RSA

RSA on ensimmäinen julkisen avaimen salausmenetelmä, jonka esittivät tutkijat Rivest, Shamir ja Adlemann v.1978. (kuvassa)

V. 2015 RSA on ollut standardina jo 37 vuotta, ja pysynee standardina vielä jonkin aikaa. Lähes kaikki SSL – yhteydet käyttävät RSA:ta digitaaliseen allekirjoitukseen ja avaimesta sopimiseen. Turvallisena pidetyt avainpituudet ovat tosin nousseet tuona aikana. Tänään turvallisena pidetty RSA:n julkinen avain on 2048 bittinen. RSA:n kaupallisia versioita kehittää yhtiö RSA Laboratories

RSA:n avaimet

Jokaisella käyttäjällä on seuraavat RSA- avaimet:

Julkiset avaimet:	modulus **n** = p*q, kahden alkuluvun p ja q tulo eksponentti **e** väliltä 1.. n-1 *jolle on ehtona, että GCD(e, (p-1)(q-1)) = 1*
Yksityinen avain:	**d** = e^{-1} mod (p-1)(q-1)

Huom. SSL (TLS) – protokollassa käytetyssä RSA – versiossa, jota palvelinsertifikaatit käyttävät, julkinen eksponentti e on kaikille käyttäjille sama vakio $2^{16} + 1$ eli 65537, joten RSA – avaimia on tällöin vain kaksi: julkinen n ja yksityinen d

VIESTIN SALAUS $c = m^e \bmod n$

Viesti esitetään kokonaislukuina m. Salauksessa käytetään vastaanottajan julkisia avaimia n ja e.

PURKU $m = c^d \bmod n$

Vastaanottaja purkaa salauksen yksityisellä avaimellaan d

RSA:n matemaattinen perustelu

* Kun n = p*q, missä p ja q ovat alkulukuja, niin kertolaskuryhmän Zn alkioiden lukumäärä φ(n) = (p-1)((q-1)

- Eulerin lauseen mukaan, kaikille Z_n^*:n alkioille m on voimassa
  $m^{\varphi(n)} = m^{(p-1)(q-1)} = 1 \bmod n$

- Jos kaksi eksponenttia on valittu siten, että
  $e*d \bmod \varphi(n) = 1$ $\quad (d = e^{-1} \bmod (p-1)(q-1))$

niin

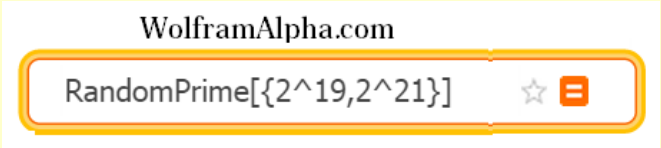
$$
\begin{aligned}
& (m^e)^d \bmod n \\
= \; & m^{ed} \bmod n \\
= \; & m^{1+k\varphi(n)} \bmod n \\
= \; & m \cdot m^{k\varphi(n)} \bmod n \\
= \; & m * (m^{\varphi(n)} \bmod n)^k \\
= \; & m*1 \bmod n \\
= \; & m
\end{aligned}
$$

- Mikä potenssilla e salataan, voidaan d:llä purkaa, ja päinvastoin.

Huom! Matemaatikko huomauttaisi, että yo. todistus ei päde tapauksessa, jossa viesti m ei ole kertolaskuryhmän Z_n^* alkio eli kun n:llä ja m:llä on yhteisiä tekijöitä. Käytännössä viesti m olisi tällöin joko p:n tai q:n monikerta. Tähän todettakoon, että RSA toimii tässäkin epätodennäköisessä tapauksessa. Todistuksia tälle löytyy netistä.

Esim. Luodaan RSA – avaimet, joissa modulus n on n. 40 bittiä

Julkiset avaimet:
Luodaan WolframAlphalla 2 alkulukua, joiden bittimäärä on 19 – 20.



Toistetaan kahdesti käsky

=> saadaan esim. p = 2 009 243 ja q = 1 951 867

Näistä saadaan julkinen avain tulona
n = pq = 3921775106681

Eksponentiksi voidaan valita SSL:n tapaan vakio $2^{16}+1$ eli
e = 65537

Yksityinen avain d = e^{-1} mod (p-1)(q-1):
d= 65537^{-1} mod 2009242*1951866

=> d = 903652380629

Testataan salausta RSA -algoritmilla

Käytetään edellä luotuja RSA – avaimia:

Salataan viesti m = 123987

Salakirjoitus lasketaan kaavalla $c = m^e \bmod n$:

123987^65537 mod 3921775106681 = 1287124623171

Purku tapahtuu purkuavaimella d laskemalla $m = c^d \bmod n$

1287124623171 ^903652380629 mod 3921775106681 =123987

=> RSA toimii

Viestin koodaaminen kokonaisluvuiksi m

Salattavat viestit voivat olla esim. tekstiä. Teksti muutetaan kokonaisluvuiksia lohkoissa. Lohkojen merkit muutetaan ensin merkkien ASCII- koodeiksi. Näistä muodostetaan 256 – kantaisen lukujärjestelmän kautta kymmenjärjestelmän kokonaislukuja m, jotka sitten salataan RSA:lla

Esim. Olkoon viesti ”Kemi”
Sen merkkien ASCII koodit ovat {75, 101, 109, 105}

Muuntaminen kokonaisluvuksi tapahtuu tulkitsemalla koodit 256- kantaisen lukujärjestelmän kertoimiksi:

$m = 75*256^3 + 101*256^2 + 109*256 + 105 = 1264938345$

Lukujen dekoodaus merkeiksi

- Vaatii lukujärjestelmämuunnosten osaamista, tässä tapauksessa kymmenjärjestelmästä 256- kantaiseen
- Lisäksi tarvitaan merkkien ASCII kooditaulukkoa.

