Author(s):	Männistö, Jukka
Title:	Lukutaito ja taitavat luvut
Main work:	Monilukutaito kaikki kaikessa
Editor(s):	Kaartinen, Tapani
Year:	2015
Pages:	238-262
ISBN:	978-951-44-9847-3
Publisher:	Tampereen yliopiston normaalikoulu
Item Type:	Article in Compiled Work
Language:	fi
URN:	URN:NBN:fi:uta-201510052330

Jukka Männistö

LUKUTAITO JA TAITAVAT LUVUT

Taitava luku

Maailman muutosten mukana matematiikan opetus on muuttunut. Kuitenkin itse matematiikka on perusteiltaan alati samankaltaista. Matematiikka konstruoi vaikeatajuisia ja mutkikkaita struktuureja, mutta lähes poikkeuksetta kaikille luomuksille löytyy sovelluksia ja vieläpä hyvin syvällisiä.

Luotto- ja pankkikortit ovat hyvin suosittu tapa maksaa laskuja, siirtää rahaa ja tehdä ostoksia verkkokaupassa. Internetissä ostoksia tehtäessä pääsääntöisesti laskun maksamisen loppuvaiheessa kysytään joko luottokortin CVV- tai CVC-turvakoodi. Mikä luku tämä pakollinen CVV (Card Verification Value) tai CVC (Card Verification Code) oikeastaan on? Miten taitava luku se on?

American Express, Mastercard ja VISA ovat joutuneet tietoturvan vuoksi kehittämään yhä taitavampia lukuja, lukusysteemejä, salaisia laskentamalleja ja algoritmeja, jotta luottokorttien käyttö verkossa olisi ongelmatonta. Itse luottokortin numerosarja on sinänsä jo taitavasti konstruoitu ja lukuna se sisältää valtavan määrän informaatiota, mutta se ei yksin riitä. Esimerkiksi VISA-kortin takaa löytyvän kolminumeroisen CVV-luvun tarkoitus on varmistaa, että käyttöhetkellä luottokortti on oikeasti käyttäjän hallussa.

Luottokortin sarjanumero ja viimeinen voimassaoloaika ovat kohokuvioina ja kulkevat sähköisesti ja tallentuvat eri yhteyksissä, mutta CVV-luku ei tallennu mihinkään. Tämän luvun on luottoyhtiö itse koodannut ja kirjoittanut sen ainoastaan luottokorttiin, mutta ei esimerkiksi magneettijuovalle. Luvun konstruoinnissa käytetään sekä TripleDES että ns. kolmannen polven Crypto Express2 salakirjoitusmenetelmiä ja lisäksi kortin omaa sarjanumeroa ja voimassaoloaikaa sekä myös luottoyhtiön omia salausavaimia. Tiettävästi tätä kokonaisuutta ei ole voitu murtaa.

CVV-luku on luottokortin vahvistusnumero ja turvatoimi, jolla kortin myöntäjä voi vahvistaa haltijan henkilöllisyyden ja kortin hallinnan luottokorttimaksun yhteydessä. Luvun avulla tarkistetaan, että luottokortin tilinumero ei ole varastettu. Lukua käytetään varmistamaan, että kortti on läsnä ostostilanteissa eli tavoitteena on suojautua luottokorttipetoksilta. CVV-luku suojaa niitä vastaan, jotka ovat saaneet haltuun luottokortin numeron, mutta eivät ole saaneet haltuun itse korttia. Lisäehto käytölle on aina, että yhteys on SSL-salattu.

Luottokorttien numeroita ei arvota satunnaisesti, vaan kaikki luvut ovat seurausta matemaattisista säännöistä, joten satunnaiset luvut, joilla ei ole määrättyä rakennetta, järjestelmän on helppo havaita. Käytön aikana luottokortin laskutusosoite varmistetaan ja CVV-luku varmistaa, että ainoastaan kortinhaltija voi käyttää korttia ja näin petoksellinen toiminta tulee mahdottomaksi.

Internetistä löydettävillä sivustoilla ja myös verkosta ladattavilla ohjelmilla voidaan luoda kortin ulkoisista tiedoista näennäisesti kelvollinen CVV-luku. Kuitenkaan tällaiset luvut eivät läpäise validiteettitestejä, kun yrittää käyttää lukuja vaikkapa verkkokaupoissa. Tämä osoittaa, että pankkien- ja luottoyhtiöiden salaustaso on korkea. Sinänsä esimerkiksi **EFT Calculator** on mainio ohjelma, jolla voi perehtyä mm. pankkimaailman ihmeellisiin lukuihin.

Tietoteknisessä yhteiskunnassa tiedonsiirto ja tietoturvallisuus ovat yhä korostuneemmassa asemassa. Esimerkiksi PIN-koodi on SIM-korttiin liittyvä turvakoodi, jonka tarkoituksena on estää puhelimen väärinkäyttö. Monissa Internet-sovelluksissa huolia yksityisyyden

menettämisestä käsitellään käyttämällä salattuja yhteyksiä asiakkaiden ja palvelinten välillä *https*-protokollan avulla. Tämä rajoittaa pelkän verkkopohjaisen valvonnan tehtävää: valvontapiste pystyy verkon keskellä havaitsemaan vain pakettien lähde- ja kohdeosoitteen. Näin esimerkiksi NSA:n (National Security Agency eli Yhdysvaltain kansallinen turvallisuusvirasto) ja FRA:n (Förvarets radioanstalt eli Ruotsin puolustusvoimien radiotiedustelu) kaltaiset virastot lähettävät palveluntarjoajille määräyksiä vapauttaa sisältö kohdasta, johon *https*-protokolla päättyy eli käytännössä palveluntarjoajan pilvipalvelusta [Kan].

Siirrettäessä tietoa paikasta toiseen ja sen kulkiessa jonkin väliaineen kautta vastaanottajalle toiveena on aina, että lopputulos on virheetön. Olipa kyse pankkitoiminnasta, satelliitteja ohjaavista radiosanomista tai vaikkapa yksinkertaisesti sähköpostitunnisteista ja niiden salasanoista, niin tavoitteena on, että muuttunut ja virheellinen informaatio havaitaan ja vielä mieluummin sekä havaitaan että korjataan automaattisesti.

Koska matematiikan näkökulmasta kirjoitettu teksti tai vaikkapa kännykkään puhuttu puhe aina digitalisoidaan, niin väistämättä on toteutukset osattava hoitaa myös binäärijärjestelmässä. Tämä puolestaan johtaa siihen, että koulumatematiikan kymmenjärjestelmässä koodauksen ja dekoodauksen hyvä havainnollistaminen on ongelmallista, sillä pääsääntöisesti sovellukset edellyttävät sekä polynomialgebran että matriisilaskennan ymmärtämistä ja osaamista. Nykyisin käytetyt kryptausmenetelmät perustuvat lukuteorian, lukugeometrian, algebran, kommutatiivisen algebran, ryhmäteorian ja myös äärellisten kuntien tuloksiin [Ruo].

