Aiheet:

# Modernit salausmenetelmät

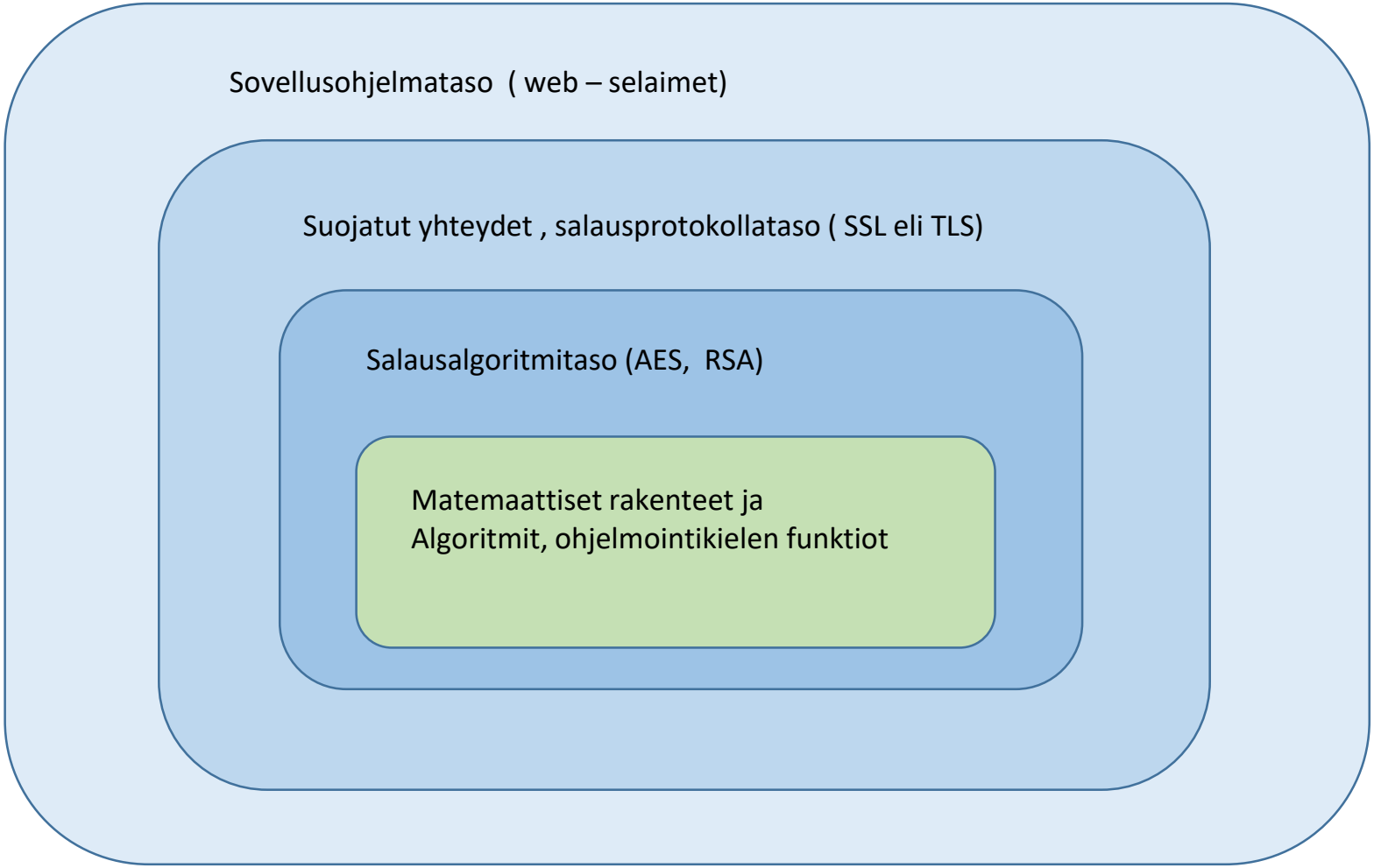
**Salaus eri tasoilta tarkasteltuna**

- Sovellusohjelmataso
- Protokollataso
- Salausalgoritmitaso
- Matemaattisten rakenteiden taso

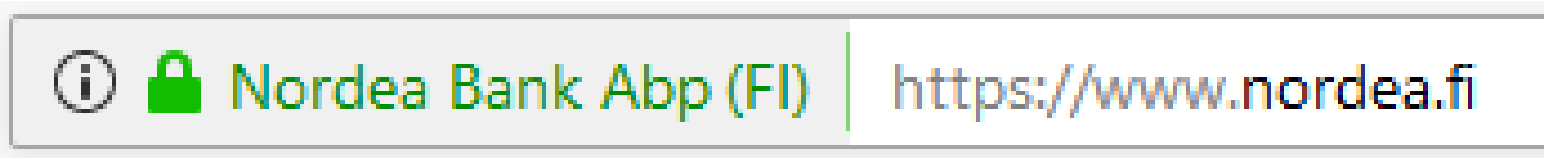
**Synkroninen jonosalaus**

**Lohkosalaus**

# Salaus eri tasoilla



# Sovellusohjelmataso (esim. selain)




Kun selaimen URL – rivillä on vihreä lukon kuva, se merkitsee, että yhteys on suojattu ja palvelin on varmennettu.  Salausohjelmistona on usemmiten TLS ,  jonka varhaiset versiot tunnettiin nimellä SSL.

