TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Toni Kauppinen

Opinnäytetyö

OpenPGP-standardia käyttävän salausrajapinnan suunnittelu

Työn ohjaaja Lehtori Tony Torp
Työn teettäjä Kilosoft Oy, valvojana diplomi-insinööri Kimmo Hakkarainen
Tampere 1/2009

TAMPEREEN AMMATTIKORKEAKOULU
Tietotekniikka
Ohjelmistotekniikka

Tekijä(t)	Toni Kauppinen
Työn nimi	Salausrajapinnan suunnittelu käyttäen OpenPGP-standardia
Sivumäärä	34 sivua
Valmistumisaika	1/2009
Työn ohjaaja	Lehtori Tony Torp
Työn tilaaja	Kilosoft Oy, valvojana diplomi-insinööri Kimmo Hakkarainen

TIIVISTELMÄ

Työn tarkoituksena on suunnitella sähköpostin salaava ja purkava rajapinta. Rajapintaa suunniteltaessa apuna tulee käyttää Open C -ohjelmistorajapintaa, joka toteuttaa OpenPGP-salausstandardin. Tällaisella rajapinnalla pystyttäisiin purkamaan GnuPG-sovelluksella salatut viestit sekä salaamaan viestit, jotka myöhemmin voitaisiin purkaa GnuPG-sovelluksella. Lisäksi tehtävänä on tutustua OpenPGP-salausstandardiin ja sitä käyttäviin PGP- ja GnuPG-salaussovelluksiin.

Työn alkuosassa esitellään salaamisen taustoja ja yleisimpiä salaustekniikoita ja salausalgoritmeja. Työn keskivaiheilla kerrotaan sähköpostin salaukseen liittyvistä asioita sekä esitellään kaksi, yleisesti käytössä olevaa, sähköpostin salaussovellusta. Työn loppuosassa kerrotaan salausrajapinnan suunnittelusta sekä esitellään johtopäätöksiä.

Avainsanat OpenPGP, salaus, sähköposti, PGP, GnuPG

TAMPERE POLYTECHNIC
Computer Systems Engineering
Software engineering

Writer(s)	Toni Kauppinen
Thesis	Designing encryption interface using OpenPGP standard
Pages	34 pages
Graduation time	1/2009
Thesis supervisor	Tony Torp (lecturer)
Co-operating Company	Kilosoft Ltd. Supervisor Kimmo Hakkarainen (MSc)

ABSTRACT

The purpose of this thesis is design an interface, which can encrypt and decrypt an email note. Open C Application Programming Interface (API) includes OpenPGP-standard and that's why it's a main tool when had to design that kind of interfaces. Consequently, with this interface man can decrypt email notes, which are encrypted with GnuPG-software and encrypt email notes, which can be later decrypted with GnuPG. Also one of the goals is to get to know with OpenPGP standard and software, which are using that.

First of all this thesis goes through some backgrounds of encryption and encryption algorithms. In the middle part, there are some issues of email encryption and short description of the two most used encryption programs. And finally, in the last part the thesis tells about designing the encryption interface and some conclusions about that.

Keywords OpenPGP, encryption, email, PGP, GnuPG

Esipuhe

Tämä tutkintotyö on tehty syksyllä 2008 Kilosoft Oy:lle. Sain mahdollisuuden tutustua sähköpostin salaukseen ja lisäksi sain opiskella salaustekniikoita sekä oppia käyttämään monipuolisemmin Symbian-sovellusalustaa.

Kiitän Kilosoft Oy:n toimitusjohtajaa Jyrki Oksasta mielenkiintoisen työn mahdollistamisesta. Työn ohjaamisesta kiitokset kuuluvat puolestaan Kimmo Hakkaraiselle ja lehtori Tony Torpille.

Tampereella 7. tammikuuta 2009,

Toni Kauppinen

Sisällysluettelo

Lyhenteet ja termit

Alice	Kryptologian epävirallinen standardi kuvaamaan lähettäjää.
Bob	Kryptologian epävirallinen standardi kuvaamaan vastaanottajaa.
Eve	Kryptologian epävirallinen standardi kuvaamaan salakuuntelijaa. Muotoiltu sanasta Eavesdropper, salakuuntelija.
Kryptologia	Salauksiin ja niiden purkamiseen erikoistunut tieteenala. Nimi tulee kreikankielisistä sanoista kryptos (piilotettu) sekä logos (sana).
Kryptografia	Salakirjoituksen käyttö. Kryptografia on kryptologian osa-alue.
Kryptoanalyysi	Salauksen murtaminen. Kryptoanalyysi on kryptologian osa-alue.
Symbian	Mobiililaitteille suunnattu sovellusalusta.
Middleware	Ohjelmisto, joka yhdistää ohjelmistokomponentteja tai sovelluksia.
PGP	Pretty Good Privacy. Philip Zimmermannin ohjelmoima salaussovellus.
POSIX	Unix-pohjaisille käyttöjärjestelmille kehitetty standardi.
Open C	Kokonaisuus, joka sisältää POSIXin ja avoimen lähdekoodin standardikirjastot.
OpenPGP	Yleisesti käytetty sähköpostin salausstandardi.
GnuPG	Gnu Privacy Guard. Avoimeen lähdekoodiin perustuva salaussovellus.
MAC	Message Authentication Code. Tiiviste, jolla symmetrisellä salauksella salattu viesti todennetaan.
DSS	Digital Signature Standard. Tiiviste, jolla asymmetrisellä salauksella salattu viesti todennetaan.
DES	Data Encryption Standard. Symmetrinen salausalgoritmi.
RSA	Asymmetrinen salausalgoritmi. Lyhenne tulee tämän algoritmin kehittäjien sukunimien alkukirjaimista (Rivest, Shamir, Adleman).
PKI	Public Key Infrastructure, järjestelmä jolla hallitaan varmenteita.

1 Johdanto

Nykypäivänä on entistä tavallisempaa, että yritysten työntekijät hoitavat kommunikointinsa projektihenkilöstön sekä asiakkaidensa kanssa mobiililaitteilla. Tämän ovat mahdollistaneet nykyaikaiset älypuhelimet, koska niissä on puhelu- ja videopuheluominaisuudet sekä teksti- ja multimediaviestit ja sähköpostiviestin lukumahdollisuus.

Helppo ja tehokas viestintä tuo mukanaan myös suuria tietoturva-aukkoja yrityksille. Matkapuhelimissa olevat tiedot ovat useimmiten suojaamattomia, joten kuka tahansa joka saa puhelimen haltuunsa, voi tutkia näitä tietoja. Näin ollen, jos yrityksen työntekijällä on matkapuhelimeensa tallennettuna jotain salaista tietoa yrityksestä, se on voitu saada julkiseksi tehokkaiden viestintäkanavien avulla parhaimmillaan muutamassa minuutissa.

Teksti- ja multimediaviestien avulla ei välttämättä saada suuria vahinkoja aikaiseksi, mutta sähköpostiviestit liitetiedostoineen voivat sisältää todella arkaluontoisia asioita.

Tutkintotyön tavoitteena on suunnitella sähköpostin salaava ja purkava rajapinta. Rajapinta toteutetaan OpenPGP-standardin mukaisesti, jotta älypuhelimella voitaisiin purkaa sähköpostiviestejä, jotka on salattu käyttämällä GnuPG:tä. Lisäksi toiminnon on myös onnistuttava toiseen suuntaan.

Mobiilialustana käytetään Nokian S60-ohjelmistoalustaa, koska se tuo OpenC-kirjastojen avulla salauskirjaston, joka toteuttaa kattavan valikoiman eri kryptografisia algoritmeja, joita tässä työssä tullaan käyttämään.

2 Salaus

Salauksella tarkoitetaan yleensä viestin salaamista siten, ettei ulkopuolinen viestin haltuunsa saanut pysty avaamaan sitä. Salaus (encryption) on prosessi, jossa selväkielinen teksti (plaintext) muutetaan ei-selkokieliseksi tekstiksi (chiphertext). Purkaminen (decryption) on puolestaan käänteinen prosessi salaukseen nähden. /1/

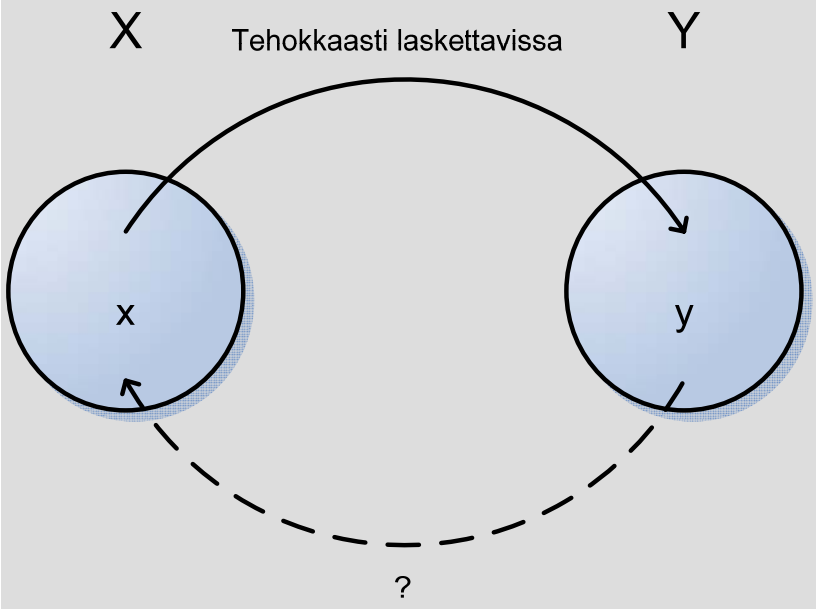
2.1 Avaimeton salaus

Salauksen yhteydessä käytettävät yksisuuntaiset funktiot, tiivistefunktiot ja satunnaisbittigeneraattorit ovat avaimettomia salauksia. Yhteinen tekijä näiden funktioiden välillä on se, etteivät ne käytä salaista parametria (avainta).