Helpommin muunnokset onnistuvat wolframalpha.com online – laskimella seuraavien esimerkkien mukaisesti.

m = 1264938345

Muunnetaan tämä 256- kantaiseksi luvuksi:
IntegerDigits[1264938345,256] antaa
{75, 101, 109, 105}

Nämä ASCII koodit muutetaan merkeiksi:
FromCharacterCode[{75, 101, 109, 105}] antaa
Kemi

RSA:n turvallisuus

RSA voidaan yleisen käsityksen mukaan murtaa vain, jos hyökkääjä onnistuu selvittämään julkisen avaimen n alkulukutekijät p ja q.

RSA:n turvallisuus perustuu suurten kokonaislukujen tekijöihin jaon vaikeuteen.

Paras tunnettu yli 100 numeroisten kokonaislukujen tekijöihinjakomenetelmä on GNFS (General Number Field Sieve).

RSA laboratories yhtiöllä oli netissä vuoteen 2009 saakka yleisölle tarkoitettuja haastelukuja, joiden tekijöihinjaosta se maksoi kymmenien tuhansien dollarien palkkion. Haastelukujen tarkoituksena oli mitata RSA:n kryptoanalyysin kehitystä. Viimeisistä haasteluvuista sai palkkion T.Kleinjung työryhmineen.

Suurin faktoroitu haasteluku RSA768 on seuraava 768 – bittinen kokonaisluku:
n=123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3

On arvioitu, että NSA kykenisi murtamaan tällä hetkellä 1024- bittisiä RSA- avaimia. Siksi RSA:n avaimenpituussuositus on nostettu yleisesti 2048 bittiin.

PK-salauksen avainpituudet

Kuvaus	RSA	DH, Elgamal	ECC
Voidaan murtaa perustekniikoilla	816 bits	816	**128**
Voidaan murtaa lyhyessä ajassa	1008	1008	**144**
Teoriassa riittävä	1248	1248	**160**
Yleisesti katsotaan ehdottomaksi minimiksi	1776	1776	**192**
Minimitason takaava turvallisuus	2432	2432	**224**
Riittävä taso paitsi huippusalaisia asiakirj.	3248	3248	**256**
Riittävä myös top secret asiakirjoihin	15424	15424	**512**

VM.n tietoturvaryhmän suositus 2008

Johtopäätökset: RSA:n avainpituudet kasvavat liian suuriksi

Älykorteissa ja pienissä laitteissa em. on ongelma. Muisti ei riitä ja laskenta on hidasta.

Elliptic curve cryptosystem ECC on pienemmän avainpituuden vuoksi suositus tulevaisuuden julkisen avaimen salaukseksi.

ECC : Elliptisten käyrien salaus

Elliptic curve cryptography

Seuraavassa esitellään teoriaa elliptisten käyrien salausmenetelmistä, jotka tulevat olevaan julkisen avaimen järjestelmien seuraava sukupolvi. Matematiikan mahdollisesta vaikeudesta ei kannata välittää, koska sen osaamista ei tällä kurssilla vaadita. Tarkoituksena on yleiskäsityksen muodostuminen ECC:n toiminnasta.

Sykliset ryhmät xy-tason käyrillä

Jo 1700 – luvulta saakka on tiedetty, että toisen asteen käyrien pisteet ja siitä astelukua korkeampien, elliptisen käyrien pisteet muodostavat syklisiä ryhmiä. 1990-luvulla kryptologit alkoivat tutkia, voitaisiinko näitä ryhmiä käyttää samaan tapaan kuin ryhmää Z_p*.

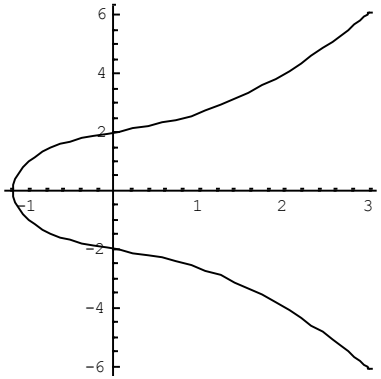
Voitaisiinko Diffie- Hellman avaimenvaihto tai Elgamal toteuttaa elliptisen käyrän pisteiden muodostamassa syklisessä ryhmässä ?

Onko näissä ryhmissä olemassa yhtä vaikeasti ratkaistavaa ongelmaa kuin on Diskreetin Logaritmin ongelma DLP ryhmässä Zp*, johon mm. DH:n turvallisuus perustuu?