Salaaminen

Koodaus

Viestin salausta kutsutaan kryptaukseksi ja tavoitteena on, että satunnainen lukija tai salakuuntelija ei saa selville mitään viestin sisällöstä eli hän ei pysty murtamaan kryptausta.

Matematiikan ja kryptologian yhteys on historiallisesti pitkä. Saksalainen Arthur Schebius ideoi koneen, joka sekä salasi että purki tekstejä automaattisesti. Scherbiuksen sähkömekaaninen kiekkokone tarjottiin ensimmäisen kerran myyntiin Enigma-nimisenä jo vuonna 1923. Saksan armeija alkoi käyttää Enigmaa kaikkiin salaisiin yhteydenottoihin, jotka saattoivat olla sieppaukselle alttiita. Viestimuseosta Riihimäeltä löytyy jopa yksi aito sodan ajan Enigma. Puolestaan **Enigma Simulator v7.0** on verkosta ladattava ohjelma, jolla voi tutkia tämän mekaanisen laitteen salausta ja purkamista. On arvioitu, että toisen maailmansodan kulkuun ehkä eniten vaikuttanut yksittäinen laite on juuri Enigma. Äärimmäisen vaikean Enigman salaaman sanoman matemaattisen murtamisen keskeisimpinä tutkijoina olivat puolalainen Marian Rejewski ja englantilainen Alan Turing [Kee].

1970-luvulla otettiin käyttöön avoin, nopea ja tehokas ensisijaisesti tietokoneille suunniteltu julkisen avaimen idea. Eräissä lukiomatematiikan kirjoissakin esitellään alkeellinen RSA-salakirjoitus (Rivest-Shamir-Adleman public-key cryptosystem), joka on julkinen salakirjoitusmenetelmä. Tämä tarkoittaa, että järjestelmään kuuluu kaksi eri avainta: julkinen avain ja salainen avain. Julkinen avain voidaan antaa muille ja julkisesti, jotta he voivat lähettää salattuja viestejä. Tämän tiedoston ei siis tarvitse olla piilossa. Salainen avain on periaatteessa salainen tiedosto, jolla voidaan avata muiden lähettämät salatut sähköpostiviestit ja sitä ei pidä koskaan antaa kenellekään ulkopuoliselle. Verkossa julkaistaan RSA-menetelmällä tuotettu ja salaukseen tarvittava luku ja kuka tahansa voi kirjoittaa RSA-menetelmällä tekstin, mutta ainoastaan avainparin haltija ymmärtää kirjoituksen. Salaisen avaimen laskeminen julkisesta koodausavaimesta tulee olla äärimmäisen vaativaa eli julkinen osa avainta voidaan hyvin esitellä verkossa tai käyntikortissa. Menetelmä on nykyisin laajassa käytössä ja se perustuu ensisijaisesti alkulukujen käyttöön ja kongruenssilaskentaan [Sal].

Lisää tehoa salaukseen syntyi, kun käyttöön otettiin täysin avoin, erittäin nopea ja tehokas tietokoneille tarkoitettu kryptausmenetelmä DES (Data Encryption Standard). Tämä salausmenetelmä valittiin

1976 Yhdysvaltain liittovaltion standardiksi ja myöhemmin se on levinnyt maailmanlaajuiseksi menetelmäksi. DES on luonteeltaan ns. symmetrinen lohkosalain, jossa lohkon pituus on 64 bittiä ja itse avaimen tehollinen pituus 56 bittiä. Avain on yhteinen sekä kryptauksessa että dekryptauksessa ja avaimia on yhteensä

$$2^{56} = 72057594037927936$$

erilaista eli nykyajan tehoilla mitattuna aivan liian vähän. Läpikäymällä koko avainavaruus ns. *Brute-force*-menetelmällä salaus on kyetty murtamaan alle vuorokaudessa.

Erilaisiin käyttäjätunnuksiin liittyvien salasanojen muodon tärkeydestä saa hyvän kuvan, kun esittää taulukkomuodossa salasanojen murtonopeuksia. Seuraavassa taulukossa havainnollistetaan tavallisen pöytäkoneen (Pentium III 933 MHz) tehoa murtaa salasanoja ja toisaalta salasanan muodostamiseen käytettyjen perusaakkosten, isojen kirjainten, numeroiden ja erikoismerkkien käytön tärkeyttä [Jär].

Salasana	Merkistö	Kulunut aika
muikku	pienet	1 min 40 s
Muikku	pienet, isot	8 h 58 min
Mui789	pienet, isot, numerot	1 vrk 13 h 43 min
Mui-89	pienet, isot, numerot, erikoismerkit	19 vrk 13 h 27 min

Salasanoja murtavia ohjelmia on saatavilla Internetistä, mutta tällaisten ohjelmien käyttö työpaikoilla ja kouluissa on ehdottomasti kiellettyä ja jopa rikollista. Kotikoneellaan kukin voi harrastusmielessä tehdä kokeiluja suljetussa ympäristössä ja perehtyä näin tietoturvaan. Esimerkiksi palkittu salasanojen tutkimusohjelma **LC4 4.00** tarjoaa hyvän mahdollisuuden jo *trial*-versiolla perehtyä asiaan. On myös huomattava, että mm. järjestelmänvalvojan, verkon ylläpitäjän ja pääkäyttäjän on osattava salasanojen murtamista ennaltaehkäisevänä toimintana ja tällöin ei ole kyse hakkeroinnista, vaan kyvystä estää hakkerointi.

Kryptologian ja sen tarvitseman matematiikan kehitys on ollut tavattoman nopeaa ja edellä kuvatut klassiset menetelmät ovat saaneet rinnalleen moderneja julkisen avaimen menetelmiä. Esimerkiksi NTRU perustuu lukuhilan pienimmän vektorin etsimiseen [Hof] ja Menezes-Vanstone puolestaan logaritmin laskemiseen elliptisen käyrän määrittämässä syklisessä ryhmässä [Men-1]. Näin osa klassisista menetelmistä on menettänyt paljon merkitystään. Voidaan vielä mainita, että jopa Yhdysvaltain maavoimien kenttämanuaali Basic Cryptanalysis: FM 34-40-2 on nykyisin saatavilla verkosta [Bas]. Uudet salausmenetelmät ovat äärimmäisen korkean tason sovelluksia ja kryptauksen murtaminen edellyttää pääsääntöisesti merkittäviä teoreettisia läpimurtoja myös itse matematiikassa.

Matemaattinen vaatimus salaukselle ja näin myös tietoturvalle on pääsääntöisesti se, että kryptausfunktion tulee olla injektiivinen eli se ei kryptaa kahta selväkielistä tekstiä samaksi kryptotekstiksi. Toisaalta kryptauksessa tulee olla satunnaisuutta, jolloin kryptausfunktio voi eri kerroilla kryptata saman tekstin eri kryptoteksteiksi ja näin kyseessä ei varsinaisesti ole koulutason matemaattinen funktio vaan ns. moniarvoinen satunnaisfunktio.

