Tampereen yliopisto

Mikko Eerola

REPPUONGELMA JA MERKLEN–HELLMANIN SALAUS

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Maaliskuu 2024

Tiivistelmä

Mikko Eerola: Reppuongelma ja Merklen–Hellmanin salaus
Kandidaattitutkielma
Tampereen yliopisto
Matematiikan ja tilastollisen data-analyysin kandidaattiohjelma
Maaliskuu 2024

Tutkielmassa perehdytään reppuongelmaan ja sitä hyödyntävään Merklen–Hellmanin salausmenetelmään. Tarkoitus on esitellä tarvittavia pohjatietoja ja rakentaa niiden päälle salausmenetelmä, jota voidaan käyttää tiedon salaamiseen. Esitietoina käydään läpi lukuteorian perustuloksia, erityisesti jaollisuutta sekä kongruenssia. Nämä perustulokset ovat keskiössä koko tutkielman läpi. Lisäksi esitellään Eukleideen algoritmi ja sen laajennus. Laajennettua algoritmia käytetään myöhemmin salausmenetelmän yhteydessä laskemaan käänteisalkioita jäännösluokissa.

Lukija johdatetaan myös kryptografiaan salausmenetelmän yleisen määritelmän kautta. Lisäksi annetaan yksinkertainen historiallinen esimerkki salausmenetelmästä, Caesar-salaus. Yleisesti salausmenetelmä on joko symmetrinen tai epäsymmetrinen eli niin sanottu julkisen avaimen salausmenetelmä. Työssä esiteltävä Merklen–Hellmanin salausmenetelmä on epäsymmetrinen eli tiedon salaamiseen käytetään eri avainta kuin tiedon purkamiseen. Tämä mahdollistaa salausavaimen julkaisemisen, josta sen nimitys tulee. Julkisen salausmenetelmän salausfunktion pitää olla yksisuuntainen, mikä tarkoittaa sitä, että salausfunktio on helppo laskea mutta sen kääntäminen on vaikeaa. Yksisuuntaisten funktioiden erikoistapauksena esitellään salaovifunktiot, joiden kääntäminen onnistuukin helposti jonkin lisätiedon avulla. Tällaiset funktiot ovat keskiössä kryptografiassa. Samalla esitellään myös tässä tutkielmassa käytettäviä merkintöjä, sopimuksia ja muunnoksia, joihin esimerkit perustuvat.

Lopullisen salausmenetelmän määrittelemiseksi perehdytään reppuongelmaan. Reppuongelma on pohjimmiltaan hyvin yleinen optimointiongelma. Esimerkiksi pakattaessa laukkua matkalle on syytä optimoida mukaan otettavien esineiden hyöty yhteistilavuuden pysyessä alle matkalaukun tilavuuden. Reppuongelmasta on monia variaatioita, joista Merklen–Hellmanin salausmenetelmä hyödyntää osajoukko-

ongelmaa. Reppuongelmat ovat NP-täydellisiä ongelmia, mikä tarkoittaa, että ratkaisujen etsiminen laskennallisesti on hyvin raskasta. Tästä johtuen reppuongelmaa voidaan hyödyntää julkisen salausmenetelmän pohjana. Salauksen purku hyödyntää salaovena erikoistilannetta, jossa esineet on valittu erityisesti kasvavasti, eli jokainen esine on suurempi kuin sitä pienemmät esineet yhteensä. Tällaiseen tilanteeseen esitellään hyvin suoraviivainen ratkaisualgoritmi.

Lopuksi esitellään Merklen–Hellmanin salausmenetelmä, jossa yhdistyvät tutkielman esitiedot ja reppuongelma. Luvussa käydään läpi kattava esimerkki Merklen–Hellmanin salausmenetelmän käytöstä yksinkertaisessa tilanteessa.

Avainsanat: kryptografia, lukuteoria, Merklen–Hellmanin salausmenetelmä, reppuongelma

Sisällys

1 Johdanto

Tässä tutkielmassa esitellään monessa käytännön sovelluksessa esiin nouseva reppuongelma ja siihen perustuva kryptografinen salausmenetelmä. Reppuongelmasta pyritään antamaan havainnollistavia ja käytännöllisiä esimerkkejä. Tutkielmassa käydään myös kattava esimerkki esiteltävän salausmenetelmän toiminnasta ja lähetetään salattu viesti, joka puretaan takaisin luettavaksi.

Kryptografia perustuu pitkälti lukuteoriaan ja algebraan, joten luvussa 2 annetaan näistä aiheista esitietoja. Luvussa määritellään jaollisuus, kongruenssi, jäännösluokka ja käänteisalkiot jäännösluokassa. Lisäksi käsitellään kyseisten käsitteiden ominaisuuksia, joita hyödynnetään myöhemmin tutkielmassa.

Aliluvussa 2.3 esitellään kryptografian alalla paljon hyödynnetty Eukleideen algoritmi sekä sen laajennus. Eukleideen algoritmilla voidaan laskea nopeasti kahden kokonaisluvun a ja b suurin yhteinen tekijä. Laajennetulla algoritmilla saadaan selville muun muassa käänteisalkio luvulle b jäännösluokan $\mathbb{Z}_n$ suhteen.

Luvussa 3 esitellään lyhyesti kryptografia käsitteenä sekä tarkastellaan tutkielmassa käytettäviä merkintöjä. Kryptografia käsittelee viestien salaamista ja salauksen purkamista. Salausmenetelmälle annetaan kattava määritelmä ja esitellään symmetrisen ja epäsymmetrisen salausmenetelmien ero. Julkisen avaimen salausmenetelmä perustuu yksisuuntaisiin funktioihin, jolloin salausavaimesta voi tehdä julkisen. Yksisuuntaisuus takaa, että avoimuudesta huolimatta kolmas osapuoli ei saa selville käyttäjän yksityistä avainta.

Luvussa 4 esitellään reppuongelma. Ongelmaan on saattanut törmätä jo monta kertaa sitä tiedostamattaan, sillä reppuongelma on monissa käytännön sovelluksissa eteen tuleva valintaongelma. Se tulee esille, kun esimerkiksi yritetään optimoida esineiden pakkaamista rajallisen kokoiseen reppuun, mistä nimikin juontaa. Jokaisella esineellä on tilavuus ja hyöty mutta yhteistilavuus ei saa ylittää repun kokoa ja samalla kokonaishyöty tulisi maksimoida. Reppuongelmasta on olemassa monia eri versioita, mutta työssä keskitytään niin sanottuun osajoukko-ongelmaan.

Reppuongelman ratkaiseminen on yleisessä tilanteessa hidasta, sillä kyseessä on NP-täydellinen ongelma. Kuitenkin aliluvussa 4.2 esitellään reppuongelman erikoistapaukseen ratkaisualgoritmi, jota hyödynnetään myöhemmin tarkasteltavassa salausmenetelmässä.

Luvussa 5 käydään läpi 1970-luvun loppupuolelta peräisin oleva kryptografinen

salausmenetelmä. Salausmenetelmä on nimetty sen kehittäjien Ralph C. Merklen ja Martin E. Hellmanin mukaan. Salausmenetelmä perustuu reppuongelmaan ja sen ratkaisemisen haastavuuteen yleisessä tilanteessa. Lopuksi käydään kattava esimerkki, miten salausmenetelmä toimii lyhyelle viestille.

