RSA-salausmenetelmä

LuK-tutkielma

Tapani Sipola

Op. nro. 1976269

Matemaattisten tieteiden laitos

Oulun yliopisto

Syksy 2017

Sisältö

Johdanto

Tietotekniikan kehittyessä ja yleistyessä myös käyttäjältä toiselle siirrettävän yksityisen tiedon määrä on kasvanut. Miljoonien ihmisten henkilö- ja pankkititoja siirtyy Internetin ja muiden tietoverkkojen kautta päivittäin verkkopankeille, -kaupoille ja muille käyttäjille. On sanomattakin selvää, että tällaisen viestinnän salaminen on ensisijaisen tärkeää näiden palvelujen asiakkaiden yksityisyyden ja turvallisuuden kannalta.

Jotta viestintä voitaisiin pitää salassa on viestivien osapuolten sovittava salauksen käytöstä. Perinteisten salausmenetelmien ongelmana on kuitenkin, että salauksesta sopiminen vaatii itsessään turvallisen viestintäkanavan, ja jos tällainen kanava olisi helposti käytettävissä, ei salausmenetelmälle olisi tarvetta. Julkisen avaimen salausmenetelmät kuitenkin mahdollistavat viestinnän salaamisen ilman että lähettäjän olisi sovittava salauksesta vastaanottajan kanssa erikseen.

Tämä tutkielma käsittelee RSA-salausta, joka on tunnetuimpia julkisen avaimen salausmenetelmiä. Sen kehittivät Ron Rivest, Adi Shamir ja Leonard Adleman vuonna 1977. Sen turvallisuus pohjautuu siihen, että suurten lukujen jakaminen tekijöihin on huomattavasti vaikeampaa kuin niiden kertominen keskenään. [1, 2]

Aluksi kerrotaan matemaatisesta salakirjoituksesta yleisesti ja esitellään tarvittavia määritelmiä, lauseita ja algoriritmeja. Sitten esitellään RSA-algoritmi ja todistetaan sen virheettömyys. Lopuksi perustellaan miksi RSA on turvallinen salausmenetelmä.

1 Salausmenetelmien yleisiä periaatteita

Salakirjoituksen takoituksena on selväkielisen viestin muuntaminen sellaiseen muotoon, että vain viestin vastaanottaja voi purkaa ja lukea viestin. Jotta selväkielistä viestiä voitaisiin käsitellä matemaattisesti, se on aluksi muunnettava numeeriseen muotoon. Ennen tietokoneita tämä suoritettiin esimerkiksi taulukoilla, joilla kukin kirjain liitettiin johonkin numeroon. Käsiteltäessä viestiä tietokoneella viestitiedoston binääridataa voidaan käsitellä salausmenetelmällä suoraan, joten tälle ei ole nykyään tarvetta.

Olkoon M viestiyksiköiden joukko ja C salattujen viestiyksiköiden joukko. Valitaan salausfunktio $E : M \to C$, joka on bijektio. Nyt $D = E^{-1}$ on avausfunktio. Selväkielinen viesti muutetaan viestiyksiköksi $m \in M$, josta lasketaan salattu viesti $c = E(m) \in C$. Salattu viesti c lähetetään vastaanottajalle, joka purkaa salauksen laskemalla $D(c) = m$ ja muuttaa tämän selväkieliseksi. Funktioiden E ja D tulee olla suhteellisen helposti laskettavissa ja viesti m ei saa selvitä salakirjoituksesta c ilman että avausfunktio D tunnetaan. Useimmiten viesti hajotetaan useammaksi viestiyksiköksi, jotka kaikki salataan ja avataan erikseen, esimerkiksi kirjain kirjaimelta tai tietyn merkkimäärän pituisissa "blokeissa".