Esimerkki koodauksesta

Selväkielisen tekstin koodaaminen digitaaliseen muotoon tarkoittaa yksinkertaistettuna sitä, että teksti kulkee läpi ketjun

Teksti…ASCII…Binääri…Alkulukusalaus…Salattu sanoma.

Valitaan koodattavaksi sanaksi **Golf,** joka jo sinällään on monimerkityksinen. Käytetään ASCII-taulukon 7-bittistä perusmerkistöä ja lukuväliä 33-126, joka kirjoitettuina merkkeinä on sekä aakkoston että numerot sisältävä väli: 33 = ! (=huutomerkki) . . . 126 = ~ (=tilde).

Merkki	Dec	Hex	Binääri	**8-bittinen binääri**
G	71	47	1000111	01000111
o	111	6F	1101111	01101111
l	108	6C	1101100	01101100
f	102	66	1100110	01100110

Ennen sanoman lähettämistä teksti salataan ja nyt käytetään salausavaimena alkulukua $131_{10} = 10000011_2$. Kerrotaan salausavaimella kunkin kirjaimen binäärikoodi.

Tulo	Binääritulo	Dec-arvo	16-bittinen koodi
131x 71	10000011 x 01000111	9301	0010010001010101
131x111	10000011 x 01101111	14541	0011100011001101
131x108	10000011 x 01101100	14148	0011011101000100
131x102	10000011 x 01100110	13362	0011010000110010

Nyt käytössä on salattu sanoma binäärimuodossa, kun yhdistetään nämä neljä 16-bittistä sanaa yhdeksi viestintäkanavaan lähetettäväksi digitaaliseksi koodiksi. Koodi on seuraavassa ryhmitelty kahdeksan bitin ryhmiin, jolloin saadaan aikaan merkkijono

00100100 01010101 00111000 11001101 00110111 01000100 00110100 00110010.

Jos merkkijonoa yrittää tulkita ASCII-koodiston avulla, niin tulos on enemmän kuin outo

36 = $, 85 = U, 56 = 8, 205 = Ĺ, 55 = 7, 68 = D, 52 = 4, 50 = 2

eli sanaa $U8Ĺ7D42 on vaikea lukea, tulkita ja ymmärtää sanaksi Golf.

Dekoodaus

Dekoodauksessa viestin vastaanottaja purkaa eli dekryptaa viestin ja tätä varten hän tarvitsee avaimen. On helppo ymmärtää, että salakuuntelijalle sekä kryptausavain että dekryptausavain ovat arvokkaita. Näitä käyttäen salakuuntelija kykenee sekä lähettämään valeviestejä että purkamaan aitoja viestejä. Edellä mainittu RSA-menetelmä on ns. epäsymmetrinen kryptausmenetelmä eli että mainitut avaimet eivät ole samat ja että millään suhteellisen pienellä määrällä työtä ei kryptausavaimesta voi saada aikaiseksi dekryptausavainta. Toisaalta viestin sisällön vaatima turvallisuuden tason nosto on myös synnyttänyt uusia ja tehokkaita tapoja koodata digitaalista materiaalia [Kiv].

Esimerkki dekoodauksesta

Salatun sanoman purkaminen eli dekoodaus digitaalisesta muodosta takaisin selväkieliseksi tarkoittaa yksinkertaistettuna sitä, että binäärikoodi kulkee läpi ketjun

Binäärikoodi...Ryhmittely...Avaimen käyttö...Binääri...ASCII...Teksti.

Olkoon nyt vastaanotettu salattu sanoma eli binäärikoodi

00100011 11010010 00110101 10111011 00111000 01001010
00111000 01001010 00110001 10100011 00110101 10111011
00111010 01010110.

Oletetaan, että salaus on ollut edellisen esimerkin kaltainen ja symmetrinen eli nyt dekoodausavainkin on $131_{10} = 10000011_2$. Jaetaan avaimella kukin vastaanotettu 16-bittinen merkkijono.

16-bittinen koodi	Binääriosamäärä	**8-bittinen binääri**
0010001111010010	0010001111010010 / 10000011	01000110
0011010110111011	0011010110111011 / 10000011	01101001
0011100001001010	0011100001001010 / 10000011	01101110
0011100001001010	0011100001001010 / 10000011	01101110
0011000110100011	0011000110100011 / 10000011	01100001
0011010110111011	0011010110111011 / 10000011	01101001
0011101001010110	0011101001010110 / 10000011	01110010

Puretaan 8-bittinen muoto tavanomaisen ASCII-taulukon 7-bittiseen muotoon ja tehdään saaduille luvuille merkkimuunnos.

8-bittinen binääri	Binääri	Hex	Dec	Merkki
01000110	1000110	46	70	F
01101001	1101001	69	105	i
01101110	1101110	6E	110	n
01101110	1101110	6E	110	n
01100001	1100001	61	97	a
01101001	1101001	69	105	i
01110010	1110010	72	114	r

Näin vastaanotetun salatun sanoman dekoodaus tuotti tulokseksi selväkielisenä luettavan merkkijonon **Finnair**.

Tietoturva ja yksityisyyden suoja

Meidän jokaisen yksityisyydellä on perustuslain turvaama asema ja suoja. Erityisesti yksityiselämä ja kotirauha kuuluvat yksityisyyden suojan piiriin. Myös henkilötiedot on suojattu ja erityisesti kirjeen, puhelun ja muun luottamuksellisen viestin salaisuus on loukkaamaton.

Sähköpostin ja sen liitteiden salaaminen voidaan tänä päivänä tehdä parhaiten **PGP** (Pretty Good Privacy, ”melko hyvä yksityisyys”) järjestelmällä. PGP perustuu julkisen avaimen salaukseen ja on sovellus RSA-menetelmästä, jossa salaus tapahtuu vastaanottajan julkisella

avaimella ja salatun sanoman avaaminen julkista avainta vastaavalla salaisella avaimella. Näin salauksessa voidaan luetella henkilöt, jotka voivat avata viestin.

PGP-ohjelmasta on olemassa sekä kaupallinen että ilmainen versio. Modernein ilmainen versio GnuPG (GNU Privacy Guard, lyhyesti GPG) on vapaa ohjelma ja se on OpenPGP-määritysten mukainen ja sen kehittämistä mm. Saksan valtio on tukenut. Tällä hetkellä Windowsille on saatavana graafisella käyttöliittymällä varustettu Symantecin ylläpitämä PGP Desktop, jolloin työskentely onnistuu hiirellä.