Tutkielman lukijalta odotetaan ainoastaan lukion matematiikan osaamista. Esitietojen lähteenä on käytetty M. Ericksonin ja A. Vazzanan kirjaa *Introduction to Number Theory*. Kryptografian teorian ja esimerkkien lähteinä on käytetty J. A. Buchmannin kirjaa *Introduction to Cryptography* sekä M. W. Baldonin, C. Ciliberton, G. M. Piacentini Cattaneon kirjaa *Elementary Number Theory, Cryptography and Codes*. Reppuongelman määrittelyn lähteenä on S. Martellon ja P. Tothin kirja *Knapsack Problems: Algorithms and Computer Implementations* ja reppuongelman sovelluksen sekä Merklen–Hellmanin salausmenetelmän läpikäynti pohjautuu jo mainittuun Baldonin, Ciliberton ja Piacentini Cattaneon kirjaan.

2 Esitietoja lukuteoriasta

Luvussa 2 esitellään pääaiheen ymmärtämiseen tarvittavia esitietoja. Kryptografian matemaattinen pohja perustuu pitkälti lukuteoriaan ja algebraan, joten aliluvuissa 2.1 ja 2.2 annetaan näiden aiheiden perusmääritelmiä ja -tuloksia [3, s. 15–66]. Osiossa 2.3 esitellään puolestaan kryptografiassa paljon sovellettu Eukleideen algoritmi ja siihen liittyviä esitietoja [2, 3, s. 12–21, s. 26–32].

2.1 Jaollisuus

Määritelmä 2.1. Olkoot a ja b kokonaislukuja. Sanotaan, että a jakaa luvun b, jos $b = ka$ jollakin kokonaisluvulla k. Tällöin merkitään $a \mid b$. Luvun a sanotaan olevan luvun b jakaja sekä luvun b luvun a monikerta. Jos luku a ei jaa lukua b, merkitään $a \nmid b$.

Lause 2.1 (Jakoyhtälö)**.** *Olkoot a ja b kokonaislukuja ja $b > 0$. Tällöin on olemassa yksikäsitteiset kokonaisluvut q ja r siten, että*

$$a = bq + r \quad ja \quad 0 \leq r \leq b. \tag{2.1}$$

Kokonaislukua q kutsutaan osamääräksi ja kokonaislukua r jakojäännökseksi.

Todistus. Osoitetaan aluksi kokonaislukujen q ja r olemassaolo. Olkoon

$$S = \{\, a - bk \mid k \in \mathbb{Z} \text{ ja } a - bk \geq 0 \,\}.$$

Huomataan, että S ei ole tyhjä joukko: Kun $a \geq 0$, niin valinnalla $k = 0$ luku a on joukon S alkio. Kun $a < 0$, niin valinnalla $k = -a$ saadaan

$$a - b(-a) = \underbrace{(-a)}_{>0}\,\underbrace{(-1 + b)}_{\geq 0},$$

mikä on joukon S alkio. Jos S sisältää luvun 0, niin on olemassa kokonaisluku q siten, että $a = bq$. Tässä tapauksessa valitaan kokonaisluvuksi $r = 0$. Oletetaan sitten, että joukko S ei sisällä lukua 0. Olkoon nyt luku r joukon S pienin alkio ja valitaan luku q siten, että $r = a - bq$. Jos $r \geq b$, niin

$$0 \leq r - b = a - bq - b = a - (q + 1)b.$$

Tästä nähdään, että luku $a - (q+1)b$ kuuluu joukkoon S. Lisäksi se on pienempi kuin r, mikä on ristiriita, joten $0 \leq r \leq b$. Siispä luvut r ja q ovat olemassa.

Osoitetaan seuraavaksi lukujen yksikäsitteisyys. Oletetaan, että

$$a = bq_1 + r_1 = bq_2 + r_2, \quad 0 \leq r_1 < b \text{ ja } 0 \leq r_2 < b. \tag{2.2}$$

Symmetrisyyden nojalla oletetaan lisäksi, että $r_2 \geq r_1$. Yhtälöstä (2.2) saadaan

$$b(q_1 - q_2) = r_2 - r_1 \geq 0. \tag{2.3}$$

Oletuksen mukaan $r_2 < b$, joten $r_2 - r_1 < b$. Siis yhtälö (2.3) saadaan muotoon

$$0 \leq b(q_1 - q_2) < b,$$

josta edelleen

$$0 \leq q_1 - q_2 < 1.$$

Siispä $q_1 = q_2$. Tästä seuraa myös, että $r_1 = r_2$. Siispä on osoitettu, että jakoyhtälön määräämät kokonaisluvut ovat olemassa ja ne ovat yksikäsitteiset. □

Esimerkki 2.1.

(a) Luku 3 jakaa luvun 12 eli $3 \mid 12$, sillä $12 = 3 \cdot 4$.

(b) Luku 3 ei jaa lukua 17 eli $3 \nmid 17$. Kuitenkin jakoyhtälön mukaan saadaan $17 = 3 \cdot 5 + 2$. Siis jaettaessa lukua 17 luvulla 3 osamäärä on 5 ja jakojäännös 2.

Määritelmä 2.2. Olkoot a ja b kokonaislukuja siten, että ainakin toinen niistä on eri suuri kuin nolla. Lukujen a ja b *suurin yhteinen tekijä* on suurin kokonaisluku, joka jakaa molemmat luvuista a ja b. Tätä merkitään $\text{syt}(a, b)$.

Määritelmä 2.3. Kokonaislukujen a ja b sanotaan olevan *suhteellisia alkulukuja* tai keskenään jaottomia, jos $\text{syt}(a, b) = 1$.

Esimerkki 2.2.

(a) lukujen 6 ja 4 suurin yhteinen tekijä on 2, eli $\text{syt}(6, 4) = 2$.

(b) $\text{syt}(35, 28) = 7$.

(c) Luvut 15 ja 8 ovat suhteellisia alkulukuja, sillä $\text{syt}(15, 8) = 1$.

2.2 Kongruenssi

Määritelmä 2.4. Olkoot a, b ja n kokonaislukuja ja $n \geq 2$. Luvut a ja b ovat kongruentteja modulo n, jos $n \mid (a - b)$. Tällöin merkitään $a \equiv b \pmod{n}$.

Lause 2.2. *Olkoot a, a', b, b' ja n kokonaislukuja. Jos $a \equiv a' \pmod{n}$ ja $b \equiv b' \pmod{n}$, niin*

$$a + b \equiv a' + b' \pmod{n}$$

ja

$$ab \equiv a'b' \pmod{m}.$$

Todistus. Ks. [3, s. 58]. □

Lause 2.3. *Kongruenssirelaatio ≡ (modulo n) on ekvivalenssirelaatio joukossa $\mathbb{Z}$.*

Todistus. Olkoot a, b, c ja n kokonaislukuja.

Refleksiivisyys: Nähdään, että $a - a = 0 = 0 \cdot n$ eli n jakaa erotuksen $a - a$. Siispä $a \equiv a \pmod{n}$.

Symmetrisyys: Oletetaan, että $a \equiv b \pmod{n}$. Määritelmän mukaan $a - b = nk$, jollakin kokonaisluvulla k. Siispä $b - a = -(nk) = n(-k)$, joten $b \equiv a \pmod{n}$.

Transitiivisuus: Oletetaan, että $a \equiv b \pmod{n}$ ja $b \equiv c \pmod{n}$. Määritelmän mukaan oletukset voidaan kirjoittaa muodossa $a - b = nk$ ja $b - c = nl$, joillakin kokonaisluvuilla k ja l. Tästä saamme edelleen

$$(a - b) + (b - c) = nk + nl,$$

josta

$$a - c = n(k + l).$$

Siispä $a \equiv c \pmod{n}$. □

Määritelmä 2.5. Ekvivalenssirelaation ≡ (modulo n) ekvivalenssiluokkia kutsutaan *jäännösluokiksi*.

