hyväksymispäivä arvosana

arvostelija

Tunnistusprotokollat

Sini Ruohomaa

Helsinki 11.12.2003

LuK-tutkielma

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty
Matemaattis-luonnontieteellinen

Laitos — Institution — Department
Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Sini Ruohomaa

Työn nimi — Arbetets titel — Title
Tunnistusprotokollat

Oppiaine — Läroämne — Subject

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
LuK-tutkielma	11.12.2003	22 sivua

Tiivistelmä — Referat — Abstract

Tunnistaminen on nyky-yhteiskunnassa keskeinen ongelma, sillä eri henkilöillä on erilaiset oikeudet tehdä tiettyjä asioita, kuten esimerkiksi siirtää rahaa tietyltä tililtä. Täyttä varmuutta henkilöllisyydestä saavutetaan hyvin harvoin. Samassa huoneessa olevan henkilön tunnistamiseksi on tarjolla varsin monipuolinen valikoima erilaisia välineitä, mutta kun henkilöllisyydestä pitäisi varmistua verkon yli, tilanne mutkistuu melkoisesti.

Tämä tutkielma käsittelee tunnistamista verkossa sekä tunnistamisjärjestelmissä huomioon otettavia uhkia ja ongelmia. Tutkielmassa käydään läpi yhteiseen salaisuuteen, julkisen avaimen kryptografiaan ja digitaalisiin allekirjoituksiin sekä nollatietotodistuksiin perustuvat järjestelmät. Esimerkkeinä kahdesta jälkimmäisestä järjestelmätyypistä käsitellään julkisen avaimen järjestelmät RSA ja ElGamal sekä henkilöllisyyden nollatietotodistuksen mahdollistavat Fiat-Shamir ja BDLP (Brandt et al).

ACM Computing Classification System (CCS):
C.2.0 [Computer-communication networks],
E.3 [Data encryption]

Avainsanat — Nyckelord — Keywords
tunnistusprotokollat, henkilöllisyys, varmenne, allekirjoitus, nollatietotodistus

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

Henkilöllisyys on meille tärkeä asia: sen avulla voidaan erotella erilaisiin toimiin oikeutetut henkilöt toisistaan. Kun ihmiset eivät tunne toisiaan nimeltä, henkilöllisyyden todistamiseen apuvälineitä, kuten kuvallisia muovikortteja, joiden avulla esimerkiksi pankkivirkailija voi vakuuttua asiakkaan henkilöllisyydestä tuntematta tätä. Nykyaikana joudumme ajoittain todistamaan henkilöllisyytemme myös puhelimitse tai verkon välityksellä, tai toisen ihmisen sijaan koneelle, kuten esimerkiksi pankkiautomaatille.

Henkilöllisyyden todistamista varten on olemassa joukko eri menetelmiä, jotka soveltuvat erilaisiin tilanteisiin. Voimme perustaa henkilöllisyytemme todistamisen kolmeen asiaan: a) siihen, mitä *olemme* (esim. silmän verkkokalvon kuvaus tai sormenjäljet), b) siihen, mitä *tiedämme* (esim. salasana tai tunnusluku) tai c) siihen, mitä *omistamme* (esim. sinettisormus tai avainkortti). Mikään näistä ei sovi kaikkiin tunnistamistarpeisiin, joten usein päädytään ainakin kahden edellämainitun periaatteen yhdistelmään. Pankkiautomaattikortissa salaisine tunnuslukuineen yhdistyvät b) ja c), kun taas kuvallisessa henkilökortissa yhdistyvät kasvokuva ja itse kortti, eli a) ja c). Tunnistaminen on vaikeaa ja epävarmaa, ja sen suhteen joudutaankin tekemään kompromisseja. Automaattikortin ja tunnusluvun käyttö ei suinkaan todista, että pankkiautomaattia käyttää Matti Meikäläinen, vaan se kertoo vain, että kyseisellä henkilöllä on hallussaan Matin pankkikortti ja tunnusluku—pankille kuitenkin riittää tieto, että muiden kuin Matin on vaikea saada käsiinsä näitä molempia.

Verkon yli viestittäessä on hyvä pitää mielessä, että mikään ei rajoita sitä, mitä vastapuoli voi tehdä laitteistollaan. Kortinlukijoista ja muista verkkoon kytkettävistä apulaitteista ei ole mitään apua, jos niiden lausunnon voi kuka tahansa väärentää. Esimerkiksi Matti ja Liisa voivat toistensa tunnistamisen varmistamiseksi kytkeä tietokoneisiinsa kalliit laitteet, jotka DNA-testin, avainkortin ja salasanan perusteella tunnistavat käyttäjänsä äärimmäisen luotettavasti. Jos tämä erinomainen tunnis-

tin kuitenkin vain lähettää Matin tunnistettuaan verkkoon viestin “tämä on selvästi Matti”, mikä estää naapurin Niiloa tekemästä pientä ohjelmaa, joka lähettää aina vastapuolelle viestin “tämä on selvästi Matti”? Siispä tunnistuslaitteiden pitää ensin todistaa, että ne ovat oikeasti tunnistuslaitteita, ja ettei niitä ole peukaloitu.

Tässä tutkielmassa jätetään huomiotta olemukseen ja omistamiseen perustuvat tunnistusmenetelmät ja käsitellään kolmea erilaista lähtökohtaa tietoperustaiselle tunnistamiselle: perinteistä salasanan ilmoittamista, julkisen avaimen kryptografian avulla toteutettavia digitaalisia allekirjoituksia ja henkilöllisyyden nollatietotodistuksia. Matti ei käsiteltävien protokollien kannalta eroa koneesta, jota hän käyttää henkilöllisyytensä todistamiseen. Vaikka tutkielmassa puhutaan “Matista” ja “Liisasta” tai “palvelimesta”, viestit kulkevat lähes aina jonkinlaisten ohjelmien ja laitteiden välillä. Ne ovat yleensä yhteydessä toisiinsa julkisen verkon välityksellä, joten viestintää voidaan myös salakuunnella.

Luvussa 2 käsitellään salasanoihin perustuvia järjestelmiä, joissa tunnistuksen perustana on kyky toistaa aiemmin sovittu salaisuus. Luvussa 3 käsitellään julkisen avaimen salaus- ja allekirjoitusjärjestelmiä, joiden avulla salakuuntelun uhka saadaan minimoitua. Luvussa esitellään myös esimerkkijärjestelmiä ja pohditaan menetelmiä, joilla avaimen oikeellisuudesta voidaan vakuuttua. Luvussa 4 käsitellään nollatietoprotokollia, joiden avulla salaisuuden tunteminen voidaan todistaa ilman, että kukaan kuunteleva taho saa todistuksesta uutta tietoa. Luvussa esitellään myös kaksi esimerkkijärjestelmää. Lopuksi luvussa 5 on yhteenveto ja pohdintaa siitä, mitä muita ongelmia viestinnässä verkon yli tulee vielä ratkaista.