Useimmiten salaus- ja avausfunktiot muodostetaan niinsanotun salausavaimen avulla. Salausavain k on parametri, joka määrittää salaus- ja avausfunktiot E_k ja D_k, joilla $D_k = E_k^{-1}$ kaikilla $k \in K$. K on niin sanottu avainjoukko.
Esimerkki 1.1. Caesar-salauksessa korvataan tekstin jokainen kirjain sillä kirjaimella, joka on aakkosissa ennalta sovitun määrän askelia alkuperäisen kirjaimen jälkeen. Aakkoston jokainen kirjain numeroidaan, esimerkiksi

englannin aakkostossa "A" = 0, "B" = 1,..., "Z" = 25. Salausfunktio on $E_k(m) = (m + k) \pmod{26}$ ja avausfunktio on $D_k(c) = (c - k) \pmod{26}$. Luku $k \in \{1, 2, \ldots, 25\}$ on salausavain, joka määrittää käytettävät salaus- ja avausfunktiot, eli sen kuinka monta askelta aakkostossa kuljetaan eteen tai taakse.

Salausavaimen tarkoitus on helpottaa salauksesta sopimista ja salauksen muuttamista. Koska avain määrittelee sekä avaus- että salausfunktion, näistä sopimiseksi riittää että kullakin osapuolella on avain tiedossa. Tämän takia salauksesta sopimista sanotaan yleensä avaimenvaihdoksi.

Kuten johdannossa mainittiin, perinteisissä salausmenetelmissä avaimenvaihto on suoritettava turvallista kanavaa pitkin. Mikäli ulkopuolinen saa avaimen tietoonsa, hänen on helppo muodostaa avausfunktio ja lukea salatut viestit. Julkisen avaimen salausmenetelmät on kuitenkin suunniteltu siten, että avausfunktio on huomattavan työläs laskea salausavaimesta. Tällöin salausavain voidaan lähettää tarvittaessa täysin julkista kanavaa pitkin. Vaikka avain joutuisikin vääriin käsiin, avausfunktion laskemiseen menisi kuukausia tai jopa vuosia, missä vaiheessa salatun viestin paljastumisesta ei enää olisi välttämättä haittaa viestin lähettäjälle ja vastaanottajalle.

Julkisen avaimen salausmenetelmä tarvitsee kuitenkin toimiakseen niin sanotun salaluukun. Salaluukku on tieto, joka tekee avausfunktion laskemisen salausavaimesta helpoksi. Ilman tätä tietoa avausfunktion luominen olisi sen omistajalle yhtä raskasta kuin viestin murtajalle, mikä tekisi salauksesta hyvin kömpelöä. Funktio jonka luomiseen olisi käytetty vuosi olisi seuraavana vuonna jo murrettu.

2 Määritelmiä ja lauseita

Määritelmä 2.1 (Kongruenssi)**.** Olkoon a, b ja c kokonaislukuja. Luku a on *kongruentti* luvun b kanssa modulo c, jos ja vain jos $c|a-b$. Tällöin käytetään merkintää

$$a \equiv b \pmod{c}.$$

Määritelmä 2.2 (Eulerin funktio)**.** Eulerin funktion $\varphi : \mathbb{N} \to \mathbb{N}$ arvo on niiden positiivisten kokonaislukujen k lukumäärä joille pätee

$$0 < k \leq n, \quad \mathrm{syt}(k, n) = 1$$

Lause 2.3 (Eulerin-Fermat'n Lause)**.** *Olkoon* $a, n \in \mathbb{Z}_+$. *Jos* $\mathrm{syt}(a, n) = 1$, *niin*

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Jatkossa tähän lauseeseen viitataan Eulerin lauseena.

3 Nopea potenssiinkorotus

RSA-salaus vaatii tehokkaan tavan laskea hyvin suurien potenssien kongruensseja. Tätä varten käytetään nopean potenssiinkorotuksen menetelmää, joka toimii seuraavasti.