Kokonaisluvun a jäännösluokkaa merkitään $[a]$. Tähän joukkoon kuuluvat siis kaikki kokonaisluvut, jotka ovat muotoa $a + kn$, missä k on jokin kokonaisluku. Kaikkien jäännösluokkien joukkoa merkitään $\mathbb{Z}_n$.

Esimerkki 2.3. Olkoon $n = 5$. Lukua 0 vastaava ekvivalenssiluokka, eli jäännösluokka $[0]$, koostuu luvuista, joille pätee $a \equiv 0 \pmod{5}$. Siis

$$[0] = \{\ldots, -5, 0, 5, 10, 15, \ldots\}.$$

Vastaavasti lukua 3 vastaava jäännösluokka on

$$[3] = \{\ldots, -7, -2, 3, 8, \ldots\}.$$

Kun näin määritellään vielä jäännösluokat $[1]$, $[2]$ ja $[4]$ saadaan

$$\mathbb{Z}_5 = \{[0], [1], [2], [3], [4]\}.$$

Huomautus. Yllä olevan esimerkin voi yleistää. Jakoyhtälön nojalla jokaiselle kokonaisluvulle a voidaan löytää luku r siten, että $0 \leq r < n$ ja $a \equiv r \pmod{n}$. Lisäksi, jos $0 \leq a < b < n$, niin $n \nmid b - a$ eli $b \not\equiv a$, joten $[a] \neq [b]$. Siispä saadaan yleisesti, että $\mathbb{Z}_n = \{[0], [1], \ldots, [n-1]\}$.

Mitä tahansa jäännösluokan jäsentä voidaan kutsua kyseisen luokan edustajaksi. Kuitenkin edellisen huomautuksen nojalla, jokaiselle jäännösluokalle voidaan valita epänegatiivinen ja lukua n pienempi edustaja. Usein sovelluksissa kannattaa työskennellä pelkästään juuri edustajien kanssa. Merkintä $a = b \pmod{n}$, missä $0 \leq b < n$ ja luku a ei kuulu kyseiseen väliin, tarkoittaa, että $a \equiv b \pmod{n}$ ja b on jäännösluokan $[a] = [b]$ edustaja. Esimerkiksi $17 = 2 \pmod{3}$.

Lause 2.4. *Kongruenssilla*

$$ax \equiv b \pmod{n}$$

on ratkaisu joukossa $\mathbb{Z}_n$, jos ja vain jos $\operatorname{syt}(a, n) \mid b$. *Jos ratkaisuita on olemassa, niiden lukumäärä on täsmälleen* $\operatorname{syt}(a, n)$.

Todistus. Ks. [3, s. 63]. □

Olkoon $\mathbb{Z}_n^*$ joukon $\mathbb{Z}_n$ osajoukko, joka sisältää jäännösluokat, joiden edustajat ovat suhteellisia alkulukuja luvun n suhteen. Lauseesta 2.4 seuraa, että yhtälöllä

$$ax \equiv 1 \pmod{n}$$

on ratkaisu, jos ja vain jos $a \in \mathbb{Z}_n^*$. Lisäksi kyseinen ratkaisu on yksikäsitteinen. Tämän avulla voidaan määritellä käänteisalkiot joukossa $\mathbb{Z}_n^*$.

Määritelmä 2.6. Olkoon $a \in \mathbb{Z}_n^*$. Luvun a käänteisluku joukossa $\mathbb{Z}_n^*$ on luku $b \in \mathbb{Z}_n^*$, jolle

$$ab \equiv 1 \pmod{n}.$$

2.3 Eukleideen algoritmi

Eukleideen algoritmilla voidaan laskea kokonaislukujen a ja b suurin yhteinen tekijä. Laskun suorittamiselle on toki muitakin tapoja mutta Eukleideen algoritmi on hyvin nopea eli siinä on yleensä vähän askelia [3, s. 52].

Eukleideen algoritmi hyödyntää jakoyhtälöä 2.1, jonka mukaan jokainen kokonaisluku a voidaan kirjoittaa muodossa

$$a = bq + r, \tag{2.4}$$

missä $0 \leq r < b$. Algoritmi perustuu seuraavaan havaintoon.

Apulause 2.1. Olkoot a, b, q ja r kokonaislukuja. Jos $a = bq + r$, niin $\text{syt}(a, b) = \text{syt}(b, r)$.

Todistus. [3, s. 27] Oletetaan, että d jakaa molemmat luvut a ja b. Tällöin luku d jakaa myös luvun r, sillä $r = a - bq$. Vastaavasti, jos d jakaa luvut b ja r, niin yhtälön (2.4) perusteella luku d jakaa tällöin myös luvun a. On siis osoitettu, että luvuilla a ja b sekä b ja r on samat yhteiset tekijät. Koska syt on suurin yhteisistä tekijöistä, tulos seuraa. □

Eukleideen algoritmissa oikeastaan vain toistetaan jakoyhtälöä ja apulauseen 2.1 huomiota. Algoritmi loppuu, kun jako menee tasan eli jakojäännös on nolla.

Esimerkki 2.4. [3] Luvuilla $a = 306$ ja $b = 252$ saadaan algoritmissa seuraavat laskutoimitukset:

$$\begin{aligned} 306 &= 252 \cdot 1 + 54, \\ 252 &= 54 \cdot 4 + 36, \\ 54 &= 36 \cdot 1 + 18, \\ 36 &= 18 \cdot 2 + 0. \end{aligned}$$

Apulauseen 2.1 tulosta toistamalla saadaan, että $\text{syt}(306, 252) = \text{syt}(252, 54) = \text{syt}(54, 36) = \text{syt}(36, 18) = \text{syt}(18, 0) = 18$.

Algoritmi 2.1 (Eukleideen algoritmi)**.** Lukujen a ja b syt voidaan selvittää seuraa-

villa laskutoimituksilla:

$$\begin{aligned}
a &= bq_1 + r_1 && 0 \leq r_1 < b,\\
b &= r_1q_2 + r_2 && 0 \leq r_2 < r_1,\\
r_1 &= r_2q_3 + r_3 && 0 \leq r_3 < r_3,\\
&\vdots\\
r_i &= r_{i+1}q_{i+2} + r_{i+2} && 0 \leq r_{i+2} < r_{i+1},\\
&\vdots\\
r_{n-1} &= r_{n-1}q_n + r_n && 0 \leq r_n < r_{n-1},\\
r_{n-1} &= r_nq_{n+1} + 0.
\end{aligned}$$

Algoritmin täytyy päättyä epäyhtälöiden perusteella, sillä jakojäännös pienenee aidosti jokaisella iteraatiolla. Lopuksi saadaan $\text{syt}(a, b) = r_n$.

2.4 Laajennettu Eukleideen algoritmi

Lause 2.5. *Olkoot a ja b kokonaislukuja, joista ainakin toinen ei ole nolla. Tällöin* $\text{syt}(a, b)$ *on lausekkeen* $ax+by$*, missä x ja y ovat kokonaislukuja, pienin positiivinen arvo.*

Todistus. Ks. [3, s. 22] □

Yhtälöllä $ax+by = \text{syt}(a, b)$ on siis olemassa kokonaislukuratkaisu. Tilanne voitaisiin yleistää ja esittää kysymys, onko yhtälöllä $ax+by = c$ kokonaislukuratkaisuja, kun c on jokin positiivinen kokonaisluku. Tällaistä kokonaislukukertoimista ensimmäiseen asteen polynomiyhtälöä kutsutaan *Diofantoksen yhtälöksi* [3]. Kuitenkin tämän työn kannalta yleinen tilanne ei ole mielenkiintoinen.

