RSA-algoritmin pätevyyden todistus

Ron Rivest, Adi Shamir ja Leonard Adleman esittivät vuonna 1978 salakirjoitusmenettelyn, jossa tietylle henkilölle osoitetut viestit voidaan salakirjoittaa hänen ilmoittamallaan *julkisella avaimella* ja hän yksin voi lukea ne vain omassa tiedossaan olevan *salaisen avaimen* avulla. Menettely sai nimekseen *RSA-salakirjoitus* esittäjien sukunimien alkukirjaimien mukaan.

Salattava viesti muunnetaan ensin numeeriseen muotoon jollakin tavalla. Tätä varten viestin merkit yleensä ryhmitetään ja kutakin ryhmää tulee vastaamaan positiivinen kokonaisluku. Vastaavuus on injektiivinen, ts. kokonaisluku voidaan yksikäsitteisellä tavalla muuntaa takaisin merkkiryhmäksi. Varsinainen salausalgoritmi kohdistetaan erikseen jokaiseen kokonaislukuun.

Salausalgoritmi perustuu isojen alkulukujen käyttöön. Mitä isompia luvut ovat, sitä vaikeammin murrettava algoritmi saadaan, mutta sitä enemmän tarvitaan myös laskentatyötä. Lukujen 10-järjestelmäesityksessä on tyypillisesti useita satoja numeroita. Algoritmin teorian kannalta lukujen suuruudella ei kuitenkaan ole merkitystä.

Olkoot p ja q kaksi alkulukua ja olkoon näiden tulo $n = pq$.

Valitaan kokonaisluvut e ja d siten, että

$$\mathrm{syt}(e, (p-1)(q-1)) = 1 \quad \text{ja} \quad de \equiv 1 \pmod{(p-1)(q-1)}.$$

Salausavain muodostuu tällöin luvuista e ja n, purkuavain luvuista d ja n. Viesti (kokonaisluku) v muunnetaan salattuun muotoon c kaavalla

$$c \equiv v^e \pmod{n}.$$

Tässä on oltava $v < n$. Salauksen purku tapahtuu kaavalla

$$v \equiv c^d \pmod{n}.$$

Salauksen pitävyys perustuu siihen, että luvun d laskeminen on vaikeata, vaikka n ja e tunnettaisiin. Laskeminen edellyttää nimittäin lukujen p ja q tuntemista, ts. luvun n tekijöihin jakoa. Tekijöihin jakoon ei tunneta mitään nopeaa algoritmia, vaan laskenta kestää nopeilla tietokoneillakin kuukausia, jos luvuissa p ja q on satoja numeroita. Toisaalta ei ole myöskään todistettu, että nopeaa algoritmia tekijöihin jakoon ei ole olemassa.

RSA-algoritmin pätevyys voidaan osoittaa näyttämällä, että $v' \equiv v \pmod{n}$, jos

$$c \equiv v^e \pmod{n} \quad \text{ja} \quad v' \equiv c^d \pmod{n}.$$

Purkamalla kongruenssit saadaan seuraavaa, missä merkinnät k_i tarkoittavat kokonaislukuja:

$$\begin{aligned} c &= v^e + k_1 n; \\ v' &= c^d + k_2 n = (v^e + k_1 n)^d + k_2 n = v^{ed} + k_3 n \\ &= v^{1+k_4(p-1)(q-1)} + k_3 n = v \cdot \left(v^{(p-1)(q-1)}\right)^{k_4} + k_3 n, \end{aligned}$$

koska $ed \equiv 1 \pmod{(p-1)(q-1)}$.

Lukuteoreettisen Eulerin lauseen mukaan on $v^{\varphi(n)} \equiv 1 \pmod{n}$, missä φ on Eulerin funktio ja $\mathrm{syt}(v, n) = 1$. Koska n on kahden alkuluvun p ja q tulo, on $\varphi(n) = (p-1)(q-1)$. Tällöin on

$$\begin{aligned} v' &= v \cdot \left(v^{(p-1)(q-1)}\right)^{k_4} + k_3 n \\ &= v(1 + k_5 n)^{k_4} + k_3 n = v(1 + k_6 n) + k_3 n = v + k_7 n. \end{aligned}$$

On siis $v' \equiv v \pmod{n}$, jos $\mathrm{syt}(v, n) = 1$.

Jäljellä on enää tapaukset $v = p$ ja $v = q$, koska joka tapauksessa on $v < n$. (Itse asiassa tulos $v' \equiv v \pmod{n}$ pätee kaikilla arvoilla v, mutta jos $v \geq n$, salausalgoritmi ei toimi muusta syystä.) Jos $v = p$, voidaan laskea seuraavasti:

$$\begin{aligned} v' &= v \cdot \left(v^{q-1}\right)^{k_4(p-1)} + k_3 n = p \cdot \left(p^{q-1}\right)^{k_4(p-1)} + k_3 n \\ &= p(1 + k_5 q)^{k_4(p-1)} + k_3 n = p(1 + k_6 q) + k_3 n \\ &= p + k_6 pq + k_3 n = p + k_7 n = v + k_7 n. \end{aligned}$$

Tässä on käytetty tulosta $p^{q-1} \equiv 1 \pmod{q}$, mikä seuraa Fermat'n pienestä lauseesta (joka on Eulerin lauseen erikoistapaus). Tässäkin tapauksessa on siis $v' \equiv v \pmod{n}$. Tapaus $v = q$ on täysin analoginen.

Harjoitustehtäviä

Osoita, että $v' \equiv v \pmod{n}$ myös, jos $v \geq n$. Miksi RSA-algoritmissa ei kuitenkaan voida sallia, että $v \geq n$?

Simo K. Kivelä 03.02.2005