Olkoon tehtävänä laskea $a^p \pmod{n}$, missä $p > \varphi(n)$, $a < n$ ja n on suuri. Selvästi $p = q\varphi(n) + r$ joillakin $q \in \mathbb{Z}, r \in \mathbb{N}$ ja $r < \varphi(n)$. Tästä saadaan

$$a^p = a^{q\varphi(n)+r} = a^{q\varphi(n)}a^r,$$

josta edelleen Eulerin lauseen avulla saadaan

$$a^{q\varphi(n)}a^r \equiv 1^q a^r \equiv a^r \pmod{n}.$$

Tehtävässä potenssi p voidaan siis korvata pienemmällä potenssilla r jolle pätee $p \equiv r(\mathrm{mod}\ \varphi(n))$. [1]

Esimerkki 3.1. Lasketaan $5^{100000000000000} \pmod{12830603}$. Nyt

$$\varphi(12830603) = \varphi(3571) * \varphi(3593) = 3570 * 3592 = 12823440$$

ja

$$100000000000000 = 7798219 * 12823440 + 6546640$$

jolloin

$$5^{100000000000000} = 5^{7798219*12823440+6546640} \equiv 5^{6546640} \pmod{12830603}. [1]$$

Seuraavaksi selvitetään eksponentin r binääriesitys

$$\sum_{i=0}^{k} r_i 2^i = r,$$

missä $r_i \in \{0,1\}$ kaikilla $0 \leq i < k$ ja $r_k = 1$.

Tämän jälkeen lasketaan luvut $a_i \equiv a^{2^i} \pmod{n}$, $0 \leq i \leq k$ rekursiivisesti kaavoista

$$a_0 = a,$$
$$a_i \equiv a_{i-1}^2 \pmod{n}.$$

Lukujen a_i avulla lasketaan

$$\begin{aligned} a^r &= a^{r_k 2^k + r_{k-1} 2^{k-1} + \cdots + r_0} \\ &= (a^{2^k})^{r_k} (a^{2^{k-1}})^{r_{k-1}} \ldots (a^1)^{r_1} \\ &\equiv a_k^{r_k} a_{k-1}^{r_{k-1}} \ldots a_0^{r_0} \pmod{n}. \end{aligned}$$

Tällä tavalla, sen sijaan että jouduttaisiin kertomaan a itsellään r kertaa, tarvitsee vain laskea $\lfloor \log_2 r \rfloor$ luvun neliöt ja korkeintaan yhtä monen luvun tulo. [1]

Esimerkki 3.2. Lasketaan $7^{85} \pmod{391}$. Koska $85 = 2^6 + 2^4 + 2^2 + 1 = 64 + 16 + 4 + 1$, riittää laskea luvut a_i, $0 \leq i \leq 6$. Tällöin saadaan

$$\begin{aligned} a_1 &\equiv 7^2 && \equiv 49 \pmod{391}, \\ a_2 &\equiv 7^4 && \equiv 49^2 && \equiv 55 \pmod{391}, \\ a_3 &\equiv 7^8 && \equiv 55^2 && \equiv 288 \pmod{391}, \\ a_4 &\equiv 7^{16} && \equiv 288^2 && \equiv 52 \pmod{391}, \\ a_5 &\equiv 7^{32} && \equiv 52^2 && \equiv 358 \pmod{391} \text{ ja} \\ a_6 &\equiv 7^{64} && \equiv 358^2 && \equiv 307 \pmod{391}. \end{aligned}$$

Tämän jälkeen on helppo laskea

$$7^{85} = 7^{64+16+4+1} \equiv 307 \cdot 52 \cdot 55 \cdot 7 \equiv 324 \cdot 385 \equiv 11 \pmod{391}. \text{ [1]}$$

4 RSA-salaus

4.1 Toimintaperiaate

Jokaiselle käyttäjälle muodostetaan julkinen ja yksityinen avain seuraavasti:

1. Valitaan kaksi suurta (vähintään 100 numeroista) alkulukua p ja q. Näiden alkulukujen pitää täyttää tiettyjä kriteerejä, joista puhutaan myöhemmin.
2. Lasketaan $n = pq$.
3. Valitaan luku e, jolle pätee $\mathrm{syt}(e, \varphi(n)) = 1$
4. Lasketaan luku d, jolla $de \equiv 1 \pmod{\varphi(n)}$
5. Julkaistaan salausavain (n, e) ja pidetään avausavain d vain avaimen omistajan tiedossa.