Yhtälön $ax + by = \text{syt}(a, b)$ ratkaisun löytämiseksi laajennetaan Eukleideen algoritmia. Käytetään samoja merkintöjä r_i ja q_i kuin Eukleideen algoritmissa. Rakennetaan nyt jonot (x_i) ja (y_i) siten, että $x = (-1)^n x_n$ ja $y = (-1)^{n+1} y_n$ on yhtälön ratkaisuna toimiva kokonaislukupari.

Asetetaan $x_0 = 1$, $x_1 = 0$, $y_0 = 0$ ja $y_1 = 1$. Olkoot lisäksi

$$x_{i+1} = q_i x_i + x_{i-1}, \quad y_{i+1} = q_i y_i + y_{i-1} \quad \text{ja } 1 \leq i \leq n.$$

Lause 2.6. *Olkoot a ja b kokonaislukuja sekä* (x_i)*,* (y_i) *ja i kuten edellä. Nyt Eukleideen algoritmin kohdassa* $0 \leq i \leq n + 1$ *pätee* $r_i = (-1)^i x_i a + (-1)^{i+1} y_i b$.

Todistus. Ks. [2, s. 16] Todistetaan induktiolla, että väite pätee kaikille $0 \leq i \leq n+1$. Todetaan aluksi, että

$$r_0 = a = 1 \cdot a - 0 \cdot b = x_0 \cdot a - y_o \cdot b$$

sekä

$$r_1 = b = -0 \cdot a + 1 \cdot b = -x_1 \cdot a + y_1 \cdot b.$$

Olkoon $i \geq 2$ ja oletetaan, että väite pätee kaikilla indekseillä k, joilla $0 \leq k \leq i$. Nyt jakoyhtälön (2.4) nojalla $r_{i-2} = q_{i-1}r_{i-1} + r_i$. Tästä termejä siirtämällä ja käyttämällä induktio-oletusta saadaan

$$\begin{aligned} r_i &= r_{i-2} - q_{i-1}r_{i-1} \\ &= (-1)^{i-2}x_{i-2}a + (-1)^{i-1}y_{i-2}b - q_{i-i}((-1)^{i-1}x_{i-1}a + (-1)^i y_{i-1}b), \end{aligned}$$

josta termejä yhdistelemällä saadaan

$$\begin{aligned} r_i &= (-1)^i a(x_{i-2} + q_{i-1}x_{i-1}) + (-1)^{i+1}b(y_{i-2} + q_{i-1}y_{i-1}) \\ &= (-1)^i x_i a + (-1)^{i+1}y_i b. \end{aligned}$$

Siispä induktioperiaatteen nojalla väite pitää paikkansa. □

Erityisesti saadaan siis

$$r_n = (-1)^n x_n a + (-1)^{n+1} y_n b,$$

joten ollaan saatu esitettyä $r_n = \text{syt}(a, b)$ lukujen a ja b lineaarikombinaationa. Laajennettu Eukleideen algoritmi suorittaa siis samat askeleet, kuin algoritmi 2.3 mutta se pitää samalla kirjaa kertoimista x_i ja y_i.

Huomautus. Laajennettua Eukleideen algoritmia voidaan käyttää myös etsimään määritelmän 2.6 mukainen käänteisalkio jäännösluokassa. Jos a ja n ovat suhteellisia alkulukuja, eli $\text{syt}(a, n) = 1$, saadaan lineaariyhtälö

$$ax + ny = \text{syt}(a, n) = 1, \tag{2.5}$$

josta termejä siirtämällä saadaan

$$ax - 1 = (-y)n.$$

Kongruenssin määritelmän nojalla

$$ax \equiv 1 \pmod{n}.$$

Siispä löytämällä ratkaisu yhtälölle 2.5 saadaan laskettua luvulle a käänteisalkio x jäännösluokassa $\mathbb{Z}_n$.

3 Kryptografia

Kryptografialla tarkoitetaan viestien ja datan salaamista kolmansilta osapuolilta matemaattisin keinoin. Viestillä tarkoitetaan kryptografiassa mitä tahansa dataa. Viestin salaamiseen käytetään salaimia, jotka perustuvat hyvin pitkälti matematiikkaan ja erityisesti lukuteoriaan sekä algebraan. Tässä osioissa tarkastellaan kryptografian matemaattista pohjaa ja esitellään esimerkki yksinkertaisesta salaimesta.

Aluksi kuitenkin huomioita tässä kandityössä käytettävistä merkinnöistä ja menetelmistä:

- Kaikki kirjaimet käsitellään pieninä kirjaimina ja muunnetaan kokonaisluvuiksi taulukon 3.1 mukaan.
- Kryptografiassa käsitellään viestejä usein lohkoina. Yksinkertaisimmillaan tämä tarkoittaa, että selväteksti jaetaan saman pituisiin lohkoihin ja täytetään tarvittaessa loppupäästä sovitulla merkillä. Esimerkiksi, jos lohkon pituus on 2 ja selväteksti on *hei*, niin lohkoina selväteksti on *he ix* eli numeroiksi muunnettuna 0704 0823. Tässä x oli ennalta sovittu täytemerkki. Lohkomalla selväteksti voidaan peittää sanapituudet ja yhtenäistää salauksen toimintaa [1].
- Yksinkertaisimmillaan lohkot käsitellään jokainen erikseen. Salaaminen ja purkaminen tapahtuu siis salaamalla yksittäinen lohko ja toistamalla tämä jokaiselle lohkolle.
- Työssä käytetään suomalaisia aakkosia, eli $\Sigma = \{\text{a,b,c,}\ldots\text{,å,}\acute{}\text{a,}\acute{}\text{o}\}$. Aakkoston pituus on siis 29.
- Olkoon lohkon pituus s ja p jokin selvätekstin lohko. Tällöin yksinkertaisesti merkitään $p = x_1x_2\ldots x_s$, missä $x_1, x_2, \ldots, x_s$ ovat luvuiksi muunnettuna lohkon kirjaimet.
- Kryptografiasta tutulla tavalla [2], käytetään viestin lähettäjästä nimeä Alice tai A, vastaanottajasta nimeä Bob tai B ja mahdollisesta kolmannesta osapuolesta nimeä Catherine tai C.

Jatketaan antamalla salausmenetelmälle yleinen määritelmä. Määritelmä kuvaa peruskäsitteitä, joista salausmenetelmä rakentuu.

Taulukko 3.1. Muunnostaulukko: kirjain-luku-bitti [1].

a → 00 = 00000	k → 10 = 01010	u → 20 = 10100
b → 01 = 00001	l → 11 = 01011	v → 21 = 10101
c → 02 = 00010	m → 12 = 01100	w → 22 = 10110
d → 03 = 00011	n → 13 = 01101	x → 23 = 10111
e → 04 = 00100	o → 14 = 01110	y → 24 = 11000
f → 05 = 00101	p → 15 = 01111	z → 25 = 11001
g → 06 = 00110	q → 16 = 10000	å → 26 = 11010
h → 07 = 00111	r → 17 = 10001	ä → 27 = 11011
i → 08 = 01000	s → 18 = 10010	ö → 28 = 11100
j → 09 = 01001	t → 19 = 10011	

Määritelmä 3.1. Salausmenetelmä tai kryptosysteemi on monikko $(\mathcal{P}, C, \mathcal{K}, \mathcal{E}, \mathcal{D})$, jolla on seuraavat ominaisuudet:

1. $\mathcal{P}$ on joukko, jota sanotaan *selväkieleksi*. Sen alkiot ovat *selvätekstejä*.
2. C on joukko, jota sanotaan *salakieleksi*. Sen alkiot ovat *salatekstejä.*
3. $\mathcal{K}$ on joukko, jota sanotaan *avainavaruudeksi*. Sen alkiot ovat *avaimia.*
4. $\mathcal{E} = \{E_k : k \in \mathcal{K}\}$ on perhe funktioita $E_k : \mathcal{P} \to C$. Sen alkiot ovat *salausfunktioita*.
5. $\mathcal{D} = \{D_k : k \in \mathcal{K}\}$ on perhe funktioita $D_k : C \to \mathcal{P}$. Sen alkiot ovat *purkufunktioita*.
6. Jokaiselle $e \in \mathcal{K}$ on olemassa $d \in \mathcal{K}$ siten, että $D_d(E_e(p)) = p$ jokaisella $p \in \mathcal{P}$.