2.1.1 Yksisuuntainen funktio

Nykyaikana yksisuuntainen (one-way-funktio) funktio on keskeinen tekijä salauksessa. Yleisesti sanottuna funktio on yksisuuntainen, jos funktio X:n on *helppo* toteuttaa Y, mutta *vaikea* invertoida sitä. Tilanne on havainnollistettu kuviossa 1. Kompleksisuusteorian mukaan termi *helppo* tarkoittaa, että laskenta voidaan suorittaa tehokkaasti. Puolestaan termi *vaikea* tarkoittaa, että laskennalle ei ole todistettavasti tiedossa tehokasta suoritustapaa.

Kuviossa 1 on esitetty graafisesti yksisuuntaisen funktion toiminta. Funktio X toteuttaa tehokkaasti Y:n, mutta Y:stä ei pystytä toteuttamaan X:ää, koska tarvittavaa lisätietoa ei ole saatavilla.



Kuvio 1. Yksisuuntainen funktio. /1/

On myös olemassa sellaisia yksisuuntaisia funktioita, jotka voidaan tehokkaasti laskea toiseen suuntaan, jos vain tiedossa on hieman lisätietoa. Tällaisia funktioita kutsutaan takaportti-funktioiksi (trapdoor-funktio). Suoritettu funktio saadaan palautettua tehokkaasti alkuperäiseen tilaansa takaportin avulla.

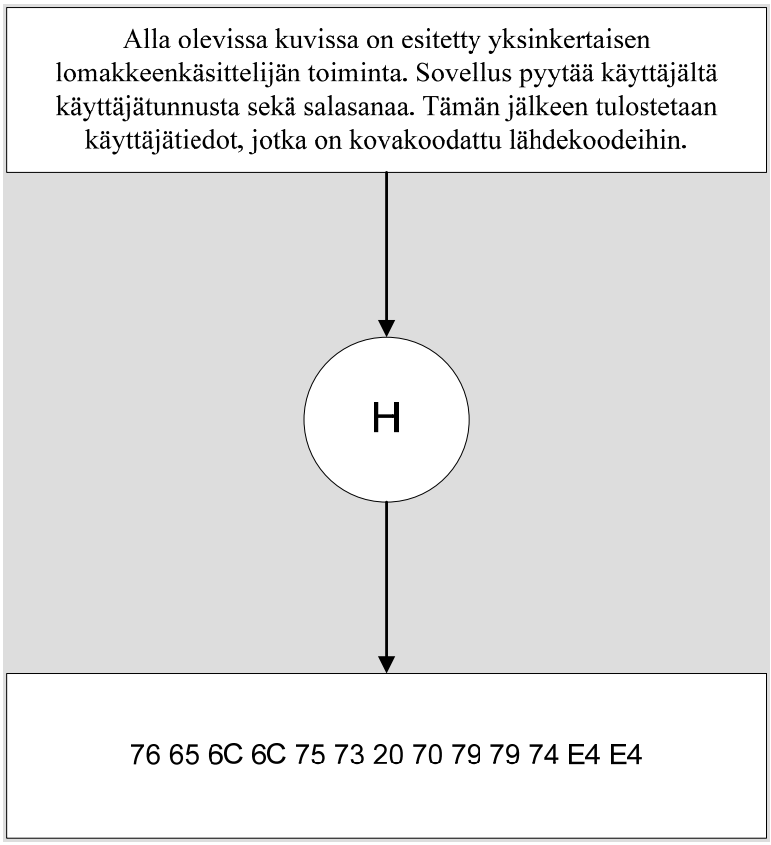
Esimerkki analogiasta

Riippulukko on takaportilla varustettu yksisuuntainen funktio. Riippulukon ollessa auki, sen voi sulkea kuka tahansa. Mutta avoimeksi lukon saa ainoastaan se, jolla on avain lukkoon. Riippulukko ilman avaimen reikää kuvaa yksisuuntaista funktiota ilman takaporttia. /1/

2.1.2 Tiivistefunktio

Tiivistefunktioita (hash-funktio) käytetään yleisesti ohjelmistotekniikassa ja niistä on saatavilla monia eri toteutuksia. Tiivistefunktio on tehokkaasti laskeva funktio, jolle annetaan mielivaltaisen kokoinen merkkijono, josta se generoi tietyn kokoisen merkkijonon. Kuviossa 2 on selvitetty tiivistefunktion toimintaa. Funktio saa

sisääntulona mielivaltaisen merkkijonon (ylempi laatikko), josta se generoi tietyn kokoisen tiivisteen (alempi laatikko). Kryptografiassa tiivistefunktion generoima tiiviste on ulostulossa salattuna.



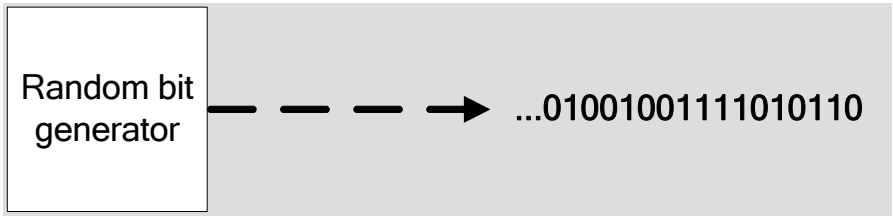
Kuvio 2. Tiivistefunktion toiminta. /1/

2.1.3 Satunnaisbittigeneraattori

Salaustekniikoiden sovelluksissa käytetään paljon satunnaisia lukuja. Satunnaisuudella pyritään hakemaan yleensä tarkoituksen ja säännönmukaisuuden puutetta. Siksi satunnaislukuja käytetään tietotekniikassa mm. suojaamaan tietoliikenneprotokollia toistohyökkäyksiltä ja yleisesti käytettävien tilapäisavaimien generoimista varten. Satunnaisuus on kaikkein tärkein tekijä ja edellytyksenä turvalliseen salausjärjestelmään. /2/

Satunnaisbittigeneraattori on laite tai algoritmi, jonka ulostulona (output) saadaan tilastollisesti peräkkäisiä itsenäisiä ja puolueettomia bittejä. Tämänkaltainen ulostulo on mahdollista vain silloin, kun generaattorilla ei ole minkäänlaista sisääntuloa (input).

Kuviossa 3 selvennetään satunnaisbittigeneraattorin toimintaa. Generaattorilla ei ole minkäänlaista sisääntuloa, mutta ulostuloksi saadaan satunnainen bittijono. /1/



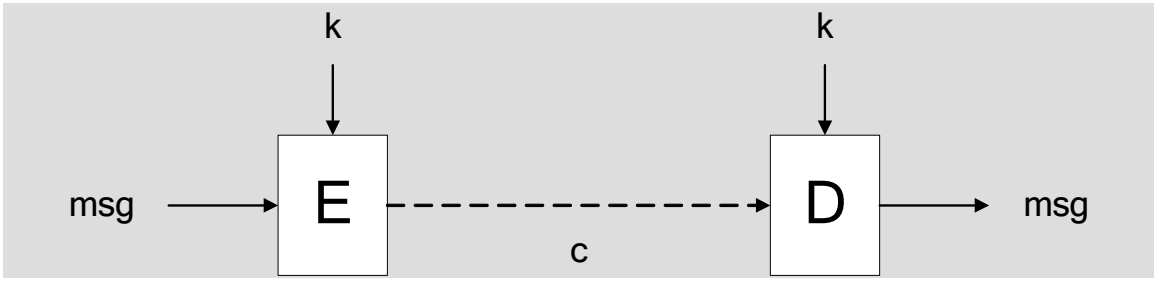
Kuvio 3. Satunnaisbittigeneraattori. /1/

2.2 Salaisen avaimen salaus

Symmetrinen salaus on salaisen avaimen salausjärjestelmä. Siinä kahdella toisistaan riippumattomalla osapuolella on yksi yhteinen salaisuus (salausavain).

2.2.1 Symmetrinen salaus

Yleensä salauksella tarkoitetaan symmetristä salausta. Symmetrinen salaus muodostuu selväkielisestä viestistä (msg), salatusta viestistä (c), salausavaimesta (k) jolla viesti salataan sekä salausprosessista (E) ja purkamisprosessista (D). Kuviossa 4 on kuvattu symmetrisen salauksen toimintaa.



Kuvio 4. Viestin salaus symmetrisen salauksen avulla. /1/

Käytetyimpiä symmetrisiä salausjärjestelmiä

DES, Data Encryption Standard /1/, /2/

- IBM:n ja NSA:n yhteistyössä kehittämä salain. Salaajan avaimen pituus on 56 bittiä, lohkon koko 64 bittiä ja kierrosten lukumäärä 16. Vuonna 1977 DES:stä tuli Yhdysvaltojen virallinen salausstandardi.

3-DES /2/

- 3-DES-salaimessa DES suoritetaan kolme kertaa peräkkäin ja avaimen pituutta saadaan kasvettua (3x56) 166 bittiin. Lohkon koko pysyy ennallaan ja salaaja on kolme kertaa hitaampi, kuin DES.

AES, Advanced Encryption Standard (Rijndael) /1/, /2/

- AES on Joan Daemenin ja Vincent Rijmenin kehittämä salain. Avaimen pituus voi olla mikä tahansa 32:n moninkerta, lohkon koko on 128 bittiä ja kierrosten lukumäärä on 10, 12 tai 14 avaimen koon mukaan. Koska menetelmä on uusi, ei sen heikkouksia tunneta ja siksi AES:n käyttö ei ole yleistynyt.

