Salaiset vaalit ja matemaattinen kryptografia[1]

HANNU NURMI JA ARTO SALOMAA

ABSTRACT
Secret ballot elections and mathematical cryptography

We outline a voting system based on modern mathematical cryptography. The system is designed to satisfy the following conditions in addition to those traditionally imposed upon secret balloting systems: a. the balloting may take place in computer networks, b. each voter may check that his/her vote has been correctly counted, c. each voter may cancel his/her previous vote and give it to another candidate within a specified period of time, and d. a voter may correct the mistakes made by the system. We argue that all these conditions can be satisfied without jeopardizing the ballot secrecy.

1. Johdanto

Salaiset vaalit kuuluvat demokratiaan, vaikka tuolloin tällöin näkee väitettävän, että demokratia toimisi paremmin, jos kaikki äänestykset olisivat julkisia. Argumentit julkisen äänestyksen puolesta ovat heikoimmillaan silloin, kun äänestyksiin mennään kollektiivisten hyödykkeiden hankintapäätösten tekemiseksi. Mutta myös henkilövaaleissa voidaan salaista äänestämistä pitää vapaan mielipiteenmuodostuksen välttämättömänä ehtona, ja juuri siitähän vaaleissa pitäisi olla kysymys.

Vaalien salaisuutta vaalitaan nykyjärjestelmissä monin tavoin. Äänestäminen tapahtuu erityisesti tähän tarkoitukseen varatuilla paikoilla vaaliviranomaisten huolehtiessa toimituksen häiriöttömyydestä. Vaaliviranomaiset valitaan siten, ettei mikään poliittinen katsomuskanta olisi yksinomaisesti edustettuna. Erityistä huolta kannetaan siitä, ettei vaalitulosten laskennassa olisi mahdollista tunnistaa eri äänestäjien vaalilippuja.

Vaalilautakuntien ja muiden vaaliviranomaisten monijäsenisyydellä katsotaan voitavan estää kollektiivinen vaalivilppi, ts. menettely, jossa tietyn äänestysalueen vaalilautakunnan jäsenet manipuloivat ko. alueen vaalin tulosta yhteisestä sopimuksesta lisäämällä tai poistamalla tietyille ehdokkaille annettuja ääniä oikeat äänet väärennettyihin vaihtaen.

Seuraavassa esitellään matemaattisen kryptografian tuloksiin tukeutuvia menettelyjä salaisten vaalien toimeenpanemiseksi tietokoneverkoissa. Pyrkimyksenä on sellainen järjestelmä, joka takaa nykyjärjestelmien edut vähentäen samalla vaaliviranomaisten vilpin mahdollisuuksia.

2. Mitä ominaisuuksia haluamme?

Asetamme tyydyttävälle vaalimenettelylle seuraavat vaatimukset:

1. Vain kukin äänestäjä itse tietää, mitä ehdokasta tai vaihtoehtoa hän on äänestänyt. Hieman yleisemmin muotoiltuna: vain kukin äänestäjä itse tietää valitsemansa äänestysstrategian. Tätä ilmeistä vaatimusta voidaan pitää salaisten vaalien määrittelevänä ominaisuutena.

[1] Suomen Akatemia on tukenut tässä artikkelissa raportoitua työtä.

2. Vain äänioikeutetut äänestäjät saavat äänestää.

3. Kultakin äänioikeutetulta hyväksytään vain yksi ääni. Tavallisimmat käytössä olevat vaalimenettelyt täyttävät kaikki yllä mainitut ehdot. On erityisesti syytä korostaa, että äänioikeutettujen ja äänioikeuttaan käyttäneiden luettelon julkisuudella on tärkeä tehtävä paitsi ehtojen 2 ja 3 takaamisessa myös sellaisen vaalivilpin ehkäisemisessä, jossa annettuihin ääniin esim. laskentavaiheessa lisätään todellisuudessa äänioikeuttaan käyttämättömien »ääniä».

Seuraavat lisäehdot täyttäviä menettelyjä sitävastoin ei ole käytössä.

4. Äänestäminen voidaan suorittaa tietokoneverkossa, joten kukin äänestäjä voi vapaasti valita päätteen, jonka ääressä äänensä antaa.

5. Kukin äänestäjä voi itse tarkistaa, onko hänen äänensä laskettu oikein, ts. sen ehdokkaan tai vaihtoehdon hyväksi, jota hän on tarkoittanut. Yleisemmin ottaen äänestäjän on voitava tarkistaa, että laskentajärjestelmä on oikein ymmärtänyt hänen äänestysstrategiansa.

6. Jokainen äänisoikeutettu voi tietyn ajan kuluessa muuttaa mieltään äänestysstrategiastaan, ts. hän voi vaihtaa antamansa äänen toiseen vaalisalaisuuden vaarantumatta.

7. Jos äänestäjä huomaa äänensä tulleen lasketuksi väärin, hän voi huomauttaa laskentajärjestelmälle tästä vaalisalaisuuden säilyttäen.

Vaatimukset 1.—7. täyttävän järjestelmän konstruoinnissa käytetään hyväksi matemaattisen kryptografian menetelmiä. Siksi on syytä lyhyesti selostaa ao. tutkimusalan kysymyksenasetteluja ja tuloksia.

3. Laskutehtävien vaikeudesta

Salakirjoitus on kiinnostanut ihmistä luultavasti lähes yhtä kauan kuin kirjoittaminenkin (ks. Kahn 1967). Pyrkimyksenä salakirjoitusjärjestelmissä on taata se, että vain legitiimit viestien vastaanottajat voivat lukea viestejä, kun taas salakuuntelijoille viestien sisältö jää käsittämättömäksi. Kryptosysteemien hyvyys määräytyy olennaisesti sen mukaan, miten vaikeaa järjestelmän murtaminen on ulkopuoliselle, jolla on käytettävissään vain salakirjoitettua tekstiä: mitä vaikeampaa murtaminen on, sitä parempi kryptosysteemi.