Tarkastellaan seuraavaksi yksinkertaisia esimerkkejä salausmenetelmistä. Ensimmäiseksi tarkastelemme Caesar–salausta, joka on yksi tunnetuimmista historiallisista salaimista. Julius Caesar käytti vastaavaa salausta viestien lähettämiseen, mistä nimi juontaa [1, s. 321].

Esimerkki 3.1 (Caesar–salaus)**.** Caesar–salauksessa Selväkieli, Salakieli ja avainavaruus koostuvat kaikki aakkosista eli $(\mathcal{P}, C, \mathcal{K}) = (\Sigma, \Sigma, \Sigma)$. Samastetaan jokainen kirjain numeroon taulukon 3.1 mukaisesti. Olkoon avain $e \in \mathbb{Z}_{29}$. Nyt salausfunktio

E_e on

$$E_e \colon \Sigma \to \Sigma, E_e(x) = (x + e) \pmod{29}.$$

Vastaava purkufunktio avaimella $d \in \mathbb{Z}_{29}$ on

$$D_d \colon \Sigma \to \Sigma, D_d(x) = (x - d) \pmod{29}.$$

Caesar–salauksessa avaimille pätee $e = d$, sillä kyseessä on symmetrinen salausmenetelmä, josta lisää esimerkin jälkeen.

Olkoon selväteksti p ”kryptografia” ja avain $e = 3$. Caesar–salauksessa salataan yksittäinen kirjain kerrallaan, joten lohkokoko on 1. Lasketaan edellä määritellyllä salausfunktiolla E_e ja taulukkoa 3.1 apuna käyttäen ($k = 10$),

$$E_3(k) \xrightarrow{3.1} E_3(10) = 10 + 3 = 13 \xrightarrow{3.1} n \pmod{29}.$$

Toistetaan lasku muillekin kirjaimille ja saadaan salakirjoitus c ”nuäswrjudild”.

Puretaan sitten saatu salakirjoitus. Käytetään avainta $d = 3$, purkufunktiota D_d sekä taulukkoa 3.1 ja saadaan

$$D_3(n) \xrightarrow{3.1} D_3(13) = 13 - 3 = 10 \xrightarrow{3.1} k \pmod{29}.$$

Tehdään lasku muillekin salakirjoituksen kirjaimille ja saadaan selväteksti ”kryptografia”.

Salaus- ja purkufunktioissa lasketaan jakojäännöstä (mod 29), sillä jos salattaisiin esimerkiksi kirjainta ä avaimella 4, saataisiin

$$E_4(\text{ä}) \xrightarrow{3.1} E_4(27) = 27 + 4 = 31 = 2 \xrightarrow{3.1} c \pmod{29}.$$

3.1 Erilaiset salausmenetelmät

Jos Alice haluaa lähettää Bobille salatun viestin, käyttää Alice salausavainta $e \in \mathcal{K}$ ja Bob käyttää avainta e vastaavaa purkuavainta d purkaakseen salatun viestin. Jos käytetyssä salausmenetelmässä avaimet e ja d ovat aina samat tai toisesta voidaan helposti laskea toinen, sanotaan salausmenetelmää symmetriseksi [2].

Jos Alice ja Bob käyttävät symmetristä salausmenetelmää, kummankin heistä on tiedettävä käytetty avain ennen kuin he voivat aloittaa salatun viestinnän. Tämä on kuitenkin hyvin ongelmallista varsinkin, jos viestijöitä on monta. Caesar–salaus on esimerkki symmetrisestä salauksesta.

Epäsymmetrisessä salausmenetelmässä salausavain e ja purkuavain d ovat erilliset ja avaimen d laskeminen avaimesta e on käytännössä mahdotonta [2]. Siispä

salausavaimen voi asettaa julkiseksi. Jos siis Alice julkaisee oman salausavaimensa ja pitää purkuavaimen salaisena, voi kuka tahansa lähettää Alicelle salatun viestin, jonka vain Alice voi purkaa. Salausavainta e voidaankin kutsua julkiseksi avaimeksi ja purkuavainta d yksityiseksi avaimeksi. Epäsymmetrisiä salausmenetelmiä kutsutaan siksi myös julkisen avaimen salausmenetelmiksi. Julkisen avaimen salausmenetelmästä esimerkkejä ovat tässä työssä esiteltävä Merklen–Hellmanin salausmenetelmä sekä tunnettu RSA-salausmenetelmä [2, s. 71].

Näin on siis vältetty symmetristen salausmenetelmien avaintenvaihto-ongelma. Julkiset salausavaimet voidaan säilyttää julkisessa hakemistossa, josta kuka tahansa voi lukea haluamansa julkisen salausavaimen. Esimerkiksi internet-ympäristössä tämä tekee salattujen viestien lähettämisestä yksinkertaista.

Tehokkuussyistä todellisuudessa usein yhdistetään symmetrinen ja epäsymmetrinen salausmenetelmä. Julkisen avaimen salaus on usein turhan raskas laskea koko viestille, joten viesti salataankin symmetrisellä salauksella käyttäen istuntoavainta, jonka Alice luo kyseistä viestin vaihtoa varten. Alice salaa symmetrisellä salauksella viestin m ja Bobin julkisella salausavaimella istuntoavaimen. Alice lähettää salakirjoituksen c ja salatun istuntoavaimen Bobille. Bob pystyy nyt purkamaan omalla yksityisellä avaimellaan istuntoavaimen, jolla edelleen Bob purkaa salakirjoituksen c ja saa viestin m [2, s. 140].

3.2 Julkisen avaimen salausfunktioista

Aiemmin mainittiin julkisen salausmenetelmän yhteydessä, että purkuavaimen d laskeminen salausavaimesta e on käytännössä mahdotonta. Käsitteelle voi antaa tarkemman, joskin ei siltikään matemaattisesti tarkan määritelmän.

Määritelmä 3.2. [1, s.343]. Funktio $f : \mathcal{P} \to C$ on *yksisuuntainen*, jos sen laskeminen on helppoa[1], mutta alkion $x \in \mathcal{P}$ löytäminen satunnaiselle $y \in f(\mathcal{P})$, jolle $y = f(x)$, on vaikeaa[2] ja käytännössä mahdotonta.

Yksisuuntaisia funktioita on matematiikassa paljon mutta kryptografian ja julkisen salausmenetelmien kannalta mielenkiintoisia ovat niin sanotut salaovifunktiot. Salaovella tarkoitetaan, että jonkin lisätiedon avulla arvojoukon alkiolle löydetään helposti alkukuva.

[1] Tietokoneella laskettaessa nopeaa, esimerkiksi polynomisessa ajassa tapahtuvaa

[2] Tietokoneella hyvin hidasta

Määritelmä 3.3. [1, s.344]. Yksisuuntaisen funktion $f\colon \mathcal{P} \to \mathcal{C}$ sanotaan olevan *salaovifunktio*, jos jollakin lisätiedolla voidaan kaikille $y \in f(\mathcal{P})$ löytää helposti $x \in \mathcal{P}$, jolle $y = f(x)$.