Toukokuussa 2013 NSA:n sopimustyöntekijä Edward Snowden toi julkisuuteen valtaisan joukon salaisia asiakirjoja. Toimittaja Glenn Greenwald ja dokumentaristi Laura Poitras käsittelivät ensimmäisinä asiakirjoja ja kesäkuun 7. päivänä The Guardian julkaisi artikkelin PRISM-ohjelmistosta. Tämän jälkeen The Guardian ja The Washington Post olivat pääosassa salaisten asiakirjojen julkaisemisessa. Snowden itse siirtyi julkisuuteen tulonsa jälkeen Hong Kongista Moskovaan.

PRISM on ehkä kuuluisin NSA:n valmistama tietokoneohjelmisto. Sen tarkoitus on paljastaa terroristeja ja se on erityisesti vakoilua varten suunniteltu. Ohjelman oikea nimi on US-984XN ja PRISM on sen koodinimi. NSA:lla on PRISM:n ansiosta käytössä tiedonkeruu suoraan seuraavien yhdysvaltalaisten palveluntarjoajien palvelimilta: Microsoft (HotMail, OneDrive, Outlook, jne.), Google (Google+, Docs, Drive, Calender, Gmail, jne.), Yahoo, Facebook, PalTalk, YouTube, Skype, AOL ja Apple.

Muita vastaavia ohjelmia ovat esimerkiksi Britannian GCHQ:n (Government Communication Headquarters) kanssa toteutettu yhteishanke Project Bullrun, jonka päämääränä on murtaa Internetin kaikki erilaiset salausjärjestelmät. Lisäksi asiakirjoista paljastui lukematon määrä muita erityisohjelmia mm. Vagrant, Magnetic, Mineralize, Lifeserver, Dropmire, Crypto Enable ja Dewsweeper. Edelleen esimerkiksi Microsoft on tehnyt FBI:n kanssa yhteistyötä ja laatinut jo vuonna 2012 järjestelmän, jolla voi murtaa varsin laajalti käytetyn SSL-salauksen [Gre].

Millaista tietoa ja yksityisyyden suojaan kuuluvaa materiaalia on haluttu saada luettavaksi? Tavoitteena on ollut selvittää mm. sähköpostitunnuksia, -salasanoja, -viestejä ja liitetiedostoja, piilokopio-osoitteita, matkapuhelinnumeroita, fax-numeroita, osoitekirjoja, Chat-tietoja, IP-paikannustietoja, pankkiasiointeja, lääketieteellisiä asiakirjoja, tehdyt lentomatkat sekä valvoa Internet-palvelimia, nettipalveluohjelmia, satelliitteja, kuituoptiikkakaapeleita, vedenalaisia kaapeleita, puhelinjärjestelmiä sekä yksittäisten ihmisten henkilökohtaisia tietokoneita. Edelleen esimerkiksi Thieving Magpie ja Homing Pigeon ohjelmilla kyetään kuuntelemaan ja valvomaan jopa kaupallisten reittilentojen aikana tapahtuvaa Internet- ja puhelinviestintää. Lisäksi tavoitteena on ollut hankkia kyky levittää haittaohjelmia, joilla voi tarkkailla näppäinpainalluksia ja ruutukuvia.

Yksityisyyden suojassa ei ole ainoastaan kyse puheluiden, sähköpostien ja Internet-chattien sisältöjen lukemisesta. Lukutaito Internetin avulla tapahtuvassa valvonnassa on viety uudelle tasolle. Tiedon keräys, verkon selaus- ja hakuhistorian selvittäminen tarkoittaa **metadatan** konstruointia. Tämä tarkoittaa sisältöä koskevaa tietoa, mutta ei itse sisältöä. Näin esimerkiksi voidaan rohkeasti uutisoida, että ”Yksittäisen ihmisen viestejä ei aiota valvoa”, kun Suojelupoliisi ja Puolustusvoimat tahtovat tiukentaa verkkovalvontaa [AL].

Toisaalta metadatan avulla voidaan lukea kuka lähetti sähköpostin ja kenelle, mikä oli viestin aihe, missä viestin lähettäjä oli, ketkä saavat viestistä piilokopion, mikä on viestien aikataulutus ja frekvenssi. Vastaavasti matkapuhelinliikenteestä voidaan rakentaa yksilön sosiaalinen verkko ja hänen käyttäytyminen. Pelkästään metadataa lukemalla voidaan konstruoida osin mielipiteet ja jopa osa ajattelusta ja asenteista. Esimerkiksi Facebook on manipuloinut salaa satojen tuhansien ihmisten uutisvirtaa tutkiakseen, voiko nähty sisältö muokata jopa käyttäjien tunnetiloja [AL]. Metadatan lukutaito on kasvanut Internetin myötä, mutta siitä on samalla tullut merkittävä uhka yksityisyyden suojalle. Tosin NSA on Snowdenin tietovuodon myötä korostanut, että metadatan lukeminen ei ole yhtä tunkeilevaa kuin on itse sisältöjen lukeminen.

Tosiasiallisesti metadataa lukemalla kyetään synnyttämään huomattavan selkeä kuva ihmisen elämästä, kumppaneista ja kaikista toimista ja samalla saadaan haltuun mitä intiimeintä ja yksityisintä tietoa. Metadatan lukeminen antaa mahdollisuuden selvittää mitä kukin tekee, sanoo, ajattelee, suunnittelee ja mitä ostoksia tekee ja mitä mieltymyksiä omaa. Lukuprosessi on näin ymmärryksen konstruointia halutusta kohteesta, se on kohteen tiedustelua ja analysointia, vaikka esim. XKeyscore mahdollistaa jopa henkilön verkon käytön reaaliaikaisen tarkkailun.

Saksan Griesheimissa Darmstadin naapurissa sijaitsee yksi NSA:n Euroopan suurimmista tiedonkeräyskeskuksista European Cryptologic Center - ECC. Vastaavia pienempiä keskuksia on mm. Frankfurt am Mainissa, Bad Ailingissa, Wiesbadenissa, Stuttgartin Vaihingenissa sekä luonnollisesti Berliinissä. Tosin tämä kaikki on osin seurausta myös II:n maailmansodan jälkeisistä tapahtumista [DS].

Eräs keskeisimmistä ohjelmista on ollut em. XKeyscore, joka on mahdollistanut tiedon sisällön lukemisen ja tutkimisen kolmesta päivästä jopa muutamaan viikkoon ja ns. metadatan konstruoinnin ja tutkimisen aina 90 vuorokauteen asti. Kaikkiaan tällainen NSA:n toiminta, salainen ja moniulotteinen tietojen kerääminen sekä lukeminen ja analysointi ja jopa liittokansleri Merkelin puhelimen salakuuntelu rikkovat Yhdysvaltain Saksalle antamaa lupausta, jonka mukaan saksalaisia ei tarkkailla ilman, että näitä epäiltäisiin mistään [DS].