IDEA /2/

- IDEA on Xuejia Lain ja James Masseyn kehittämä salain. Lyhenne tulee sanoista International Data Encryption Algorithm. Avaimen pituus on 128 bittiä, lohkon koko 64 ja kierrosten määrä 8,5. Salaus on erittäin turvallinen, mutta hidas. Sitä käytetään oletussalaimena PGP-ohjelmassa.

Blowfish, Twofish /1/, /2/, /6/

- Blowfish on Bruce Schneierin kehittämä salain. Avaimen pituus voidaan valita 32:n ja 448 bitin väliltä, lohkon koko on 64 bittiä ja kierrosten lukumäärä on 16. Blowfish on patentoimaton ja rojaltivapaa. Twofish on Schneierin työryhmän edelleen kehittämä salain, jonka avaimen pituus voi olla 128, 192 tai 256. Lohkon koko on 128 ja kierrosten lukumäärä on 16. Twofishin rakenne muistuttaa DESiä.

2.2.2 Avaimellinen tiiviste

MAC-koodin avulla varmistetaan viestin eheys sekä lähettäjän henkilöllisyys, kun käytetään symmetristä salausta. Varmistus tehdään lisäämällä viestin loppuun salaisella avaimella salattu tiiviste, joka on generoitu tiivistefunktion avulla viestistä. On myös mahdollista käyttää julkisen avaimen salausjärjestelmän digitaalista allekirjoitusta, mutta toisinaan ei ole tarvetta tai edes suotavaa käyttää niin raskasta toimenpidettä. Tiivisteen purkamista varten vastaanottajalla on oltava tiedossa, millä avaimella tiiviste on salattu. /1/

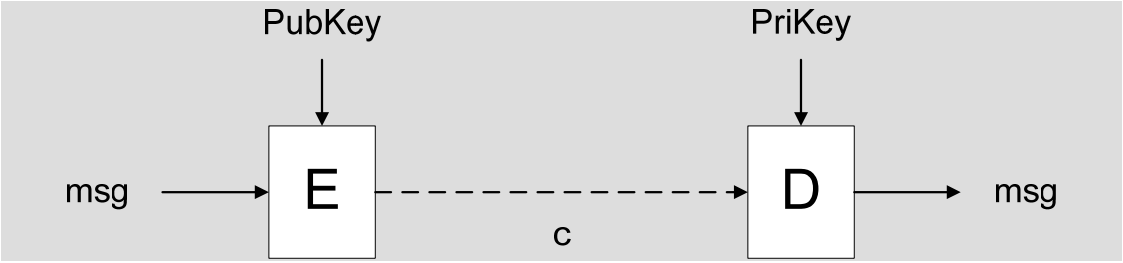
2.3 Julkisen avaimen salaus

Asymmetrinen salaus on julkisen avaimen salausjärjestelmä. Siinä kahdella toisistaan riippumattomalla osapuolella on avainpari, joka koostuu yksityisestä ja julkisesta avaimesta.

2.3.1 Asymmetrinen salaus

Asymmetrisestä salauksesta löytyy täysin samanlaiset toiminnot kuin symmetrisestä salauksesta. Poikkeuksena kuitenkin on se, että lähettäjä salaa viestin (msg) vastaanottajan julkisella avaimella (PubKey) ja vastaanottaja purkaa salatun viestin (c) omalla yksityisellä avaimellaan (PriKey). Asymmetrisen salauksen salausprosessi (E) ja purkamisprosessi (D) on havainnollistettu kuviossa 5.

Asymmetrinen salaus on takaportilla varustettu yksisuuntainen funktio, jossa julkinen avain toimii yksisuuntaisena funktiona ja yksityinen avain toimii tämän käänteisfunktiona eli takaporttina. Kuviossa 5 on kuvattu asymmetrisen salauksen toimintaa käyttäen edellisessä kappaleessa mainittuja symboleita.



Kuvio 5. Viestin salaus asymmetrisen salauksen avulla. /1/

Järjestelmää, jossa toisella avaimella salataan ja toisella puretaan, kutsutaan julkisen avaimen salaukseksi. Vaikka salausavaimet ovat erilaiset toisiinsa nähden, eivät ne ole silti mielivaltaisia. Avaimet liittyvät toisiinsa matemaattisella tavalla, mistä seuraa se, että ulkopuolisen on lähes mahdotonta yhdistää niitä.

Kautta aikojemme salaustekniikkamme ovat perustuneet lähettäjän ja vastaanottajan yhteiseen salaisuuteen. Kun tietokoneet alkoivat yleistyä, tuli mahdolliseksi käyttää tekniikkaa, jossa viestin salaamiseen käytetty avain voi olla julkinen, mutta purkuavain pidettiin salassa. /1/, /2/

Julkisen avaimen levittämistä ei tarvitse pelätä, koska avaimella voidaan ainoastaan salata viesti ja tarkistaa allekirjoitetun viestin oikeellisuus. Siksi onkin aivan tavallista, että ihmiset levittävät julkisia avaimiaan mm. omilla verkkosivuillaan, yleisillä avain-palvelimilla, keskustelukanavilla ja salaamattomien sähköpostiviestien liitetiedostoina. Yksityinen avain on pidettävä visusti omana tietonaan sillä se, mitä julkisella avaimella salataan, voidaan purkaa ainoastaan sitä vastaavalla yksityisellä avaimella.

Salauksen jälkeen ainoastaan julkista avainta vastaava yksityinen avain pystyy purkamaan viestin takaisin selväkieliseksi. Tästä syystä, jos lähettäjä haluaa säilyttää alkuperäisen viestin, täytyy siitä tehdä kopio ennen salausta.

Esimerkki asymmetrisen salauksen käytöstä

Alice haluaa lähettää Bobille salaista tietoa. Aluksi hän kirjoittaa selväkielisen viestin, jonka hän salaa Bobin julkisella avaimella. Tämän jälkeen Alice lähettää viestin verkon kautta Bobille. Evellä olisi mahdollisuus napata viesti verkosta, mutta viestin purkaminen ei onnistu, koska hänellä ei ole tiedossa Bobin yksityistä avainta. Siispä Bob vastaanottaa salatun viestin ja purkaa sen selväkieliseksi omalla yksityisellä avaimellaan.

Käytetyimpiä asymmetrisiä salausjärjestelmiä

RSA /1/, /2/

- RSA-salausalgoritmi perustuu suuriin alkulukuihin. Vuonna 1977 kolme nuorta matemaatikkoa kuvasivat tämän algoritmin. Nimi on johdettu heidän sukunimiensä alkukirjaimista (Rivest, Shamir ja Adleman). Verrattuna muihin julkisen avaimen salausjärjestelmiin, RSA mahdollistaa sekä asymmetrisen salauksen sekä digitaalisen allekirjoituksen. Tämä tarkoittaa sitä, että samalla algoritmilla voidaan salata, purkaa kuin myös allekirjoittaa viestejä ja todentaa digitaalisia allekirjoituksia.

Elliptiset käyrät (ECC) /1/, /2/

- Julkisen avaimen salausjärjestelmä, joka perustuu algebralliseen elliptisien käyrien rakenteeseen kaikissa äärellisissä kunnissa. Lyhenne tulee sanoista Elliptic Curve Cryptography. ECC-salaus on hitaampi, mutta turvallisempi kuin RSA. RSA:sta joudutaan luopumaan, jos nopea tekijöihinjako keksitään. Tästä seuraa se, että ECC on mitä todennäköisimmin RSA:n korvaaja.

2.3.2 Asymmetrisen salauksen rajoitukset

Vaikka julkisen avaimen tekniikka on jo itsestään hyvä salauskeino, löytyy siitä silti monia rajoituksia. Seuraavassa on käyty läpi muutamia sen rajoitteita.

1. Pitkät avaimet

Asymmetrisen avaimen tekniikassa tarvitaan huomattavasti pidempiä avaimia kuin mitä symmetrisen avaimen tekniikassa. Avaimen ulkoa muistaminen (jopa useita tuhansia bittejä) ei tule kuuloonkaan, joten yleensä se tallennetaan tiedostoon, joka salataan

symmetrisellä salauksella. Muistinvarainen säilytys ei ole mahdollista ja siksi avaimen käyttöön tarvitaan aina fyysinen laite tai tiedosto. /2/

2. Riippuvuus matematiikkaan

Asymmetriset menetelmät pohjautuvat matematiikkaan ja yleisimmät niistä suorittavat lukujen jakamista tekijöihin. Nykyään ei ole tiedossa tarpeeksi nopeaa tapaa suorittaa tekijöihin jakoa, mutta ei ole myöskään pystytty todistamaan, ettei niin voitaisi tulevaisuudessakaan tehdä. /2/

3. Hitaus

Kun verrataan symmetristä DES-salausalgoritmia asymmetriseen RSA-salausalgoritmiin, niin voidaan huomata selkeä nopeusero näiden kahden välillä. Kun jokin satunnainen aineisto salataan käyttämällä DESiä ja siihen menee aikaa 10 sekuntia, niin RSA:lla salattaessa sama kestää n. 17 minuuttia. Tämä johtuu siitä, että asymmetrisissä salauksissa lasketaan suurilla luvuilla (jopa yli 12 miljoonaa digittiä). /2/, /4/

4. Forward search -hyökkäys

Forward search -hyökkäyksen käyttö on mahdollista silloin, kun salattu selväteksti on lyhyt tai siitä tiedetään suurin osa. Silti tämän hyökkäyksen toteuttaminen on työläs prosessi, koska kokeiltavien vaihtoehtojen määrä on suuri. Lisäksi kestää kauan, koska Eve joutuu aina salaamaan selvätekstin Bobin avaimella ja vertaamaan saatua tulosta lähetettyyn salatekstiin.