Työssä käsiteltävä reppuongelman ratkaiseminen on esimerkki salaovifunktiosta, mihin palataan myöhemmin. Toinen klassinen esimerkki on kahden ison alkuluvun tulon jakaminen takaisin tekijöihin. Tähän perustuu RSA salausmenetelmä [2]. Siis alkulukujen p ja q tulo $r = pq$ on helppo laskea, mutta tekijöiden selvittäminen pelkästään tulosta r on vaikeaa. Kuitenkin, jos tiedetään toinen alkuluvuista p tai q on toisen laskeminen sen jälkeen helppoa [2].

4 Reppuongelma

Kuvitellaan vaihtoon lähtevä opiskelija, jolla on lennolle mukaan vain yksi iso matkalaukku. Mukaan olisi kiva ottaa vaikka mitä, mutta opiskelijan pitää päättää, mitkä tavarat mahtuvat laukkuun ja mitkä toisivat eniten iloa kokoonsa suhteutettuna. Tällaista valintaongelmaa kutsutaan reppuongelmaksi.

Vastaava ongelma tulee monessa käytännön yhteydessä esille, esimerkiksi rahdin pakkaaminen, pääoman budjetointi ja sijoitusportfolion hajauttaminen [4]. Reppuongelma on tästä syystä hyvin tutkittu aihe.

Reppuongelmasta on monia eri versioita. Onko käytössä yksi vai useampi kassi? Onko esineitä rajallisesti vai rajattomasti? Pitääkö kassin tulla täyteen? Työssä tarkastellaan yksinkertaisinta reppuongelmaa eli niin sanottua 0-1 reppuongelmaa sekä sen erityistapausta osajoukko-ongelmaa. Matemaattisesti 0-1 reppuongelma voidaan määritellä tarkemmin seuraavasti.

Määritelmä 4.1. [4, s. 1]. Olkoon käytössä n esinettä ja olkoon x vektori, joka saa binäärisiä arvoja x_i, missä $i = 1, \ldots, n$ ja joilla on merkitys

$$x_i = \begin{cases} 1, & \text{jos esine } i \text{ valitaan,} \\ 0, & \text{muuten.} \end{cases}$$

Olkoon lisäksi p_i esineen i antama hyöty, w_i sen koko ja c repun koko. Reppuongelmana on nyt valita kaikista mahdollisista binäärivektoreista x se, joka täyttää ehdon

$$\sum_{i=1}^{n} w_i x_i \leq c$$

ja maksimoi summan

$$z = \sum_{i=1}^{n} p_i x_i.$$

Esimerkki 4.1. Oletetaan, että halutaan sijoittaa euroina c verran pääomaa ja tarjolla on n sijoituskohdetta. Olkoon p_i sijoituksesta i odotettava tuotto ja w_j sen hinta euroina. Nyt selvästi reppuongelman maksimaalinen ratkaisu kertoo, miten sijoitukset kannattaa tehdä.

0-1 reppuongelmasta saadaan niin sanottu osajoukko-ongelma, kun asetetaan edellisessä määritelmässä hyöty $p_i = w_i$ kaikilla $i = 1, \ldots, n$ [4, s. 105]. Nimitys

tulee siitä, että nyt etsitään osajoukkoa $I \subseteq \{1, \ldots, n\}$, jolle repun kokonaispaino on mahdollisimman suuri ylittämättä lukua c, eli

$$\sum_{i \in I} w_i \leq c.$$

Osajoukko I kuvaa tässä siis valintoja eli määritelmässä vektoria x.

Jos edelleen rajoitetaan summa yhtä suureksi repun koon kanssa eli

$$\sum_{i=1}^{n} w_i x_i = c, \tag{4.1}$$

saadaan eräs diofantoksen yhtälö. Tällöin ei ongelma ei ole enää maksimaalisuus vaan ratkaisuiden olemassaolo. Kuten tiedetään diofantoksen yhtälöillä ei aina ole ratkaisuita tai ratkaisut eivät välttämättä ole yksikäsitteisiä. Tässä työssä ei syvennytä ratkaisuiden olemassaoloon eikä yksikäsitteisyyteen ja lukija voi perehtyä tarkemmin aiheeseen itse [3, s. 303].

Yhtälön 4.1 tapaista reppuongelmaa hyödynnetään Merklen–Hellmanin salausmenetelmässä, kun käsitellään esineiden sijasta positiivisia kokonaislukuja. Siis olkoot $a_1, a_2, \ldots, a_n$ ja c positiivisia kokonaislukuja. Osajoukko-ongelma saadaan nyt muotoon, voidaanko löytää vektori x ($x_i \in \{0, 1\}$) siten, että

$$\sum_{i=1}^{n} a_i x_i = c. \tag{4.2}$$

Esimerkki 4.2. Olkoot $n = 6$ ja $(a_1, a_2, \ldots, a_6) = (2, 3, 7, 9, 13, 15)$

(a) Olkoon lisäksi $c = 27$. Nyt huomataan, että $2+3+7+15 = 27$, $3+9+15 = 27$ ja $2 + 3 + 9 + 13$. Siispä osajoukko ongelmaan on ainakin kolme ratkaisua. Nämä voidaan merkitä myös määritelmän tavalla $x = (1, 1, 1, 0, 0, 1)$, $y = (0, 1, 0, 1, 0, 1)$ ja $z = (1, 1, 0, 1, 1, 0)$.

(b) Olkoon $c = 1$. Koska $c < a_i$ kaikilla i, niin ratkaisuita x, missä $x_i \in \{0, 1\}$, ei ole olemassa.

4.1 Reppuongelman ratkaisemisen haastavuudesta

Miksi reppuongelmaa voidaan hyödyntää julkisen avaimen salausmenetelmänä? Tämä vaatii ongelmalta yleensä kaksi ominaisuutta: ongelman pitää olla yleisesti vaikea ja hidas laskea, jotta avainta ei pysty murtamaan, sekä ongelmaan pitää olla saatavilla jokin salaovi, joka tekee viestin purkamisesta helppoa ja nopeaa. Siis reppuongelmaan perustuvan salausfunktion pitää olla yksisuuntainen salaovifunktio.

Huomautus. Reppuongelma ja sen tässä esitellyt muunnelmat ovat on niin sanottuja $\mathcal{NP}$-täydellisiä ongelmia [4]. Tämä tarkoittaa lyhyesti sitä, että niille ei ole olemassa polynomiaikaisia ratkaisualgoritmeja esineiden lukumäärän n suhteen, mutta annetun ratkaisun vahvistaminen on kuitenkin nopeaa [1].

Yleisessä tilanteessa[3] 0-1 reppuongelmaan tai osajoukko-ongelmaan ei tunneta parempia algoritmeja kuin kokeilla kaikkia mahdollisia ratkaisuita [1]. Kaikkia mahdollisia ratkaisuita osajoukko-ongelman tilanteessa on 2^n, jos kaikkia esineitä on n kappaletta. Reppuongelma on siis vaikeustason kannalta hyvä kandidaatti salausmenetelmän perustalle, kunhan n on tarpeeksi suuri. Entäpä salaovi?