2 Yhteinen salaisuus

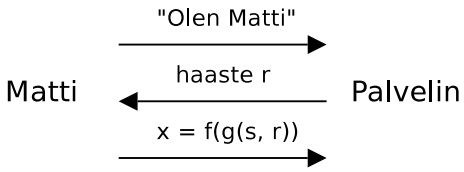
Verkossa tunnistamisen kenties suoraviivaisin keino perustuu yhteiseen salaisuuteen. Jos Matti haluaa lukea sähköpostinsa, palvelin kysyy todennäköisesti häneltä sala-

sanaa. Matti todistaa siis henkilöllisyytensä palvelimelle osoittamalla tietävänsä aiemmin sen kanssa sovitun yhteisen salaisuuden. Sama toistuu, kun Matti asioi verkkopankissa. Matin pitää kertoa oikea salasana annetusta luettelosta, jotta pankin palvelin vakuuttuisi hänen henkilöllisyydestään.

Salasanan tarkistaminen on varsin yksinkertainen tapahtuma molemmille osapuolille. Matin tarvitsee vain muistaa salasanansa ja lähettää se ennalta sovitulla hetkellä, esimerkiksi heti yhteyden solmimisen ja käyttäjätunnuksensa lähettämisen jälkeen tai pyydettäessä. Vastapuoli vertaa tätä omaan käsitykseensä siitä, mikä salasanan pitäisi olla. Toisaalta Matin pitää voida luottaa kysyjään ja verkkoon matkan varrella, sillä hyökkääjä voi joko salakuunnella yhteyttä tai yrittää saada Matin kertomaan salasanansa palvelimelle, joka onkin hyökkääjän hallussa. Lisäksi hän voi hyökätä suoraan palvelimelle ja yrittää saada kaikkien käyttäjien salasanat haltuunsa. Tämän ehkäisemiseksi palvelimelle tallennetaan vain riittävät tiedot salasanan oikeellisuuden tarkistamiseksi, jotta hyökkääjä ei voi helposti päätellä niiden perusteella oikeaksi kelpaavaa salasanaa [DiH76].

Salasanaan voidaan esimerkiksi ensin lisätä satunnainen osa, niin sanottu suola, ja käyttää siihen sitten turvallista hajautusfunktiota, minkä jälkeen palvelimelle tallennetaan tieto siitä, mitä salasanaan lisättiin ja hajautusfunktion tulos. Kun Matti lähettää salasanansa, suolaus (*salting*) ja hajautus toistetaan sekä tuloksia verrataan. Mikäli hajautuksen tulos on sama, salasana hyväksytään.

Toisaalta salasanaa s lähetettäessä voidaan myös käyttää apuna hajautusta tai mitä tahansa muuta ns. yksisuuntaista funktiota (ks. luku 3) f, jossa $y = f(x)$ on helppo laskea, mutta x:n ratkaiseminen y:n perusteella on huomattavasti vaikeampaa. Hajautusfunktioksi käy esimerkiksi turvallinen hajautusmetodi SHA-1 [SHA95]. Palvelin, jolle Matti yrittää todistaa henkilöllisyyttään, lähettää ensin Matille haasteena jonkin satunnaisen merkkijonon r. Matti liittää tämän ennalta sovitulla tavalla $g(s, r)$ omaan salasanaansa s ja laskee näin saamansa uuden merkkijonon perusteella yksisuuntaisella funktiolla arvon $y = f(g(s, r))$, jonka lähettää palvelimelle. Tämä

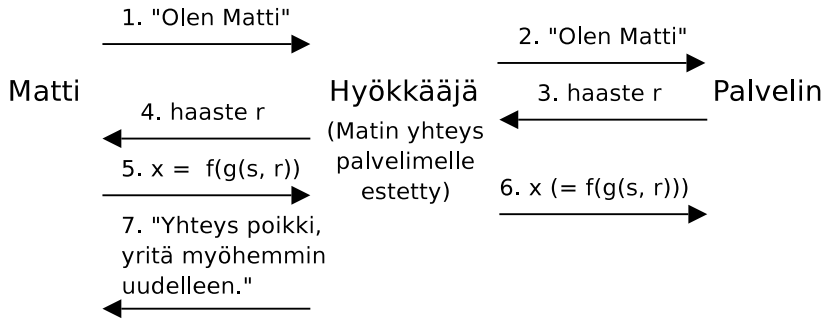


Kuva 1: Tavallinen viestinvaihtotilanne Matin ja palvelimen välillä.

viestien vaihto on esitetty kuvassa 1.

Viestinvaihtoa salakuunteleva hyökkääjä ei saa tietää Matin salasanaa, vaan hän voi vain tallentaa satunnaisosan ja sen perusteella lasketun vastauksen. Tästä on Mattina esiintymään pyrkivälle hyökkääjälle hyötyä vain, jos palvelin tarjoaa myöhemmin hyökkääjälle samaa haastetta r. Tällöin hyökkääjä voi toistaa vastauksen, jonka kuuli aiemmin Matilta. Toisaalta koska palvelimen pitää nyt voida yhdistää salasana vastaavalla tavalla erilaisiin haasteisiin Matin vastauksen tarkistamiseksi, sen tarvitsee tallentaa salasana kokonaisuudessaan. Tällöin hyökkäys palvelimelle salasanojen selvittämiseksi voi jälleen muuttua houkuttelevammaksi verrattuna edellä kuvailtuun tilanteeseen, jossa palvelimelle voidaan tallentaa vain hajautusfunktion tulos suolatusta salasanasta.

Salakuuntelun vaikeuttamisesta huolimatta Matin on luotettava siihen, että kysyjä on oikealla asialla. Hyökkääjä voi muutoin toimia aktiivisena välimiehenä, ja todistaa palvelimelle olevansa Matti, Matin itsensä avustamana: Matin ottaessa yhteyden palvelimelle hyökkääjä asettuukin osapuolten väliin ja esittäytyy itse palvelimelle Mattina. Kun palvelin lähettää hyökkääjälle haasteen, tämä lähettää sen eteenpäin Matille, joka laskee oikean vastauksen salasanaansa perustuen ja lähettää sen takaisin hyökkääjälle luullen tätä kohdepalvelimekseen. Hyökkääjän tarvitsee nyt vain välittää Matin oikea vastaus viralliselle palvelimelle, joka täten uskoo hyökkääjän olevan Matti. Lopuksi hyökkääjä voi halutessaan ilmoittaa Matille, että yhteys on katkennut tai varmistaa muuten, ettei palvelimelle päädy kahta Mattia yhtäaikaa. Tämä on esitetty kuvassa 2. Tämän ehkäisemiseksi voi olla tarpeen valita erityinen



Kuva 2: Hyökkääjä asettuu Matin ja palvelimen väliin, ja uskottelee molemmille olevansa toinen heistä.

alkutunnistusmetodi, jota vastaan aktiivisen välimiehen hyökkäys ei onnistu ja jolla Matti voi vakuuttua viestivänsä haluamansa palvelimen kanssa.

3 Julkisen avaimen salaus- ja allekirjoitusjärjestelmät

Yhteiseen salaisuuteen perustuvat menetelmät ovat sinänsä hankalia, että niitä ei voi käyttää ennen kuin osapuolet ovat saaneet viestittyä salaisuuden toisilleen. Tätä varten tarvitaan salakuuntelulta ja muulta häirinnältä suojattu kanava, joka puolestaan usein vaatii jonkinlaista salausavainten vaihtoa, eikä viestintä tahdo päästä alkuun. Tässä luvussa kuvaillaan julkisen avaimen menetelmiä, jotka perustuvat eräänlaiseen salaisuuden jakamiseen kahtia. Toinen osa pitää yhä pitää salassa, mutta toinen voidaan julkaista vapaasti. Turvalliseen viestintään pyrkivät osapuolet voivat suojata viestintänsä salauksella tai jopa jatkossa suoraan vakuuttua toistensa henkilöllisyydestä näiden jaettujen salaisuuksien avulla. Kumpikin kehittää oman salaisuutensa, ja näiden julkiset osat toimitetaan toiselle osapuolelle. Henkilöllisyys todistetaan tällöin salaisen osan tuntemisen avulla. Julkisen avaimen kryptografiassa näitä salaisuuden kahta osaa kutsutaan julkiseksi ja salaiseksi avaimeksi.