Esimerkki Forward search -hyökkäyksestä

Alice haluaa lähettää lyhyen viestin Bobille ja salaa sen Bobin julkisella avaimella. Eve kaappaa salatun viestin. Eve testaa erilaisilla selväteksteillä ja vertailee, mikä niistä antaisi Bobin avaimella salattuna samanlaisen salatekstin, kuin Alice oli lähettänyt. Kun samanlainen tulos ilmenee, on Eve saanut tietää, mitä viestissä lukee. /2/

5. Luottamuksen synty

Julkisen avaimen järjestelmä ei takaa sitä, että Bob on todella Bob. Asian voi varmistaa esimerkiksi tapaamalla Bobin ja hänen kanssaan julkisia avaimia vaihtamalla. Tässäkin tilaisuudessa henkilöllisyys tulisi todentaa voimassa olevan henkilöllisyystodistuksen avulla. Uusimmissa julkisen avaimen järjestelmissä avaimet voidaan identifioida tarkemmin, esimerkiksi lisäämällä valokuva käyttäjästä avaimeen. /2/

2.3.3 Digitaalinen allekirjoitus

Joissakin asymmetrisissä salausalgoritmeissa, kuten RSA:ssa, on mahdollisuus käyttää digitaalista allekirjoitusta. Julkinen ja yksityinen avain ovat symmetrisiä, joten ne toimivat molempiin suuntiin. Alicen allekirjoittama viesti on kenen tahansa purettavissa, jolla on Alicen julkinen avain. Viestin purkauduttua selkeään muotoon, Bob voi olla vakuuttunut siitä, että viesti on tullut Alicelta.

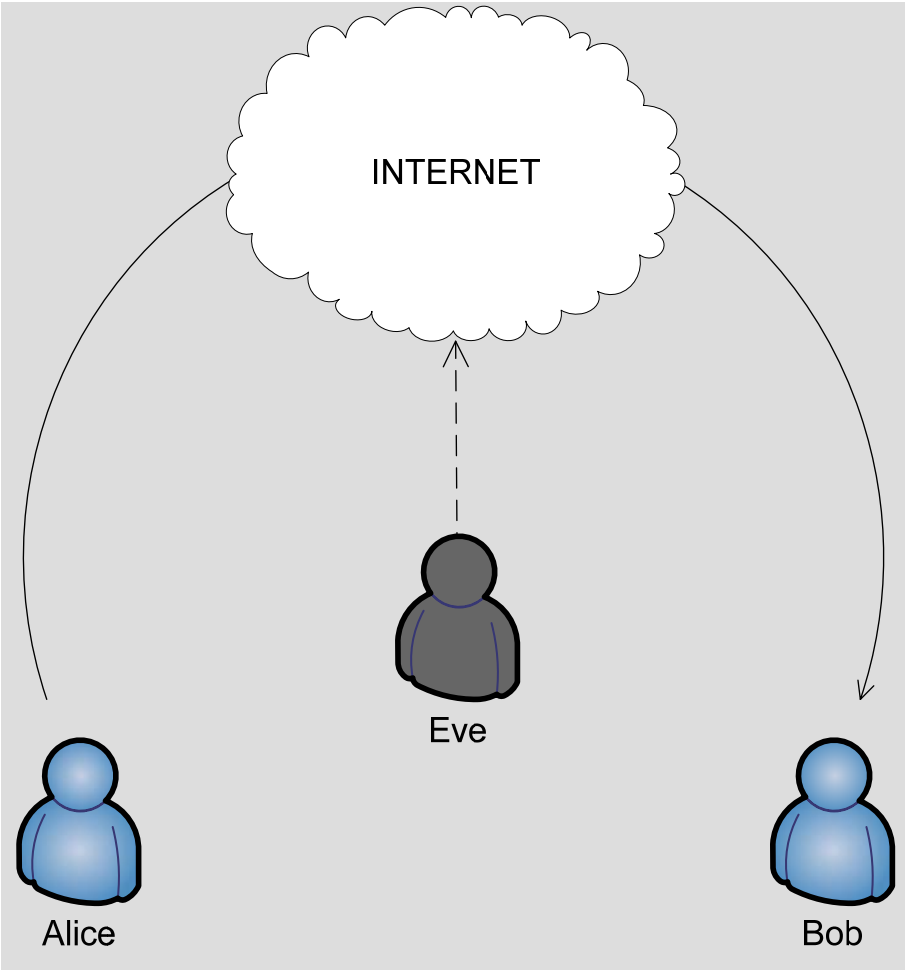
Allekirjoituksessa salataan tiiviste, joka saadaan laskettua alkuperäisestä viestistä. Even on siis mahdotonta laatia samanlaista viestiä, josta saataisiin laskettua sama tiivistearvo kuin alkuperäisestä viestistä. Tiiviste on lyhyt ja siksi sen käsittely sujuu nopeasti. /2/

2.3.4 PKI

PKI on järjestelmä jolla hallitaan varmenteita. Tähän kokonaisuuteen kuuluu varmenteiden myöntäminen, jakelu, hallinnointi sekä ylläpito. PKI voi olla EU:n, yhden yrityksen tai vaikkapa kaveripiirin laajuinen. Järjestelmän ytimenä toimii CA (Certification Authority) joka myöntää varmenteita. Jos henkilö haluaa avainpariinsa varmennuksen, hän antaa avainparin varmentajalle, joka tarkistaa tiedot ja allekirjoittaa ne omalla yksityisellä avaimellaan. /2/

3 Sähköpostin salaus

Käytämme lähes päivittäin sähköpostia lähettäessämme tärkeitä viestejä eri tahoille. Useimmiten ne lähetetään ilman minkäänlaista suojausta, ikään kuin postikortti ilman kirjekuorta. Postikortin voi lukea jokainen, joka pääsee siihen käsiksi sinä aikana, kun se kulkeutuu lähettäjältä vastaanottajalle: postin lajittelija, posteljooni ja pahimmassa tapauksessa jopa naapuri, jos postikortti on jaettu väärään osoitteeseen. Lähes sama koskee myös salaamatonta sähköpostiviestiä. Poikkeuksena on vain se, että viestin voi lukea maailmalla kuka tahansa, joka pystyy kuuntelemaan sitä verkkoa, jota viestin välitykseen käytetään. Kuviossa 6 Eve salakuuntelee Internetiä, jonka välityksellä Alice lähettää viestin Bobille. Jos viesti on salaamaton ja Eve saa napattua viestin, hän voi lukea sen.



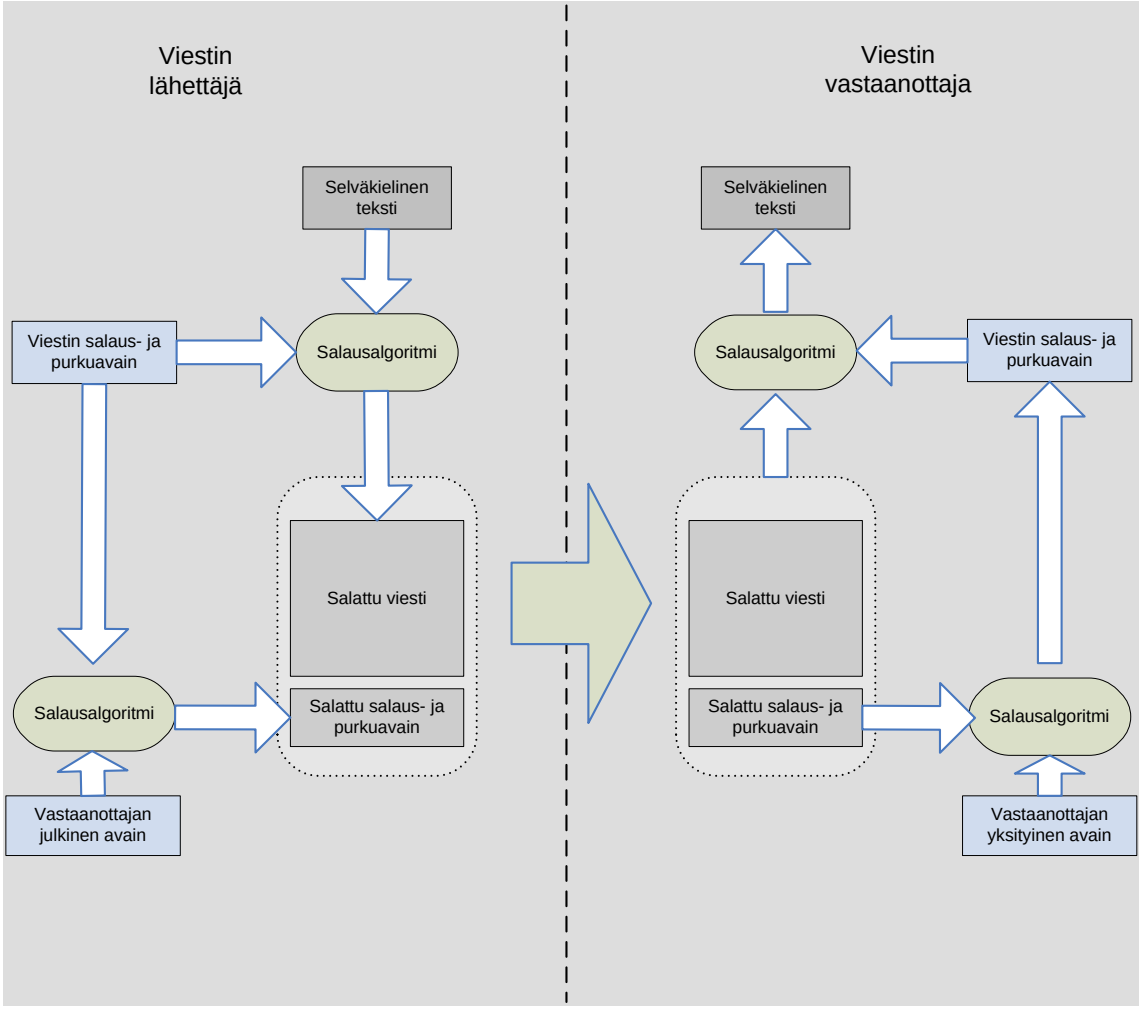
Kuvio 6. Viestin lähetys Internetin välityksellä.