# Salausprotokollataso ( TLS)

**Klikkaamalla lukon kuvaa löytyy lisätietoa yhteydestä:**

**Tekniset tiedot**

Yhteys salattu (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, 256-bittinen avain, TLS 1.2)

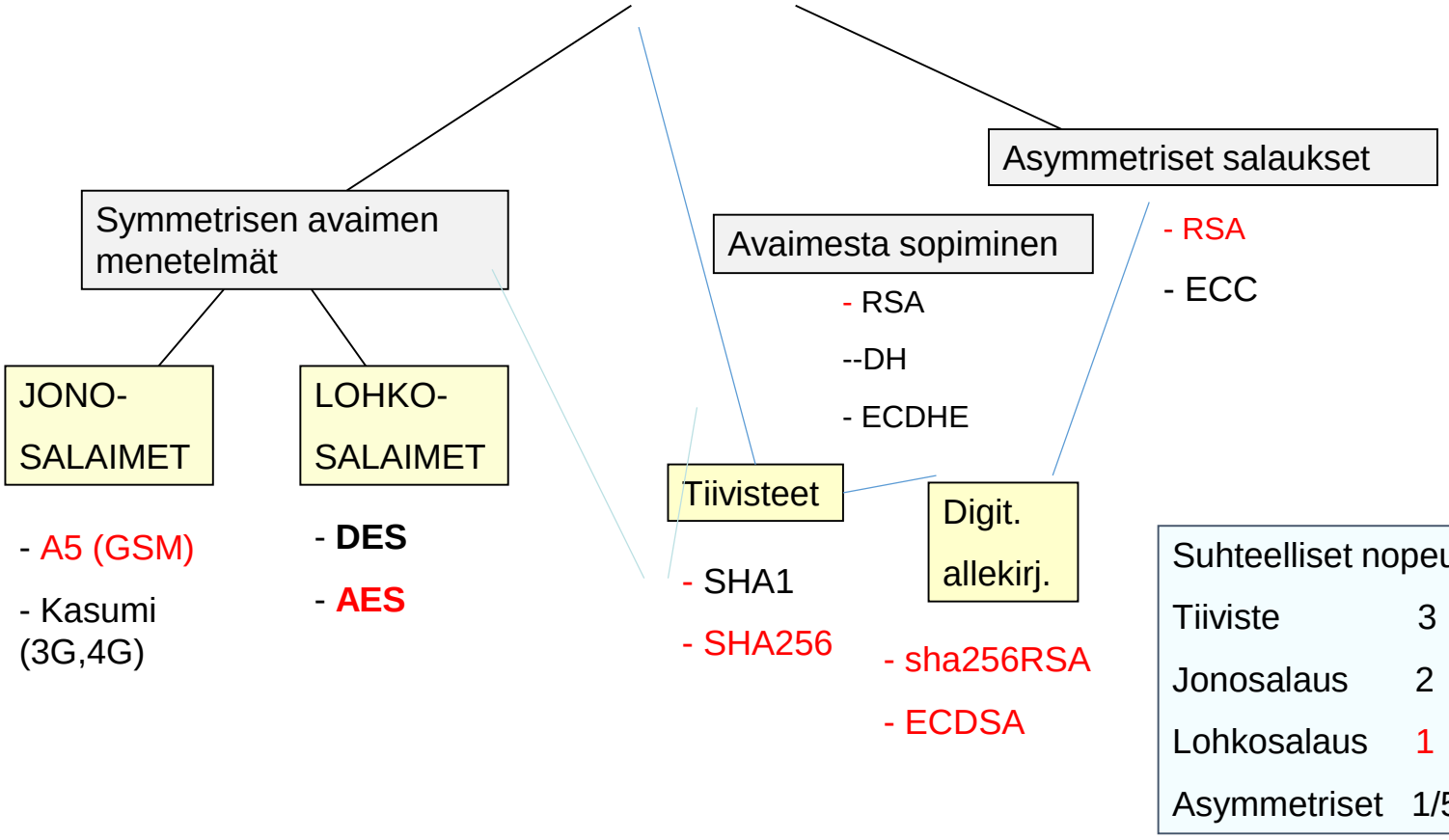
**Salausohjelmistona on TLS ver 1.2** , joka on tyypillinen **hybridisalausohjelmisto**, joka käyttää useita eri tyyppisiä salausalgoritmeja hyödyntäen eri algoritmien parhaita puolia