On ehdottoman tärkeää, että pelkästään julkista avainta tarkastelemalla on erittäin vaikeaa päätellä mitään salaisesta avaimesta. Yleensä julkisen avaimen järjestelmä perustuu johonkin laskennallisesti vaikeaan ongelmaan, josta on johdettu yksisuuntainen funktio. Näitä käsitellään aliluvussa 3.1. Vaikka julkisen avaimen järjestelmä vaikuttaakin lupaavalta ratkaisulta tunnistusongelmaan, senkin pitää päästä jotenkin alkuun: henkilöllisyyden tarkistamista tarvitaan taas, jotta esimerkiksi Liisa ei voisi levittää omaa julkista avaintaan Matin nimissä.

3.1 Yksisuuntaiset funktiot

Salaisuus voidaan jakaa kahtia esimerkiksi yksisuuntaisen metodin avulla. Nämä ovat usein funktioita, mutta joissakin sovelluksissa voidaan myös käyttää sopivaa epädeterminististä metodia, jossa yhtä parametriä x voi vastata kaksi tai useampi eri arvo y.

Metodin f katsotaan olevan yksisuuntainen, mikäli annetun arvon x perusteella on laskennallisesti helppoa ratkaista y siten, että $y = f(x)$, mutta mikäli tiedossa on vain y ja f, on laskennallisesti vaikeaa ratkaista x samasta yhtälöstä [DiH76]. Tätä metodia hyödynnetään siten, että salaisen avaimen hallussapitäjä pääsee kulkemaan "helppoa tietä" tuottaakseen avainparinsa, salauksen, digitaalisen allekirjoituksen tai todistuksen henkilöllisyydestään, kun taas salaista avainta tuntematon hyökkääjä joutuu kulkemaan vastavirtaan ja ratkaisemaan laskennallisesti vaikean vastaongelman. Esimerkiksi sopivasti rajoittuneessa, laskimettomassa ja vähäisen muistiinpanotilan maailmassa voitaisiin valita $f(x) = x^2$. Luvun kertominen itsellään on suhteellisen yksinkertainen päässälasku, kun taas tämän maailman hyökkääjän on pakko ratkaista vastaongelma eli neliöjuuren määrittäminen arvaamalla jokin luku $x' \in \mathbf{R}$ ja kokeilla sitten neliöimällä, oliko arvaus oikein.

Hyökkääjällä voi olla myös muita vaihtoehtoja kuin vastaongelman suora ratkaisu, mutta nämä vaihtoehdot ovat yleensä yhtä vaikeita kuin varsinaisen vastaongelman

ratkaiseminenkin. Mikäli hyökkääjä voi saada selville salaisen avaimen tai minkä tahansa viestin ratkaisemalla jonkin vaihtoehtoisen ongelman, joka on huomattavasti vastaongelmaa helpompi, järjestelmä on murrettu. Kun edellisen esimerkkimme maailmassa keksitään esimerkiksi Newtonin iteraatiomenetelmä, jonka avulla sokean kokeilun sijaan hyökkääjä voi varsin yksinkertaisia yhteen- ja jakolaskuja käyttäen päästä äärettömän lähelle neliöjuurta sitä varsinaisesti "ratkaisematta", vastaongelman ratkaisu helpottuu huomattavasti. Voinemme olettaa, että päässälaskuna ei voida neliöidä kovinkaan monidesimaalisia lukuja, joten tietyssä pisteessä hyökkääjä voi vain arvioida oikean arvon x pyöristämällä. Neliöjuurijärjestelmä on siis murrettu.

3.2 Julkisen avaimen järjestelmien yleinen toimintaperiaate

Julkisen avaimen kryptografia esiteltiin ensi kertaa 1970-luvun puolivälissä [DiH76]. Jokaisella käyttäjällä on salausmetodipari, joista toinen, $S(x)$, on pidettävä salassa, kun taas toisen, $J(x)$, voi huoletta julkaista. Julkisella metodilla salatun viestin voi muuttaa salaisella metodilla takaisin selkokieliseksi: $S(J(x)) = x$. Erityisesti pätee, että jos j ja s ovat kaksi sopivasti valittua arvoa, julkisen avaimen järjestelmä voidaan toteuttaa yksisuuntaisella funktiolla, jossa $f(s, f(j, x)) = x$. Tällöin j toimii julkisena ja s salaisena avaimena. Kun joku haluaa lähettää salaisen viestin Matille, hän salaa sen käyttäen Matin julkista avainta. Tätä salattua viestiä ei voi avata ilman Matin salaista avainta—lähettäjäkään ei siis voi purkaa salausta.

Jotkin julkisen avaimen salausalgoritmit mahdollistavat myös digitaaliset allekirjoitukset, joiden aitouden kuka tahansa voi tarkistaa hankittuaan itselleen luotettavan julkisen avainkappaleen. Digitaaliset allekirjoitukset toimivat tietynlaisissa tilanteissa tunnistamisen apuvälineinä, mutta aivan yksinkertaista niiden käyttö ei ole. Esimerkiksi jos Matti todistaa henkilöllisyytensä aina vain allekirjoittamalla viestin, jossa lukee "Hei, olen Matti", mikä estää hyökkääjää kuuntelemasta linjaa, ottamasta allekirjoitetun viestin talteen ja lähettämästä sen uudelleen?

Vaikka julkisen avaimen järjestelmässä julkinen avain voidaan huoletta lähettää turvattomankin kanavan yli, vastaanottaja ei voi täysin luottaa siihen, että julkisen avaimen lähettäjä on se, kuka väittää olevansa, ellei tämä ole ensin todistanut henkilöllisyyttään. Toisaalta suorittamalla tämän vaihdon kerralla kunnolla, esimerkiksi tapaamalla henkilökohtaisesti, osapuolet voivat jatkossa hoitaa kaiken keskinäisen tunnistamisensa samalla järjestelmällä. Lisäksi on kehitetty erilaisia tukijärjestelmiä, jotka mahdollistavat julkisten avainten levittämisen erinäisten luotettujen kolmansien osapuolten, varmentajien, kautta [Koh78, MWR89]. Tästä lisää luvussa 3.4.