Sähköpostiviestinnän luottamuksellisuutta voidaan parantaa salaamalla viesti ennen sen lähettämistä. Viesti voidaan myös allekirjoittaa digitaalisesti, jolloin voidaan varmistua siitä, kuka viestin on lähettänyt, ja että lähettäjä on juuri se, joka hän väittää olevansa. Tällöin on käytettävä sellaista allekirjoitusmenetelmää, joka perustuu varmenteisiin. Allekirjoituksella voidaan varmistaa myös se, ettei kukaan ole päässyt muuttamaan viestin sisältöä matkan varrella – toisin sanoen, viestin eheys on säilynyt. /3/

Salaustapoja on useita eri vaihtoehtoja, mutta juuri sähköpostiviestien salaukseen on vakiintunut julkisen avaimen salaus eli asymmetrinen salaus.

4 PGP ja GnuPG

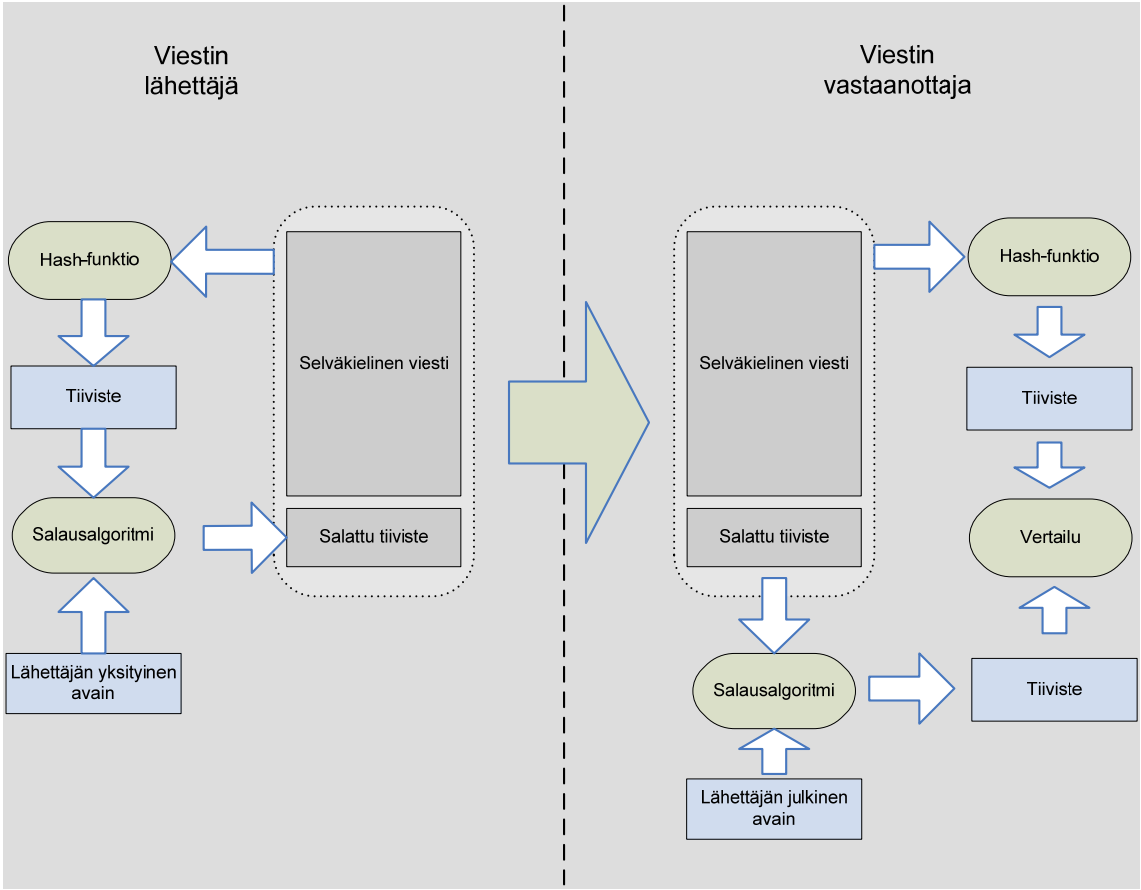
PGP (Pretty Good Privacy) on Phil Zimmermannin vuonna 1991 ohjelmoima salausohjelma. Se pyrkii mahdollistamaan vahvan salauksen myös muillekin ihmisille kuin diplomaateille ja sotilaille. PGP käyttää julkisen avaimen salausjärjestelmää, jolla voidaan salata, purkaa ja allekirjoittaa viestejä. PGP käyttää sekä symmetristä että asymmetristä salausta. Kuviossa 7 on esitettynä PGP:llä salatun viestin muodostaminen ja sen purkaminen takaisin luettavaan muotoon.



Kuvio 7. Viestin lähetys PGP:llä. /3/

Viesti salataan käyttäen symmetristä salausalgoritmia, jonka istuntoavain puolestaan salataan asymmetrisen salausalgoritmin julkisella avaimella. Tämän jälkeen salatusta viestistä ja sen salaaman algoritmin istuntoavaimesta muodostetaan viesti. Käyttäjän näkökulmasta PGP toimii täysin asymmetrisesti. /3/

Viestin allekirjoittaminen tapahtuu salausalgoritmin yksityisellä avaimella. Selväkielisestä viestistä otetaan tiivistefunktion avulla tiiviste, joka salataan yksityisellä avaimella. Tämän jälkeen salatusta tiivisteestä ja selväkielisestä viestistä muodostetaan sähköpostiviesti. Allekirjoitusprosessi on kuvattuna kuviossa 8. /3/



Kuvio 8. Viestin allekirjoitus PGP:llä. /3/

PGP:n kaupallistuttua ja kehitysvastuun siirtyminen Yhdysvaltoihin 2000-luvun vaihteessa käynnisti se avoimeen lähdekoodin perustuvan GnuPG kehityksen. GnuPG on yhteensopiva PGP:n kanssa, mutta se ei käytä lainkaan kaupallisia tai lisensoituja kirjastoja. /2/

GnuPG on komentorivipohjainen salausmoottori, josta on tehty lukuisia porttauksia eri alustoille. Sovelluksen käyttöä on helpotettu erilaisilla graafisilla käyttöliittymillä, jotka generoivat saadut tehtävät komentorivikäskyiksi. GnuPG toteuttaa OpenPGP-protokollan (salatun viestinnän protokolla) vaatimat määritykset. /6/

5 OpenPGP ja Open C-kirjasto

5.1 Lyhyt kuvaus OpenPGP-standardista /7/

OpenPGP käyttää vahvaa kombinaatiota, julkisen avaimen salausta ja symmetristä salausta luomaan turvatut palvelut elektronista kommunikaatiota ja tiedon tallentamista varten. Nämä palvelut sisältävät tietosuojan, avainhallinnan, todennuksen ja digitaalisen allekirjoituksen.

5.1.1 Tietosuojaa salauksen avulla

OpenPGP luo tietosuojan yhdistämällä symmetrisen avaimen salauksen ja julkisen avaimen salauksen. Kun suojausta tehdään, aluksi objekti salataan symmetrisellä salausalgoritmilla. Jokaista symmetristä avainta käytetään vain kerran ja yhdelle objektille kerrallaan. Jokaiselle objektille generoidaan istuntoavain satunnaisesti. Koska tätä avainta käytetään vain kerran, sisällytetään se viestiin, joka lähetetään vastaanottajalle. Avain salataan vastaanottajan julkisella avaimella.

Salaamisen sekvenssi on seuraavanlainen:

1. Lähettäjä luo viestin.
2. Lähettävä sovellus generoi istuntoavaimen kyseistä viestiä varten.
3. Istuntoavain salataan jokaisen vastaanottajan julkisella avaimella. Nämä salatut istuntoavaimet aloittavat viestin.
4. Lähettävä sovellus salaa viestin istuntoavaimen avulla. Salattu viesti muodostaa viestin loppuosan. Useimmiten viesti myös pakataan.
5. Vastaanottava sovellus purkaa istuntoavaimen käyttämällä vastaanottajan yksityistä avainta.
6. Vastaanottava sovellus purkaa viestin istuntoavaimella. Jos viesti on pakattu, se puretaan takaisin alkuperäiseen muotoonsa.

5.1.2 Todentaminen digitaalisen allekirjoituksen avulla

Digitaalinen allekirjoitus käyttää tiivistekoodia tai viestin tiivistealgoritmia ja julkisen avaimen allekirjoitusalgoritmia. Sekvenssi on seuraavanlainen:

1. Lähettäjä luo viestin.
2. Lähettävä sovellus generoi tiivistekoodin viestistä.
3. Lähettävä sovellus generoi allekirjoituksen tiivistekoodista lähettäjän yksityisen avaimen avulla.
4. Binaarinen allekirjoitus liitetään viestiin.
5. Vastaanottava sovellus tekee kopion viestin allekirjoituksesta.
6. Vastaanottava sovellus generoi uuden tiivistekoodin vastaanotetusta viestistä ja vertailee sitä viestin allekirjoitukseen. Jos vertailu onnistuu, viesti hyväksytään muuttumattomana.