Tietokoneiden myötä moni aiemmin lähes mahdoton tehtävä on muuttunut mahdolliseksi. Niin myös kryptosysteemien kohdalla. Kaikki kryptosysteemit ovat murrettavissa, jos vain on riittävästi hyvää onnea ja lisäksi hyvät laskulaitteet. Kryptosysteemin laatijan onkin nykyään varauduttava siihen, että mahdollisella salakuuntelijalla on paljon laskennallista kapasiteettia käytössään. Tavoite murtamattomissa olevan systeemin laatimisesta on saanut väistyä realistisemman päämäärän tieltä: hyvän kryptosysteemin tulee asettaa parhain mahdollisin laskulaittein varustetut salakuuntelijatkin laskennallisesti kohtuuttoman työläiden tehtävien eteen. Sitävastoin kryptosysteemin legitiimien käyttäjien — siis sanomien lähettäjien ja vastaanottajien — ratkaistaviksi tulevien tehtävien tulee olla helppoja.

Laskutehtävien helppous ja vaikeus ovat algoritmien kompleksisuusteorian käsitteitä. Tarkastellaan algoritmia funktion f(n) arvojen laskemiseksi. Tietyllä argumentin n arvolla algoritmi käyttää tietyn määrän laskentaresursseja, esim. aikaa tai muistitilaa. Argumenttia voidaan tällöin tarkastella algoritmin syötteenä ja funktion arvoa sen tulostuksena. Tietyn funktion arvojen laskemiseksi voi tietysti olla useampia kuin yksi algoritmi. Laskutehtävän, määrää f(n), laskennollisen vaikeuden selvittämiseksi on olennaista tietää, miten tehokkaimman tunnetun algoritmin käyttämien laskentaresurssien määrä vaihtelee syötteen n koon, k(n), kasvaessa. Syötteen n koko, k(n), on yksikertaisesti luvun n numeroiden lukumäärä (esim. binaari- tai desimaaliesityksenä). Esim. jos n = 1991, niin k(n) = 4.

Laskutehtävän f(n) sanotaan kuuluvan luokkaan *P*, jos ja vain jos f(n):n laskemiseksi on olemassa sellainen algoritmi, joka selviytyy tehtävästä siten, että sen käyttämien laskennallisten resurssien määrä on polynomifunktio k(n):stä. Jos esim. algoritmi A tarvitsee f(n) laskemiseksi $k(n)^{100} + k(n)^{99} + 90892$ resurssiyksikköä, niin f(n) on *P*:ssä. Jos taas kaikki tunnetut f(n):n laskemisalgoritmit tarvitsevat sellaisen määrän resursseja, jota voidaan kuvata lausekkeella, jossa k(n) esiintyy eksponenttina (esim. $2^{k(n)} + 1$), niin f(n) ei todennäköisesti ole *P*:ssä.

P:hen kuuluvia tehtäviä — siis sellaisia joiden laskemiseksi on olemassa polynomiajassa toimiva algoritmi — on tapana kutsua laskennallisesti helpoiksi (computationally tractable). Näiden ohella on sellaisia tehtäviä, joiden laskemiseksi ei tunneta polynomiaikaista algoritmia, mutta jotka ovat helppoja hyvälle arvaajalle. Tällä tarkoitetaan sitä, että f(n):n arvot voidaan arvata ja arvausten oikeellisuus tarkistaa polynomiajassa. Tällaiset tehtävät ovat *NP*:ssä. Esimerkiksi käy

lukujen tekijöihinjako. Tämän tehtävän suorittamiseksi ei tunneta polynomiaikaista algoritmia, mutta jos onnistuu arvaamaan tietyn luvun tekijät, niin ratkaisun oikeellisuuden tarkistaminen kyllä sujuu helposti (siis polynomiajassa) kertomalla arvatut tekijät keskenään ja katsomalla, onko tulona alkuperäinen luku.

Yksi tärkeimmistä matematiikan ongelmista on, onko $P = NP$. Tätä ei tällä hetkellä tiedetä, mutta yleisesti lähdetään siitä, että näin ei ole asianlaita, ts. uskotaan, että on olemassa peruuttamattomasti *NP*:ssä olevia ongelmia. Hyvin mielenkiintoinen osajoukko *NP*:ssa muodostuu sellaisista tehtävistä, joilla on se ominaisuus, että jos jokin niistä osoittautuukin olevan *P*:ssä (ts. ao. funktion laskemiseksi löydetään polynomiajassa toimiva algoritmi), niin $P = NP$ vastoin yleistä otaksumaa. Tätä osajoukkoa kutsutaan *NP*-täydellisten ongelmien joukoksi. Mm. ns. kauppamatkustajaongelma on *NP*-täydellinen ongelma (ks. Garey & Johnson 1979).

Modernien kryptosysteemien perusidea on se, että viestien lähettäjien ja legitiimien vastaanottajien ongelmat — ts. viestien saattaminen salakirjoitusasuun eli kryptaus ja salakirjoitetun viestin selväkieliseksi kääntämien eli dekryptaus — ovat *P*:ssä, siis laskennallisesti helppoja. Sitävastoin salakuuntelijoiden on ratkaistava vaikeita ongelmia saadakseen selville viestien sisällön. Varmimmaksi vakuudeksi kryptosysteemit suunnitellaan yleensä siten, että salakuuntelijan ratkaistavaksi tuleva ongelma on *NP*-täydellinen. Tätä ratkaisua voidaan perustella sillä, että *NP*-täydelliset ongelmat ovat viime vuosina olleet erittäin intensiivisen tutkimuksen kohteena, ja siitä huolimatta polynomiajassa toimivia algoritmeja niiden ratkaisemiksi ei ole löydetty. Toisaalta ratkaisun varjopuolena on se, että minkä hyvänsä *NP*-täydellisen ongelman osoittautuminen ongelmaksi *P*:ssä vie pohjan pois myös ao. kryptosysteemiltä.

