Hajautetun avaimen salausmenetelmät

LuK-tutkielma
Pauli Pauna
Y57935150
Matemaattisten tieteiden laitos
Oulun yliopisto
Kesä 2023

Sisällys

Johdanto

Hajautetun avaimen salausmenetelmät on suhteellisen uusi aihe kryptografian alalla. Adi Shamir ja George Blakley esitteli hajautetun avaimen salausmenetelmät vuonna 1979, molemmat omissa erillisissä teoksissaan. Tätä oli vaikea aluksi uskoa, sillä tilanteet joissa näitä kaivataan, voi löytää jo lasten leikeistä. Salainen arkku, joka tarvitsee kaksi avainta. Siinä se on yksinkertaisimmillaan. Se, mitä nämä kaksi tutkijaa tekivät, oli aiheen määrittely. Mitä ominaisuuksia näiltä salausmenetelmiltä vaaditaan ja mihin ongelmiin ne ovat ratkaisu. Jos huonoa salausmenetelmää pidetään salausmenetelmänä, menettää koko aihe merkityksensä. Samoin tekee myös huono hajautetun avaimen salausmenetelmä. Siksi oli erittäin tärkeää, että aihe määriteltiin tarkemmin.

Hajautetun avaimen salausmenetelmien tärkein ominaisuus:

- Jos avaimen osia ei ole tarpeeksi avaimen rakentamiseen, ei avaimesta tiedetä mitään.

Vaatimukset:

- Tarvitaan jakaja joka tietää avaimen sekä jokaisen osan siitä.

Tutkielmassa on käsitelty pääasiassa teoksen "Cryptography: theory and practice"[1], kappaleita 11.2 "Key Predistribution"ja 11.5 "Treshold Schemes". Teoksen lisäksi olen suunnitellut ja toteuttanut erilaisia algoritmeja viimeisen kappaleen esimerkkien ja teorian tueksi. Näiden avulla olen saanut vietyä teoriaa hieman pidemmälle teoksen vastaavasta kappaleesta.

Ensimmäisessä kappaleessa annan yleisen määritelmän hajautetun avaimen salausmenetelmille, joka antaa pohjan koko loppu tutkielmalle. Toisessa kappaleessa tutustumme Shamirin tapaan luoda näitä määriteltyjä salausmenetelmiä, joissa avaimena on jokin luku. Kolmannessa kappaleessa näytän, miten voidaan toteuttaa hajautetun avaimen salausmenetelmiä vielä helpommin kuin Shamirin menetelmällä, jos asetetaan joitain rajoituksia. Tämä antaa syyn siihen, miksi neljännessä kappaleessa lähdetään liikkeelle triviaaleista tapauksista, joita lähdetään laajentamaan yleispätevämmiksi. Neljännessä kappaleessa myös vaihdetaan avain luvuista kuviksi, joka tuottaa omia ongelmiaan.

Pohjatietona tutkielmassa tarvitsee hallita lähinnä kongruenssin käsite, sillä joka sovellutuksessa (myös visuaalisissa) käsitellään jäännösluokkien joukkoja. Ensimmäiset kolme kappaletta ovat hyvin normaalia lukuteoriaa, jossa käsitellään lukuja ja niiden laskutoimituksia. Viimeisessä kappaleessa sen sijaan käsitellään asiaa visuaalisesti, ja itse matematiikka jää hyvinkin piiloon.

Sen sijaan aihetta lähestytään enemmän algoritmi ongelmana. Siellä myös lisätään enemmän tai vähemmän "keinotekoisia"ongelmia, esimerkiksi "lohkot ja avain piirretään läpinäkyvälle kalvolle".

Tutkielma antaa perusteet hajautetun avaimen salausmenetelmiin, ja sen pohjalta voi jatkaa edistyksellisiin sekä yleispätevämpiin tapoihin luoda visuaalisia hajautetun avaimen salausmenetelmiä.

1 Yleisesti

Käydään alkuun havainnollistava esimerkki. Yrityksellä on käytössä arkaluonteinen järjestelmä, johon pääsyn toimitusjohtaja haluaa antaa kolmelle työntekijälle. Kenellekään heistä ei kuitenkaan haluta antaa mahdollisuutta käyttää sitä yksinään, eikä toisaalta haluta myöskään, ettei sitä voisi käyttää ilman kaikkia kolmea. Tarvitaan ratkaisu millä ketkä tahansa kaksi ihmistä näistä kolmesta pääsevät käyttämään järjestelmää. Tällainen ratkaisu on hajautetun avaimen-salausmenetelmä, jolle annetaan seuraavaksi yleinen määritelmä.