5.2 Symbian Open C /8/

Open C tuo S60 3rd Editionille kattavan kokoelman standardeja C-ohjelmistorajapintoja. Open C antaa mahdollisuuden luoda middlewareja ja sovelluksia S60 3rd Edition laitteille, vaikka ohjelmoijalla olisi vain hiukan tai ei ollenkaan kokemusta Symbian C++ kehityksestä. Ohjelmistorajapinnat tuovat mahdollisuuden portata helposti Open C työpöytäsovelluksia S60 alustalle. Open C sisältää seuraavat kirjastot:

Standardi C-kirjastot

Libcrypt	Kryptografiakirjasto, joka sisältää funktiot salaamaan/purkamaan datalohkoja sekä viestien ja salasanan tiivistämiseen.

Middleware-kirjastot

Libz	’zlib’-tiivistyskirjasto sisältää keskusmuistin tiivistämis- ja purkamisfunktiot.
Libcrypto	OpenSSL-salauskirjasto toteuttaa laajan valikoiman eri salausalgoritmeja.
Libssl	OpenSSL-ssl-kirjasto toteuttaa SSL- ja TLS-protokollat.
Libglib	Yleishyödyllisiä työkaluja, jotka sisältävät hyödyllisiä tietotyyppejä, makroja, tyyppi muunnoksia, merkkijono ja tiedosto työkaluja.

Geneeriset kirjastot

libcrt0	Käännöksen aikana tarvittava kirjasto.
libwcrt0	Käännöksen aikana tarvittava kirjasto.

Libcrypto

Libcrypto on salauskirjasto joka toteuttaa laajan valikoiman salausalgoritmeja, joita käytetään monissa Internet-standardeissa. Palvelut, joita tämä kirjasto toteuttaa, ovat OpenSSL-toteutukset SSL:stä, TLS:stä ja S/MIME:stä. Lisäksi kirjasto toteuttaa SSH-, OpenPGP- sekä muita standardeja.

Salauskirjasto sisältää muutamia rajoitteita, jotka on otettava huomioon, kun sitä käytetään. Rajoituksia ovat mm. seuraavat:

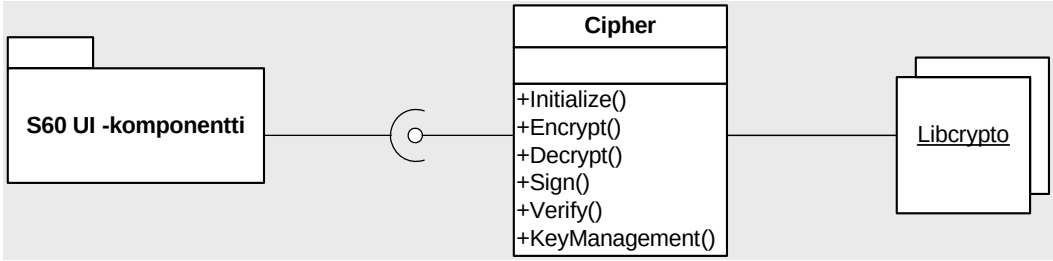
- Patentilla suojatut algoritmit kuten Rc5, IDEA, Blowfish, CAST, RIPEMD, MDC2, ECC, ECDH ja ECDSA eivät sisälly kirjastoon.
- Sertifikaattien pitäisi olla .der muodossa Symbianin sertifikaatti varastossa, jotta vältyttäisiin sertifikaattiristiriidoilta.
- Jotkut kirjaston funktioista vaativat enemmän muistia, kun sovelluksella on oletuksena käytössään. Suositeltu oletusmuistin koko on 10 kB.

6 Salausrajapinnan suunnittelu

Tässä kappaleessa esitellään suunniteltava salausrajapinta, tärkeimmät sille asetetut vaatimukset sekä suunnittelussa huomioon otettavat rajoitukset.

6.1 Rajapinnan kuvaus

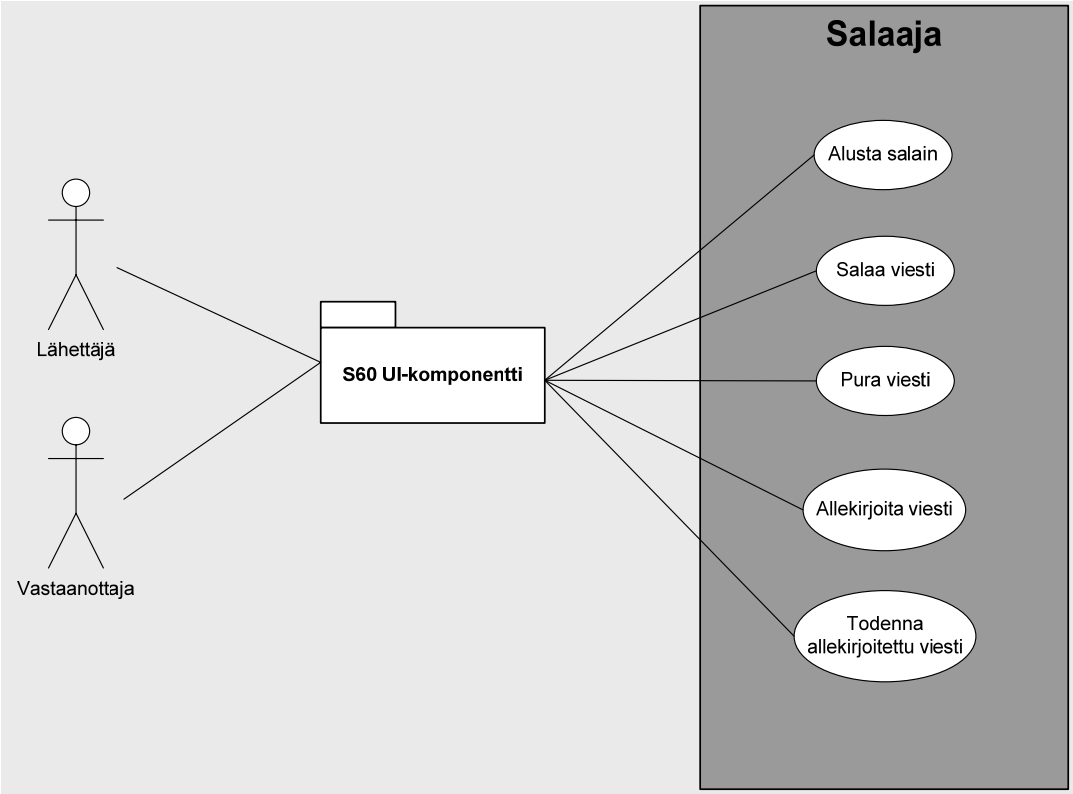
Salausrajapinta kostuu yhdestä Symbian C++-luokasta, joka pitää sisällään funktiokutsut libcrypto-kirjastoon. Luokasta luodaan instanssi halutussa S60 UI -komponentissa, joka hoitaa myös luokan metodien kutsumisen. UI-komponentti tuhoaa luodun instanssin sen jälkeen, kun salausrajapinta ei ole enää käytössä. Kuviossa 9 on esitetty korkealla tasolla salausrajapinnan ja libcrypto-kirjaston välinen yhteys.



Kuvio 9. Komponenttien väliset yhteydet.

Salausrajapinta on yhteydessä Open C:n libcrypto-kirjastoon ja tämän takia sillä voidaan hallita OpenPGP-standardin mukaisia sähköpostiviestejä. Rajapinnassa on myös ominaisuus, jonka avulla voidaan hallita käyttäjän tiedossa olevia julkisia avaimia.

Kuviossa 10 on esitetty salausrajapinnan käyttötapaukset käyttötapauskaavion avulla. Käyttötapausten avulla voidaan mallintaa rajapinnan toiminnallisuuksia.



Kuvio 10. Salausrajapinnan käyttötapaukset.