# Salausalgoritmitaso

**Alla on Nordean TLS – yhteyden algoritmit ja niiden tehtävät**

| **Tehtävä salausprotokollassa** | **Salausalgoritmi** |
|---|---|
| Palvelimen autentikointi | **RSA** |
| Istuntoavaimesta sopiminen | **ECDHE** |
| Tiedonsiirron salaus | **AES256 GCM-moodi** |
| Digitaalinen allekirjoitus | **sha384RSA** |

# Salausalgoritmityypit

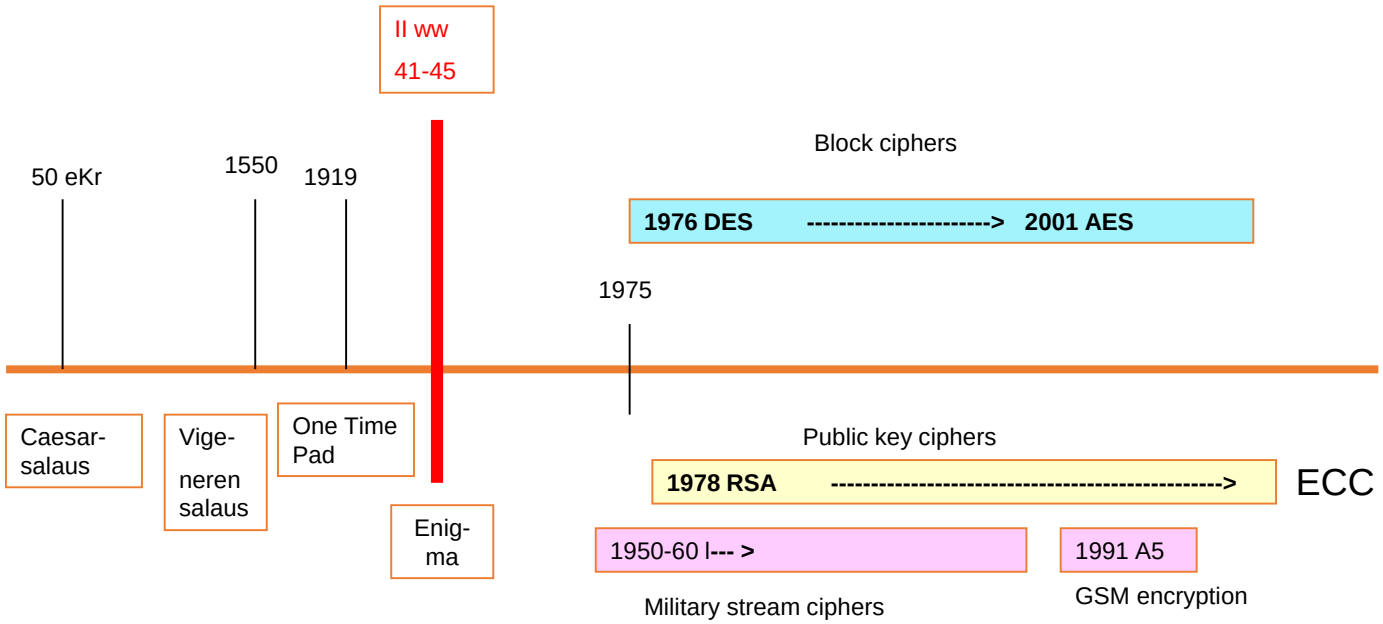


# 4. Matemaattisten rakenteiden taso

TLS – yhteyden algoritmit sisältävät runsaasti matematiikkaa:

| | |
|---|---|
| **1. Fermat'n ja Eulerin teoreemat, Laajennettu Eukleideen algoritmi** | |
| **2. Satunnaislukujen generointi** | **pseudosatunnaisuuden käsite** |
| **3. Alkulukujen generointi, alkulukutestit** | **Fermat'n testi, Rabin- Miller testi** |
| **4. Jakojäännösaritmetiikka** | **mod funktio** |
| **5. Nopea potenssiin korotus mod n** | **ns. Powermod - algoritmi** |
| **6. Käänteislukujen laskeminen mod n** | **Eucleideen laajennettu algoritmi** |
| **7. Generoivat alkiot joukossa $Z_p^*$** | **ryhmäteoriaa** |
| **8. Elliptiset käyrät** | |