Määritelmä 1.1. Valitaan avain k, sekä positiiviset kokonaisluvut t ja w, joille pätee $t \leq w$. Menetelmää, jossa w määräiselle ryhmälle R jaetaan avain k siten, että mikä tahansa t määräinen osaryhmä voi laskea k arvon, mutta mikään sitä pienempi ei, on nimeltään *hajautetun avaimen (t, w)- salausmenetelmä.*

Määritelmän mukaisen menetelmän voi lausua esimerkiksi "kaksi neljästä-salausmenetelmä", joka kuvaa neljää osallista joista kahta tarvitaan salauksen purkuun. Näissä avaimen k valitsee jakaja D. Jakajana täytyy toimia luotettava taho, jolla on pääsy salauksen taakse, sillä hän on ainoa osallinen joka saa salauksen auki yksinään. Oletamme että $D \notin R$. Kun D jakaa avaimen k ryhmälle R, hän antaa jokaiselle osapuolelle tietoa avaimesta. Kutsutaan tästä eteenpäin tätä tietoa lohkoksi l. Nämä lohkot tulee jakaa salassa, niin ettei kukaan osapuolista tiedä kuin oman lohkonsa.

Nyt myöhemmin osajoukon $B \subseteq R$ osalliset keräävät lohkonsa kasaan ja yrittävät laskea avainta k. Jos $|B| \geq t$, he voivat laskea avaimen lohkojensa funktiona. Sen sijaan jos $|B| < t$, he eivät onnistu avaimen laskemisessa.

Määritelmä 1.2. Käytetään *w määräiselle ryhmälle* merkintää

$$R = \big(r_i : 1 \leq i \leq w\big),$$

kaikkien mahdollisten avainten joukolle merkintää K, ja *kaikkien mahdollisten lohkojen joukolle* merkintää L.

Nyt voidaan huomata, että esimerkissä oli kyse $(2, 3)$-salausmenetelmästä, jossa jakajana D oli toimitusjohtaja. Mahdollisten avainten joukko K on esimerkiksi seuraavassa kappaleessa $\mathbb{Z}_p$, ja viimeisessä kappaleessa mielivaltaisen kokoinen, nelikulmainen valko-musta pikselikuva. Salaus perustuu siihen, että kun kaikki tarvittavat lohkot ovat kasassa, voidaan varmaksi sanoa että mikä oikea avain on. Jos kaikki lohkot eivät ole käytettävissä, avaimesta ei tiedetä mitään, joten mahdollisten avainten joukkoa ei voi supistaa. Kun

tutkimme näitä menetelmiä (yleisesti), haluamme tarkastella ehdotonta turvallisuutta. Emme siis rajoita minkään kokoisen ryhmän laskentatehoa, kun avainta k yritetään laskea. Näillä ehdoilla saamme lopputulokseksi kvanttiturvallisia salausjärjestelmiä.

2 Shamirin salainen jako

Tutustutaan (t, w)-salausmenetelmään nimeltä "Shamir Threshold Scheme"(Shamir 1979). Valitaan mahdollisten avainten joukoksi $K = \mathbb{Z}_p$, missä $p \geq w + 1$ on alkuluku. Valitaan myös mahdollisten lohkojen joukoksi $L = \mathbb{Z}_p$, joten nyt sekä avain että jokainen lohko tulee olemaan joukosta $\mathbb{Z}_p$. Sitten jakaja luo satunnaisen polynomin $a(x)$, joka on enintään astetta $t - 1$, ja jonka vakiotermi on avain k. Jokainen osallinen r_i saa lohkokseen luvut x_i ja y_i joille pätee $a(x_i) = y_i$.

Algoritmi 2.1. Luodaan Shamirin hajautetun avaimen salausjärjestelmä:

1. D valitsee w toisistaan eroavaa, nollasta poikkeavaa alkiota joukosta $\mathbb{Z}_p$, joita merkitään x_i, $1 \leq i \leq w$. Nyt D antaa jokaiselle ryhmän jäsenelle r_i alkion x_i. Alkiot x_i ovat julkisia.

2. Nyt D haluaa jakaa avaimen $k \in \mathbb{Z}_p$. D valitsee salaa ja satunnaisesti $t - 1$ alkiota joukosta $\mathbb{Z}_p$, joita merkitään $a_1, ..., a_{t-1}$.

3. Jokaista r_i kohti D laskee arvon $y_i = a(x_i)$, missä

$$a(x) = k + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$

4. Jokaiselle r_i D jakaa salaisen y_i.

Esimerkki 2.2. Luodaan Shamirin menetelmällä (3, 5)-salausjärjestelmä.

1. Täytyy olla $p \geq 5 + 1$, valitaan $p = 7$. Valitaan satunnaisesti joukosta $\mathbb{Z}_7$ arvot $x_1 = 3, x_2 = 1, x_3 = 6, x_4 = 4, x_5 = 2$, ja annetaan nämä osallisille r_i.