Tämän lisäksi XKeyscore paljastaa, että jo pelkkä Internet-käytön anonymisoivan Tor-verkon käyttö tai pelkkä yksityisyyttä suojaavien ohjelmien etsiminen hakuroboteilla Internetistä vievät käyttäjän IP-osoitteen eli henkilökohtaisen verkko-osoitteen NSA:n tietokantoihin. Tor-verkko on muokatulla selainohjelmistolla toteutettu salausjärjestelmä, jossa nettikäyttäjien liikenne reititetään monien palvelimien kautta ennen kuin nettiliikenne tulee luettavassa muodossa ulos ”tavalliseen internetiin”. Tämä tekee käyttäjästä tuntemattoman tai ainakin erittäin hankalan jäljittää. Alun perin Tor-verkon katsottiin lisäävän sananvapautta mm. diktatuureissa. Lisäksi Toria käyttävät työkaluna

muun muassa Yhdysvaltain huumepoliisi sekä tietoturvayhtiöt, jotka välttelevät paljastumista työssään [IS].

Ei siis ole ihme, että Edward Snowden on sanonut: ”En halua elää maailmassa, jossa meillä ei ole yksityisyyttä eikä vapautta ja jossa Internetin ainutlaatuinen arvo tuhotaan.” Snowdenin kaltaiset tietovuotajat luokitellaan usein heppoisin perustein surkimuksiksi, vieraantuneiksi ja epäonnistuneeseen elämään turhautuneiksi [Gre]. Ehkä historiallisesti merkittävä NSA-tietovuoto ansaitsee hieman syvällisemmän pohdinnan erityisesti siitä, mitä on metadatan lukutaito.

Yksityisyys on vapaan elämän välttämätön edellytys! On siis huomattava, että yhteiskunnan vapautta mitataan sillä, miten toisinajattelijoita ja muita marginaaliryhmiä kohdellaan eikä sillä miten se kohtelee arvoilleen uskollisia. Tällaista aikaa odotellessa voisi toivoa, että teknologiayhteisö kehittäisi yhä tehokkaampia ja käyttäjäystävällisempiä nimettömyys- ja salausohjelmia.

Virheet luvuissa

Virheen tunnistus

On olemassa erilaisia menetelmiä tutkia vastaanotettujen lukujen ja merkkijonojen oikeellisuutta, mutta itse sisällön ymmärtämisen kanssa pelkällä virheen tunnistuksella ei ole tekemistä. Virheen tunnistus tarkoittaa, että itse lukuun tai merkkijonoon sisällytetään alkeellista tarkistamista varten matemaattinen informaatio, jota hyödynnetään laskennallisesti. Näin ennen luvun sisällön käyttöä ja ymmärtämistä voidaan varmentaa, että onko itse luku ylipäätään edes rakenteeltaan oikein. Luvusta tuleekin tehdä taitava!

Tavanomaisin ja samalla alkeellisin tekniikka merkkijonon oikeellisuuden tarkistamiseksi on binäärimatematiikasta tuttu ns. pariteetintarkistus. Hieman hienostuneempi tapa on käyttää hyväksi ns. jäännösluokkia **modulo** jokin alkuluku. Jäännösluokka-algebraan tutustutaan lukion pitkässä matematiikassa.

Kuinka taitava luku on IBAN-pankkitilinumero?

IBAN-pankkitilinumeron tarkistuksessa käytetään kongruenssilaskennan menetelmää eli laskenta suoritetaan kokonaislukujen jakojäännöksillä käyttäen joko yhtä tai useampaa kiinteää jakajaa eli ns. modulia. Kokonaisluvut jakautuvat *m* jäännösluokkaan sen mukaan miten ne ovat kongruentteja *i*:n kanssa modulo *m* (i=0,...,m-1).

Esimerkiksi koulumatematiikan kerto- ja jakolaskut suoritetaan tietokoneissa kongruenssilaskennan ja ns. Karatsuban algoritmin lisäavulla hyvin nopeasti.

Suomessa kansainvälinen IBAN-pankkitilinumero (International Bank Account Number) otettiin käyttöön kansainvälisessä maksuliikenteessä jo vuonna 2001. Vuonna 2007 IBAN tuli pakolliseksi EU:n kansainvälisessä maksuliikenteessä ja 2008 se otettiin käyttöön Suomessa myös sisäisessä maksuliikenteessä. IBAN-tilinumero on pakollinen mm. tilisiirtolomakkeissa, e-laskuissa ja palkanmaksussa. Se on pakollinen myös yksityisille tilinumeroille, mutta pääsääntöisesti verkkopankit tekevät vanhalle tilinumerolle muunnoksen automaattisesti.

Vanhan tilinumeron muuttaminen IBAN-tilinumeroksi

Vanha pankin tilinumero (esim. 573008-331741) muokataan 14-numeroiseen muotoon. Väliviivan kohdalle lisätään nollia niin monta, että luvusta tulee 14-numeroinen. Niiden pankkien tilinumeroihin, jotka alkavat joko luvulla viisi tai neljä, nollien lisäys tehdään väliviivan jälkeisen numeron jälkeen. Näin saadaan esimerkiksi tulos 57300830031741.

Tämän jälkeen tilinumeron loppuun lisätään maatunnus (Suomi = FI) ja kaksi nollaa. Näin esimerkin tilinumero on muotoa 57300830031741FI00.

Numerosarjaan lisättävän maatunnuksen mukaiset kirjaimet vaihdetaan kansainvälisesti sovitun taulukon mukaan numeerisiksi.

A = 10	B = 11	C = 12	D = 13	E = 14
F = 15	G = 16	H = 17	I = 18	J = 19
K = 20	L = 21	M = 22	N = 23	O = 24
P = 25	Q = 26	R = 27	S = 28	T = 29
U = 30	V = 31	W = 32	X = 33	Y = 34
Z = 35				

Maatunnuksen kirjainten vaihto numeerisiksi

Suomen tunnus FI vaihdetaan luvuksi 1518. Näin saadaan 20-numeroinen pankkitilinnumero 57300830031741151800, jolle muodostetaan oma tarkiste käyttäen kongruenssilaskentaa.

Luvun 57300830031741151800 **jakojäännös modulo 97** on 37 ts.

$$57300830031741151800 \equiv 37 \pmod{97}.$$

Jakojäännöksen laskenta yksinkertaisella funktiolaskimella on hieman työläs, mutta lukiolaisen CAS-laskimen *mod*-komennolla tulos syntyy helposti.

Jakojäännöstä käyttäen lasketaan **tarkiste** 98 – 37 = **61**. Koska tarkistuslaskennassa jakolaskun mennessä tasan tavoitteena on saada jakojäännökseksi nollan sijaan kuitenkin yksi, niin tarkisteen mahdolliset arvot ovat 2, 3, ..., 98 ja lisäämällä etunollat saadaan lopulta kaksinumeroiset tarkisteet 02, 03, ..., 98.

Lopuksi maatunnus ja edellä laskettu tarkiste lisätään 14-numeroisen tilinumeron alkuun. IBAN-tilinumero esitetään tavallisesti ryhmiteltynä ja esimerkin tapauksessa saadaan lopulta tilinumero

FI**61** 5730 0830 0317 41.

