvaa ja samalla turvallisuus paranee, niin algoritmit vievät enemmän aikaa.

Lähteet

[AASA+15] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status report on the first round of the nist post-quantum cryptography standardization process. 2015. URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. *CRYSTALS-Kyber: Algorithm Specifications And Supporting Documentation (version 3.02)*. Submission to round 3 of the NIST post-quantum project, 2021. URL: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. *Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[BDK+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabem Gregor Seiler, and Damien Stehlé. *CRYSTALS -Kyber: a CCA-secure module-lattice-based KEM*. IEEE computer society, 2018.

[BVG12] Zvika Brakerski, Vinod Vaikuntanathan, and Craig Gentry. *Fully homomorphic encryption without bootstrapping*. In Innovations in Theoretical Computer Science, 2012.

[CETU20] Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel M. Tabia, and Dominique Unruh. Relationships between quantum ind-cpa notions. *IACR Cryptol. ePrint Arch.*, 2020:596, 2020.

[CGK98] Isaac L. Chuang, Neil A. Gershenfeld, and Mark Kubinec. Experimental implementation of fast quantum searching. *Physical Review Letters*, 80:3408–3411, 1998.

[Dan09] Quynh Dang. *Randomized Hashing for Digital Signature*. NIST Special Publication 800-106, 2009.

[DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. 439:553–558, 1992.

[Fib15] Sha-3 standard: Permutation-based hash and extendable-output functions. 2015. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. *A modular analysis of the Fujisaki-Okamoto transformation*. IACR Cryptology ePrint Archive report 2017/604, 2017.

[Kes16] Gary C. Kessler. *An Overview of Cryptography (Updated Version)*. Embry-Riddle Aeronautical University, 2016. URL: https://commons.erau.edu/cgi/viewcontent.cgi?article=1137&context=publication.

[LCR+09] Dale Liu, Max Caceres, Tim Robichaux, Dario V. Forte, Eric S. Seagren, Devin L. Ganger, Brad Smith, Wipul Jayawickrama, Christopher Stokes, and Jan Kanclirz. *Next Generation SSH2 Implementation*. Syngress, 2009.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. *On Ideal Lattices and Learning with Errors Over Rings*. Springer Berlin Heidelberg, 2010.

[Mol11] Richard A. Mollin. *Algebraic Number Theory (2nd edition)*. Chapman and Hall/CRC, 2011.

[oST15] National Institute of Standards and Technology. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Function*. National Institute of Standards and Technology, 2015. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[ref21] Crystals kyber reference implementation github, 2021. URL: https://github.com/pq-crystals/kyber.

[Reg09] Oded Regev. *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. Journal of the ACM 56.6, 2009.

[Sim20] Tom Simonite. China stakes its claim to quantum supremacy. 2020. URL: https://www.wired.com/story/china-stakes-claim-quantum-supremacy.

[Tuo96] Pekka Tuominen. *Todennäköisyyslaskenta I*. Limes ry, 1996.

[Wu15] David J. Wu. *Fully Homomorphic Encryption: Cryptography's Holy Grail*. XRDS: Crossroads, The ACM Magazine for Students, 2015.