2. Valitaan avaimeksi $k = 3$. Nyt valitaan $t - 1$ alkiota joukosta $\mathbb{Z}_7$, $a_1 = 2, a_2 = 5$.

3. Jokaiselle r_i, lasketaan arvo $y_i = a(x_i)$.

$$y_1 = a(x_1) = 3 + \sum_{j=1}^{2} a_j 3^j = 3 + 6 + 45 = 54 = 5 \bmod 7$$

$$y_2 = a(x_2) = 3 + \sum_{j=1}^{2} a_j 1^j = 3 + 2 + 5 = 10 = 3 \bmod 7$$

$$y_3 = a(x_3) = 3 + \sum_{j=1}^{2} a_j 6^j = 3 + 12 + 180 = 195 = 6 \bmod 7$$

$$y_4 = a(x_4) = 3 + \sum_{j=1}^{2} a_j 4^j = 3 + 8 + 80 = 91 = 0 \bmod 7$$

$$y_5 = a(x_5) = 3 + \sum_{j=1}^{2} a_j 2^j = 3 + 4 + 20 = 27 = 6 \bmod 7$$

4. Jokaiselle r_i, jaetaan nyt salainen lohko y_i.

Kun myöhemmin osalliset $r_1, r_2, ..., r_t$ haluavat purkaa salauksen, he keräävät lohkonsa yhteen ja ratkaisevat näin saadun yhtälöparven.

Esimerkki 2.3. Edellisen esimerkin järjestelmän haluavat nyt avata osalliset r_1, r_2 ja r_4, ratkaisuilla $a(3) = 5$, $a(1) = 3$ ja $a(4) = 0$. Tiedetään että avain k on enintään $(t-1) = 2$ asteisen polynomin $a(x)$ vakiotermi ja

$$a(x) = a_0 + a_1 x + a_2 x^2.$$

Näillä tiedoilla he saavat tehtyä yhtälöparven

$$\begin{cases} 5 = a_0 + 3a_1 + 9a_2 \bmod 7 \ (1) \\ 3 = a_0 + a_1 + a_2 \bmod 7 \ (2) \\ 0 = a_0 + 4a_1 + 16a_2 \bmod 7 \ (3) \end{cases}$$

ja tästä laskettua yksikäsitteisen a_0 joukossa $\mathbb{Z}_7$:

$$(1)\ ja\ (2) \rightarrow a_1 = 1 - 4a_2 \ (4)$$

$$(1)\ ja\ (4) \rightarrow a_0 = 3a_2 + 2 \ (5)$$

$$(3),\ (4)\ ja\ (5) \rightarrow a_2 = 5 \ (6)$$

$$(5)\ ja\ (6) \rightarrow a_0 = 3.$$

On selvää että näissä käytettävissä järjestelmissä on oltava yksikäsitteinen ratkaisu, niin kuin edellisessä esimerkissä. Tämä yleispätevyys voidaan todistaa usealla tavalla. Lähdeteoksen mukaan paras tapa on soveltaa Lagrangen interpolaatiopolynomin kaavaa polynomeille. Sen perusteella haluttu t:n pisteen kautta kulkeva, enintään astetta $t-1$ oleva polynomi $A(x)$ on yksikäsitteinen, ja se antaa kaavan jolla $A(x)$ voidaan laskea.

Lause 2.4 (Lagrangen interpolaatiopolynomin kaava)**.**
Oletetaan että p on alkuluku, $x_1, x_2, ..., x_t$ ovat eriäviä alkioita joukosta $\mathbb{Z}_p$, ja $y_1, y_2, ..., y_t$ ovat alkoita joukosta $\mathbb{Z}_p$. Nyt on olemassa enintään $t-1$ asteinen yksikäsitteinen polynomi $A(x) \in \mathbb{Z}_p[x]$, jolle pätee $A(x_i) = y_i$, $1 \leq i \leq t$. Polynomi $A(x)$ on Lagrangen interpolaatiopolynomi, ja se voidaan esittää muodossa

$$A(x) = \sum_{\mathrm{j}=1}^{\mathrm{t}} y_j \prod_{1 \leq \mathrm{h} \leq \mathrm{t}, \mathrm{h} \neq \mathrm{j}} \frac{x - x_h}{x_j - x_h}.$$

Edellistä kaavaa on vaikea tulkita pelkällä intuitiolla, mutta on helppo tarkastella miten se toimii asettamalla $x = x_i$. Huomataan nyt, että jokainen summattava saa arvon 0, lukuunottamatta indeksiä $j = i$, jolla tulo-operaatio saa arvon 1 ja $A(x)$ arvon y_i. Tästä seuraa että $A(x_i) = a_i$. Tarkempi todistus sivuutetaan.

Shamirin hajautetun avaimen salasmenetelmässä jakaja valitsee enintään $t-1$ asteisen polynomin, jonka vakioterminä on lopullinen avain k. Nyt t osallista voivat soveltaa Lagrangen interpolaatiopolynomin kaavaa lohkoilla $(x_{\mathrm{n}_1}, y_{\mathrm{n}_1}), (x_{\mathrm{n}_2}, y_{\mathrm{n}_2}), ..., (x_{\mathrm{n}_\mathrm{t}}, y_{\mathrm{n}_\mathrm{t}})$, missä n kuvaa heidän lohkojen muodostamaa ryhmää. Kaavalla he saavat muodostettua polynomin $A(x)$, ja selvitettyä sen vakiotermin eli avaimen k. Polynomin $A(x)$ kaava on nyt

$$A(x) = \sum_{\mathrm{j}=1}^{t} \Big(y_{\mathrm{n}_\mathrm{j}} \prod_{1 \leq \mathrm{h} \leq \mathrm{t}, \mathrm{h} \neq \mathrm{j}} \frac{x - x_{\mathrm{n}_\mathrm{h}}}{x_{\mathrm{n}_\mathrm{j}} - x_{\mathrm{n}_\mathrm{h}}} \Big) \bmod p.$$