4.2 Erityisesti kasvavat jonot

Määritelmä 4.2. Jonon positiivisia kokonaislukuja $a_1, a_2, \ldots, a_n$ sanotaan olevan *erityisesti kasvava*, jos on voimassa

$$\begin{aligned} a_1 &< a_2, \\ a_1 + a_2 &< a_3, \\ &\vdots \\ a_1 + a_2 + \cdots + a_{n-1} &< a_n. \end{aligned}$$

Oletetaan, että osajoukko-ongelman tilanteessa joukko, mistä valitaan, on erityisesti kasvava. Tällöin, jos ratkaisu on olemassa, se on yksikäsitteinen ja sen voi löytää polynomisessa ajassa [1, s. 347]. Ongelman ratkaisemiseen voi hyödyntää seuraavaksi annettua algoritmia. Ideana on hyödyntää jonon $a_1, \ldots, a_n$ erityistä kasvavuutta, sillä isoin luku a_n pakko valita, jos repun koko c on suurempi kuin a_n. Tämä idea takaa myös yksikäsitteisyyden.

Algoritmi 4.1. [1, s. 347–348].

1. Olkoot repun koko c ja $a_1, \ldots, a_n$ positiivia kokonaislukuja, jotka muodostavat erityisesti kasvavan jonon. Määritetään x_n huomaamalla, että välttämättä

$$x_n = \begin{cases} 1, & \text{jos } c \geq a_n, \\ 0, & \text{jos } c < a_n. \end{cases}$$

[3] Monessa erikoistapauksessa on olemassa nopeampia algoritmeja [4]. Tässä työssä ei kuitenkaan välitetä niistä, vaan oletetaan, että tilanne on yleinen.

2. Koska jono $a_1, \ldots, a_{n-1}$ on myös erityisesti kasvava, saadaan x_{n-1} määritettyä samalla tavalla, sillä välttämättä

$$x_{n-1} = \begin{cases} 1, & \text{jos } c - a_n x_n \geq a_{n-1}, \\ 0, & \text{jos } c - a_n x_n < a_{n-1}. \end{cases}$$

3. Yleisesti, kun ollaan löydetty $x_n, \ldots, x_{j+1}$, voidaan jatkaa ja saadaan

$$x_j = \begin{cases} 1, & \text{jos } c - \sum_{i=j+1}^{n} a_i x_i \geq a_j, \\ 0, & \text{jos } c - \sum_{i=j+1}^{n} a_i x_i < a_j. \end{cases}$$

4. Jos $c - \sum_{i=j+1}^{n} a_i x_i = 0$, niin yksikäsitteinen ratkaisu $x = (x_1, \ldots, x_n)$ on löydetty. Jos taas $c - \sum_{i=j+1}^{n} a_i x_i > 0$, mutta jokainen luvuista $a_{j-1}, \ldots, a_1$ on suurempi kuin $c - \sum_{i=j+1}^{n} a_i x_i$, ratkaisua ei ole olemassa. Muussa tapauksessa suoritetaan kohta kolme uudelleen.

Algoritmi on selvästi polynomiaikainen, joten sitä voi hyödyntää salausmenetelmässä viestien purkamiseen.

Esimerkki 4.3. [1, s. 348]. Olkoon $a = (1, 4, 6, 13, 25)$ jono positiivisia kokonaislukuja. Olkoon lisäksi repun koko $c = 26$. Jono a on erityisesti kasvava, sillä $1 < 4$, $1 + 4 = 5 < 6$, $1 + 4 + 6 = 11 < 13$ ja $1 + 4 + 6 + 13 = 24 < 25$. Käytetään algoritmia 4.1.

Koska $a_5 = 25 < 26 = c$, niin $x_5 = 1$. Siirrytään seuraavaan jonon alkioon. Nyt $c - a_5 x_5 = 26 - 25 \cdot 1 = 1 < 13 = a_4$, joten $x_4 = 0$. Jatketaan samalla tavalla, ja koska $1 < a_2 < a_3$ täytyy olla $x_2 = 0 = x_3$. Suoritetaan lasku vielä kerran jonon ensimmäiselle alkiolle. Nyt siis $c - (a_5 x_5 + \cdots + a_2 x_2) = 1 \geq 1 = a_1$ eli valitaan x_1=1. Nyt $c - \sum_{i=1}^{n} a_i x_i = 0$, joten on löydetty yksikäsitteinen ratkaisu $x = (1, 0, 0, 0, 1)$.

Jos olisi asetettu $a_1 = 2$, ratkaisuita ei olisi ollut olemassa. Tällöin algoritmin lopuksi olisi saatu $c - (a_5 x_5 + \cdots + a_2 x_2) = 1 > 0$ ja $a_i > 1$ kaikilla $i = 1, \ldots, 5$. Siis viimeisessä kohdassa luku 1 olisi pitänyt esittää muodossa $2 \cdot x_1$, missä $x_1 \in \{0, 1\}$, mikä ei selvästi ole mahdollista.

Nyt on esitelty reppu- ja osajoukko-ongelma ja jälkimmäiselle hyödynnettävä salaovi. Seuraavaksi näiden pohjalta rakennetaan salausmenetelmä.

5 Merklen–Hellmanin salausmenetelmä

Merklen-Hellmanin salausmenetelmä on peräisin 1970–luvun lopulta ja sen todettiin jo 80–luvun alkupuolella olevan murrettavissa polynomisessa ajassa [1]. Salausmenetelmää on sen jälkeen paranneltu, mutta tässä työssä esitellään alkuperäinen menetelmä. Siis lukijan ei tule hyödyntää esiteltyä salausmenetelmää kryptografisessa tarkoituksessaan.

Ennen kuin päästään esittelemään itse salausmenetelmä, niin tehdään havaintoja reppuongelman ratkaisun muodosta. Ratkaisussa määritellään binääriarvoinen vektori $x = (x_1, \ldots, x_N)$. Toisaalta taulukossa 3.1 on esitelty kirjaimia vastaavat binääriluvut. Huomataan, että jos vektori x pituus N vastaisi salattavan lohkon pituutta, voisi jokaista mahdollista N pituista binääriarvoista vektoria pitää mahdollisena reppuongelman ratkaisuna. Tai toisinpäin, voisi jokaista reppuongelman ratkaisua käsitellä lohkona. Salauksessa käytetäänkin siis kirjaimista binäärimuotoja ja lohkon kokona ratkaisun kokoa.

5.1 Merklen–Hellmanin salausmenetelmän määrittely

Määritellään seuraavaksi Merklen–Hellmanin salausmenetelmä määritelmän 3.1 mukaisesti. Salausmenetelmässä selvä- ja salakieli koostuvat kirjaimista eli $(\mathcal{P}, C) = (\Sigma, \Sigma)$.

Koska kyseessä on julkisen avaimen salausmenetelmä, on avain moniosainen. Yksityinen avain koostuu erityisesti kasvavasta jonosta $a \in \mathbb{Z}^N$ sekä kokonaisluvuista $m > 2a_N$ ja w, jotka ovat suhteellisia alkulukuja. Nämä on siis pidettävä salassa.

Yksityisestä avaimesta saadaan laskettua julkinen avain $b \in \mathbb{Z}^N$ asettamalla

$$b_i = wa_i \pmod{m} \qquad \text{jokaisella } i = 1, \ldots, N.$$

Siis avainavaruus $\mathcal{K}$ on pari $((a, m, w), b)$, jossa parin ensimmäinen alkio on yksityinen purkuavain ja toinen alkio julkinen salausavain.