IBAN-tilinumero on taitavasti ja matematiikkaa hyödyntäen konstruoitu. Tilinumeron sisällä on informaatio, jota hyödyntäen kyetään tarkistamaan itse tilinumeron oikeellisuus. Vastaava kongruenssiperusteinen tarkistusmenetelmä on sosiaaliturvatunnuksessa, jossa tarkiste-

merkki saadaan jakamalla syntymäajan ja yksilönumeron muodostama yhdeksännumeroinen luku luvulla 31.

Koska kaikista pankkitilinumeroista on jo olemassa IBAN-muoto, niin tärkeämpää on ymmärtää ja osata tarkistaa onko annettu tilinumero oikein.

IBAN-tilinumeron tarkistaminen

Siirretään maatunnus ja tarkiste tilinumeron loppuun, jolloin tilinumerosta FI61 5730 0830 0317 41 saadaan tilinumero 57300830031741FI61 ja josta korvauksen jälkeen edelleen tilinumero 573008300317411518 61.

Jaetaan luku 573008300317411518 61 luvulla 97 ja **jos jakojäännös on yksi** ts. jos

$$573008300317411518 61 \equiv 1 \pmod{97},$$

niin IBAN-tilinumero on oikein. Tarkista oma pankkitilinumerosi! Tarkista myös, onko saksalainen pankkitilinumero DE83 1670 5000 2612 13 IBAN-menetelmällisesti oikeassa muodossa.

Kuinka taitava luku on SIM-kortin ICCID-tunniste?

Kaikille kännykän käyttäjille tuttu SIM-kortti (Subscriber Indentification Module) on sekä integroitu piirikortti että älykortti. Kortin tarkoituksena on yksilöidä ja tunnistaa tietoliikenneliittymän käyttäjä ja erityisesti erottaa liittymän tilaaja itse fyysisestä puhelimesta tai muusta vastaavasta laitteesta. Juuri SIM-kortin ansiosta esimerkiksi GSM-verkon käyttäjät voivat vaihtaa hyvin helposti puhelinta tai operaattoria.

Koska SIM-kortti on käyttäjä- ja liittymäkohtainen ja koska sille voidaan tallentaa tietoja, niin siltä vaadittava keskeinen ominaisuus on

tietoturva. Yhteydenotto tulee suojata ja lisäksi kortti sisältää operaattorikohtaisia salaisia avaimia, jotka mm. tarkistetaan puhelun yhteydessä.

SIM-kortti sisältää tallennetun datan lisäksi mm. sarjanumeron ICCID (Integrated Circuit Card Identifier), tilaajatunnisteen IMSI (International Mobile Subscriber Identity), henkilökohtaisen salasanan PIN (Personal Identification Number) ja perinteisen käytön lukituskoodin PUK (Personal Unblocking Key). Jokainen SIM-kortti tunnistetaan kansainvälisesti piirikortin ICCID-tunnisteen avulla. Tämä tunniste on sekä tallennettu kortille että painettu tekstimuodossa valmistusprosessin yhteydessä kortin pintaan. ICCID-tunniste on 22 numeroa pitkä luku ja rakenteeltaan standardin E.118 mukainen.

Tarkistenumero lasketaan **Luhnin algoritmia** käyttäen.

Luhnin algoritmi on tarkistussumman laskemiseen tarkoitettu matemaattinen menetelmä, jota käytetään varmistamaan erilaisia tunnistelukuja. Algoritmin loi IBM:n matemaatikko Hans Peter Luhn jo vuonna 1954. Algoritmin tehtävänä on tunnistaa virheet itse numerosarjassa, mutta algoritmia ei ole esimerkiksi suunniteltu suojautumaan tahalliselta hyökkäykseltä. Näin itse viestinnän sisällön suojaus ja salaus on eri asia kuin tämä pelkkä SIM-kortin identifiointi.

Tarkisteen konstruointi

SIM-kortin 21-numeroisen sarjanumeron loppuun määritetään tarkistenumero Luhn algoritmilla seuraavasti:
Sarjanumeron numerot kerrotaan alkaen oikealta vasemmalle vuorotellen kahdella ja vuorotellen yhdellä.

Sarja-numero	2	5	6	7	2	1	1	0	2	3	9	5	2	7	8	9	4	7	0	2	8
Kerroin	×2		×2		×2		×2		×2		×2		×2		×2		×2		×2		x2
Tulo	4	5	12	7	4	1	2	0	4	3	18	5	4	7	16	9	8	7	0	2	16

Saaduista tuloista lasketaan yksittäiset numerot yhteen, jolloin saadaan summa

4+5+1+2+7+4+1+2+0+4+3+1+8+5+4+7+1+6+9+8+7+0+2+1+6 = 98.

Saatu summa kerrotaan luvulla yhdeksän ja lasketaan tulon arvo modulo 10. Näin saadaan tulos

$$9 \cdot 98 = 882 \equiv 2 \pmod{10}$$

ts. tulon arvon 882 viimeinen numero kaksi on SIM-kortin ICCID-tunnisteen tarkiste. Lopulta itse ICCID-tunniste saa ryhmiteltynä muodon

2567 2110 2395 2789 4702 82,

joka valmistusvaiheessa tulostetaan mm. SIM-kortin pintaan ja lisäksi se talletetaan myös integroidulle piirikortille.

ICCID-tunnisteen oikeellisuuden tarkistaminen

Alkaen taas oikealta vasemmalle kerrotaan järjestyksessä parittomien paikkojen luvut yhdellä ja parillisten paikkojen kahdella ja summataan saatujen tulojen numerot yhteen.

Sarjanumero	2	5	6	7	2	1	1	0	2	3	9	5	2	7	8	9	4	7	0	2	8	2
Kerroin	×2		×2		×2		×2		×2		×2		×2		×2		×2		×2		x2	
Tulo	4	5	12	7	4	1	2	0	4	3	18	5	4	7	16	9	8	7	0	2	16	2

Kun saaduista tuloista lasketaan yksittäiset numerot yhteen, niin saadaan summa

4+5+1+2+7+4+1+2+0+4+3+1+8+5+4+7+1+6+9+8+7+0+2+1+6+2 = 100.

Koska saatu summa sata on kymmenellä jaollinen ts. $100 \equiv 0 \pmod{10}$, niin koko ICCID-tunniste on oikein laadittu.

Tietoliikenteessä voitaisiin sanoa, että se on vastaanotettaessa ollut oikein – tai on ainakin todennäköisesti oikein!