Osallisten ei kuitenkaan tarvitse laskea koko polynomia $A(x)$, vaan pelkästään vakiotermi. Käytetään nyt hyväksi tietoa $A(0) = k$, jotta voidaan supistaa kaavaa ja uudeksi kaavaksi saadaan

$$A(0) = k = \sum_{\mathrm{j}=1}^{t} \Big(y_{\mathrm{n}_\mathrm{j}} \prod_{1 \leq \mathrm{h} \leq \mathrm{t}, \mathrm{h} \neq \mathrm{j}} \frac{x_{\mathrm{n}_\mathrm{h}}}{x_{\mathrm{n}_\mathrm{h}} - x_{\mathrm{n}_\mathrm{j}}} \Big) \bmod p.$$

Jos nyt valitaan

$$b_{\mathrm{n_j}} = \prod_{1 \le \mathrm{h} \le \mathrm{t}, \mathrm{h} \neq \mathrm{j}} \frac{x_{\mathrm{n_h}}}{x_{\mathrm{n_h}} - x_{\mathrm{n_j}}} \bmod p,$$

$1 \le j \le t$, niin saadaan

$$k = \sum_{\mathrm{j}=1}^{t} b_{\mathrm{n_j}} y_{\mathrm{n_j}} \bmod p.$$

Nyt avain on lineaarikombinaatio t tunnetusta lohkosta. Täytyy huomata, että jokainen $b_{\mathrm{n_j}}$ on laskettavissa, sillä jokainen tarvittava $x_{\mathrm{n_h}}$ on tunnettu. Havainnollistetaan tätä käyttämällä edellisten esimerkkien järjestelmää.

Esimerkki 2.5. Osalliset r_1, r_3 ja r_5 haluavat avata järjestelmän. Lagrangen interpolaatiopolynomin kaavaa hyödyntääkseen heidän pitää selvittää $b_{\mathrm{n_1}}$, $b_{\mathrm{n_3}}$ ja $b_{\mathrm{n_5}}$. Nyt

$$b_{\mathrm{n_1}} = \prod_{1 \le \mathrm{k} \le \mathrm{t}, \mathrm{k} \neq 1} \frac{x_{\mathrm{n_k}}}{x_{\mathrm{n_k}} - x_{\mathrm{n_1}}} = \frac{x_3 x_5}{(x_3 - x_1)(x_5 - x_1)} = \frac{6 \cdot 2}{(6-3)(2-3)}$$

$$\equiv 5 \cdot 3^{-1} \cdot 6^{-1} \equiv 5 \cdot 5 \cdot 6 \equiv 3 \bmod 7,$$

$$b_{\mathrm{n_3}} = \frac{x_1 x_5}{(x_1 - x_3)(x_5 - x_3)} = \frac{3 \cdot 2}{(3-6)(2-6)} \equiv 6 \cdot 4^{-1} \cdot 3^{-1} \equiv 4 \bmod 7,$$

$$b_{\mathrm{n_5}} = \frac{x_1 x_3}{(x_1 - x_5)(x_3 - x_5)} = \frac{3 \cdot 6}{(3-2)(6-2)} \equiv 4 \cdot 1^{-1} \cdot 4^{-1} \equiv 1 \bmod 7.$$

Nyt he saavat laskettua avaimeksi

$$k = \sum_{\mathrm{j}=1}^{t} b_{\mathrm{n_j}} y_{\mathrm{n_j}} = 3 \cdot 5 + 4 \cdot 6 + 1 \cdot 6 \equiv 3 \bmod 7.$$

Jos nyt $t - 1$ osallista yrittävät aukaista Shamirin hajautetun avaimen salausmenetelmällä luotua järjestelmää, eivät he saa selville mitään avaimesta. Tiedetään, että avain $k \in \mathbb{Z}_p$. Kun käytetään lohkojen kanssa tietoa $A(0) = k$, saadaan t pistettä. Määritelmän 2.4. mukaan t pistettä määrittävät yksikäsitteisen $t - 1$ asteisen polynomin (muistetaan että jakajan luoma $A(x)$ on $t - 1$ asteinen polynomi). Koska tämä on tosi kaikille k, ei voida supistaa mitään mahdollisia arvoja pois mahdollisten avainten joukosta.