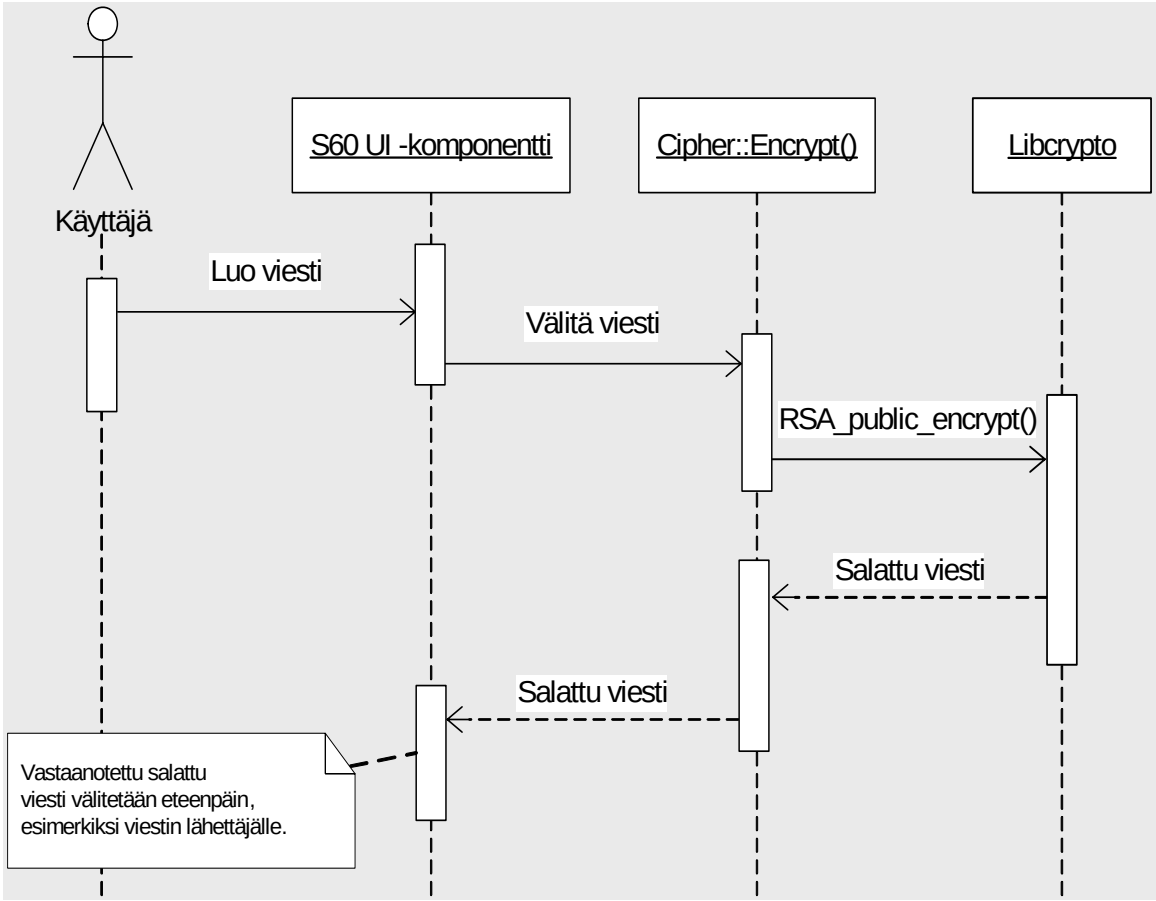
Kuviosta 10 käy ilmi, että käyttäjä pystyy käyttämään salausrajapintaa vain S60 UI-komponentin välityksellä. Näin saadaan eroteltua salauslogiikka käyttöliittymän toiminnallisuuksista. Seuraavassa käydään läpi kaikki käyttötapaukset sekä kerrotaan tarkemmin niiden toiminnallisuuksia.

Käyttötapaus 1: Alusta salain

Kun käyttäjä ensimmäisen kerran käyttää salausrajapintaa, alustetaan se käyttäjän määrittämillä parametreilla. Parametreilla määritellään salausalgoritmin salasana (passphrase) ja salausavain pari, jota käytetään viestin salaukseen, purkamiseen, allekirjoittamiseen tai allekirjoituksen todentamiseen.

Käyttötapaus 2: Salaa viesti

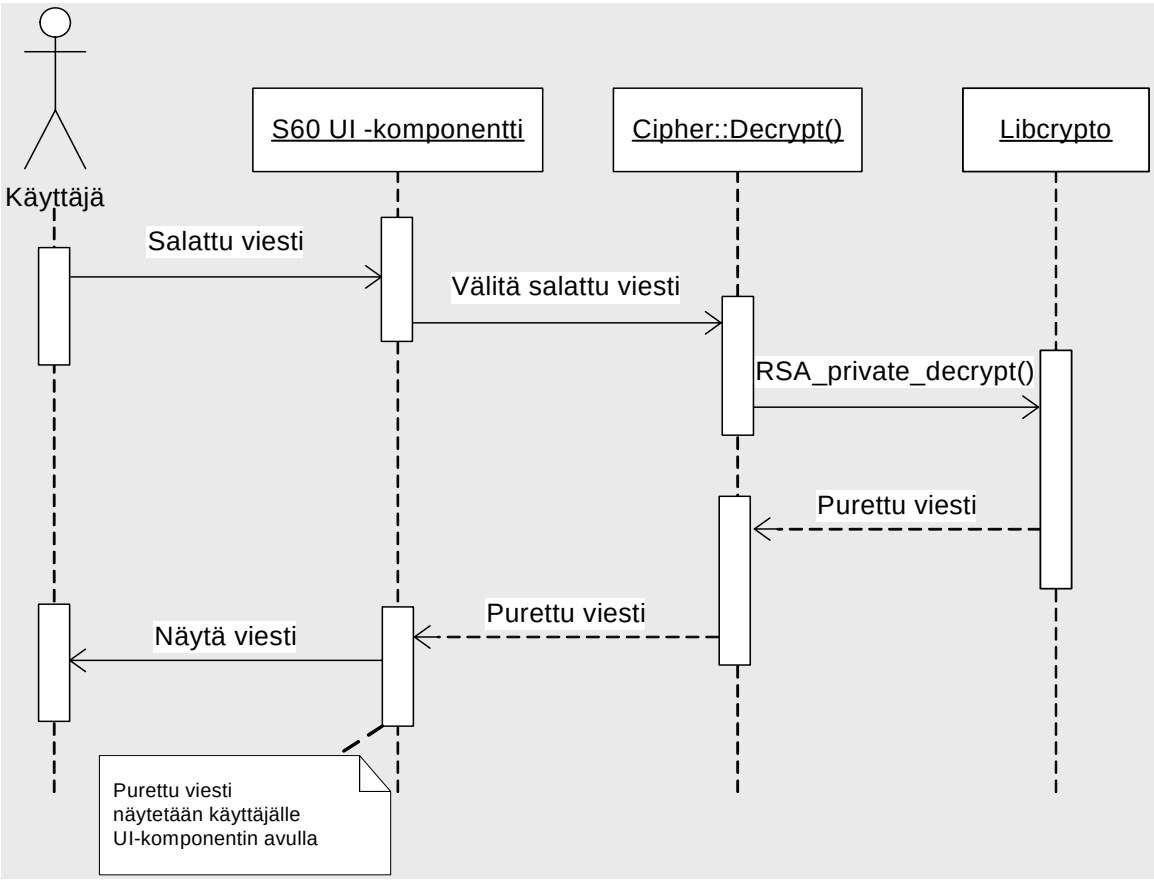
Käyttäjä luo viestin UI-komponentilla, joka välittää merkkijonon salausrajapinnalle. Rajapinta käyttää *RSA_public_encrypt()*-funktiokutsua, joka salaa viestin valitulla salausavaimella. Tämän jälkeen salattu viesti palautetaan takaisin UI-komponentille, joka huolehtii sen välittämisestä eteenpäin. Kuviossa 11 on kuvattu edellä mainitut vuorovaikutukset sekvenssikaavion avulla.



Kuvio 11. Salaamisprosessi.

Käyttötapaus 3: Pura viesti

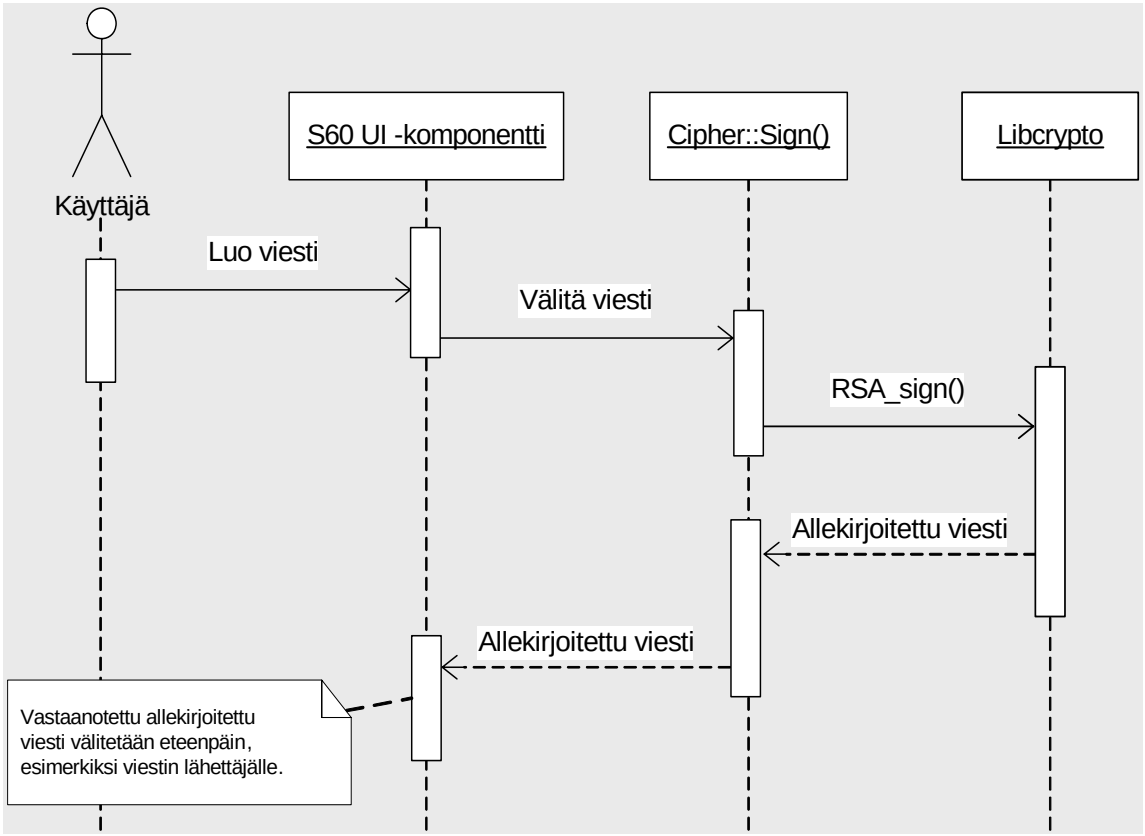
Käyttäjä välittää vastaanotetun salatun viestin UI-komponentilla salausrajapinnalle. Rajapinta käyttää *RSA_private_decrypt()*-funktiokutsua, joka purkaa viestin valitulla purkuavaimella. Tämän jälkeen purettu viesti palautetaan takaisin UI-komponentille, joka näyttää puretun viestin käyttäjälle. Kuviossa 12 on kuvattu edellä mainitut vuorovaikutukset sekvenssikaavion avulla.