3.3 Esimerkkijärjestelmiä

RSA lienee julkisen avaimen salausjärjestelmistä tunnetuin. Ron Rivest, Adi Shamir ja Leonard Adleman kehittivät järjestelmän sekä viestien salausta että digitaalisia allekirjoituksia varten, ja se perustuu lukujen tekijöihin jakamisen ongelmaan [RSA78]. RSA-algoritmia voidaan käyttää viestien allekirjoittamiseen vaihtamalla julkisen ja salaisen avaimen paikkoja algoritmissa: jos avaimenhaltija "salaa" viestin käyttäen salaista avaintaan d julkisen avaimen e sijaan ja lähettää tuloksen alkuperäisen viestin mukana, kuka tahansa voi purkaa salauksen väitetyn lähettäjän julkisen avaimen avulla ja verrata saamaansa selkotekstiä alkuperäiseen viestiin. Mikäli ne täsmäävät, lähettäjän allekirjoitus on "aito"—lähettäjällä on siis hyvin todennäköisesti hallussaan oikea salainen avain. Oikeannäköisiä allekirjoituksia voidaan toki luoda myös takaperoisella menetelmällä: valitaan tai arvotaan ensin allekirjoitus ja lasketaan siitä salausavaimen avulla "viesti", joka on aloittelevan väärentäjän kannalta satunnainen bittijono, johon hän ei voi juurikaan vaikuttaa. Mikäli viestin on sovittu olevan aina tietyssä muodossa (esim. luettavana tekstinä), vastaanottaja voi rauhassa hylätä tällaiset viestit.

Taher ElGamalin 1985 kehittämä ElGamal-menetelmä perustuu diskreettien loga-

ritmien määrittämisen vaikeaksi oletettuun ongelmaan [ElG85]. Sykliset lukuryhmät koostuvat rajallisesta määrästä alkioita siten, että ryhmän viimeisen alkion jälkeen palataan ryhmän ensimmäiseen alkioon. Esimerkiksi ryhmä $\mathbf{Z}_7$ koostuu luvuista 0, 1, 2, 3, 4, 5 ja 6, joille pätevät mm. $6 + 1 = 0$ ja $3 * 4 = 5$. Tällaisessa ryhmässä potenssiin korottaminen on varsin helppoa, mutta vastakkainen suunta, diskreetin logaritmin määrittäminen, on laskennallisesti vaikeampaa. Esimerkiksi $5^2 = 25 \equiv 4 \pmod 7$, mutta miten ratkaista tehokkaasti $log_5 4 \pmod 7$? ElGamal on suunniteltu digitaalisia allekirjoituksia varten, mutta samoja avaimia voidaan käyttää myös muunnetussa, salaukseen kelpaavassa ElGamal-järjestelmässä.

3.3.1 RSA

Avaimen luontia varten tarvitaan kaksi suurta[1] ja satunnaista alkulukua, p ja q. Näiden tulon, $n = pq$, laskemisen jälkeen salauksessa käytettäväksi julkiseksi avaimeksi valitaan satunnaisesti luku e, jolla ei ole yhteisiä tekijöitä luvun $\phi(n) = (p-1)(q-1)$ kanssa. Tässä $\phi(x)$ on Eulerin ϕ-funktio [Jac85, s. 103-105]. Salauksen purkamiseen käytettävä salainen avain d saadaan yhtälöstä

$$ed \equiv 1 \pmod{\phi(n)}$$

(eli kongruenssin[2] määritelmän mukaan $ed = k\phi(n)+1$ pätee jollekin kokonaisluvulle k). Luku d on kokonaisluku, eikä sillä ole yhteisiä tekijöitä luvun $(p-1)(q-1)$ kanssa.

Viestin m salaamiseksi se on jaettava osiin, joista kukin on lukua n pienempi. Mikäli jokin viestin osa esimerkiksi vastaisi lukua $n+3$, sitä ei RSA-käsittelyn jälkeen voisi enää erottaa luvusta 3, koska tulokset pelkistetään aina välille $0..n-1$. Kukin osa m_i salataan erikseen, ja samaa kokoluokkaa olevat tulokset liitetään yhteen salatuksi

[1] "Suuri alkuluku" tässä kontekstissa on reilusti yli sadan numeron pituinen.

[2] Lauseke $x \equiv y \pmod z$ kertoo, että jos luku x jaetaan luvulla z, jakojäännös on y, eli että $x = k * z + y$ pätee jollekin kokonaisluvulle k.

viestiksi c. Kukin osa c_i saadaan yhtälöstä

$$c_i = m_i^e \mod n$$

Salauksen purkamiseksi kukin osa c_i käsitellään jälleen erikseen:

$$m_i = c_i^d \mod n$$

Kaikille kokonaisluvuille x, y ja z pätee, että jos $y \equiv 1 \pmod{\phi(z)}$ ja x:llä ei ole yhteisiä tekijöitä luvun z kanssa, on $x^y \equiv x^1 = x \pmod z$ [Jac85, s. 103-105]. Yllä kuvailtu metodi siis purkaa salauksen, koska (1) mikäli viestillä m ja moduluksella n ei ole yhteisiä tekijöitä, pätee

$$c_i^d = (m_i^e)^d = m_i^{ed} = m_i^{k\phi(n)+1} \equiv m_i * 1 = m_i \pmod n.$$

Lisäksi (2) mikäli viestillä ja moduluksella on yhteisiä tekijöitä, on lopputulos sama, mutta hieman monimutkaisemmin johdettuna. Tämä on käsitelty tarkemmin artikkelissa [RSA78].

Julkinen avain koostuu lukuparista n ja e, ja salainen avain luvusta d.

Esimerkki selventänee järjestelmän toimintaa. Liisa haluaa lähettää Matille viestin, joka voidaan esittää kymmenjärjestelmän lukuna $m = 999$. Matti on toki jo aiemmin luonut avaimensa. Aluksi Matti on valinnut satunnaisesti kaksi alkulukua, $p = 11$ ja $q = 5$. Nämä luvut ovat aivan liian pieniä varsinaisen salausjärjestelmän käyttöön, mutta kelpaavat esimerkiksi. Lukujen tulo on $n = pq = 11 * 5 = 55$. Tämän jälkeen Matti on valinnut satunnaisesti luvun $e = 3$, jolla ei ole yhteisiä tekijöitä luvun $\phi(n) = (p-1)(q-1) = 10 * 4 = 40 = 2^3 * 5$ kanssa. Salainen purkuavain d on saatu kongruenssiyhtälöstä $ed = 3 * d \equiv 1 \pmod{40}$, josta voidaan ratkaista pienellä skriptillä $d = 27$. Tällöin saadaan haluttu tulos $ed = 3 * 27 = 81 = 2 * 40 + 1 \equiv 1 \pmod{40}$. Koska $d = 27 = 3^3$, ei myöskään luvulla d ole yhteisiä tekijöitä luvun $\phi(n) = 40$ kanssa. Matin julkinen avain on siis lukupari $n = 55$ ja $e = 3$, mikä on myös Liisan tiedossa, sekä salainen avain $d = 27$.

Muutetaan nyt viesti m 55-kantaiseksi moduluksen n mukaan: $999 = 18 * 55 + 9$. Toisin sanoen m koostuu kahdesta osasta, $m_0 = 18$ ja $m_1 = 9$. Salaamme näistä ensimmäisen osan m_0. Toisen osan salaaminen tapahtuu vastaavasti, mutta välituloksena saadut luvut ovat turhan suuria tässä esitettäviksi. Liisan on siis selvitettävä $c_0 = m_0^e \mod n$, josta hän saa tulokseksi $c_0 = 18^3 = 5832 \equiv 2 \pmod{55}$. Nyt hän voi lähettää Matille salatun viestin ensimmäisen osan $c_0 = 2$. Kun Matti vastaanottaa luvun c_0, hän purkaa salauksen kaavan $m_0 = c_0^d \mod n$ avulla. Matti saa tulokseksi $m_0 = 2^{27} = 134\ 217\ 728 \equiv 18 \pmod{55}$. Voimme todeta Matin tuloksen $m_0 = 18$ vastaavan Liisan salaamaa lukua. Mikäli Matti haluaa vielä muuntaa saamansa viestin kymmenkantaiseksi vastaanotettuaan myös viestin toisen osan m_1, hänen tarvitsee vain laskea $m_0 * 55 + m_1 = 18 * 55 + 9 = 999$.