On osoitettu, että Luhnin algoritmi havaitsee kaikki yhden numeron virheet ja lähes kaikki vierekkäiset vaihtoparivirheet 09-90, mutta on olemassa tilanteita, joista algoritmi ei selviä. Tällaisia ongelmatapauksia ovat esimerkiksi 22-55, 33-66 ja 44-77. Itse asiassa kesällä 2013 saksalainen mobiilialan turvallisuusasiantuntija Karsten Nohl esitteli SIM-korttien salauksessa havaittuja ongelmia, jotka liittyivät mm. DES-salaukseen [Yle]. Ongelmat olivat hyvin syvällisiä ja havaittujen turvallisuusaukkojen avulla ulkopuolisen tahon on mahdollista saada haltuunsa jopa kortin digitaalinen avain. Näin puhelimen sisältämä data on luettavissa ja sitä voidaan mm. salakuunnella. Ongelman taustat ovat RSA-salauksen kaltaisessa ns. DES-salauksessa. Itse SIM-kortin varsinainen salausavaimen generointi toteutetaan nykyisin ns. A8-algoritmilla ja käyttäjän tunnistamiseen käytetään ns. A3-algoritmia. Isona ongelmana edelleenkin on se, että kiinteässä verkossa ja tukiasemien radiolinkeissä (A5-algoritmi) itse data liikkuu käytännössä salaamattomana. Alkujaan kaikki GSM-tekniikan algoritmit olivat salaisia, mutta algoritmien leviämistä Internet on edesauttanut ja nykyään nämäkin algoritmit ovat saatavilla verkon kautta esimerkkitoteutuksineen [Sda].

Kuinka taitava lukusarja on VISA-kortin numero?

Siinä missä sosiaaliturvatunnuksessa on tarkisteosa, niin aivan vastaavasti myös luottokorttien numerosarjaan on rakennettu sisälle tarkistemerkki – taitava luku, jonka avulla välittömästi saadaan selville onko kyseinen luottokortin numero ylipäätään mahdollinen.

Lähes poikkeuksetta kaikkien luottokorttien tarkiste lasketaan edellä kuvattua Luhnin algoritmia käyttäen. Näin voidaan tarkistaa onko vastaanotettu VISA-kortin numero sisäiseltä rakenteeltaan oikein.

OIKEIN

VISA -tunnus	6	3	9	2	2	5	7	3	3	0	2	1	2	5	0	6
Kerroin	x2		x2		x2		x2		x2		x2		x2		x2	
Tulo	12	3	18	2	4	5	14	3	6	0	4	1	4	5	0	6

Saaduista tuloista lasketaan yksittäiset numerot yhteen ja näin saadaan summa

$$1+2+3+1+8+2+4+5+1+4+3+6+0+4+1+4+5+0+6 = 60,$$

joka on jaollinen kymmenellä.

VÄÄRIN

VISA -tunnus	6	3	9	2	2	5	7	5	3	0	2	1	2	5	0	6
Kerroin	x2		x2		x2		x2		x2		x2		x2		x2	
Tulo	12	3	18	2	4	5	14	5	6	0	4	1	4	5	0	6

Saaduista tuloista lasketaan yksittäiset numerot yhteen ja näin saadaan summa

$$1+2+3+1+8+2+4+5+1+4+5+6+0+4+1+4+5+0+6 = 62,$$

joka ei ole jaollinen kymmenellä. Tällaisessa tapauksessa maksujärjestelmä ei hyväksy esitettyä VISA-kortin numeroa, mutta toisaalta ei myöskään tiedä miltä osin tunnus on virheellinen.

Kuinka taitava luku on maksettavan laskun viitenumero?

Suomessa pankkialan käytänteitä koordinoi Finanssialan Keskusliitto. Kotimaisen laskutusjärjestelmän hallinnoinnin käyttöön se on määritellyt viitenumeron tarkistamiseksi tarkisteen matemaattisen muodon.

Viitenumeron pituus tulee olla vähintään 4-numeroinen (3+tarkiste) ja sallittu enimmäispituus on 19+1 numeroa. Laskuttaja saa varsin vapaasti muodostaa viitenumeron ja tavallisesti siihen sisällytetään tiedot mm. laskun numerosta ja asiakasnumerosta. Tavoitteena on, että maksu välittyy maksunsaajan viitteellisten suoritusten vastaanottamiseen tarkoitetulle tilille.

Viitenumeron tarkisteen muodostaminen

Perusviitetietojen muodostaman luvun numerot kerrotaan oikealta vasemmalle painoarvoilla 7, 3, 1, 7, 3, 1, 7, 3, 1, ... Tämän jälkeen saadut tulot lasketaan yhteen ja summa vähennetään seuraavasta täydestä kymmenestä. Saatu erotus on tarkiste, joka merkitään viitenumeron viimeiseksi numeroksi ja jos erotus on 10, niin tarkiste on 0.

Perusviitetieto	1	9	5	6	0	9	0	9	2	0	1	4
Kerroin	1	3	7	1	3	7	1	3	7	1	3	7
Tulo	1	27	35	6	0	63	0	27	14	0	3	28

Saatujen tulojen summa on 1+27+35+6+0+63+0+27+14+0+3+28=204. Seuraavasta nollaan päättyvästä luvusta vähennetään saatujen tulojen summa eli 210-204=6, jolloin erotuksen arvo kuusi on viitenumeron tarkiste.

Tavallisesti viitenumero tulostetaan tilisiirtolomakkeelle viiden numeron ryhmiin ja etunollia ei tulosteta. Näin esimerkin lopullinen viitenumero olisi

195 60909 20146.

Viitenumeron tarkistaminen

Viitetietojen muodostaman luvun numerot kerrotaan oikealta vasemmalle painoarvoilla 1, 7, 3, 1, 7, 3, 1, 7, 3, 1, ... ja saatujen tulojen summa tulee olla kymmenellä jaollinen. Esimerkiksi viitenumerossa 521 85634 on virhe, sillä tulojen summa

$$1\cdot4 + 7\cdot3 + 3\cdot6 + 1\cdot5 + 7\cdot8 + 3\cdot1 + 1\cdot2 + 7\cdot5 = 144$$

ei ole kymmenellä jaollinen.

Monilla pankeilla on verkkosivuillaan laskureita, joilla voidaan generoida toimivia viitenumeroita tai vaikkapa tarkistaa tilisiirtolomakkeen viitenumeron oikeellisuus [Akt].

Miksi tämän tyyppisissä ongelmanratkaisuissa käytetään alkulukuja 2, 3, 5, 7, 11, 13, 17, jne.? Erityisesti jo viitenumeron alkeellisesta tarkistuslaskennasta huomataan, että yleisessä muodossa syntyy kolmen muuttujan **Diofantoksen kongruenssiyhtälö**

$$\mathbf{7x + 3y + z \equiv 0 \pmod{10}},$$

jolle ratkaisun löytyminen on äärimmäisen epätodennäköistä, kun ottaa vielä huomioon, että luvut x, y ja z, ovat tiettyjen luonnollisten lukujen summia.