4. Julkisen avaimen kryptosysteemit

Ennen Diffien ja Hellmanin (1976) merkittävää tutkimusta kaikki kryptosysteemit perustuivat siihen, että jokainen, joka tietää, miten viestit saatetaan salakirjoitusasuun, tietää myös, miten ne »avataan». Näitä systeemejä kutsutaan nykyään klassisiksi. Diffie ja Hellman totesivat, että on mahdollista erottaa toisistaan yhtäältä kryptausfunktio ja dekryptausfunktio, ts. sääntö, jota noudattaen selvätekstit saaatetaan salakirjoitusasuun, ja sääntö, jolla salakirjoitusta käännetään selväkieliseksi. Heidän varsinainen oivalluksensa oli siinä, että nuo kaksi funktiota voidaan määritellä siten, että kryptausfunktion tietäminen ei lainkaan helpota dekryptausfunktion etsimistä. Näin ollen kryptausfunktiot voidaan julkistaa. Tästä johtuen onkin tapana puhua julkisen avaimen kryptosysteemeistä (ks. Salomaa 1990).

Julkisen avaimen kryptosysteemeissä kukin käyttäjä julkistaa oman kryptausavaimensa, mutta pitää omana tietonaan dekryptausavaimensa. Olettakaamme, että henkilö A lähettää salaisen viestin henkilölle B. Hän käyttää silloin B:n julkaisemaa kryptausavainta lähettämänsä tekstin kääntämiseksi salakirjoitusasuun. Viestien kryptaamisessa lähtökohtana on niiden numeerisesti koodattu versio, joka saadaan, kun kirjaimet esitetään kokonaislukuina 0—27. Julkinen kryptausavain ilmoittaa, miten näitä lukujonoja pitää muuntaa tekstin saattamisessa salakirjoitettuun asuun. Luonnollisesti kryptaustehtävän tulee olla helppo eli *P*:ssä. Sitävastoin dekryptauksen tulee olla vaikeaa eli palautua ongelmaksi *NP*:ssä muille kuin ao. viestin vastaanottajalle.

Ehkä tunnetuin julkisen avaimen kryptosysteemi, RSA, perustuu siihen, että lukujen tekijöihinjako on laskennallisesti vaikea ongelma (Rivest, Shamir & Adleman 1978). Viestien numeerisesti koodattu versio w kryptataan tässä systeemissä seuraavasti.

Kryptaaja A muodostaa ensinnä kahden suuren alkuluvun tulon $n = pq$. Luvut p ja q hän pitää omana tietonaan, kun sitävastoin n julkistetaan. Kun A tietää p:n ja q:n voi hän myös muodostaa tulon $(p—1)(q—1)$. Merkitsemme tätä tuloa E(n):llä. A etsii nyt sellaisen luvun m, että sillä ja E(n):llä ei ole yhteisiä tekijöitä. Lukuteoriasta tiedetään, että tällöin on olemassa yksikäsitteinen luku t siten, että $mt = 1 \pmod{E(n)}$. Tämä tarkoittaa, että kun mt jaetaan E(n):llä, niin jakojäännös on 1. A julkistaa myös t:n. Viestin numeerisesti koodattu versio w kryptataan siten, että w jaetaan ensin sopivan pituisiin lohkoihin ja jokaisen lohkon L kohdalla suoritetaan muunnos: $L \dashrightarrow L^t \pmod{n}$. Tulos on L:n kryptattu versio.

L^t:n dekryptaus on vaikeaa sellaiselle, joka ei tunne n:n tekijöihin jakoa. Sen sijaan A:lle tehtävä on helppo. Hän yksinkertaisesti suorittaa muunnoksen: $L^t \pmod{n} \dashrightarrow (L^t)^m \pmod{n}$. Tulos on L eli alkuperäisen viestin lohko. Se, että tulos todella on L osoitetaan tämän artikkelin liitteessä 1.

RSA:ssa siis salakielisen viestin kääntämien selväkieliseksi tekstiksi on salakuuntelijalle *NP*-täydellinen laskutehtävä. Sen sijaan sellaiselle henkilölle, joka tuntee n:n tekijöihin jaon, tehtävä on helppo. Tekijöihin jaon vaikeuteen perustuvien kryptosysteemien ohella on kehitelty varsin monia muunlaisia julkisen avaimen systeemejä (ks. Salomaa 1990). RSA lienee kuitenkin laajimmin käytetty järjestelmä.

On syytä mainita, että RSA:ssa esiintyvien lukujen p ja q edellytetään olevan varsin suuria; nykyään pidetään sata-numeroisia lukuja riittävän turvallisina. Modulaaristen kertolaskujen suorittaminen näin suurilla luvuilla ei tietenkään ole mahdollista muuten kuin tietokoneella. RSA:n kaupallinen menestys perustuukin juuri siihen, että tarpeellisia RSA-siruja on saatavana.