# Modernien salausalgoritmien aikajana

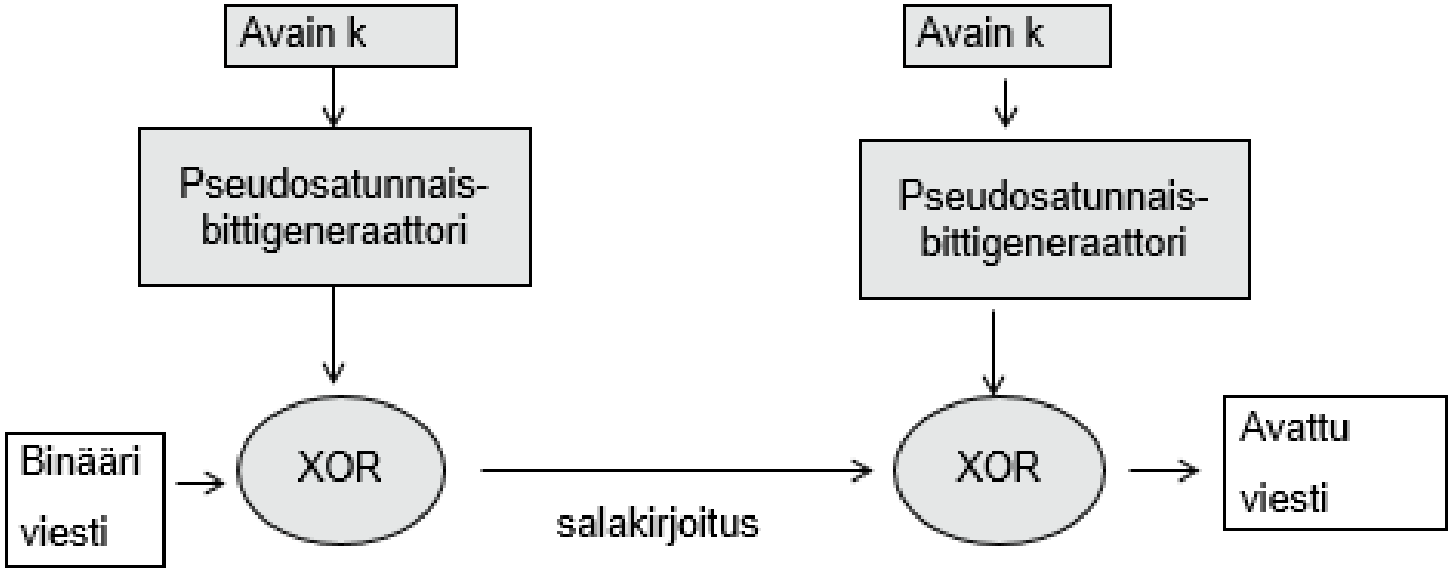



Salausmenetelmät kehittyvät konservatiivisesti. Algoritmeja ei yleensä vaihdeta ennen kuin on pakko. RSA on ollut standardi jo 39 vuotta julkisen avaimen salauksissa. Lohkosalausstandardia on vaihdettu vain kerran viimeisen 40 vuoden aikana.

# Symmetriset salausalgoritmit

1. **Synkroninen jonosalaus**
   - GSM – salaus A5
   - LFSR - pseudosatunnaisbittien generaattorina
   - 3G ja 4G salaus Kasumi
2. **Lohkosalaus**
   - Diffuusio ja konfuusioperitaatteet
   - Permutaatio-substituutioverkot
   - Lohkosalauksen iteraatiokaavio
   - Lohkosalaimen toimintamoodit
   - DES ja Feistelin silmukka
   - AES salausstandardi

# Synkroninen jonosalaus




Esim. **GSM puhelujen salaus A5 (1991)** on synkroninen jonosalaus. Puhelin sisältää mikropiirin, joka tuottaa avaimeen perustuvan bittijonon, joka yhdistetään XOR piirin kautta binäärimuotoiseksi muunnettuun puheeseen. Periaate on sama kuin One Time Pad salauksessa lukuunottamatta sitä, että avain ei ole täysin satunnainen vaan tuotetaan mikropiirillä tai ohjelmalla.

# GSM salaus A5

Perusesimerkki synkronisesta lohkosalauksesta on GSM –salausalgoritmi A5. Sen avainpituus on 64 bittiä, joka on liian lyhyt ollakseen turvallinen.
GSM –puhelimessa pseudosatunnaista bittijonoa tuotetaan 1950 – luvulta peräisin olevilla mikropiireillä nimeltä LFSR : Linear Feédback Shift Register.

Suomi oli 1990 – luvun alussa maailman kärjessä mobiilitekniikassa:
Kuva esittää maailman ensimmäistä GSM-puhelua 1.7.1991 Helsingistä Tampereelle, soittajana Suomen silloinen pääministeri Harri Holkeri, operaattori oli tuolloin Radiolinja.

# Pseudosatunnaisuuden käsite

Synkroninen jonosalain tarvitsee mikropiirin, joka tuottaa bittijonoa, joka on täysin deterministinen, mutta joka täyttää tilastollisen satunnaisuuden kriteerit. Tällaista jonoa nimitetään **PN – jonoksi** (pseudo noise)

Käsitteen määritteli 1950 –luvulla **Samuel W Golomb**.