Salatun viestin p lähettämiseksi Alicelle, Bob käyttää Alicen julkista avainta b seuraavasti. Ensin Bob muuttaa viestinsä taulukon 3.1 mukaisesti binääriseksi jonoksi. Bob pilkkoo saadun jonon N mittaisiin lohkoihin. Täytemerkkinä jonon lopussa käytetään ykköstä, eli jos viimeinen lohko on lyhempi kuin N, lisätään sen loppuun tarvittava määrä ykkösiä. Tämän jälkeen Bob käyttää lohkoon $p_j = x_1 \cdots x_N$

salausfunktioita $E_b : \mathcal{P} \to C, E_b(p_j) = b_1x_1 + \cdots + b_Nx_N = c \pmod{m}$. Nyt Bob voi lähettää lohkoista yhdistetyn salatekstin c Alicelle.

Salatekstin c purkamiseksi Alicen on pitänyt laskea purkuavain $k_d = (m, w')$, jossa $ww' = 1 \pmod{m}$. Tällainen luku w' on olemassa lauseen 2.4 perusteella, sillä oletettiin, että $\text{syt}(m, w) = 1$. Luvun w' Alice voi laskea nopeasti käyttäen laajennettua Eukleideen algoritmia, joka on esitelty aliluvussa 2.4.

Kuvaillaan purkufunktio $D_k : C \to \mathcal{P}$ sanallisesti. Aluksi Alice laskee salatekstin c ja luvun w' tulon, jota merkitään kirjaimella v. Siis

$$v = w'c = w'(b_1x_1 + \cdots + b_Nx_n) \pmod{m}.$$

Luvun w' ja lukujen b_i määritelmien perusteella saadaan

$$v = (w'b_1x_1 + \cdots + w'b_Nx_N) = (w'wa_1x_1 + \cdots + w'wa_Nx_N).$$

Jos käytetään summamerkintää ja tietoa, että $w'w = 1 \pmod{m}$, niin saadaan

$$v \equiv \sum_{i=1}^{N} a_ix_i \pmod{m}.$$

Lisäksi jonon a erityisen kasvavuuden perusteella $a_1 + \cdots + a_N < a_N + a_N = 2a_N < m$. Tästä seuraa, että $\sum_{i=1}^{N} a_ix_i$ on edustaja eli

$$v = \sum_{i=1}^{N} a_ix_i \pmod{m}.$$

Nyt Alice voi käyttää algoritmia 4.1. Kuvituksellisen repun koko on siis v ja erityisesti kasvava jono on Alicen valitsema jono a. Näin Alice saa ratkaistuksi luvut x_i eli lohkon p_j.

Alicella on siis tiedossa jokaista selvätekstin lohkoa vastaava binäärinen jono $p_j = x_1 \ldots x_N$. Nyt Alice voi yhdistää lohkot sekä muuttaa binääriesityksen takaisin kirjaimiksi ja lukea Bobin lähettämän viestin selvätekstinä.

Jos Catherine haluaisi selvittää, mitä Bobin lähettämässä viestissä sanotaan, pitäisi Catherinen ratkaista reppuongelma $c = \sum_{i=1}^{N} b_ix_i$. Siis hänen pitäisi löytää vektori x, jolle edellinen yhtäsuuruus on voimassa. Jos yksityinen avain on valittu riittävän hyvin, ei jono b ole erityisesti kasvava. Tämä tarkoittaa, että Catherinen pitäisi ratkaista reppuongelma yleisessä tilanteessa.

Huomautus. Merklen–Hellmanin salausmenetelmän ongelmana on juuri yksityisen avaimen valitseminen riittävän hyvin. Vaikka jono b ei olisikaan erityisesti kasvava, se ei siltikään ole välttämättä tarpeeksi yleinen ja viestin pystyy purkamaan usein polynomisessa ajassa [1].

5.2 Esimerkki tekstin salaamisesta ja purkamisesta

Esimerkki 5.1. Vrt. [1, s. 350]. Olkoon lohkon pituus $N = 4$. Valitaan yksityinen avain (a, m, w) siten, että $a = (1, 2, 4, 27)$, $m = 61$ ja $w = 17$. Eukleideen algoritmilla 2.3 voidaan tarkistaa, että luvut m ja w ovat keskenään jaottomia. Siispä yksityinen avain on $((1, 2, 4, 27), 61, 17)$.

Julkinen salausavain b voidaan nyt laskea yksityisestä avaimesta. Määritelmän mukaan $b_i = wa_i$, joten $b = (14, 34, 68, 459) = (17, 34, 7, 32) \pmod{m}$.

Lisäksi yksityinen purkuavain $k_d = (m, w')$ voidaan laskea aliluvussa 2.4 kuvatulla algoritmilla ratkaisemalla yhtälö $ww' = 1 \pmod{m}$. Näin saadaan $k_d = (61, 18)$.

Olkoon sitten Bobin salattava viesti p ”krypto”. Muutetaan selväteksti binääriseen muotoon taulukon 3.1 avulla. Saadaan

$$01010\ 10001\ 11000\ 01111\ 10011\ 01110,$$

jossa välilyönnit ovat selkeyden vuoksi binäärimuotoisten kirjainten erottimena.

Muodostamalla $N = 4$ pituisia lohkoja saadaan

$$0101\ 0100\ 0111\ 0000\ 1111\ 1001\ 1011\ 1011,$$

jossa viimeiset kaksi ykköstä on lisätty, sillä $30 = 4 \cdot 7 + 2$.

Käytetään lohkoittain salausfunktiota. Ensimmäiselle lohkolle saadaan

$$c_1 = E_d(p_1) = \sum_{i=1}^{4} b_i x_i = 14 \cdot 0 + 34 \cdot 1 + 7 \cdot 0 + 32 \cdot 1 = 66 = 5 \pmod{m}.$$

Sama voidaan toistaa muillekin lohkoille, ja saadaan salateksti

$$5\ 34\ 12\ 0\ 29\ 49\ 56\ 56.$$

Tämän Bob voi nyt lähettää Alicelle.

Tarkastellaan sitten, miten Alice voi purkaa Bobin lähettämän salatekstin c. Purku tapahtuu myös lohko kerrallaan, joten tarkastellaan esimerkkinä ensimmäistä lohkoa $c_1 = 5$.

Lasketaan reppuongelmaa vastaava repun koko v. Määritelmästä saadaan $v = w'c_1 = 18 \cdot 5 = 90 = 29$. Käytetään algoritmia 4.1 ja ratkaistaan ensimmäistä salattua lohkoa c_1 vastaava selväteksti p_1 binäärimuodossa. Repun koko on siis $v = 29$ ja erityisesti kasvava jono $a = (1, 2, 4, 27)$. Algoritmilla saadaan ratkaisu $x = (1, 0, 1, 0)$. Siis ensimmäisen lohkon purettu binääriesitys on 1010.

Puretaan muut lohkot vastaavasti ja saadaan purettu viesti binäärimuodossa

0101 0100 0111 0000 1111 1001 1011 1011.

Viesti on kuitenkin vielä lohkotettu, joten yhdistetään lohkot viiden bitin pituisiksi kirjainkuvioiksi ja poistetaan lisätyt ykköset lopusta.

01010 10001 11000 01111 10011 01110

Nyt voidaan lukea purettu viesti taulukon 3.1 avulla ja saadaan selväkielinen Bobin lähettämä viesti ”krypto”.

Lähteet

[1] Baldoni, M. W., Ciliberto, C., Piacentini Cattaneo, G. M. *Elementary Number Theory, Cryptography and Codes.* Berlin: Springer, 2009.

[2] Buchmann, J. A. *Introduction to Cryptography*. New York: Springer, 2001

[3] Erickson, M., Vazzana, A. *Introduction To Number Theory*. Boca Raton: Chapman & Hall/CRC, 2008.

[4] Martello, S., Toth, P. Knapsack Problems: *Algorithms and Computer Implementations.* Chichester: Wiley, 1990.