Esimerkki 2.6. Osalliset r_1 ja r_2 keräävät lohkonsa ja tekevät yhtälöparin

$$\begin{cases} 5 = a_0 + 3a_1 + 9a_2 \bmod 7 \\ 3 = a_0 + a_1 + a_2 \bmod 7. \end{cases}$$

Nyt a_0 voidaan ratkaista vain joko a_1:n tai a_2:n avulla,

$$a_0 = 3a_2 + 4 \bmod 7$$

tai

$$a_0 = a_1 + 1 \bmod 7.$$

Nyt a_0 voi saada molemmissa tapauksissa kaikkia arvoja modulo 7, joten osalliset eivät saaneet avaimesta mitään tietoa.

3 Triviaalit tapaukset

Hajautetun avaimen (t, w)-salausmenetelmissä voidaan katsoa olevan kaksi triviaali tapausta, $t = 1$ ja $t = w$. Näihin molempiin voidaan soveltaa Shamirin menetelmiä, mutta on mahdollista toteuttaa ne yksinkertaisemmin.

Valitaan $t = 1$. Jakaja valitsee avaimen k, ja antaa sen jokaiselle osalliselle. Nyt kuka vain osallinen saa salauksen auki yksinään, joten hajautetun avaimen-salaus pelkistyy pelkäksi salaisen avaimen salaukseksi.

Jos sen sijaan valitaan $t = w$, päästään tilanteeseen, missä avaimesta voidaan jakaa toisistaan aidosti eroavat lohkot osallisille. Tällöin salausta on mielekästä tutkia hajautetun avaimen salausmenetelmänä, mutta sen toteuttaminen on huomattavasti helpompaa kuin yleisissä tapauksissa. Tarkastellaan nyt tällaisen järjestelmän luomista, kun $K = L = \mathbb{Z}_m$. Tässä triviaali tapauksessa luvun m ei tarvitse olla alkuluku, eikä ole tarpeen, että $m \geq w + 1$.

Algoritmi 3.1. Luodaan yksinkertaistettu (t, t)-salausmenetelmä.
Valitaan *mahdollisten avainten joukoksi* $K = \mathbb{Z}_m$ ja *mahdollisten lohkojen joukoksi* $L = \mathbb{Z}_m$, $m \in \mathbb{N}$. Jakaakseen avaimen k:

1. D valitsee satunnaisesti $t - 1$ alkiota joukosta $\mathbb{Z}_m$, $y_1, y_2 ..., y_{t-1}$.
2. D laskee viimeisen alkion
$$y_t = k - \sum_{i=1}^{t-1} y_i \bmod m.$$
3. D jakaa jokaiselle osalliselle r_i lohkon y_i.

Huomataan, että nyt t osallista saa laskettua avaimen kaavalla

$$k = \sum_{i=1}^{t} y_i \bmod m,$$

mutta selvästi $t - 1$ osallista ei saa. Heillä on hallussa kaikki muut lohkot paitsi satunnainen lohko y_j, ja he saavat laskettua arvon $k - y_j$. Tämä ei kuitenkaan paljasta avaimesta mitään, sillä k ja y_j voivat olla mitä vain modulo m.

Esimerkki 3.2. Triviaalin (5, 5)-salausjärjestelmän luonti

Olkoon $m = 5$ ja $k = 3$.

1. Jakaja D valitsee satunnaisesti joukosta $\mathbb{Z}_5$ lohkot $y_1 = 2, y_2 = 0, y_3 = 1, y_4 = 4$.

2. D laskee viimeisen lohkon

$$y_5 = 3 - \sum_{\mathrm{i}=1}^{4} y_i = 3 - (2 + 0 + 1 + 4) \equiv 1 \bmod 5.$$

3. D jakaa jokaiselle r_i lohkon y_i.

Esimerkki 3.3. Edellisen esimerkin osalliset $r_1, r_2..., r_5$ haluavat selvittää avaimen. He sijoittavat lohkonsa ratkaisu kaavaan ja laskevat

$$k = \sum_{\mathrm{i}=1}^{t} y_i = 2 + 0 + 1 + 4 + 1 \equiv 3 \bmod 5.$$

Esimerkki 3.4. Edellisten esimerkkien osalliset r_1, r_3, r_4 ja r_5 yrittävät selvittää avainta. He sijoittavat lohkonsa ratkaisu kaavaan ja saavat

$$k = \sum_{\mathrm{i}=1}^{t} y_i = 2 + y_2 + 1 + 4 + 1 \equiv 3 + y_2 \bmod 5.$$

Koska y_2 voi olla mitä vain modulo 5, he eivät tiedä avaimesta mitään.

Näiden triviaalien tapauksien lisäksi hajautetun avaimen salausmenetelmiä voi jakaa muihinkin tapauksiin, joihin voi olla yleispätevää ratkaisua tehokkaampi ratkaisu. Shamirin toteutus on suhteellisen yksinkertainen, ja se toimii kaikissa (t, w)-salausmenetelmissä. Joten, jos sitä voi käyttää, on tämän tarkempi jaottelu tarpeetonta. Seuraavassa kappaleessa kuitenkin tarkastellaan tilanteita, missä voi olla tarpeellista käsitellä erikseen esimerkiksi $(2, n)$- ja $(5, n)$-menetelmiä.