**PN jonon vaatimukset:**

**G1**. Jonossa on oltava n. 50% ykkösiä 50% nollia

**G2**. Suhteelliset esiintymistodennäköisyydet samaa bittiä toistaville lohkoille

| | | | |
|---|---|---|---|
| 0**1**0 | **tai** | 1**0**1 | yhden bitin lohko |
| 0**11**0 | | 1**00**1 | kahden saman bitin lohko |
| 0**111**0 | | 1**000**1 | kolmen saman bitin lohko |
| 0**1111**0 | | 1**0000**1 | neljän saman bitin lohko |

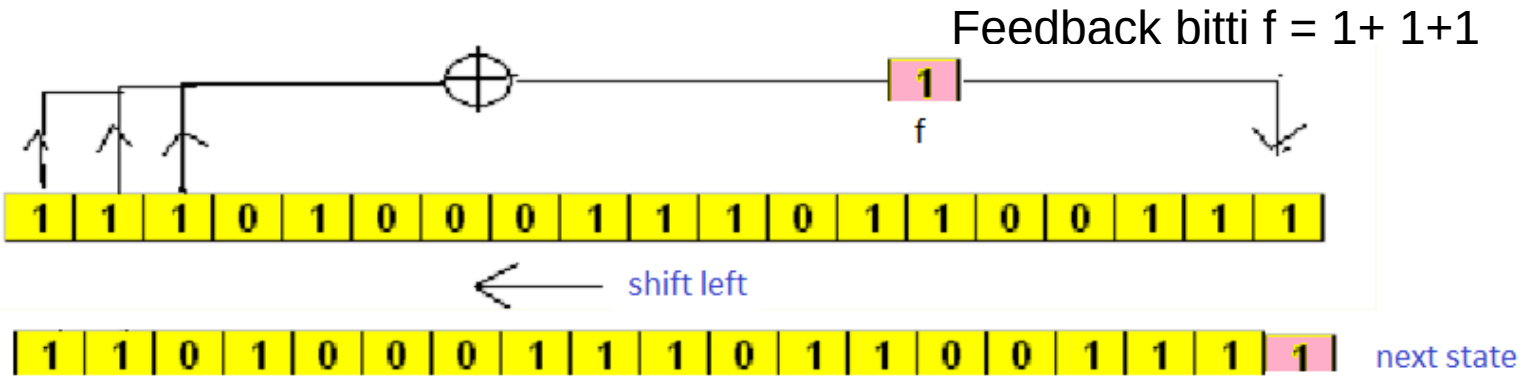
**suhtautuvat toisiinsa kuten 1/2 , 1/4 , 1/8 , 1/16**

**G3**. Kun jonoon tehdään rotaatio ja uutta ja alkuperäistä bittijonoa verrataan, niin yhtäläisyyksiä ja eroavaisuuksia pitäisi olla likimain sama määrä. Tätä ominaisuutta mittaa ns. autokorrelaatiokerroin, jonka pitäisi olla lähellä nollaa kaikille jonon rotaatioille. Tällöin jonosta ei löydy sisäistä periodisuutta.