5. Kryptografiset protokollat

Kryptografiset protokollat ovat kryptosysteemien hyväksikäyttöön perustuvia vuorovaikutusmenettelyjä, joilla viestejä lähettävät ja vastaanottavat osapuolet pyrkivät antamaan itsestään, intresseistään, resursseistaan tms. riittävästi informaatiota joidenkin yhteisesti hyväksyttyjen päämäärien saavuttamiseksi pitäen kuitenkin olennaisen tärkeinä pitämänsä salaisuudet omana tietonaan. Kryptografiset protokollat perustuvat siis kryptosysteemeihin. Näin ollen niiden luotettavuus on ratkaisevasti riippuvainen käytetyn kryptosysteemin luotettavuudesta.

Esimerkkinä kryptografisesta protokollasta tarkasteltakoon menettelyä, jossa halutaan lähettää viesti A:lta B:lle siten, että seuraavat ehdot ovat voimassa:

1. A tietää, että vain B kykenee selvittämään sanoman sisällön.
2. B tietää, että viesti tulee A:lta.
3. B tietää, ettei A voi myöhemmin kieltää lähettäneensä viestiä (Salomaa 1985).

Oletamme, että käytetty kryptosysteemi on luotettava (esim. RSA). Merkitsemme A:n ja B:n julkisia kryptausfunktioita e_A:lla ja e_B:llä. Vastaavasti A:n ja B:n salaisia dekryptausfunktioita merkitään d_A:lla ja d_B:llä. Merkitsemme lähetettävän viestin numeerisesti koodattua versiota w:llä ja tarkastelemme sen yhtä lohkoa L. Kun lohkon pituus on ennalta sovittu, voidaan menettely ilmeisellä tavalla ulottaa koko viestiin w. Oletamme, että mistä hyvänsä lohkosta L pätee se, että kun L ensin kryptataan ja tulos jälleen dekryptataan vastaavalla funktiolla, niin saadaan alkuperäinen lohko. Oletamme samoin käyvän, kun ensin suoritetaan dekryptaus ja sitten kryptaus. Vähän formaalisemmin oletamme siis, että jokaiselle lohkolle L ja jokaiselle henkilölle H:

$$d_H(e_H(L)) = e_H(d_H(L)) = L.$$

Protokolla, joka täyttää äsken mainitut ehdot 1.—3. on seuraava:

Vaihe 1. A dekryptaa L:n omalla salaisella avaimellaan eli suorittaa käännöksen L ----> $d_A(L)$.

Vaihe 2. A kryptaa vaiheen 1 tuloksen B:n julkisella avaimella: $d_A(L)$ ----> $e_B(d_A(L))$. A lähettää tuloksen B:lle.

Vaihe 3. B dekryptaa saamansa viestin salaisella dekryptausavaimellaan: $e_B(d_A(L))$ ----> $d_B(e_B(d_A(L))) = d_A(L)$.

Vaihe 4. B kryptaa vaiheen 3 tuloksen A:n julkisella kryptausavaimella: $d_A(L)$ ----> $e_A(d_A(L)) = L$.

Ehto 1 on täytetty sen vuoksi, että vain B kykenee kääntämään selväkielelle viestit, jotka on kryptattu e_B:llä. Ehto 2 taas täyttyy sen kautta, että vain A:n julkinen kryptausfunktio tuottaa B:lle mielekkään viestin vaiheessa 4. Toisaalta vain A pystyy laatimaan viestin, joka e_A:lla kryptattuna tuottaa mielekkään viestin. Tämän A tekee vaiheessa 1. Ehdon 3 voimassaolo seuraa myös vaiheesta 4. Tässä vaiheessahan B:llä on kaksi versiota L:stä: L ja $d_A(L)$. Se, että edellinen saadaan jälkimmäisestä A:n julkisella avaimella, takaa, että vain A on voinut sen lähettää. Seuraavassa tarkasteltava salaisten vaalien protokolla perustuu yllä esitetyn viestintäprotokollan systemaattiseen käyttöön.

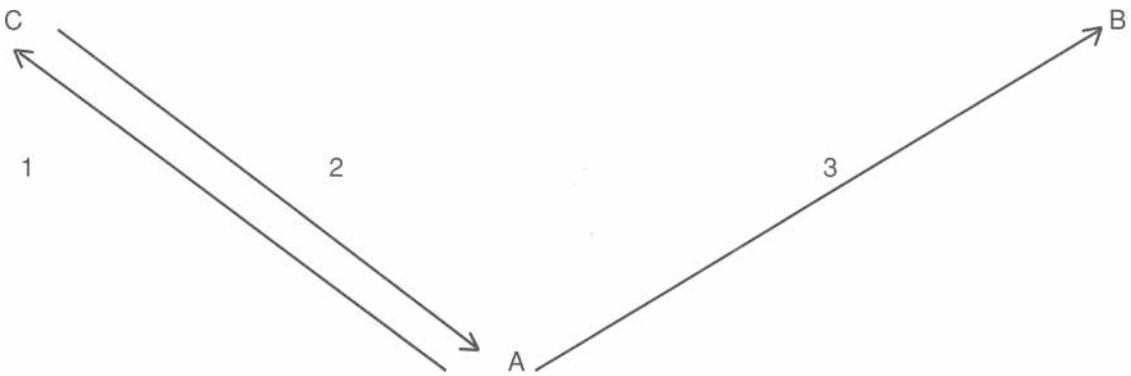
6. Salaisten vaalien protokolla

Seuraavassa esiteltävä protokolla perustuu muissa yhteyksissä esitettyihin töihin (Nurmi & Salomaa 1991; Nurmi & Salomaa 1990; Nurmi, Salomaa & Santean 1990). Ensi näkemältä turvallinen protokolla salaisten äänestysten toimeenpanemiseksi olisi yksinkertaisesti yllä esitelty siten tulkittuna, että A on äänestäjä ja B ääntenlaskentasysteemi. Tarkemmin katsottuna menettely ei kuitenkaan ole tyydyttävä; B:hän pystyy suoraan näkemään, miten A äänesti. Toisaalta menettely kyllä takaa sen, ettei kukaan ulkopuolinen pysty

tätä selvittämään. Edelleen A:n äänioikeus kyetään tarkistamaan. Haluamme kuitenkin systeemin, jossa yksittäisten äänestäjien äänet pysyvät salassa myös laskentajärjestelmältä; tämänhän jo käytössä olevat järjestelmätkin pyrkivät takaamaan.