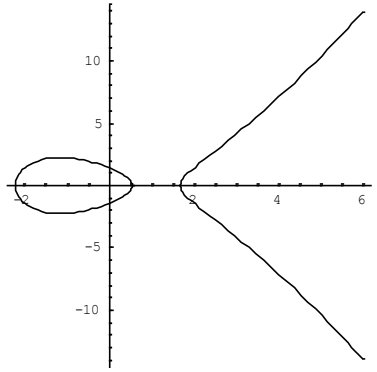
Elliptiset käyrät salauksessa

Käytetty muoto: $\mathbf{y^2 = x^3 + a\,x^2 + b}$

- Näyttävät seuraavilta



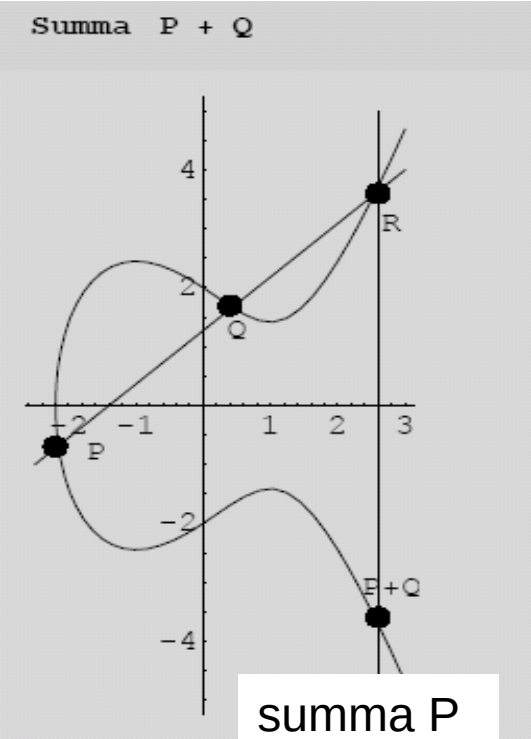
$y^2 = x^3 + 2x + 4$



$y^2 = x^3 - 4x + 2$

Käyrän pisteiden yhteenlasku

Geometrisesti summa on pisteiden P ja Q kautta piirretyn suoran ja käyrän leikkauspisteen peilikuvapiste



summa P + Q

Algebrallisesti pisteiden

$P = (x_1, y_1)$ ja $Q = (x_2, y_2)$

summa voidaan laskea kaavoilla

$P + Q = (x, y)$, missä

$$x = \lambda^2 - x_1 - x_2$$

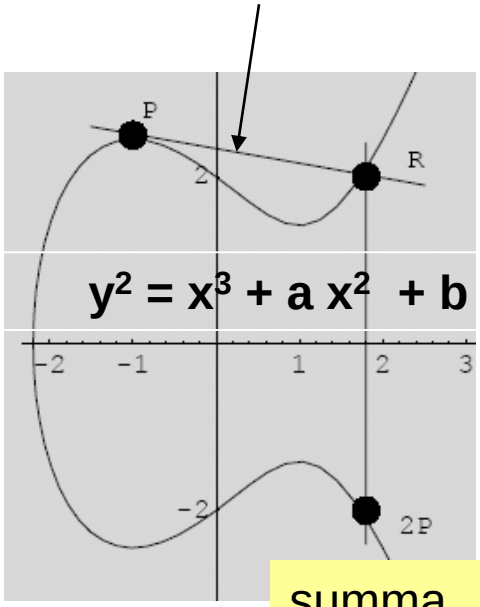
$$y = -y_1 + \lambda(x_1 - x)$$

ja λ (kuvan suoran kulmakerroin)

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

Pisteen monikerta P + P (= 2P)

Geometrisesti 2P on pisteeseen P piirretyn tangentin ja käyrän leikkauspisteen R peilikuva



summa P+P

Algebrallinen kaava pisteelle

2P =(x,y), missä P = (x_1, y_1)