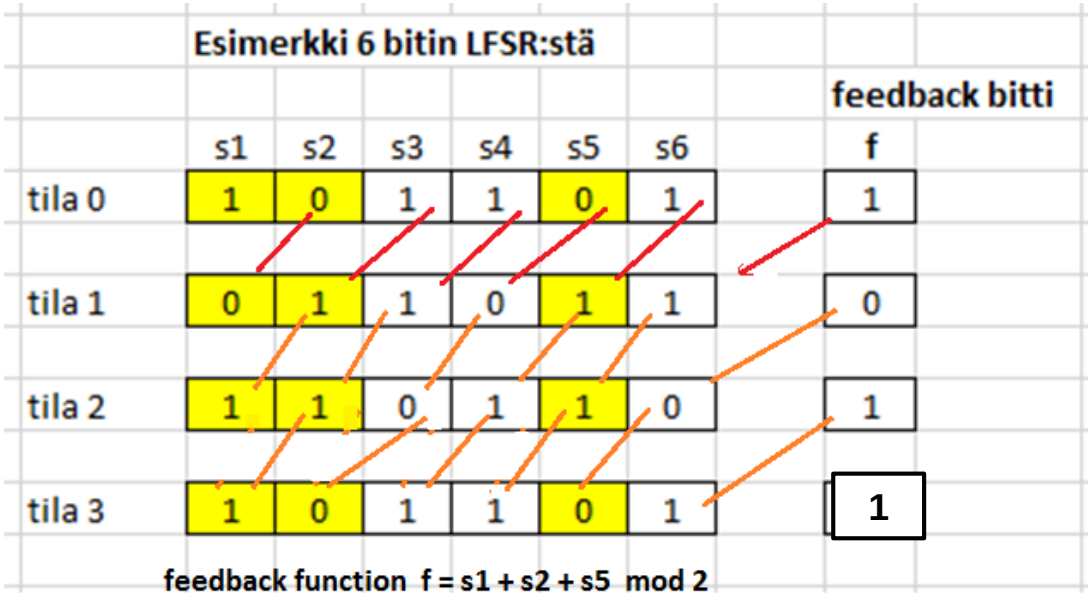
# LFSR - linear feedback shift register

**= mikropiiri, jota on käytetty mm. taskulaskimien RAND toiminnoissa**

* LFSR on n bitin rekisteri, joka sisältää siten n bitin jonon.
- Se toimii siten, että bitit siirtyvät jokaisella kellopulssijaksolla yhden pykälän vasemmalle ja uudeksi oikeanpuolimmaiseksi bitiksi tulee feedback – bitti , joka lasketaan sovituista rekisterin biteistä arvoista XOR – yhteenlaskulla juuri ennen siirtymää.



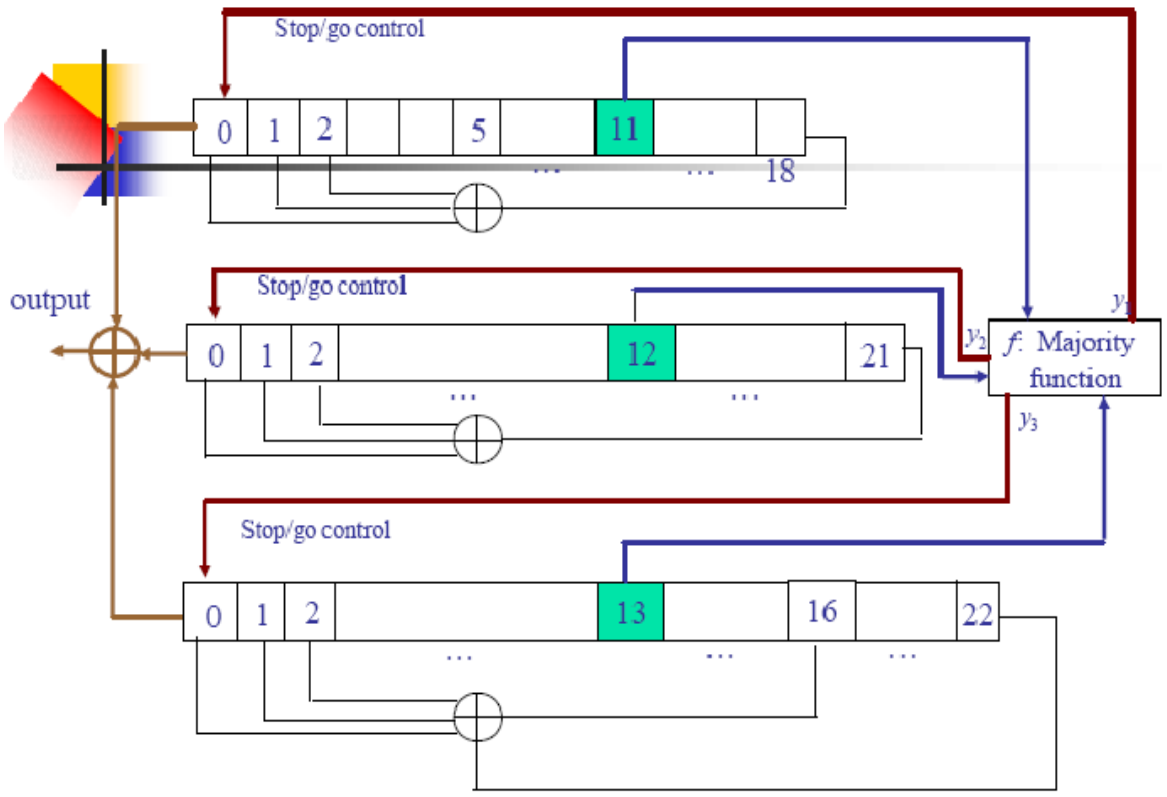
# Esim. LFSR:n toiminnasta




Feedback -bitti lasketaan kuvassa keltaisista biteistä $s_1$, $s_2$ ja $s_5$ kaavalla
$f = s_1+s_2+s_5 \bmod 2$
ennen bittien siirtymää vasemmalle.

**PN jonoa saadaan LFSR rekisterin S1 – biteistä, kun LFSR siirtyy tilasta toiseen.**

# GSM puhelimen A5 PN -generaattori




Majority funktio: Jos kaikki vihreät bitit samoja, kaikki rekisterit liikkuvat, jos kaksi kolmesta samaa bittiä, ne rekisterit liikkuvat eteenpäin.