Seuraava kuvio havainnollistaa äänestysprotokollan toimijain suhteita.

Kuvio



A:lla merkitsemme äänestäjää, B:llä ääntenlaskentajärjestelmää ja C:llä äänioikeuden tarkistavaa järjestelmää. Käymme seuraavassa läpi protokollan vaiheittain (numerot viittaavat kuvion numeroihin).

Vaihe 1. Äänestäjä lähettää C:lle viestin v, esim. »Päivää, A tässä haluaisi äänestää». Viesti v esitetään muodossa $e_C(d_A(v))$.

Vaihe 2. C toteaa A:n äänioikeuden. Jos A:lla ei ole äänioikeutta, C lähettää tätä koskevan sanoman A:lle. Muussa tapauksessa A poistetaan äänioikeutettujen luettelosta ja A:lle lähetetään viesti u, joka sisältää tietyn informaation. Se lähetään muodossa $e_A(d_C(u))$. Viesti on olennaiselta sisällöltään sama kaikille äänioikeutetuille äänestäjille.

Vaihe 3. Siinä tapauksessa, että A on todettu äänioikeutetuksi ja on saanut siis viestin u vaiheessa 2, A voi lähettää varsinaisen äänestysviestin B:lle. Se koostuu kolmesta komponentista x, y ja z. Ensimmäinen komponentti sisältää tietyllä tavalla kryptattuna u:n ja toimii siten A:n todisteena B:lle siitä, että C on todennut A:n äänioikeutetuksi. Toinen komponentti y taas sisältää viestin, jonka perusteella A voi tunnistaa oman äänensä muiden joukosta. Kolmas komponentti vihdoin on A:n äänestysstrategia. Tämänkin vaiheen komponentit lähetetään kryptattuina, mutta ainoastaan B:n julkisella kryptausfunktiolla. A:n dekryptausfunktiota ei voida käyttää, koska se paljastaisi B:lle A:n henkilöllisyyden.

Vaihe 4. B julkistaa vaalin tuloksen, kaikki äänestysstrategiat sekä kunkin äänestysstrategian kohdalla niiden henkilöiden y-komponentit, jotka ovat ao. strategian valinneet. B julkistaa myös u:n, jotta äänestäjät voisivat todeta, ettei C ole liittänyt heidän saamaansa informatioon liitettä, joka mahdollistaisi A:n tunnistamisen B:n taholta.

Vaihe 1 toimii kuten jakson 5 lopussa esitetty protokolla, joten siihen emme enää puutu. Vaiheen 2 turvallisuutta voidaan olennaisesti parantaa menettelyllä, jossa A lähettää yllä mainitun viestinsä ohessa C:lle valitsemansa suuren luvun kryptattuna C:n julkisella avaimella. Vaiheessa 2 C lähettääkin A:lle edellä mainitun informaation numeerisen version kerrottuna A:n valitsemalla luvulla, joten mahdollinen salakuuntelija ei A:n ja C:n välistä viestintää analysoimalla saa selville informaatiota, joka mahdollistaisi äänestämisen.

Vaiheessa 3 voi aktiivinen salakuuntelija yrittää seuraavanlaista menettelyä: hän sieppaa A:n B:lle lähettämän viestin, vaihtaa oman äänestysstrategiansa A:n äänestysstrategian paikalle, jättää kaiken muun viestissä ennalleen ja lähettää

näin muokatun viestin B:lle. Onko tälle ilmeisen epätoivottavalle menettelylle olemassa ehkäisykeinoa? Mielestämme on. On syytä vaatia, että A lähettää äänestysstrategiansa B:lle »sotketussa» muodossa siten, että esim. äänestysstrategian numeerinen versio kerrotaan jollakin A:n valitsemalla luvulla, jonka A ilmoittaa kryptatussa muodossa B:lle. Tämän luvun ei pidä olla sama kuin A:n vaiheessa 1 valitsema luku, jottei B:llä ja C:llä yhteistoiminnassa ole mahdollisuuksia A:n henkilöllisyyden selvittämiseen. Joka tapauksessa tällä lisävarokkeella ulkopuolisen on mahdoton esiintyä A:na ilman tietoa A:n dekryptausavaimesta, jonka oletamme pysyvän salaisuutena.

Yllä esitelty protokolla ei vielä takaa mahdollisuuksia virheiden korjauksiin. Niinpä meidän on hieman monimutkaistettava menettelyä. Sen sijaan, että B suoraan tulostaisi kaikki saamansa äänet yhtä aikaa, se antaa kullekin äänestäjälle »kuitin» siitä, miten tämän ääni allokoidaan: $p_A v(j)$. Tässä v(j) on A:n antama ääni ja p_A hänen valitsemansa suuri alkuluku. Tällä luvulla ja y-komponentilla »sekoitettuna» (esim. kerrottuna ja redusoituna ao. modulilla) A lähettää äänensäkin vaiheessa 3. Kuitti on julkinen — so. tietokoneverkossa julkistettava — viesti. Siitä käy ilmi, että B osaa dekomponoida A:lta tulleen viestin. Esim. B osaa »jakaa» viestin A:n identifiointiin käyttämällä viestillä. Kuitti voidaan antaa vaikkapa välittömästi kunkin äänen tultua B:lle. Siitä ulkopuoliset, B mukaanlukien, eivät pysty identifioimaan A:ta, mutta A toki pystyy tunnistamaan oman äänensä. Viestin tunnistettuaan A voi todeta B:n osanneen tulkita hänen äänensä oikein. Kuittausviestejä voidaan lähettää välittömästi B:n saatua äänestysviestin, sillä ne eivät paljasta, kenen ehdokkaan hyväksi ääni on annettu. Näin ollen äänten väliaikaistulostus ei anna mahdollisuuksia strategiseen äänestämiseen. Itse asiassa kaikki äänet voidaan tulostaa ensin identifiointiviesteillä sekoitetussa muodossa, jolloin kaikilla äänestäjillä on viimeinen mahdollisuus joko korjata väärin sijoitetut äänet tai muuttaa mielensä valintansa suhteen.