3.3.2 ElGamal

ElGamal-avainparin luomiseksi tarvitaan RSA-algoritmin tapaan suuri alkuluku p. Tämän jälkeen valitaan satunnaisesti kaksi lukua, g ja x, jotka molemmat ovat pienempiä kuin p, ja lasketaan $y = g^x \mod p$. Julkinen avain koostuu luvuista p, g ja y, kun taas x pitää pitää salassa. Koska p on alkuluku eikä kahden alkuluvun tulo RSA-algoritmin moduluksen n tapaan, on $\phi(p) = p - 1$ (vertaa RSA).

Viestin m allekirjoittamista varten pitää valita satunnaisesti luku k siten, ettei sillä ja luvulla $\phi(p)$ ole yhteisiä tekijöitä, ja määrittää $a = g^k \mod p$. Viestin allekirjoitus koostuu kahdesta osasta. Toinen on edellä määritetty a, joka siis riippuu avaimesta ja viestikohtaisesta satunnaisluvusta k, ja toinen b, joka ratkaistaan suoraviivaisen algoritmin avulla yhtälöstä $m = (xa + kb) \mod \phi(p)$. Koska luvulla g ei ole yhteisiä tekijöitä luvun p kanssa, pätee myös $g^m \equiv g^{xa+kb} \pmod{p}$ (katso RSA).

Allekirjoituksen tarkastamiseksi riittää varmistaa, että kongruenssi $y^a a^b \equiv g^m \pmod{p}$ pätee. Tällöin, koska $y = g^x \mod p$ ja $a = g^k \mod p$, saadaan

$$y^a a^b = (g^x)^a (g^k)^b = g^{ax+kb} \equiv g^m \pmod{p}.$$

Samaa lukua k ei tule käyttää kuin kerran; kahdella viestin ja siihen liittyvän allekirjoituksen parilla voidaan määrittää k, mikäli se ollut on sama molemmissa [ElG85]. Lisäksi tiettyyn viestin allekirjoitukseen liittyvää lukua k ei saa paljastaa, sillä sijoittamalla se viestin m ja allekirjoituksen osien a ja b lisäksi viimeksi mainittuun yhtälöön voidaan ratkaista x. Käytännössä luvun k toistoa vältetään lähinnä valitsemalla suuri p. Tällöin mahdollisia erilaisia arvoja k on riittävän monta, jotta saman arvon toistuminen on sopivan epätodennäköistä.

3.4 Voiko julkisen avaimen oikeellisuuteen luottaa?

Salasanan tietäminen ei todista tietäjän olevan jokin tietty henkilö; se kertoo vain, että kyseinen henkilö tietää syystä tai toisesta kyseisen salasanan. Julkinen avain kertoo joskus vielä vähemmän, sillä sitä ei yleensä erikseen sovita viestijäosapuolten kesken, vaan avainparin luoja tuo sen julkisuuteen itse parhaaksi katsomallaan tavalla. Matin mummo Marja ei ehkä ole koskaan koskenutkaan tietokoneeseen, mutta kuka tahansa voi julkaista lehdessä julkisen avaimensa Marja Meikäläisen nimellä. Mikäli joku sitten yrittää käyttää tätä avainta viestiäkseen Marjan kanssa, hyökkääjä voi kaapata viestin ja on itse asiassa ainoa, joka voi poistaa salauksen. Hän voi viestin luettuaan peittää jälkiään lähettämällä sen eteenpäin Marjalle uudelleensalattuna tämän omalla julkisella avaimella, mikäli Marjalla sellaista on.

Yleensä julkiseen avaimeen kuuluvat arvot katsotaan osaksi varmennekokonaisuutta, jossa ovat mukana avaimen omistajan nimi ja riittävästi muita tietoja varmaa tunnistamista varten, jonkinlaiset yhteystiedot ja omistajan käyttämien julkisten avainten arvot ja mahdolliset lisätiedot avaimista. Tällöin varmenteen aitoudesta vakuuttuminen riittää yhdistämään julkisen avaimen tiettyyn henkilöön. ISO-standardi X.509 (ISO DIS 9594-8) määrittelee varmenteen sisällön ja sisältää ohjeita varmenteiden levityksen toteuttamiseksi [MWR89].

Varmenteiden aitouden tarkistamiseen ei ole yhtä varmaa ratkaisua. Yksi yleinen

ratkaisumalli perustuu siihen, että johonkin yleisesti tunnettuun osapuoleen voidaan luottaa. Tämä osapuoli, varmentaja, allekirjoittaa varmenteen omalla avaimellaan jos ja vain jos varmenteen omistaja todistaa henkilöllisyytensä tyydyttävästi. Koska tätä tarvitsee tehdä vain hyvin harvoin, varmenteen omistaja voi saapua paikalle henkilökohtaisesti ja näyttää esimerkiksi ajokorttiaan varmentajalle. Toisaalta jos varmentajan avain anastetaan tai murretaan—tästä tulee sitä houkuttelevampaa, mitä useampi varmenne avaimesta riippuu—kaikkien pitää käydä allekirjoituttamassa varmenteensa uudelleen. Lisäksi luottamuksen keskittäminen toimii käytännössä jokseenkin huonosti taloudellisista ja poliittisistakin syistä. Tästä syystä näemmekin lähes jatkuvasti esimerkiksi WWW-selaimen varoituksia varmenteista, joiden myöntäjää ei tunnisteta. Jos varmentajan omaa julkista avainta ei ole asennettu käyttämäämme selaimeen, ei selain voi tietää, onko varmenne luotettava.

Useat yksityishenkilöt ovatkin päätyneet lähes päinvastaiseen ratkaisuun: luottamuksen verkkoon. Kukin käyttäjä allekirjoituttaa varmenteensa useilla käyttäjillä, jotka hän tavalla tai toisella voi tehokkaasti vakuuttaa henkilöllisyydestään. Ajatuksena on, että kun uusi käyttäjä haluaa varmistaa varmenteen kuuluvan Matti Meikäläiselle, hän saattaa löytää allekirjoitusten joukosta sellaisen, johon hän voi luottaa. Mikäli sellaista ei löydy, hän voi joko korjata tilanteen tapaamalla Matin ja vakuuttumalla siitä, että kyseessä on hänen avaimensa, tai jatkaa etsimistä: kenties jokin allekirjoituksista kuuluu henkilölle, jonka varmenteen oikeellisuudesta voidaan helposti vakuuttua ja jonka tiedetään ottavan muiden varmenteiden allekirjoittamisen vakavasti. Epävarmuus lisääntyy matkan varrella, mutta käyttäjä voi itse päättää, miten suuri varmuus riittää hänen käyttötarkoitukseensa. Tätä periaatetta käytetään muun muassa OpenPGP-standardiin [CDF98] perustuvissa sovelluksissa Pretty Good Privacy [PGP03] ja Gnu Privacy Guard [Gnu03].