**GSM - puhelimessa on 3 LFSR-rekisteriä,** joiden pituudet ovat 19, 22 ja 23 bittiä

**Rekisterien alkutila toimii symmetrisenä salausavaimena,** jonka pituus on 19 + 22 + 23 = 64 bittiä.

**Jokaisella kellopulssilla generaattori tuottaa yhden pseudosatunnaisbitin**, joka on rekisterien nollabittien XOR-summa.

**Murtamista vaikeuttamaan on kehitetty vielä majority funktio** f, joka lasketaan kuvan vihreistä biteistä. f määrää mitkä rekistereistä liikkuvat kellopulssin aikana, mitkä jäävät paikalleen.

# A5 PN-generaattorin bittivirta täyttää tilastollisen satunnaisuuden vaatimukset

0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1,
1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0,
1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1,
1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0,
0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0,
1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,
1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0,
1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0,

Tämä A5:n tuottama PN jono yhdistetään puhelun digitaaliseen signaaliin bitti bitiltä käyttäen XOR – yhteenlaskua. Näin syntyy salattu signaali.

Operaattorilla, joka välittää puhelun, on samanlainen A5 generaattori. Sen alkuavain on sama => se tuottaa saman PN- jonon ja salattu signaali puretaan alkuperäiseksi salaamattomaksi signaaliksi, joka jatkaa kulkuaan valokaapelia pitkin.

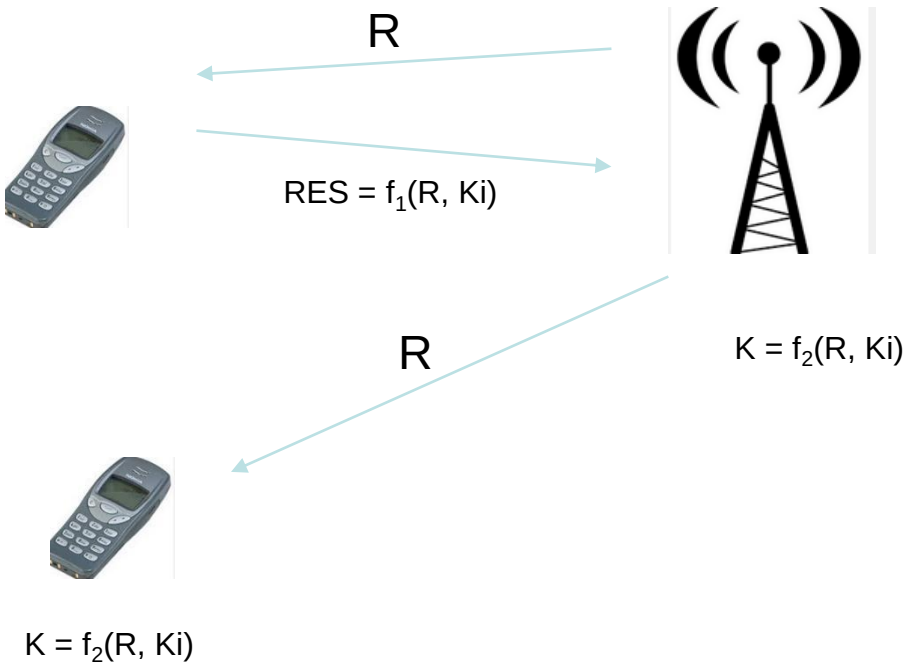
**TURVALLISUUS**

1. A5 :n avainpituus 64 on liian pieni, se voidaan murtaa Brute Forcella
2. Efektiivinen PN jonon jaksonpituus $4/3*2^{23}$
3. Brute force Pentium III PC:llä vie n. 250 h (vanha tieto)
4. GSM signaali on salattu vain maston ja puhelimen välillä – kaapelissa se on salaamaton.

# Mobiililaitteen todennus, avaimesta sopiminen GSM - puhelimessa

**Autentikointi A3.** Operaattori lähettää puhelimelle satunnaisluvun R. Puhelin laskee siitä ja SIM- avaimesta Ki vastauksen RES, jonka lähettää operaattorille. Operaattori laskee itsekin luvun RES samalla tavalla. Jos luvut täsmäävät, on käyttäjä varmennettu.

**Avaimesta sopiminen A8.** Operaattori lähettää puhelimelle satunnaisluvun R. Puhelin laskee siitä ja SIM- avaimesta Ki 64 bitin salausavaimen K, joka tulee kolmen LFSR:n muodostaman PN generaattorin alkutilaksi.