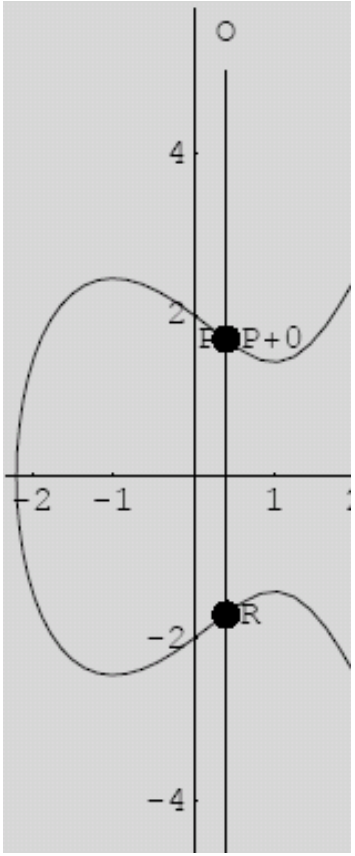
$$x = \lambda^2 - 2x_1$$

$$y = -y_1 + \lambda(x_1 - x)$$

Käyrän tangentin kulmakerroin λ

$$\lambda = \frac{3x_1^{\,2} + a}{2y_1}$$

Ryhmän neutraalialkio *O*

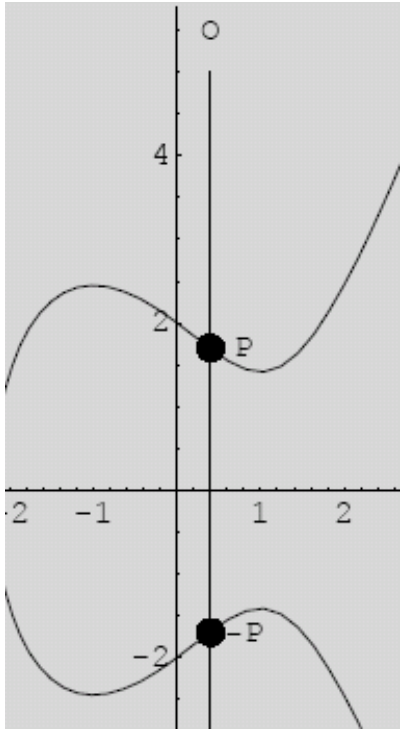


- Ryhmän määritelmän mukaan sillä pitää olla neutraalialkio. Elliptisen käyrän "Luku 0" määritellään pisteeksi O, jonka y –koordinaatti on ääretön.

Pisteelle *O* on voimassa

$$P + O = O + P = P$$

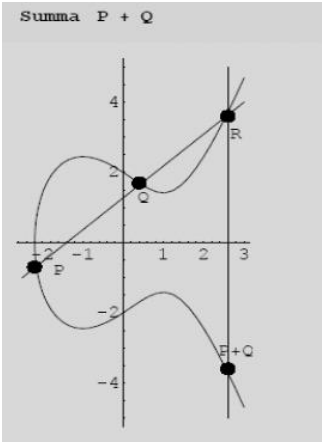
Alkioilla pitää olla käänteisalkio -P



Pisteen P käänteisalkiota merkitään –P. Se on pisteen P peilikuvapiste x – akselin toisella puolen

$$P + -P = -P + P = O$$

Muut ryhmäominaisuudet



- Seuraavat ominaisuudet ovat ilmeisiä käyrien muodon perusteella:

- Summa P + Q on olemassa kaikille käyrän pisteille

(kahden pisteen kautta kulkeva suora aina leikkaa käyrää kolmannessa pisteessä)

- P+(Q+R) = (P+Q)+R

(usean pisteen summassa yhteenlaskujen suoritusjärjestyksellä ei väliä)

- P + Q = Q + P

(summa on vaihdannainen => ryhmä on Abelin ryhmä)

Diskreetti elliptinen käyrä

Pisteet (x,y) ovat kokonaislukupareja, missä $x,y \in Z_q = \{0,1,\ldots, q-1\}$

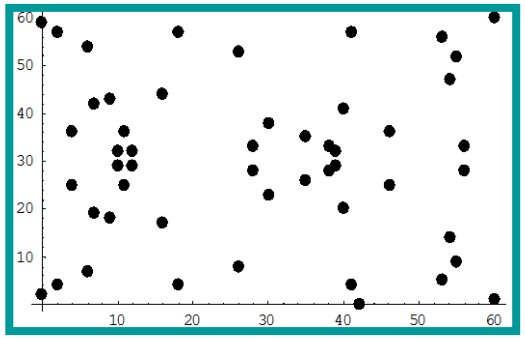
$y^2 = x^3 + a*x + b \pmod q$

Esimerkki: Käyrä $y^2=x^3+2x+4$ joukossa Z_{61} , ts. modulus q = 61

```
In[13]:= curve = {O};
         Do[If[Mod[y^2, 61] == Mod[x^3 + 2 * x + 4, 61], curve = Append[curve, {x, y}];], {x, 0, 60}, {y, 0, 60}]
         curve
         Print["Number of points ", Length[curve]]

Out[15]= {O, {0, 2}, {0, 59}, {2, 4}, {2, 57}, {4, 25}, {4, 36}, {6, 7}, {6, 54}, {7, 19}, {7, 42}, {9, 18},
          {9, 43}, {10, 29}, {10, 32}, {11, 25}, {11, 36}, {12, 29}, {12, 32}, {16, 17}, {16, 44}, {18, 4},
          {18, 57}, {26, 8}, {26, 53}, {28, 28}, {28, 33}, {30, 23}, {30, 38}, {35, 26}, {35, 35}, {38, 28},
          {38, 33}, {39, 29}, {39, 32}, {40, 20}, {40, 41}, {41, 4}, {41, 57}, {42, 0}, {46, 25}, {46, 36},
          {53, 5}, {53, 56}, {54, 14}, {54, 47}, {55, 9}, {55, 52}, {56, 28}, {56, 33}, {60, 1}, {60, 60}}

         Number of points 52
```



Ryhmässä on 52 alkiota, neutraalialkio O mukaan luettuna.