4 Nollatietoprotokollat

Julkisen avaimen salausta ja digitaalisia allekirjoituksia käytettäessä ei ole lainkaan varmaa, etteikö samalla vuodettaisi vähäisiä määriä tietoa salaisesta avaimesta. ElGamalin tapauksessa uhka on todellinen: jos allekirjoittamiseen tarvittava salainen satunnaisosa k toistuu samana kahdessa eri allekirjoituksessa, voidaan näiden ja alkuperäisten viestien avulla ratkaista allekirjoittajan salainen avain [ElG85]. Allekirjoituksen sijaan voidaankin käyttää nollatietotodistusta, jolla varmistetaan salaisen avaimen tai yhteisen salaisuuden tuntemus.

Esimerkkinä käsitellään kaksi nollatietoprotokollaa, joista ensimmäinen, Fiat-Shamirin nollatietoprotokolla, on kuvailtu 1988 julkaistussa artikkelissa [FFS88]. Jälkimmäinen protokolla mahdollistaa tunnistamisen ohella samanaikaisen salausavainten vaihdon. Tämä protokolla kehitettiin alun perin samoihin aikoihin Aarhusin yliopistossa Tanskassa, mutta viimeistellyn artikkelin julkaisu oli lykkääntynyt vuoteen 1996 asti [BDL96].

4.1 Nollatietotodistusten yleinen toimintaperiaate

Nollatietotodistuksella on tiukat vaatimukset. Sillä vakuutetaan vastapuoli siitä, että käyttäjä tuntee tietyn, henkilöllisyyttä todistettaessa tunnuksena toimivan salaisuuden. Lisäksi vastapuoli ei saa oppia todistuksen perusteella mitään sellaista, mitä ei jo tietäisi tai voisi itse helposti laskea [GMR85].

Nollatietotodistuksen ominaisuuksiin kuuluu, että ulkopuolinen voi generoida julkisesti saatavilla olevalla tiedolla täysin uskottavia todistusprotokollan viestinvaihtoja. Hän ei kuitenkaan voi todistaa tuntevansa salaisuutta, sillä hän ei aidossa todistustilanteessa esimerkiksi voi tuntea todistusta määrääviä haasteita ennalta. Muuttuvilla haasteilla ja tarvittaessa useaa todistuskierrosta käyttämällä rajataan

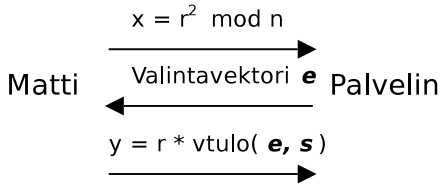
haasteiden onnekkaan arvaamisen todennäköisyys tarvittavan pieneksi.

Uskottavilla viestinvaihdoilla on kaksi ominaisuutta. Ensinnäkin viestinvaihdon tulee edetä siten kuin se etenisi "oikeassa" todistustilanteessa. Mikäli siis jollakin kierroksella jonkin osapuolen viesti ei ole protokollan mukainen vaan esimerkiksi väärä laskennan tulos, protokolla keskeytyy eikä todistusta hyväksytä. Tällaisten hylättyjen todistusten osuus kaikista generoiduista todistuksista saa kuitenkin olla vain niin suuri kuin se todennäköisesti olisi aidossa todistustilanteessa, eli todistajan imitoijan tulee myös onnistua todistuksissaan varsin usein. Toiseksi viestinvaihdoissa luotujen satunnaislukujen tulee noudattaa samanlaista jakaumaa kuin aidon todistustilanteen satunnaislukujen. Tällä varmistetaan, etteivät generoidut todistukset käsittele vain muutamaa helppoa tapausta, jolloin kuulija voisi sittenkin saada uutta tietoa todistuksesta silloin, kun käsitellään huomiotta jätettyjä tapauksia.

4.2 Fiat-Shamirin nollatietoprotokolla

Fiat-Shamirin protokolla [FFS88] henkilöllisyyden todistamiseen perustuu julkisen avaimen kryptografian tapaan yksisuuntaiselle funktiolle. Neliöjuuren määrittämisen uskotaan olevan vaikeaa syklisissä lukuryhmissä, joita kuvailtiin kappaleessa 3.3. Toisaalta luvun neliöinti ei ole kovin vaikeaa. Amos Fiat ja Adi Shamir valitsivatkin yksisuuntaisen funktion $f(x) = x^2$ protokollansa perustaksi.

Protokolla tarvitsee luotetun kolmannen osapuolen, joka valitsee satunnaisesti RSA-tyyliin moduluksen n, jolle pätee $n = pq$, missä p ja q ovat suuria alkulukuja. Kunhan n on julkaistu, tämä kolmas osapuoli voi tarvittaessa lakata olemasta. Arvojen p ja q tulee silti pysyä salassa. Tämän jälkeen kaikki, joiden on tarkoitus todistaa henkilöllisyytensä, valitsevat k salaista satunnaislukua s_i siten, että kaikille kokonaisluvuille i välillä $[0, k-1]$, pätee a) $1 \leq s_i \leq n-1$ ja b) luvulla s_i ei ole yhteisiä tekijöitä luvun n kanssa. Julkinen avain määritetään tämän jälkeen siten, että kunkin luvun s_i neliölle määritetään vastaluku $1/(s_i^2) \mod n$, ja julkisen avaimen



Kuva 3: Fiat-Shamirin nollatietoprotokolla: yhden kierroksen viestien vaihto.

osaksi valitaan satunnaisesti joko $v_i = +1/(s_i^2) \mod n$ tai $v_i = -1/(s_i^2) \mod n$. Tämän jälkeen käyttäjä julkistaa vektorin $\mathbf{v} = v_1, ..., v_k$ ja pitää salassa vektorin $\mathbf{s} = s_1, ..., s_k$. Koska neliöjuurten määrittäminen oletetaan vaikeaksi, salaista vektoria $\mathbf{s}$ ei voi käytännössä laskea julkisesta vektorista $\mathbf{v}$.

Protokollan yksi kierros koostuu kolmesta viestistä. Ensin Matti, joka haluaa todistaa henkilöllisyytensä, valitsee satunnaisesti luvun r, jolle pätee $1 \leq r \leq n-1$ ja jolla ei ole yhteisiä tekijöitä moduluksen n kanssa. Hän laskee tämän neliön r^2 ja lähettää vastapuolelle satunnaisesti joko arvon $x = +r^2$ tai arvon $x = -r^2$. Vastapuoli varmistaa ensin, että $x \neq 0$, ja valmistaa sitten satunnaisesti valintavektorin $\mathbf{e}$, joka koostuu k totuusarvosta e_i, ja lähettää vektorin Matille. Tämä vektori "valitsee" Matin salaisesta vektorista $\mathbf{s}$ luvut, jotka kerrotaan mukaan Matin vastaukseen. Matti vastaa siis laskemallaan luvulla $y = r * \Pi_{e_i=1} s_i \pmod n$. Vastapuoli hyväksyy tämän kierroksen tuloksen, jos ja vain jos $x \equiv \pm y^2 * \Pi_{e_i=1} v_i \pmod n$. Mikäli Matti on laskenut luvun y oikein, on $y^2 = (r * (\Pi_{e_i=1} s_i))^2 = r^2 * \Pi_{e_i=1} s_i^2$. Koska kaikille $i \in [0, k-1]$ pätee, että $v_i = \pm 1/(s_i^2) \mod n$, saadaan siis $x \equiv \pm(r^2 * \Pi_{e_i=1} s_i^2) * \Pi_{e_i=1} 1/(s_i^2) = \pm r^2 * \Pi_{e_i=1}(s_i^2 * \pm 1/s_i^2) = \pm r^2 \pmod n$. Yhden kierroksen viestien vaihto on esitetty kuvassa 3. Tämä viestien vaihto toistetaan t kertaa, ja vastapuoli hyväksyy todistuksen jos ja vain jos kaikki t vaihtoa tuottavat hyväksyttävän lopputuloksen [FFS88].