Olkoon $m = (m_1, m_2, \dots, m_k)$ lähetettävä viesti, missä $2 \leq m_i < n$. Kustakin visestin osasta m_i muodostetaan salakirjoituksen osa c_i laskemalla kongruenssi $c_i \equiv m_i^e \pmod{n}$, missä (e, n) on viestin vastanottajan salausavain. Salattu viesti $c = (c_1, c_2, \dots, c_k)$ lähetetään vastaanottajalle. Purkaakseen salauksen vastaanottaja käyttää avausavaintaan d laskiessaan kongruenssin $c_i^d \pmod{n}$, jolle tullaan osoittamaan että $c_i^d \equiv m_i \pmod{n}$

Lause 4.1. *Olkoon p ja q erillisiä alkulukuja ja $n = pq$. Tällöin $m^{l\varphi(n)+1} \equiv m \pmod{n}$ kun $m \in \mathbb{Z}$ ja $l \in \mathbb{N}$*

Todistus. Kun $\mathrm{syt}(m, n) = 1$ väite pätee Eulerin lauseen nojalla

$$m^{\varphi(n)} \equiv 1 \pmod{n} \quad \Rightarrow \quad m^{l\varphi(n)} \equiv 1 \pmod{n} \quad \Rightarrow \quad m^{l\varphi(n)+1} \equiv m \pmod{n}.$$

Olkoon $\mathrm{syt}(m, n) \geq 1$. Koska p ja q ovat alkulukuja, täytyy olla että $\mathrm{syt}(m, n) =$

p tai syt$(m,n) = q$ tai syt$(m,n) = n$. Oletetaan että $syt(m,n) = pq = n$. Tällöin

$$m = kn \equiv k \cdot 0 \equiv 0 \pmod n$$

jollakin $k \in \mathbb{Z}$, ja siten

$$m^{l\varphi(n)+1} \equiv 0^{l\varphi(n)+1} \equiv 0 \equiv m \pmod n$$

Oletetaan sitten että $syt(m,n) = q$. Nyt

$$m \equiv 0 \pmod q \Rightarrow m^{l\varphi(n)+1} \equiv m \pmod q \Leftrightarrow q \mid m^{l\varphi(n)+1} - m.$$

Toisaalta Eulerin lauseen nojalla

$$m^{p-1} \equiv 1 \pmod p \Rightarrow m^{(q-1)(p-1)} \equiv 1 \pmod p.$$

Koska $\varphi(n) = (q-1)(p-1)$, niin

$$p \mid m^{l\varphi(n)+1} - m.$$

Koska sekä $q \mid m^{l\varphi(n)+1} - m$, että $p \mid m^{l\varphi(n)+1} - m$ niin

$$pq \mid m^{l\varphi(n)+1} - m \Leftrightarrow m^{l\varphi(n)+1} \equiv m \pmod n.$$

Tapaus syt$(m,n) = p$ todistetaan vastaavasti kuin tapaus $syt(m,n) = q$.

$\square$

Nyt lausen 4.1 nojalla

$$c_i^d = (m_i^{e_B})^{d_B} = m_i^{e_B d_B - 1 + 1} = m_i^{l\varphi(n_B)+1} = (m_i^{\varphi(n)})^l m_i \equiv (1)^l m_i \equiv m_i \pmod n$$

ja vastaanottaja voi lukea viestin m.