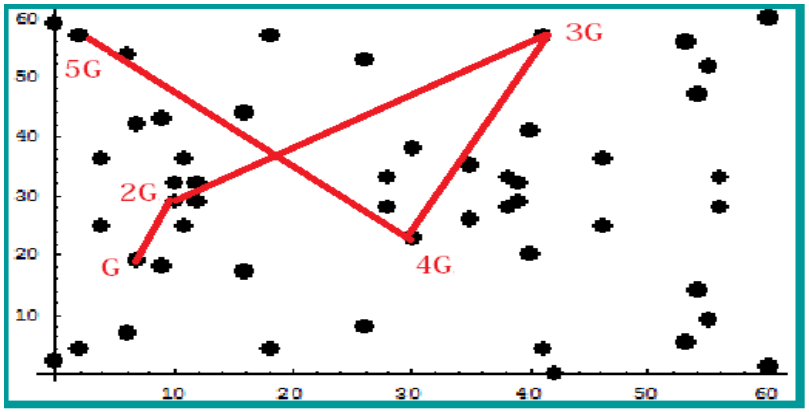
Diskreetti EC on syklinen ryhmä

Elliptiseltä käyrältä voidaan löytää generoiva alkio, joka generoi kaikki käyrän pisteet

Esimerkkikäyrämme $y^2=x^3+2x+4$ joukossa Z_{61} eräs generaattori on (7,19)

G = (7,19), 2G = (8,30) , ….., 52 G = O (viimeisenä neutraalialkio) tuottaa pisteet:

```
{{7, 19}, {8, 30}, {45, 51}, {31, 25}, {4, 58}, {36, 53}, {37, 51}, {26, 38}, {29, 20}, {40, 10},
 {5, 47}, {1, 19}, {53, 42}, {47, 22}, {12, 34}, {51, 32}, {42, 58}, {60, 30}, {14, 44}, {54, 31},
 {58, 16}, {15, 3}, {43, 53}, {23, 54}, {6, 48}, {35, 0}, {6, 13}, {23, 7}, {43, 8}, {15, 58}, {58, 45},
 {54, 30}, {14, 17}, {60, 31}, {42, 3}, {51, 29}, {12, 27}, {47, 39}, {53, 19}, {1, 42}, {5, 14},
 {40, 51}, {29, 41}, {26, 23}, {37, 10}, {36, 8}, {4, 3}, {31, 36}, {45, 10}, {8, 31}, {7, 42}, {O}}
```



syklinen ryhmä esimerkin käyrällä , generaattori (7,19)

EC:n käyttö salauksessa

* Perustuu matemaattiseen ongelmaan nimeltä “Diskreetin logaritmin ongelma elliptisillä käyrillä”, lyh. ECDLP (elliptic curve discrete logarithm problem)

ECDLP:

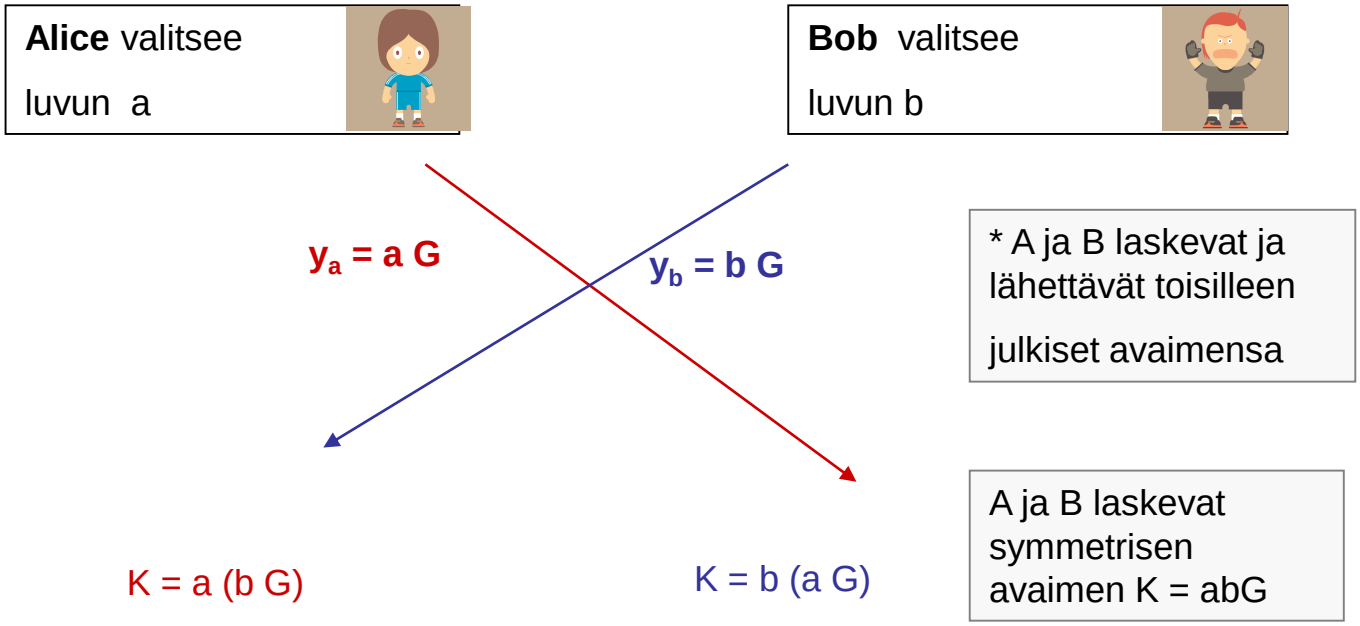
Jos tunnetaan käyrän mielivaltainen piste Y, joka on luonnollisesti generaattorin G jokin monikerta kG, on mahdotonta ratkaista kerrointa k yhtälöstä

$$Y = kG$$

mikäli käyrän modulus q ja parametrit a ja b ovat suuria eli käyrällä on riittävän monta pistettä.