Mikäli Mattina esiintyvä hyökkääjä arvaa valintavektorin $\mathbf{e}$ ennen luvun r lähettämistä, hän voi laskea ennalta tulon $z = \pm r^2 \Pi_{e_i=1} v_i \pmod n$ ja lähettää sen ensimmäisenä arvona x, ja lähettää myöhemmin lukuna y arvon r. Tällöin "todistuksen

vastaanottajan" tarkistus tuottaa suoraan yllä kuvaillun, halutun tuloksen:

$$x = z \equiv \pm r^2 \Pi_{e_i=1} v_i \pmod{n} \equiv \pm y^2 \Pi_{e_i=1} v_i \pmod{n}$$

Tämä osoittaa samalla sen, että ulkopuolinen, joka luonnollisesti voi arvata, mitä itseltään kysyy, voi helposti tuottaa aidonnäköisiä todistuksia. Todistusten ulkopuolista generointia käsitellään tarkemmin Fiatin ja Shamirin artikkelissa [FFS88].

4.3 Nollatietotunnistus ja salaisen avaimen vaihto

Julkisen avaimen salausta ei käytännössä juurikaan käytetä kaiken viestinnän salaamiseen monestakin syystä, joista yksi on algoritmien suhteellinen hitaus verrattuna moniin perinteisiin salausalgoritmeihin, joissa molemmilla osapuolilla on käytössään yhteinen, salainen avain. Näitä avaimia vaihdetaan usein jokaisen viestienvaihdon alussa esimerkiksi salaamalla ne julkisilla avaimilla, kunhan tunnistaminen on hoidettu. Tällaisia jatkuvasti vaihtuvia avaimia kutsutaan kertatunnuksiksi (*session key*). Aarhusin yliopistossa kehitetty tunnistusprotokolla mahdollistaa tämän avaimenvaihdon jo tunnistuksen yhteydessä, jolloin ylimääräistä avaimenluontia ja salausta ei tarvita. Kutsun protokollaa BDLP-tunnistusprotokollaksi sen tekijöiden mukaan. BDLP ei ole riippuvainen yksittäisestä ongelmasta, kuten Fiat-Shamir RSA:n perustasta eli luvun tekijöihin jakamisesta. Protokolla on määritelty yksisuuntaisesta ongelmasta riippumattomana, mutta esimerkiksi RSA:n mainitaan sopivan tässä käytettäväksi alijärjestelmäksi [BDL96].

Järjestelmää varten määritellään funktio $tail_{k,l}$, jolle pätee $1 \leq l \leq k$, viimeisten l bitin valintafunktiona k-bittisestä jonosta. Tarkemmin sanottuna funktiolle $tail_{k,l} : \{0,1\}^k \to \{0,1\}^l$ pätee $tail_{k,l}(x_k x_{k-1} \cdots x_1) = x_l x_{l-1} \cdots x_1$. Lisäksi käyttäjän tulee valita funktiopari P_k ja P_k^{-1}. Funktiot voidaan tässä käsittää käyttäjän salaus- ja purkufunktioiksi. P_k on julkinen, kun taas P_k^{-1} pidetään salassa. Jos käyttäjä esimerkiksi valitsee käyttävänsä pohjana RSA:ta, olisi $P_k(x) = x^e$ ja $P_k^{-1}(x) = x^d$, joissa e ja d ovat käyttäjän julkinen ja salainen RSA-avain. Lopuksi tarvitaan vielä

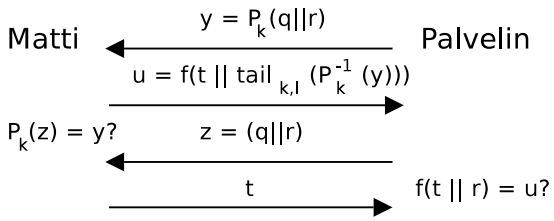
toinen käyttäjien kannalta yksisuuntainen funktio f, jota käytetään protokollassa tiettyjen viestien sekoittamiseen. Käyttäjät eivät koskaan tarvitse vastafunktiota f^{-1}, mutta järjestelmän toimivuuden todistamiseksi sen tulee olla olemassa, toimia polynomisessa ajassa ja pysyä salassa [BDL96]. Tekijät suosittavatkin, että luotettu kolmas osapuoli määrittää ja julkaisee funktion f. Käytännössä f voisi olla esimerkiksi jonkin varmentajan julkinen RSA-salausfunktio.

Syötejonojen pituudet on määritelty siten, että funktiot P_k, P_k^{-1} ja f ottavat kaikki syötteekseen sekä tuottavat tuloksenaan k-bittisiä lukuja. Näistä l viimeisen bitin oletetaan olevan niin satunnaisia, ettei niitä voi arvata. Esimerkiksi, mikäli käytettävä taustajärjestelmä olisi RSA, jossa modulus on n ja salauseksponentti e, olisivat luvun $x^e \mod n$ viimeiset l bittiä "satunnaisia". Luvun l on todistettu RSA:n tapauksessa olevan $O(\log \log n)$ [ACG84].

Itse todistus vaatii vain yhden kierroksen. Palvelin, jolle Matti haluaa todistaa itsensä, generoi haasteeksi bittijonon $s = \{0,1\}^k$. Tämän jälkeen se salaa jonon s ja jakaa tuloksen kahdeksi jonoksi q ja r, joiden pituudet ovat $k-l$ ja l. Jos siis merkitsemme jonojen liitosta merkillä $||$, on $(q||r) = P_k(s)$. Luku $a = tail_{k,l}(s)$ toimii kertatunnuksena, joten palvelin kirjaa sen muistiin, ja lähettää Matille luvun $y = P_k(q||r) = P_k(P_k(s))$.

Matti laskee salaisen avainfunktionsa avulla vastaanottamastaan luvusta y tuloksen $x = P_k^{-1}$, jolle siis pätee myös $x = (q||r)$. Hän luo satunnaisen bittijonon t, jonka pituus on $k-l$, korvaa tällä x:n $k-l$ ensimmäistä bittiä (eli luvun q) ja salaa tuloksen varmentajan yksisuuntaisella salausfunktiolla f. Hän siis lähettää vastapuolelle tuloksen $u = f(t||tail_{k,l}(x))$.

Vastaanotettuaan luvun u vastapuoli lähettää Matille aiemmin määrittämänsä luvun $z = (q||r)$. Matti voi nyt tarkistaa, että $P_k(z) = y$. Jos tämä on totta, hän lähettää aiemmin luomansa satunnaisen bittijonon t vastapuolelle. Tämä puolestaan tarkistaa, että $f(t||r) = u$, missä tapauksessa todistus voidaan hyväksyä.



Kuva 4: Brandt-Damgård-Landrock-Pedersen-nollatietoprotokolla.