Entä jos A ei löydäkään B:n antamien kuittien joukosta omaa ääntään (sekoitetussa muodossa)? Silloin A voi lähettää B:lle viestin, joka ennalta sovitun järjestelmän mukaan on tunnistettavissa korjausviestiksi. Viestistä tulee käydä ilmi, että A tuntee p_A:n ja C:n kaikille äänioikeutetuille antaman viestin. Näiden perusteella B etsii aiemmin annetuista äänistä p_A:lla sekoitetun äänestysviestin, mitätöi sen ja revisioi A:n äänen korjausviestin ohjeiden mukaisesti tulostaen lopulta myös kuittauksen. Menettely toistetaan, kunnes A:n ääni tulee oikein kuitatuksi.

Samaa menettelyä voidaan käyttää myös tietyn ajan ajanjakson kuluessa annettujen äänten revisiointiin. Silloinkin äänestäjä esittelee itsensä B:lle ilmaisemalla tuntevansa aiemmin itse valitsemansa suuren luvun ja C:n kaikille äänioikeutetuille antaman viestin.

7. Komplisoidumpi protokolla

Edellä esitellyn protokollan kannalta on tärkeää varmistaa, että C:n antamien äänestyslupien määrä ja annettujen äänien määrä on sama. Muutenhan systeemi B voi lisätä mielivaltaisen määrän »kuolleita sieluja» äänioikeutettujen joukkoon. Kirjallisuudessa on esitelty protokollia, joissa salaiset vaalit voidaan toimeenpanna tietokoneverkoissa siten, että protokollan tultua läpikäydyksi osapuolet tietävät, että kaikki äänet tulevat lasketuiksi (ks. esim. Benaloh 1987). Systeemi ei kuitenkaan mahdollista sitä, että yksittäiset äänestäjät voisivat tarkistaa oman äänensä sijoittumisen tarkoitetun ehdokkaan hyväksi.

Jaksossa 6 esitellyn systeemin heikkoutena on kuitenkin se, että äänestäjä voi äänestää useammin kuin kerran saatuaan C:ltä äänestysluvan eli viestin, jonka äänioikeutetut saavat. Äsken mainittu äänten lukumäärän laskeminen toki takaa sen, että moninkertainen äänestäminen tulee huomatuksi, mutta moninkertaiseen äänestämiseen syyllistyneitä ei voida identifioida. Siksi hahmottelemme seuraavassa toiselta pohjalta lähtevää kryptografista protokollaa, joka perustuu ANDOS-protokollaan (all-or-nothing-disclosure-of-secrets). ANDOS-ideaa ovat hahmotelleet Brassard, Crepeau ja Robert (1987). Seuraavassa esiteltävä versio perustuu Nurmen, Salomaan ja Santeanin (1990) esitykseen (ks. myös Salomaa ja Santean 1990). Protokollassa on vain yksi vaaliviranomainen, jota merkitsemme B:llä.

Nimensä mukaisesti ANDOS-protokollat perustuvat siihen, että tietyn informaation selville saamiseksi on selvitettävä kaikki sen osat. Yhdenkin osan selvittämättä jääminen on samaa kuin ei tietäisi ao. informaatiosta mitään. Esimerkiksi tietty salaisuus voidaan jakaa n:n henkilön hallussaan pitämiin osiin siten, että mikään pienempi kuin s:n henkilön koalitio ei tietonsa yhdistämällä saa selville salaisuutta, kun sitävastoin mikä hyvänsä vähintään s:n jäsenen koalitio saa

selville salaisuuden. Tällaista systeemiä sanotaan kynnysjärjestelmäksi (threshold scheme, ks. Shamir 1979). Seuraavassa esiteltävässä protokollassa hyödynnetään kynnysjärjestelmävarianttia, joka mahdollistaa salaisuuksien myynnin siten, että myyjä ei tiedä, minkä hallussaan olevista salaisuuksista on tullut myyneeksi ja mille erityiselle ostajalle. Ostaja sitävastoin voi ostaa nimenomaan häntä kiinnostavalta alueelta salaisuuden (ks. tarkemmin Salomaa ja Santean 1990). Tätä varianttia on selostettu hieman yksityiskohtaisemmin liitteessä 2.

ANDOS-äänestysprotokolla on seuraava.

Vaihe 1. Äänten laskusysteemi B julkistaa äänioikeutettujen luettelon.

Vaihe 2. Äänestäjät rekisteröityvät, ts. ilmaisevat tietyn ajan kuluessa halunsa äänestää.

Vaihe 3. B julkistaa rekisteröityneiden luettelon.

Vaihe 4. B valitsee m suurta alkulukua, missä m on huomattavasti suurempi luku kuin rekisteröityneiden lukumäärä. Lukuja merkitään 1, ..., m.