Diffie Hellman elliptisellä käyrällä

Järjestelmäparametreina ovat käyrä ja sen generaattoripiste G.



Avain K on käyrän piste (x,y), josta saadaan symmetrinen AES – avain esim. ottamalla sen x –koordinaatista 128 ensimmäistä bittiä.

Elgamal salaus elliptisellä käyrällä

Bob valitsee eksponentin b ja laskee julkisen avaimensa Yb = b G

Alice salaa viestin M seuraavasti:

1. Alice luo satunnaisen luvun a ja laskee oman julkisen avaimen Ya = a G ja salausavaimen K = a (b G)

2. Alice koodaa viestin lukupareiksi M = (m1, m2)

2. Alice salaa viestin tulona C = K*M (tulo* määritellään (a,b)*(c,d)=(ac,bd))

ja lähettää Bobille salakirjoituksen ja oman julkisen avaimensa (C, Y_a)

Purku

1. Bob laskee salausavaimen K = b Ya = b (a G)

2. Bob laskee avaimen käänteisluvun $K^{-1} = (k_1^{-1}, k_2^{-1}) \bmod q$

3. Bob purkaa salauksen $K^{-1}*C = (K^{-1}*K)\ M = M$

ECC:n turvalliset avainpituudet

- Turvaraja parametrille q on n. 200 bittiä, mikä takaa saman turvallisuuden kuin RSA:n 2048 julkinen avain => ECC:ssä avainpituudet ovat n. 10% RSA:n avainpituuksista
- ECC onkin RSA:n seuraaja, kun RSA:n avainten turvallinen koko kasvaa sellaiseksi, että RSA:n käyttö tulee mahdottomaksi (avainten generointi hidastuu, avainten talletustila esim. sirukorteissa kasvaa, laskutoimitukset hidastuvat)
- ECC:hen perustuvia TLS- versioita sertifikaatteineen on jo markkinoilla, mutta ne eivät ole vielä saavuttamassa valta-asemaa.
- Elliptisten käyrien salaus tarjoaa “koko paketin” toimintoja :

  -käyttäjien autentikointi, symmetrisestä avaimesta sopiminen, tiedon salaus, digitaalinen allekirjoitus, sertifikaatit

Esimerkki standardoidusta Elliptisestä käyrästä : käyrä EC192

Käyrä EC192 on $y^2 = x^3 - 3x + b$

Modulus on 192 –bittinen luku q =
6277101735386680763835789423207666416083908700390324961279

Generoiva alkio on piste G =
**(602046282375688656758213480587526111916698976636884684818,
174050332293622031404857552280219410364023488927386650641)**

Parametri b =
2455155546008943817740293915197451784769108058161191238065

Huom! Kurssin web – sivulla http://web.lapinamk.fi/jouko.teeriaho/crypto15.htm on linkki pdf:ään (Brasovin esitys) jossa on laskettu testiesimerkki Diffie Hellman avaimenvaihdosta yo. käyrällä EC192.

6. Tiivistefunktiot

6.1 Tiivistefunktion määritelmä
6.2 Tiivistefunktion iteraatiorakenne
6.3 Tiivistefunktioiden turvallisuudesta
6.4 HMAC – message authentication code

7. Digitaalinen allekirjoitus

7.1 Digitaalisen allekirjoituksen periaate
7.2 shaRSA – digitaalinen allekirjoitus

Tiivistefunktiot (hash -functions)

Määr: Tiivistefunktiot ovat yksisuuntaisia funktioita, jotka tuottavat viestistä määrämittaisen tiivisteen (tarkistussumman)

Käyttö:
1. Tiiviste varmistaa **tiedonsiirron eheyden** (ts. että tieto ei ole muuttunut siirron aikana).
2. Palvelimien **salasanatiedostoihin** ei tallenneta itse salasanoja vaan **salasanatiivisteet**.
3. Digitaaliset allekirjoitukset lasketaan tiivisteen avulla
4. Tietoturvaohjelmien tarkistussummat kiintolevyn kansioista ovat tiivisteitä

Hyvän tiivisteen vaatimukset:
1.h(m) on **yksisuuntainen funktio**. Tiivisteestä ei ole mahdollista laskea taaksepäin itse viestiä.
2. Kun tunnetaan yhden viestin tiiviste h(m), on mahdotonta keksiä toista viestiä jolla olisi sama tiiviste. (**collision resistance 1**)
3. Ylipäänsä on mahdotonta luoda kahta viestiä m1 ja m2 joilla olisi sama tiiviste. (**collision resistance 2**)

Tunnettuja tiivisteitä

”Chinese Xiaoun Wang broke in 2005 almost all hash functions in sight - and paradoxally the community of cryptographers loved it”

- **MD5** **- ei suositella, murrettu**
- **SHA-1** - ei suositella, murrettu
- **SHA256** - murtamaton
- **SHA512** - murtamaton