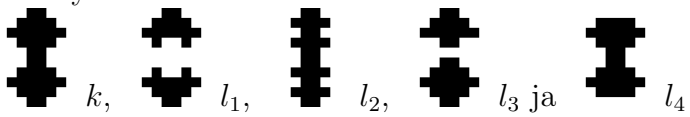
4 Visuaalinen hajautetun avaimen salausmenetelmä

Tähän mennessä olemme käsitelleet menetelmiä missä avain ja sen lohkot ovat jonkin rajallisen joukon alkioita. Naor ja Shamir ovat ehdottaneet että avain voisi myös olla vaikka nelikulmainen, mustista ja valkoisista pikseleistä koostettu kuva I, jonka lohkot ovat sellaisia musta-valkoisia kuvia jotka päällekkäin asettamalla paljastavat avaimen. Käytännössä tämä onnistuu esimerkiksi käyttämällä läpinäkyviä kalvoja.

Käydään ensimmäisenä helppo esimerkki, jossa $t = 2$. Näissä riittää yksinkertaisimmillaan, että jokaisesta lohkosta puuttuu ainakin yksi uniikki pikseli, kunhan kahden yhdisteenä saadaan silti koko avain.

Esimerkki 4.1. Luodaan primitiivinen visuaalinen "(2,4)-salausjärjestelmä".

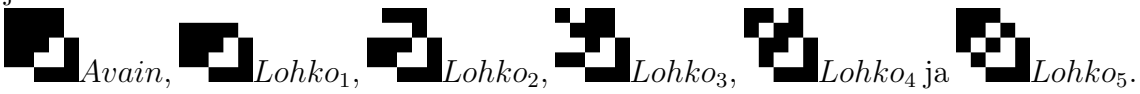
Valitaan avain k, ja luodaan sen avulla lohkot l_1, l_2, l_3 ja l_4. Avain ja lohkot ovat nyt


k, l_1, l_2, l_3 ja l_4

Kun siirrytään tarkastelemaan tapauksia missä $t > 2$, muuttuu kuvien laatiminen haastavemmaksi, sillä nyt jokaisen $(t-1)$ lohkoa sisältävän joukon yhdistelmässä pitää olla jotain avaimesta poikkeavaa, mutta millään t lohkoa sisältävällä yhdisteellä ei.

Esimerkki 4.2. Luodaan primitiivinen visuaalinen "(3,5)-salausjärjestelmä"

Valitaan avain k, ja luodaan sen avulla lohkot l_1, l_2, l_3, l_4 ja l_5. Nyt avain ja lohkot ovat


$Avain$, $Lohko_1$, $Lohko_2$, $Lohko_3$, $Lohko_4$ ja $Lohko_5$.

Nämä edellisten esimerkkien primitiiviset menetelmät ovat kuitenkin varsin kehnoja, eikä niitä voi pitää hajautetun avaimen salausmenetelminä. Hajautetun avaimen salausmenetelmien periaatteiden mukaisesti, millään alle t määräisellä joukolla ei pitäisi saada avaimesta mitään tietoa. Tämä ei kuitenkaan päde edellisten esimerkkien tapauksissa, sillä jokainen musta pikseli paljastaa että avaimessa on sama musta pikseli.

Tämän ongelman korjaamiseksi Naor ja Shamir löysivät ratkaisun. Heidän ratkaisu on siitä hieno, että se pitää mukanaan ominaisuuden missä järjestelmän voi luoda läpinäkyville kalvoille. Käydään seuraavaksi läpi heidän tapaa luoda visuaalinen hajautetun avaimen $(2, 2)$-salausmenetelmä.

Aluksi jakaja valitsee avaimen I aivan kuin primitiivisissäkin tapauksissa. Sen jälkeen luodaan kaksi kuvaa niin, että jokainen alkuperäisen kuvan pikseli korvataan 2x2 ruudukolla. Näistä kuvista saadaan kaksi lohkoa, kun jokaista alkuperäisen kuvan pikseliä kohden käytetään seuraavaa algoritmia.

Algoritmi 4.3. Luodaan $(2, 2)$-salausjärjestelmän lohkot.

1. Alkuperäinen pikseli on valkoinen.
   Valitaan lohkojen vastaaviin ruudukoihin satunnaisesti joko
    ja  , tai  ja .

2. Alkuperäinen pikseli on musta.
   Valitaan lohkojen vastaaviin ruudukoihin satunnaisesti joko
    ja  , tai  ja  .

Kun kaikki pikselit on käyty läpi, kummankaan lohkon mistään pikselistä ei voi päätellä mitään tietoa alkuperäisestä kuvasta. Lohkot yhdistämällä saadaan kuva joka vastaa alkuperäistä sillä erotuksella, että jokainen valkoinen pikseli on korvattu 2x2 ruudukolla jossa toinen sivuista on musta. Tämä menetelmä siis johtaa "kohisevaan" kuvaan, mutta ratkaisee turvallisuus ongelman mikä oli primitiivisissä menetelmissä.