Matti voi lopuksi laskea aiemmin saamastaan luvusta $x = (q||r)$ salaisen kertatunnuksen käyttämällä salaista avaintaan ja ratkaisemalla ensin alkuperäisen luvun s, joka saadaan yhtälöstä $s = P_k^{-1}(x)$. Tämän jälkeen Matti ratkaisee kertatunnuksen a valitsemalla tämän luvun l viimeistä bittiä, siis $a = tail_{k,l}(s)$. Viestien vaihto on esitetty kuvassa 4.

Mikäli Mattina esiintyvä hyökkääjä saa todistuksen alussa tietoonsa luvun s, hän voi helposti johtaa luvut q ja r Matin julkisen salausfunktion avulla. Tällöin hänen ei tarvitse juuri välittää palvelimen lähettämästä luvusta $y = P_k(q||r)$, vaan hän voi lähettää ennakkotietojensa pohjalta vastauksena suoraan luvun $u = f(t||r)$, jota varten hän valitsee satunnaisluvun t. Palvelimen takaisin lähettämä $z = (q||r)$ onkin jo hänen tiedossaan, mutta muodon vuoksi hyökkääjä voi todeta, että $P_k(z) = y$. Lopuksi hän lähettää palvelimelle luvun t, ja palvelin hyväksyy todistuksen todettuaan, että $f(t||r) = u$. Luvun s on siis ehdottomasti pysyttävä salassa todistuksen loppupuolelle asti. Tämä osoittaa samalla, että ulkopuolinen voi halutessaan generoida aidonnäköisiä todistuksia, sillä hänenkin tarvitsee vain tuntea itse asettamansa s. Todistusten generointia on jälleen käsitelty tarkemmin alkuperäisartikkelissa [BDL96].

5 Yhteenveto

Henkilön tarkka koneellinen tunnistaminen ei ole täysin suoraviivaista silloinkaan, kun käytössä on täysi valikoima kortin- tai sormenjälkien lukijoita ja niin edelleen. Verkon yli tämä on vielä vaikeampaa, sillä tunnistamisessa käytettävien laitteiden pitää niidenkin todistaa antavansa luotettavia lausuntoja. Tunnistusmenetelmät joudutaankin perustamaan salaisuuksille, joiden oletetaan olevan vain tietyn käyttäjän tiedossa.

Käyttäjän salaisuuksia voidaan käyttää monella tavalla. Perinteisessä salasanaan perustuvassa järjestelmässä käyttäjä kertoo salaisuutensa kysyjälle ja todistaa näin tietävänsä sen. Tällöin salaisuus on kuitenkin välttämättä myös kysyjän tiedossa. Julkisen avaimen kryptografia ja nollatietoprotokollat mahdollistavat salaisuuden hallussapidon todistamisen ilman, että salaisuutta tarvitsee paljastaa kysyjälle. Kysyjällä tulee silti olla käytössään jotakin salaisuudesta johdettua tietoa, jonka avulla todistuksen oikeellisuudesta voidaan varmistua. Esimerkiksi julkisen avaimen järjestelmässä täksi tiedoksi riittää julkinen avain.

Kunhan tunnistamisen ongelma on ratkaistu, käyttäjät tarvitsevat avukseen vielä muita tekniikoita varmistaakseen sen, että jatkossa henkilöllisyytensä juuri todistaneelta käyttäjältä tulevat viestit tunnistetaan varmasti häneltä tuleviksi, eikä esimerkiksi aktiivinen välimies voisi tässä vaiheessa esiintyä hänen nimissään. Tätä ja muita viestinnän suojaukseen liittyviä ongelmia ja ratkaisuja on käsitelty esimerkiksi Bruce Schneierin teoksessa Applied Cryptography [Sch95].

Lähteet

ACG84 Alexi, W., Chor, B., Goldreich, O., Schnorr, C., RSA/Rabin Bits are 1/2 + 1 Poly (Log N) Secure. 25th Annual Symposium of Foundations of Computer Science, IEEE 1984, s. 449-457. [Myös `http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=15500`, 28.11.2003.]

BDL96 Brandt, J., Damgård, I., Landrock, P., Pedersen, T., Zero-Knowledge Authentication Scheme with Secret Key Exchange. Journal of Cryptology 1998 11, s. 147-159. [Myös `http://www.springerlink.com/link.asp?id=31h3cdf32tufjg7y`, 28.11.2003.]

CDF98 Callas, J., Donnerhacke, L., Finney, H., Thayer, R., RFC 2440: OpenPGP Message Format. The Internet Society, November 1998. [Myös `ftp://ftp.rfc-editor.org/in-notes/rfc2440.txt`, 8.12.2003.]

DiH76 Diffie, W., Hellman, M., New Directions in Cryptography. IEEE Transactions on Information Theory, v. 22, n. 6, 1976, s. 644-654. [Myös `http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=22693`, 28.11.2003.]

ElG85 ElGamal, T., A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory, v. 31, n. 4, 1985, s. 469-472. [Myös `http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=22749`, 28.11.2003.]

EKW74 Evans, A., Kantrowitz, W., Weiss, E., A User Identification Scheme Not Requiring Secrecy in the Computer. Communications of the ACM, v. 17, n. 8, Aug 1974, s. 437-472. [Myös `http://portal.acm.org/toc.cfm?id=361082`, 28.11.2003.]

FFS88 Feige, U., Fiat, A., Shamir, A., Zero Knowledge Proofs of Identity. Journal of Cryptology 1, 1988, s. 77-94.

Gnu03 The Gnu Privacy Guard, 2003. http://www.gnupg.org, 8.12.2003.

GMR85 Goldwasser, S., Micali, S., Rackoff, C., The Knowledge Complexity of Interactive Proof-Systems. Proceedings of the 17th ACM Symposium on Theory of Computing, 1985, s. 291-304. [Myös http://portal.acm.org/toc.cfm?id=22145, 28.11.2003.]

Jac85 Jacobson, N., Basic Algebra I, Second Edition. W. H. Freeman and Company, New York, 1985.

Koh78 Kohnfelder, L. M., Toward a Practical Public Key Cryptosystem. Bachelor's thesis, MIT Department of Electrical Engineering, May 1978. [Myös http://theses.mit.edu/Dienst/UI/2.0/Composite/0018.mit.theses%2f1978-29/1?nsections=3, 28.11.2003.]

MWR89 Mitchell, C., Walker, M., Rush, D., CCITT/ISO Standards for Secure Message Handling. IEEE Journal on Selected Areas in Communications, v. 7, n. 4, May 1989, s. 517-524. [Myös http://ieeexplore.ieee.org/xpl/tocresult.jsp?isNumber=648, 28.11.2003.]

PGP03 PGP Corporation, 2003. http://www.pgp.com, 8.12.2003.

RSA78 Rivest, R. L., Shamir, A., Adleman, L. M., A method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, v. 21, n. 2, Feb 1978, s. 120-126. [Myös http://portal.acm.org/toc.cfm?id=359340, 28.11.2003.]

Sch95 Schneier, B., Applied Cryptography: Protocols, Algorithms and Source Code in C, Second Edition. John Wiley & Sons, 1995.

SHA95 Secure Hash Standard. Federal Information Processing Standards Publication 180-1, National Institute of Standards and Technology, 1995. [Myös http://www.itl.nist.gov/fipspubs/fip180-1.htm, 28.11.2003.]