Hash["Tämä on koeviesti, josta lasketaan tiiviste","SHA256"]
32134986657396586487192643852384396035054468584117070131813446773863278924851

Tiiviste paljastaa siirtovirheen

Alkuperäinen viesti ja sen tiiviste, joka on laskettu ennen lähetystä

```
Hash["Tämä on koeviesti, josta lasketaan tiiviste","MD5"]
301212701765142973053929339314878217659
```

Saapunut viesti, jossa yksi kirjain on muuttunut ja sen tiiviste,

```
Hash["Tämä on koeviesti, josta lasketaan tieviste","MD5"]
301633936839166156093425737348671573284
```

Tiivisteen muuttuminen paljastaa, että viestissä on tapahtunut siirtovirhe.

Turvallisen tiivisteen bittimäärä ”birthday paradox”

Turvallisen tiivisteen bittimäärän tulee olla 2 x lohkosalaimen turvallinen avainpituus, eli nykyisin 2 x 128 bittiä = 256 bittiä.

Selitys on ns. ”**syntymäpäiväparadoksi**” (birthday paradox).

Mikä on sellainen koululuokan vähimmäiskoko, että luokassa on yli 50% todennäköisyydellä kaksi henkilöä jolla on sama syntymäpäivä ? Vastaus: 23 henkeä

$$P = 1 - \prod_{k=1}^{23} \frac{365 - k + 1}{365} = 0.51$$

Voidaan osoittaa samaan tapaan, että jos meillä on n mahdollista tiivistettä, noin $\sqrt{n}$ tiivisteen joukoissa on yli 50% todennäköisyydellä ”törmäyksiä”

MAC = message authentication code

- MAC = tiiviste, jossa käytetään symmetristä salausavainta
- MAC:llä voidaan taata paitsi viestin muuttumattomuus, myös lähettäjä (koska avain tiivistetään viestin mukana)

HMAC

= tavallisin MAC: keyed hash message authentication code lasketaan viestistä m ja symmetrisesta avaimesta seuraavasti

$$HMAC(K, m) = sha(K \oplus opad || sha(K \oplus ipad || m)$$

⊕ = XOR summa

K = symmetrinen avain

Ipad = 5c5c5c5c

Opad = 363636

|| liittäminen

1101 ⊕ 0101 = 1000

Ipad ja Opad ovat vakioita

Liittäminen = kahden merkkijonon yhdistämistä

“auto”||”mat” = “automat”

MAC takaa lähettäjän autentikoinnin + viestin eheyden

Se vastaa tässä mielessä digitaalista allekirjoitusta. Esim. Verkkopankkiyhteydessä datapakettien perässä on usein HMAC tiiviste

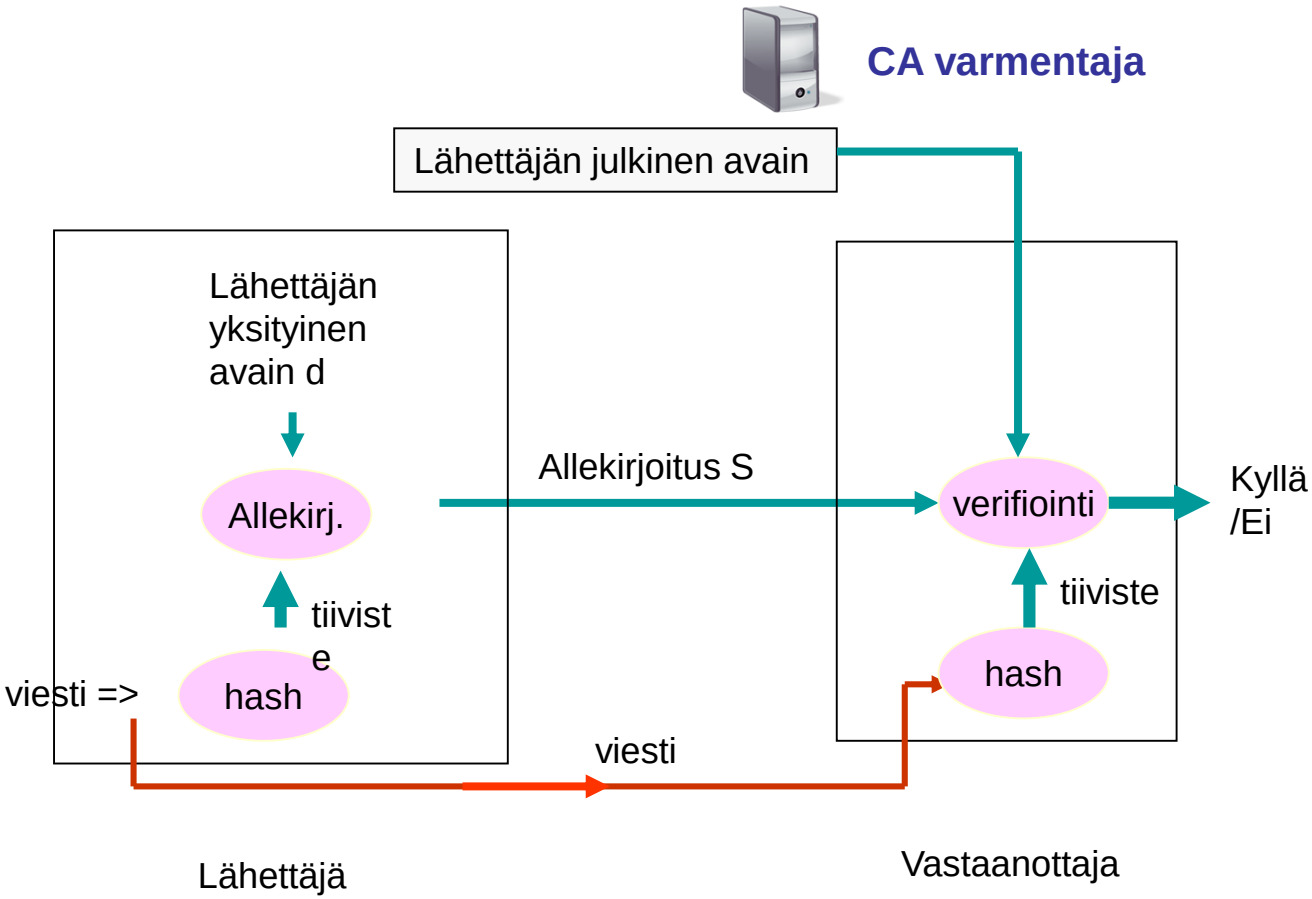
sha-HMAC – funktiota käytettään yleisesti pseudosatunnaisluku-generaattorina laitteissa, jotka tuottavat kertakäyttösalasanoja esim. verkkopankkisovelluksiin. Tästä on kerrottu enemmän osassa kalvot5.pdf