Vaihe 5. Äänestäjä A valitsee tietyn kokonaisluvun k_A väliltä [1,m]. Vaiheesta 4 johtuen voimme olettaa, että jokainen äänestäjä valitsee eri luvun. Salaisuuksien myyntiin laadittu ANDOS-protokolla pannaan toimeen A:n ja B:n välillä. Sen seurauksena A tietää k_A:n ja B tietää A:n valinneen jonkin vaiheessa 4 määritellyn luvun.

Vaihe 6. A sekoittaa kryptografisesti äänensä k_A:ta hyväksi käyttäen ja lähettää sekoitetun äänensä k_A:n kera B:lle.

Vaihe 7. B kuittaa äänen saaduksi julkaisemalla A:n sekoitetun äänen.

Vaihe 8. A lähettää B:lle sekoitusavaimen ja luvun k_A. Siten B tietää nyt, miten k_A:n valinnut äänestäjä äänestää (vaiheessa 7 hän ei vielä sitä tiedä). B ei kuitenkaan tiedä, kuka A on.

Vaihe 9. Mikäli A vaiheessa 7 huomaa virheen omalla kohdallaan, hän korjaa sen lähettämällä B:lle viestin, josta käy ilmi, että A tuntee k_A:n ja äänen sekoittamismenettelyn. Samaa menettelyä noudatetaan, mikäli A haluaa vaihtaa äänestysstrategiaansa.

Vaihe 10. Kun äänestys-, mielenmuutos- ja virheidenkorjausaika on kulunut umpeen B julkistaa äänestystuloksen ilmoittamalla kunkin äänestysstrategian (tai ehdokkaan) kohdalla ao. strategian valinneiden äänestäjien k_A:t sekä ao. äänet sekoitetussa muodossa. Strategisen äänestämisen vaikeuttamiseksi on syytä ilmoittaa tulokset vasta viimeiseksi.

8. Etuja ja haittoja

Vaikka RSA-kryptosysteemi onkin jo saavuttanut kaupallisen levityksen asteen, ei yllä esiteltyjä menettelyjä ole käytännössä kokeiltu. Eräät varjopuolet jaksojen 6 ja 7 protokollissa haittaavat niiden käytöönottoa yleisissä poliittisissa vaaleissa. Moninkertaisen äänestämisen mahdollisuus jakson 6 protokollassa tosin voidaan välttää jakson 7 monimutkaisella ANDOS-menettelyllä. Mutta äänten myyntimahdollisuuksien lisääntymistä voidaan pitää rajoituksena molempien lukujen protokollien käytölle yleisissä vaaleissa, joissa äänestäjien kiinnostus ratkaistaviin asioihin huomattavasti vaihtelee. On tosin muistettava, ettei nykyisissäkään vaalimenettelyissä ole mitään sellaista, joka tekisi äänten oston ja myynnin mahdottomaksi. Lisäksi voidaan väittää, että kaikissa sellaisissa järjestelyissä, joissa äänestäjien on mahdollista todeta äänensä tulleen oikein lasketuksi, on myös mahdollisuus äänten myyntiin.

Keskeiset edut yllä kuvatuissa järjestelyissä käyvät luonnollisesti ilmi jaksossa 2 esitetyistä ehdoista 1.—7. Oman äänen tuleminen lasketuksi nimenomaan tarkoitetun ehdokkaan hyväksi on ominaisuus, jota käytössä olevilla järjestelmillä ei ole. Samoin ajatus vaalien toimeenpanosta tietokoneverkoissa näyttää edellä sanotun perusteella mahdolliselta. Näin ollen ajatus jatkuvista vaaleista näyttäisi teknisesti mahdolliselta. Eri asia kokonaan on, voidaanko jatkuvia vaaleja muista syistä pitää perusteltuna.

Liite 1. Miksi RSA toimii?

On osoitettava, että $(L^t)^m = L \pmod n$. Erotamme kolme tapausta, joissa jokaisessa päädymme tähän tulokseen.

Tapaus 1. L ei ole jaollinen p:llä eikä q:lla. Tällöin Eulerin lauseesta seuraa, että $L^{E(n)} = 1 \pmod n$. Edelleen $mt = HE(n) + 1$, missä H on kokonaisluku, sillä $mt = 1 \pmod{E(n)}$. Näin ollen $(L^t)^m = L^{HE(n)+1} = L \pmod n$, joten saimme halutun tuloksen.

Tapaus 2. L on jaollinen sekä p:llä että q:lla. Silloin $(L^t)^m = L \pmod{pq} = L \pmod n$.

Tapaus 3. L on jaollinen yhdellä ja vain yhdellä luvuista p ja q. Rajoituksetta oletamme, että L on jaollinen p:llä. Silloin $(L^t)^m = L \pmod p$. Merkitään tätä (i):llä. Euler osoitti, että $L^{qn-1} = 1 \pmod q$. Edelleen $L^{(p-1)(q-1)} = 1 \pmod q$ ja $L^{HE(n)} = 1 \pmod q$.

Koska mt = HE(n) + 1 saamme tuloksen L^{mt}= L (mod q), kun asetamme vastaavat L:n potenssit yhtä suuriksi. Viimeksi mainitusta tuloksesta ja (i):stä seuraa, että L^{mt}= L (mod pq) = L (mod n).

Siten kaikissa tapauksissa olemme saaneet halutun johtopäätöksen, ts. dekryptaus potenssiin m korottamalla ja redusoimalla modulo n tuottaa alkuperäisen viestin lohkon L.