Esimerkki 4.4. Luodaan algoritmillä $(2, 2)$-salausjärjestelmä, kun avaimena on 20x20 px kuva. Lohkot, sekä yhdistetty kuva ovat nyt 40x40 px.

 *Avain*  $Lohko_1$  $Lohko_2$  $Yhdistetty$

Huomataan nyt, että lohkojen yhdistelmästä voi esimerkin tapauksessa nähdä pandan. Yksittäisen pikselin tarkkuudella kuvaa on vaikea tutkia, mutta kunhan avaimena on jokin helposti tunnistettava kuva, se yleensä näkyy selvästi. Jos kuitenkin halutaan lohkoilla saada alkuperäinen kuva, voidaan soveltaa seuraavaa algoritmia jokaiseen yhdistetyn kuvan 2x2 ruudukkoon (joilla korvattiin alkuperäisen kuvan pikselit).

Algoritmi 4.5. Poistetaan yhdistetyn kuvan kohina.

1. Jos 2x2 ruudukko on musta,
   älä tee mitään.

2. Jos 2x2 ruudukko on musta-valkoinen,
   korvaa ruudukon mustat pikselit valkoisilla.

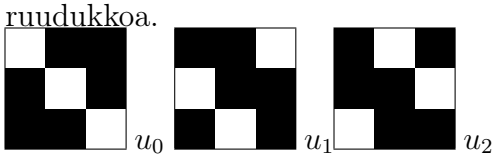
Algoritmissa 4.3. korvattavat ruudukot olivat 2x2, mutta mikään ei rajoita niitä olemasta suurempia. Tämmöisillä suuremmilla ruudukoilla ei sinänsä päästä sen valkoisempiin lopputuloksiin, mutta äärimmäisyyksiin vietynä lopputuloksessa nämä ruudukot vaikuttaisivat harmailta, jolloin yhdistetty kuva olisi selkeämpi (ei tarvitse viedä harmaaseenkaan asti). Huomataan nyt se, että tämän algoritmin tuloksena, lohkojen yhdistelmän valkoiset ruudukot olivat 50 prosenttisesti valkoisia, eli sopivasti $\frac{1}{w} = \frac{1}{2}$. Seuraavaksi tarkastellaan millaisilla muokkauksilla algoritmin saa tuottamaan visuaalisia hajautetun avaimen $(2, w)$-salausjärjestelmiä, missä $w \geq 2$. Tulemme huomaamaan, että näistä saatavien yhdistettyjen kuvien valkoiset ruudukot ovat aina vain $\frac{1}{w}$ valkoisia. Tämä johtaa w kasvaessa tilanteeseen, missä alkuperäistä avainta ei enää näe silmämääräisesti, tai siitä tulee liian suuri. Menetelmä vaatii suurta pikseli laajennusta, joten se ei ole laskennallisesti tehokkain. Se kuitenkin toteuttaa hajautetun avaimen salausmenetelmien periaatteita oikeellisesti, ja se on suhteellisen helppo toteuttaa ohjelmoimalla.

Jakaja valitsee avaimeksi kuvan I, ja jakaa sen lohkoihin niin, että jokainen pikseli on korvattu YxY ruudukolla, missä $Y = w$. Nyt hän valitsee ruudukoihin w toisistaan eroavaa kuvaa, joista jokaisen kahden yhdistelmällä saadaan kokonaan musta ruudukko. Nyt sovelletaan tätä ja luodaan algoritmi $(2, w)$-salausmenetelmälle.

Algoritmi 4.6. Luodaan $(2, w)$-salausjärjestelmän lohkot.

1. Luodaan w toisistaan eroavaa ja w pituista riviä r niin, että r_0:n ensimmäinen pikseli on valkoinen ja loput mustia, r_1:n toinen pikseli on valkoinen ja loput mustia..., $r_{\mathrm{w}-1}$:n viimeinen pikseli on valkoinen ja loput mustia.

2. Luodaan ruudukko u_0, asettamalla sen ensimmäiseksi riviksi r_0, toiseksi r_1, ja niin edelleen kunnes viimeiseksi riviksi tulee $r_{\mathrm{w}-1}$. Seuraava ruudukko u_2 luodaan siirtämällä edellisen ruudukon viimeinen rivi ensimmäiseksi (ensimmäisestä tulee toinen). Jatketaan tätä kunnes on w ruudukkoa.



3. Käydään läpi alkuperäisen kuvan pikseleitä.

   3.1. Alkuperäinen pikseli on valkoinen.
   Valitaan sama satunnainen ruudukko u_i korvaamaan pikseli jokaisessa lohkossa.

   3.2. Alkuperäinen pikseli on musta.
   Valitaan satunnaisesti numero x, niin että $0 \leq x \leq w-1$. Ensimmäiseen lohkoon valitaan u_x korvaamaan alkuperäistä pikseliä, toiseen lohkoon u_{x+1}..., viimeiseen lohkoon u_{x-1} (ruudukoiden indeksit ovat modulo w).

Esimerkki 4.7. Luodaan algoritmillä $(2, 3)$-salausjärjestelmä, kun avaimena on 20x20 px kuva. Lohkot, sekä yhdistetty kuva ovat nyt 60x60 px.