Salauksen lisäksi RSA-salausta voidaan käyttää myös viestien allekirjoittamiseen. Tämä tapahtuu laskemalla numeerisesta viestistä tiivisteluku, joka salataan lähettäjän yksityisellä avaimella d ja liitetään viestiin. Vastaanottaja avaa tiivisteluvun lähettäjän julkisella avaimella e ja laskee varsinaisesta viestistä tiivisteluvun. Jos viestistä laskettu luku on sama kuin allekirjoituksen luku, viestin tiedetään olevan julkisen avaimen omistajan lähettämä ja että viestiä ei ole muunneltu, sillä oletuksella että tiiviste on laskettu turvallisella funktiolla. Yleensä viestit sekä allekirjoitetaan että salataan.

4.2 Salauksen murtaminen

Ainoa tunnettu tapa RSA-salauksen murtamiseen on selvittää joku luvuista p, q ja $\varphi(n)$. Jos yksi näistä löydetään, muut luvuista voidaan laskea välittömästi löydetyn arvon pohjalta. Käytännössä tämä tarkoittaa luvun n jakamista tekijöihinsä, mikä on parhaimmillankin luvun n pituuden suhteen ylipolynomista kompleksisuusluokkaa oleva ongelma. Koska RSA-algoritmi on korkeintaan polynomista kompleksisuusluokkaa, tarpeeksi pitkällä avaimella salattua viestiä on käytännössä mahdoton murtaa perinteisellä laskennalla.

On kuitenkin olemassa useita erilaisia heikkouksia, joita hyökkääjä voi hyödyntää mikäli avaimet on valittu huolimattomasti. Esimerkiksi jos alkuluvut p ja q on valittu jostakin tunnetusta taulukosta, hyökkääjän on helppo käydä taulukon alkiot läpi ja löytää tekijät. Jos p ja q ovat liian lähellä toisiaan (ja $p > q$), hyökkääjä voi löytää ne helposti neliöseulamnetelmän avulla. Tällöin $(p-q)/2$ on pieni ja $(p+q)/2$ on vain hieman suurempi kuin $\sqrt{n}$. Koska

$$\left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = pq = n$$

hyökkääjän riittää etsiä sellainen kokonaisluku $x > \sqrt{n}$, jolla $x^2 - n = y^2$ jollakin kokonaisluvulla y. Tällöin $x = (p+q)/2$, $y = (p-q)/2$ ja $n = x^2 - y^2 = (x-y)(x+y)$, joten $p = x + y$ ja $q = x - y$. [2]

Esimerkki 4.2. $\sqrt{97343} = 311,998$ ja $312^2 - 97343 = 1$, joten

$$97343 = (312 - 1)(312 + 1) = 311 * 312$$

Alkulukujen p ja q valinnassa on myös otettava huomioon muiden mahdollisien avausavaimien määrä. Salausavaimella e salatun viestin voi purkaa millä tahansa luvulla d, jolla $de \equiv 1 \pmod{\text{pyj}(q-1, p-1)}$. Näin ollen mitä pienempi lukujen $p-1$ ja $q-1$ pienin yhteinen jaettava on, sitä useampi toimiva avausavain on olemassa ja sitä helpompi salaus on murtaa satunnaisella arvauksella. [2]

Tällaisia mahdollisia heikkouksia on useita, mutta ne ovat vältettävissä huolellisella avaimenvalinnalla. Edellämainittujen lisäksi esimerkiksi toistetulla salauksella on mahdollista murtaa jotkin avaimet.

Kvanttitietokoneella olisi teoriassa mahdollista suorittaa tekijöihin jako polynomisessa ajassa ja näin murtaa RSA tehokkaasti. Kvanttiteknologia ei kuitenkaan kehittynyt tarpeeksi jotta se voisi käsitellä niin isoja lukuja kuin RSA käyttää.

Lähdeluettelo

[1] J. H. Silverman: *A Friendly Introduction to Number Theory*. Pearson Prentice Hall, USA, 2006.

[2] Arto Salomaa: *Public-Key Cryptography*. Springer, Saksa, 1996

[3] Leinonen, Rinta-aho, Matala-aho, Väänänen: Salausmenetelmät kurssin luentorunko, Oulun yliopisto 2015