Liite 2. Esimerkki ANDOS-protokollasta

Myyjällä A on hallussaan salaisuudet s_1, s_2, ... , s_k, joiden yleisen luonteen hän on julkistanut kunkin s_i:n kohdalla. Siten s_1 voisi tarkoittaa esim. tietyn maan puolustusvoimien hallussa olevien ohjusten torjuntaohjusten määrää, s_2 tietyn maan presidentin päiväohjelmia tiettynä aikana tms. informaatiota, mistä jotkut tahot ovat valmiit maksamaan jotakin haluamatta paljastaa, mikä erityinen salaisuus heitä kiinnostaa.

Olettakaamme yksikertaisuuden vuoksi, että ostajia on kaksi, B ja C, ja että edellinen on kiinnostunut salaisuudesta s_j ja jälkimmäinen salaisuudesta s_m. Jokaisessa salaisuudessa on n bittiä eli salaisuudet ovat ilmaistavissa n:n pituisina binaarilukuina.Protokolla on seuraava.

Vaihe 1. Myyjä A ilmoittaa B:lle funktion f ja C:lle funktion g, mutta pitää käänteisfunktiot f^{-1} ja g^{-1} omana tietonaan.

Vaihe 2. B ilmoittaa C:lle k kappaletta satunnaisia n:n bitin lukuja x_1, ... , x_k. Samoin C ilmoittaa B:lle saman määrän satunnaisesti valittuja n:n bitin lukuja x'_1, ... , x'_k. Määrittelemme nyt *kiinteän bitti-indeksin* (KBI). Olkoon x jokin n-bittinen luku, i jokin kokonaisluku välillä [1,n] ja f injektiivinen funktio, joka kuvaa n-bittisiä lukuja toisilleen. Indeksi i on KBI x:n ja f:n suhteen, jos i. bitti x:ssä on sama kuin i. bitti f(x):ssä.

Vaihe 3. B kertoo C:lle joukon KBI_B, nimittäin kiinteät bitti-indeksit x'_j:n ja f:n suhteen. Vastaavasti C kertoo B:lle joukon KBI_C, so. kiinteät bitti-indeksit x_m:n ja g:n suhteen. KBI_B määräytyy siis C:n B:lle antaman lukujoukon j:nnen luvun ja A:n B:lle antaman funktion f kautta. Siinä on lueteltuna ne j:nnen luvun bitit, jotka kuvauksessa f säilyttävät arvonsa muuttumattomina. Samoin KBI_C määräytyy B:n C:lle antaman lukujoukon m:nnen luvun ja A:n B:lle antaman funktion g avulla.

Vaihe 4. B kertoo A:lle luvut y_1, ... , y_k, missä jokainen y_i saadaan vastaavasta x_i:stä vaihtamalla komplementikseen — so. 0 vaihdetaan 1:ksi ja päinvastoin — jokainen sellainen bitti, joka ei kuulu KBI_B:hen. Samoin C kertoo A:lle luvut y'_1, ... , y'_k, missä jokainen y'_i saadaan vastaavasta x'_i:stä vaihtamalla komplementikseen jokainen sellainen bitti, joka ei kuulu KBI_C:hen.

Vaihe 5. A ilmoittaa B:lle luvut s_i o $f^{-1}(y'_i)$, missä i = 1, ..., k. Samoin A ilmoittaa C:lle luvut s_i o $g^{-1}(y^i)$ (i=1, ... , k). Symboli o tarkoittaa suoraa summaa, ts. symbolin molemmilla puolilla olevien lukujen komponenteittaista (binaarista) summaa (tällöin 1 o 1 = 0).

Vaihe 6. Nyt B pystyy määräämään s_j:n ja C s_m:n, sillä B tietää, että x'_j= $f^{-1}(y'_j)$, ja C tietää, että x'_m= $g^{-1}(y_m)$. Siten B:lle s_j:n ja C:lle s_m:n selvittäminen käy »vähennyslaskulla». Protokollan läpikäynnin jälkeen B ja C tietävät haluamansa salaisuudet. A ei tiedä, minkä salaisuuden kukin on hankkinut.

LÄHTEET

Benaloh, J. D. C.(1987): *Verifiable Secret Ballot Elections.* Yale University, Computer Science Department, Technical Report 561.

Brassard, G., C. Crepeau & J.-M. Robert (1987): All-Or-Nothing Disclosure of Secrets, *Springer Lecture Notes in Computer Science* 263, 1987, ss. 234—238.

Diffie, W. & M. Hellman (1976): New Directions in Cryptography, *IEEE Transactions on Information Theory IT-22*, 1976, ss.644—654.

Garey, M. & D. Johnson (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: W.H.Freeman.

Kahn, D. (1967): *The Codebreakers*, Reading: Macmillan.

Nurmi, H. & A. Salomaa (1991): A Cryptographic Approach to the Secret Ballot, *Behavioral Science* 36, 1991, ss. 34—40.

Nurmi, H. & A. Salomaa (1990): *Secret Ballot Elections and Public-Key Cryptosystems*, artikkelikäsikirjoitus.

Nurmi, H., A. Salomaa & L. Santean (1990): *Secret Ballot Elections in Computer Networks*, artikkelikäsikirjoitus.

Rivest, R., A. Shamir & L.Adleman (1978): A Method of Obtaining Digital Signatures and Public-Key Cryptosystems, *ACM Communications* 21, 1978, ss. 120—126.

Salomaa, A. (1985): *Computation and Automata*, Cambridge: Cambridge University Press.

Salomaa, A. (1990): *Public-Key Cryptography*, Berlin—Heidelberg—New York: Springer-Verlag.

Salomaa, A. & L. Santean (1990): Secret Selling of Secrets with Many Buyers, *EATCS Bulletin* 42, 1990, ss. 178—186.

Shamir, A. (1979): How to Share a Secret, *ACM Communications* 22, 1979, ss. 612—613.