Avain

$Lohko_0$

$Lohko_1$

$Lohko_2$


Yhdistetty kuva mistä tahansa lohkojen kombinaatiosta

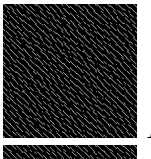
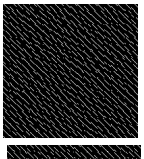
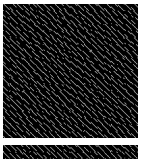
Esimerkki 4.8. Luodaan algoritmillä $(2, 10)$-salausjärjestelmä, kun avaimena on 20x20 px kuva. Lohkot, sekä yhdistetty kuva ovat nyt 200x200 px.


Avain

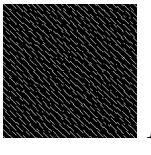
$Lohko_0$

$Lohko_1$

$Lohko_2$


$Lohko_3$

$Lohko_4$

$Lohko_5$

$Lohko_6$


$Lohko_7$

$Lohko_8$

$Lohko_9$


Yhdistetty kuva mistä tahansa lohkojen kombinaatiosta

Tässä tutkielmassa ei mennä tämän pidemmälle visuaalisissa hajautetun avaimen salausmenetelmissä. Tutustutuimme siis aluksi Shamirin ja Naorin visuaaliseen $(2, 2)$-salausmenetelmään, ja laajennettiin se toimimaan kaikissa tapauksissa $(2, w)$. Tutkimusta on tehty edellisten tapojen optimoinnista, ja on myös esitetty muita tapoja luoda samankaltaisia järjestelmiä. Jos esimerkiksi otetaan työkaluksi tietotekniikan puolelta tuttu XOR-käsite, voidaan luoda järjestelmiä, joiden pikselilaajennusta voidaan supistaa merkittävästi, ilman että yhdistetyn kuvan laatu kärsii.

Toinen suunta mihin tästä työstä voi jatkaa, on visuaalisen hajautetun avaimen (t, t)-salausmenetelmät, ja sen jälkeen (t, w)-salausmenetelmät. Näistä ainakin (t, t)-salausmenetelmät on toteutettavissa samankaltaisilla järjestelmillä, mitä on käsitelty tässä työssä, mutta niissä yhdistettyjen kuvien selkeus menee selvästi huonommaksi kuin $(2, w)$-salausmenetelmissä. Lohkojen yhdistelmän mustiksi tarkoitetut ruudukkojen kaikki pikselit ovat siis mustia, mutta valkoisiksi tarkoitettujen ruudukoiden pikseleistä valkoisia on vain hyvin pieni osa. Omissa toteutuksissani pääsin $(3, 3)$-salausmenetelmään, jonka valkoiset ruudukot oli $\frac{1}{5}$ valkoisia, ja vaadittava pikselilaajennus oli viisinkertainen.

5 Yhteenveto

Jos haluamme hajautetun avaimen salausmenetelmillä jakaa avaimen joka on kokonaisluku, voimme käyttää Shamirin menetelmiä. Sen etuina on:

- 1. Kvanttiturvallisuus. Salaus ei ole kiinni laskentatehosta.
- 2. Koko. Lohkot ovat samassa kokoluokassa kuin avain itse.
- 3. Laajennettavuus. Lohkoja on mahdollista lisätä ja poistaa, ilman että järjestelmää pitää uusia.
- 4. Hierarkia. Osallisille voi antaa eri määrän lohkoja, jolloin esimerkiksi johtajalla voi olla isompi vastuu kuin alaisella.

Silläkin on kuitenkin puutteensa. Turvallisuutta heikentää se, että lohkon aitoutta ei voi todistaa, ja että avain on kokonaisena sekä alussa että lopussa. On olemassa menetelmiä jotka korvaavat ainakin yksi kerrallaan näitä ongelmia. Jos nämä ominaisuudet ovat tärkeitä, joudutaan käyttämään jotain näistä muista menetelmistä.

Visuaalisissa hajautetun avaimen menetelmissä Shamirin keino, mikä esiteltiin lähdeteoksessa, on hyvin alkeellinen, ja toimii vain $(2, 2)$-menetelmässä. Lopussa esitelty $(2, n)$-menetelmä on sekä parannus että laajennus tähän Shamirin menetelmään, mutta siinäkin on ongelmansa. Osallisten n kasvaessa, lohkojen pikseleiden määrä kasvaa eksponentiaalisesti. Jos tämä on ongelma, tai jos tarvitaan yleisemmin toimivaa menetelmää, joudutaan tutustumaan edistyksellisempiin tapoihin. On täysin mahdollista, että näin saataisiin parempia tuloksia jo pienillä luvuilla n, eli saavutettaisiin pienempiä pikselilaajennuksia ja parempaaa selkeyttä yhdistetyille kuville.

Lähdeluettelo

[1] D. Stinson, M. Paterson: *Cryptography : theory and practice IV*. CRC Press, 2018.