Virheen korjaus

Tietokoneiden välinen tiedonsiirto, satelliittien lähettämät tiedot, erilaiset automaattien ohjausjärjestelmät sekä myös tekstiviestit ja puhelut ovat esimerkkejä tilanteista, joissa tieto siirtyy digitaalisessa muodossa jotain kanavaa pitkin. Tullessaan ulos kanavasta kaikki ei ole ehkä toiminut häiriöttä, joten informaatio on muuttunut ja

sisältää virheitä. Virheet voivat esiintyä satunnaisesti läpi koko sanoman tai ne voivat esiintyä ryöppyinä eli purskeina. Samoin kuin normaalissa puheessa väärinkäsityksiä vastaan taistellaan käyttämällä ns. liikainformaatiota. Samoin digitaalisessa tiedonsiirrossa sanomaan konstruoidaan enemmän lukuja kuin olisi tarpeen sanojen ymmärtämiseksi. Lisätyn digitaalisen massan tarkoitus on todeta ja paikallistaa virheitä sekä nykyisin automaattisesti ja matematiikkaa hyödyntäen myös korjata syntyneet virheet.

Vastaanotetun koodin tarkistaminen on mahdollista tehdä niin, että virheet sekä löytyvät että ne voidaan samalla korjata. Erityisesti purskevirheet ovat ongelmallisia. Toisaalta jos purskevirheen pituus on pieni, niin se voi olla yllättäen jopa helpommin korjattavissa kuin yksittäiset virheet.

Hamming-koodi on eräs alkeellisimmista lähtökohdista tutustua sekä virheen paljastavaan että sen myös korjaavaan tapaan koodata informaatio [Ham-1], [Ham-2]. Tavanomaisesti tavoitteena on (p,q) -koodi, jolla voidaan havaita p virhettä ja korjata q:n mittaiset purskevirheet.

BCH-koodit muodostavat tärkeän syklisten koodien luokan ja niiden merkitys on mm. se, että jokaista luonnollista lukua k kohti on olemassa vähintään k virhettä korjaava BCH-koodi.

BCH-koodit keksi vuonna 1959 ranskalainen matemaatikko Hocquenghem [Hoc] sekä hänestä riippumatta 1960 Bose ja Ray-Chaudhuri. Ensimmäisen BCH-koodin dekoodausmenetelmän keksi Peterson vuonna 1960. BCH-koodi toimii parhaimmillaan ympäristössä, jossa virheet esiintyvät koodissa tasaisesti ja satunnaisesti [Lin].

Nykyisin ehkä modernein tapa koodata sanoma on käyttää ns. Menezes-Vanstone-menetelmää. Elliptisiin käyriin pohjautuvana kryptosysteeminä menetelmän erityinen etu on, että tarvittavan avaimen koko on huomattavasti pienempi kuin esimerkiksi RSA-koodauksessa. Julkinen avain on kolmikko $k_1 = (E, \alpha, \beta)$, missä E on elliptinen käyrä yli lukukunnan Zp (missä $p>3$), α on generoiva alkio E:n syklises-

sä ryhmässä ja $\beta = a\alpha$. Salainen avain on $k_2 = a$. Viestilohko on Zp:n alkioiden pari (w1,w2) positiivisessa jäännössysteemissä esitettynä [Men-2], [Ruo].

Lopuksi

Matematiikan opetukseen on hieman hankalaa tuoda uusimpia sovelluksia salauksesta ja erityisesti koodista, joka osaa itse korjata siihen syntyneet virheet. Tällaiset koodaussysteemit edellyttävät usein jo alkeellisella tasolla varsin vaativaa lukuteoreettista matematiikan osaamista. Ne edellyttävät lukutaitoa!

Tämä ei kuitenkaan saa olla este, vaan opetusta voi elävöittää ja matematiikkaan voi motivoida, kun pyrkii tuomaan esille edes alkeellisia sovelluksia. Edellä olevassakin ohitettiin esimerkit modernista salauksesta, koska varsinainen koodaus ja dekoodaus ovat nykyaikana korkean tason matemaattisia tietokoneperusteisia toimintoja. Kuitenkin kaikille tuttujen tietoyhteiskunnan käsitteiden (CVV, IBAN, ICCID, IMEI, ISBN, PIN, PUK, SIM, VISA, jne.) edes osittainen matemaattinen käsittely ja niiden äärellä kaikenlainen puuhailu – ainakin oman kokemukseni mukaan – luovat innostuneisuuden ilmapiirin lukion matematiikan oppitunneille.

Kirjallisuus

Aamulehti 30.6.2014 ja 3.7.2014

Aktiapankin tilisiirtolomakkeen viitenumerolaskuri. http://www.aktia.fi/fi/viitenumerolaskuri

Basic Cryptanalysis: FM 34-40-2 United States Army Intelligence School http://www.umich.edu/~umich/fm-34-40-2/

Der Spiegel Nr. 25/16.6.2014 Mein Nachbar NSA: NSA-Dependancen in der Bundesrepublik

Greenwald, G. No Place to Hide: Edward Snowden, the NSA and the U.S. Surveillance State. Metropolitan Books. New York, 2014.

Hamming, R. W. Coding and Information Theory Prentice Hall. New Jersey, 1986.
Hamming, R. W. The Art of Doing Science and Engineering Taylor & Francise e-Library. Amsterdam, 2005.
Hocquenghem, A. Codes correcteurs d'erreurs Chiffers 2 (147-156). Paris, 1959.
Hoffstein-Pipher-Silverman NTRU: A ring-based public key cryptosystem Springer-Verlag. Oregon, 1998.
Ilta-Sanomat 4.7.2014
Järvinen, P. Tietoturva & yksityisyys Docendo. Jyväskylä, 2002.
Kantola, R. Internet-politiikka ja politiikka Internetissä CSC-Tieteen tietotekniikka 3/2013 (19-20). Espoo, 2013.
Keegan, J. Intelligence in War – Knowledge of the Enemy from Napoleon to Al-Qaeda Pimlico. London, 2004.
Kiviharju, M. Content-based information security (CBIS): Definitions, requirements and cryptographic architecture Defence Forces Technical Research Center, Publications 21Riihimäki, 2010.
Lin, S., Costello, D. Error Control Coding: Fundamentals and Applications Prentice-Hall. New Jersey, 2004.
Menezes, A., Vanstone, S. A. Elliptic curve cryptosystems and their implementation Journal of Cryptology, 6 (209-224), 1993.
Menezes, A., van Oorschot, P., Vanstone, S. A. Handbook of Applied Cryptography CRC Press. 1996.
Ruohonen, K. Matemaattinen kryptologia TTY:n opintomoniste No. 3, uusi sarja, 2012.
Salomaa, A. Public-Key Cryptography Springer-Verlag. Berlin, Heidelberg, 1996.
Smartcard Developer Association. http://www.scard.org/
Yle-Uutiset 22.7.2013. http://yle.fi/uutiset/tutkimus_paljastaa_sim-korttien_vanhentunut_turvakoodi_mahdollistaa_tietokaappaukset/6742382