Kuvio 12. Purkamisprosessi.

Käyttötapaus 4: Allekirjoita viesti

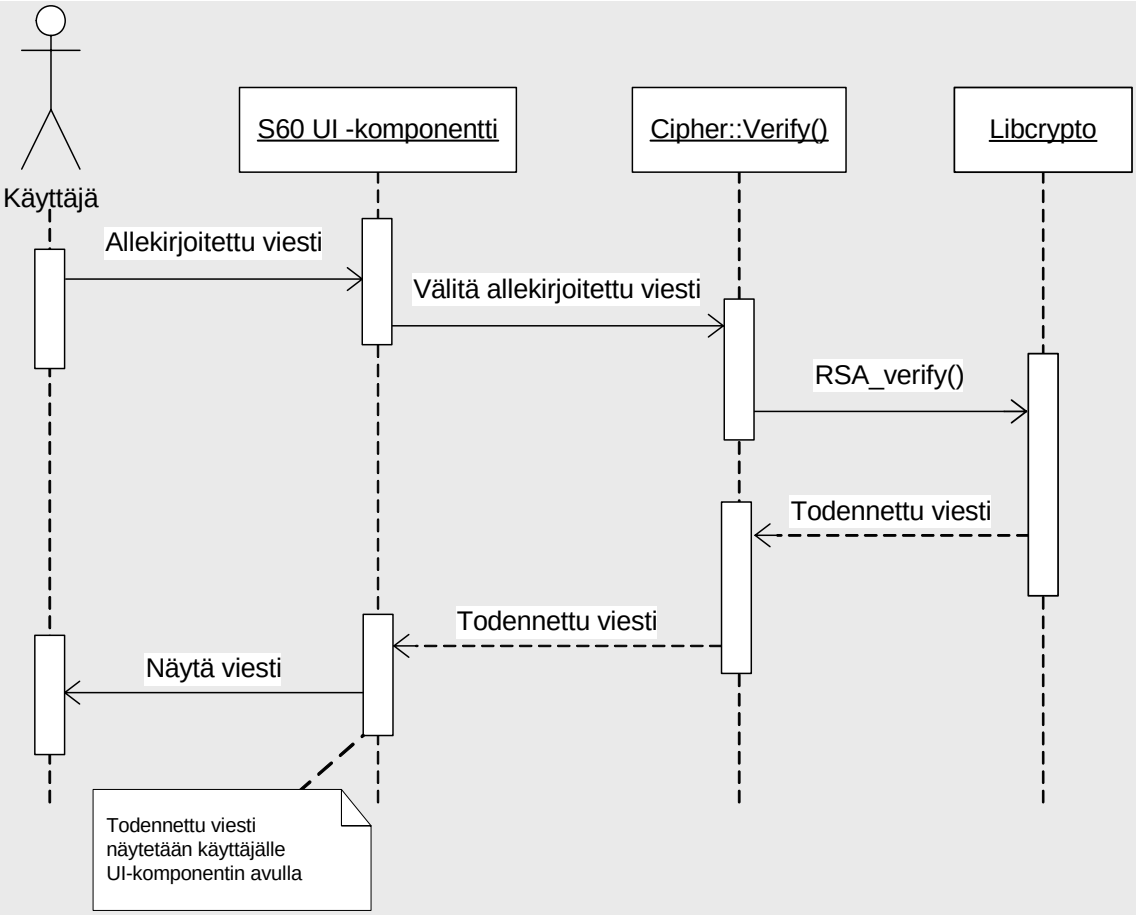
Käyttäjä luo viestin UI-komponentilla, joka välittää merkkijonon salausrajapinnalle. Rajapinta käyttää *RSA_sign()*-funktiokutsua, joka allekirjoittaa viestin valitulla yksityisellä avaimella. Tämän jälkeen allekirjoitettu viesti palautetaan takaisin UI-komponentille, joka huolehtii sen välittämisestä eteenpäin. Kuviossa 13 on kuvattu edellä mainitut vuorovaikutukset sekvenssikaavion avulla.



Kuvio 13. Allekirjoitusprosessi.

Käyttötapaus 5: Todenna allekirjoitettu viesti

Käyttäjä välittää vastaanotetun allekirjoitetun viestin UI-komponentilla salausrajapinnalle. Rajapinta käyttää *RSA_verify()*-funktiokutsua, joka todentaa viestin valitulla julkisella avaimella. Tämän jälkeen todennettu viesti palautetaan takaisin UI-komponentille, joka näyttää viestin käyttäjälle. Kuviossa 14 on kuvattu edellä mainitut vuorovaikutukset sekvenssikaavion avulla.



Kuvio 14. Allekirjoituksen todentamisprosessi.

6.2 Tärkeimmät vaatimukset

Yhteensopivuus GnuPG:n kanssa

Salausrajapinnan pitää pystyä purkamaan GnuPG-ohjelmalla salatut viestit. Lisäksi toiminnon on toimittava toiseen suuntaankin. Viestin allekirjoitus ja sen todentaminen pitää olla myös yhteensopiva GnuPG-ohjelman kanssa.

Avainten hallinta

GnuPG-ohjelmalla generoidut avaimet pitää voida ottaa käyttöön myös laitteessa, joka käyttää suunniteltua salausrajapintaa. Avainten hallinnassa pitää ottaa huomioon myös muut mahdolliset laitteessa säilytettävät avaimet. Symbian omaa sertifikaattivaraston, jossa säilötään laitteessa olevia sertifikaatteja. Tämän avulla vältytään mahdollisilta sertifikaattiristiriidoilta.

S60-sovellus

Salausrajapinta pitää toteuttaa S60-sovellusalustalle ja sen on oltava toteutettavissa standardi S60 UI -komponenttien avulla.

6.3 Suunnittelunrajoitukset

Open C

Open C-kirjastot tuovat suurimmat rajoitteet rajapintaa suunniteltaessa. Käytettäviä salausalgoritmeja pitää käyttää C-ohjelmointikielen mukaisesti. Tämä vaikeuttaa jo ennestään vaikeakäyttöistä Symbianin muistinhallintaa ja se aiheuttaa vaikeasti löydettäviä muistivuotoja. Dokumentaatiota apuna käyttäen vältytään yleisimmiltä muistivuodoilta.

OpenPGP

OpenPGP-standardi määrittelee salattavan viestin muodon. Standardin toteuttaminen onnistuu libcrypto-kirjaston avulla, koska se toteuttaa kaikki tarvittavat algoritmit viestin rakenteen muodostamista varten.

Laitteistoympäristö

Salausrajapinta toimii Symbian60v3 käyttöjärjestelmää käyttävissä matkapuhelimissa. Open C -kirjastojen takia laitteen käyttöliittymän version pitää olla 9.2 tai uudempi.

7 Yhteenveto

Työn tavoitteena oli suunnitella sähköpostin salaava ja purkava rajapinta. Rajapintaa suunniteltaessa apuna tuli käyttää Open C -ohjelmistorajapintaa, joka toteuttaa OpenPGP-salausstandardin. Lisäksi tehtävänä oli tutustua OpenPGP-salausstandardiin ja sitä käyttäviin PGP- ja GnuPG-salaussovelluksiin. Vaikka itse rajapinnan suunnittelu ei ollut suuritöinen, niin siihen liittyvän taustatiedon määrä oli huikea. Taustatietoja tutkiessani opin paljon salaamisesta ja sen vuoksi onnistuin tavoitteissa. Lisäksi taustatutkimusten perusteella erilaisten salausjärjestelmien ymmärtäminen on selkeytynyt ja aikaisempiin kysymyksiin on löytynyt vastauksia.

Seuraavana tavoitteena on ryhtyä toteuttamaan salausrajapintaa suunnitelmieni pohjalta. Tarkoituksena on toteuttaa S60-mobiililaitteelle sähköpostisovellus, joka on yhteensopiva GnuPG:n kanssa. Toteutusta on jo aloiteltu ja ensimmäisiä toiminnallisuuksiakin on jo valmiina.

Kryptologia on kaiken kaikkiaan mielenkiintoinen ja haastava tieteenala. Vaikeudestaan huolimatta se jäi kiehtomaan tätä työtä tehdessä ja toivottavasti pystyn olemaan mukana sellaisissa projekteissa, jotka käsittelevät salausta ja sen purkamista.

Lähteet

Painetut lähteet

1 Oppliger, Rolf 2005. Contemporary Cryptography. Artech House, INC.

2 Järvinen, Petri 2003. Salaus-menetelmät. Docendo Finland Oy

Sähköiset lähteet

3 Viestintävirasto [www-sivu]. [viitattu 1.12.2008] Saatavissa: http://www.ficora.fi/index/palvelut/palvelutaiheittain/tietoturva/sahkoposti.html

4 The Largest Known Primes – A Summary [www-sivu]. [viitattu 2.12.2008] Saatavissa: http://primes.utm.edu/largest.html

5 OpenPGP Message Format [tekstitiedosto]. [viitattu 8.12.2008] Saatavissa: http://www.ietf.org/rfc/rfc4880.txt

6 The Gnu Private Guard [www-sivu]. [viitattu 13.12.2008] Saatavissa: http://www.gnupg.org/

7 The Blowfish Encryption Algorithm [www-sivu]. [viitattu 13.12.2008] Saatavissa: http://www.schneier.com/blowfish.html

8 S60 Open C API Reference Guide and Developer Guides [www-sivu]. [15.12.2008] Saatavissa: http://library.forum.nokia.com/index.jsp?topic=/S60_5th_Edition_Cpp_Developers_Library/GUID-FE27AB35-C6FD-4F11-802D-0D5FCFFC2976/html/mrt/main.html