Digitaalinen allekirjoitus (digital signature)

- Tarkoitus on varmistaa, että 1) viesti on muuttumaton, 2) lähettäjä on varmennettu
- Digitaalisessa allekirjoituksessa käytetään tiivistettä ja julkisen avaimen salausta
- Tyypillinen yhdistelmä on RSA ja SHA -tiiviste

”**sha1RSA”** ja ”sha256RSA” digitaaliset allekirjoitukset ovat SSL protokollassa yleisiä

Digitaalinen allekirjoitus kaaviona



Digitaalinen allekirjoitus

Digitaalinen allekirjoitus = viestin tiiviste salattuna lähettäjän yksityisellä avaimella

Se on siten luku joka lähetetään viestin mukana.

Vastaanottaja verifioi seuraavasti:

1) purkaa digitaalisen allekirjoituksen lähettäjän julkisella avaimella ja saa viestin tiivisteen.
2) Hän laskee itse viestistä toisen tiivisteen.

Jos tiivisteet täsmäävät, lähettäjä on varmennettu ja viesti muuttumaton

RSA digitaalinen allekirjoitus kaavana:

$$S = hash(m)^d \bmod n$$

SSL- protokollassa yleisimmät: sha1RSA ja sha256RSA

Esim. Laske digitaalinen allekirjoitus shaRSA viestille ”Tämä on koeviesti”. Viestin lähettäjän RSA- avaimet ovat

n=1145765750978426008374361826929901435657745332732160379602007
d = 336699919100561613978086186807347884314069552804538190944273
(e = 65537)

1. Luodaan viestin tiiviste.

SHA "Tämä on koeviesti"

integer form	912 642 116 448 435 246 138 124 989 263 395 491 973 387 563 ... 505
hexadecimal form	9fdc 49b2 84e2 697f 4168 6c5c 6901 24df 4d24 b1f1

2. Luodaan allekirjoitus käyttämällä tiivisteeseen lähettäjän yksityistä avainta

Input:

$$912\,642\,116\,448\,435\,246\,138\,124\,989\,263\,395\,491\,973\,387\,563\,505^{336\,699\,919\,100\,561\,613\,978\,086\,186\,807\,347\,884\,314\,069\,552\,804\,538\,190\,944\,273} \bmod 1\,145\,765\,750\,978\,426\,008\,374\,361\,826\,929\,901\,435\,657\,745\,332\,732\,160\,379\,602\,007$$

Result:

646 724 005 488 510 290 167 295 062 318 966 956 372 378 831 841 439 912 526 782 <= Viestin dig. allekirjoitus

Esimerkki jatkuu: Miten allekirjoitetun viestin saaja verifioi allekirjoituksen?

1. Vastaanottaja purkaa allekirjoituksesta viestin tiivisteen lähettäjän julkisella avaimella (n,e).

Input:

$$646\,724\,005\,488\,510\,290\,167\,295\,062\,318\,966\,956\,372\,378\,831\,841\,439\,912\,526\,782^{65\,537} \bmod 1\,145\,765\,750\,978\,426\,008\,374\,361\,826\,929\,901\,435\,657\,745\,332\,732\,160\,379\,602\,007$$

Result:

912 642 116 448 435 246 138 124 989 263 395 491 973 387 563 505

2. Vastaanottaja laskee viestiosasta tiivisteen.

3. Verrataan kohdan 1 ja 2 tuloksia: Luvut täsmäävät => lähettäjä on todennettu ja viesti on tullut muuttumatta perille.