# 3G ja 4G salaus Kasumi

3G ja 4G verkoissa käytössä on salaus nimeltä Kasumi. Se on japanilaista alkuperää ja kehitetty **Misty**- salausalgoritmista. Salausalgoritmi on kehitetty perinteisellä tekniikalla, jossa sovelletaan samoja ideoita (Feistelin silmukka) kuin ensimmäisessä lohkosalausstandardissa  DES:ssä, joka lanseerattiin jo v. 1976.

- Kasumi -algoritmi tuottaa 3G ja 4G puhelimissa pseudosatunnaista bittivirtaa samoin kuin A5 – algoritmi  GSM puheluissa.

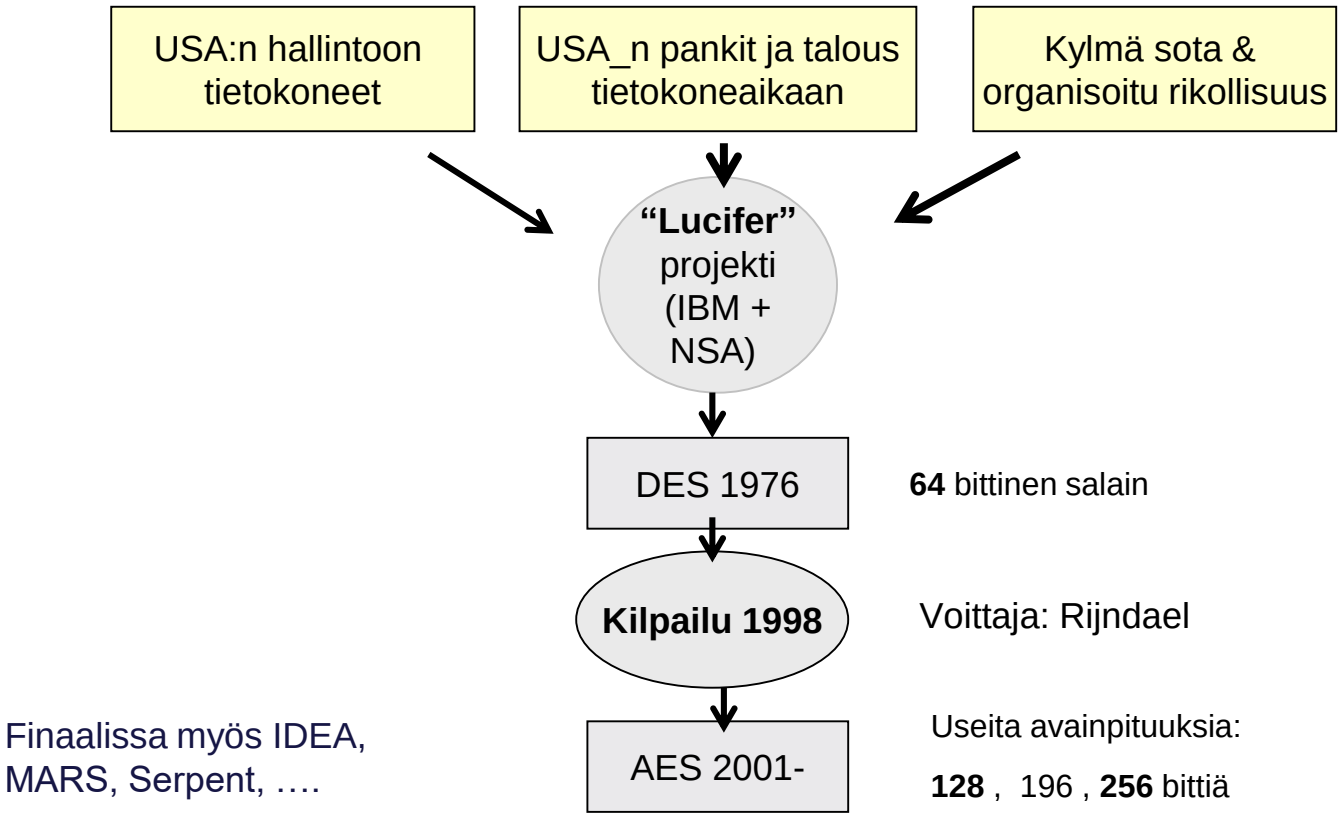
- **Eroja GSM – salaukseen:**
  * Avainpituus on 128 bittiä, jota voidaan pitää turvallisena
  - 3G ja 4G puheluissa käytetään kaksisuuntaista todennusta:  siinä sekä puhelin, että masto joutuvat todistamaan aitoutensa.  Tämän pitäisi estää valetukiasemien toiminta verkossa.
  - Kasumi kestää Brute Force hyökkäyksiä hyvin
  - Kasumistakin on löydetty heikkouksia. Toistaiseksi kuitenkin kehitetyt hyökkäykset ovat sellaisia, ettei niistä ole suurta uhkaa puhelimen käyttäjille.

# **Lohkosalaimet** (block ciphers)

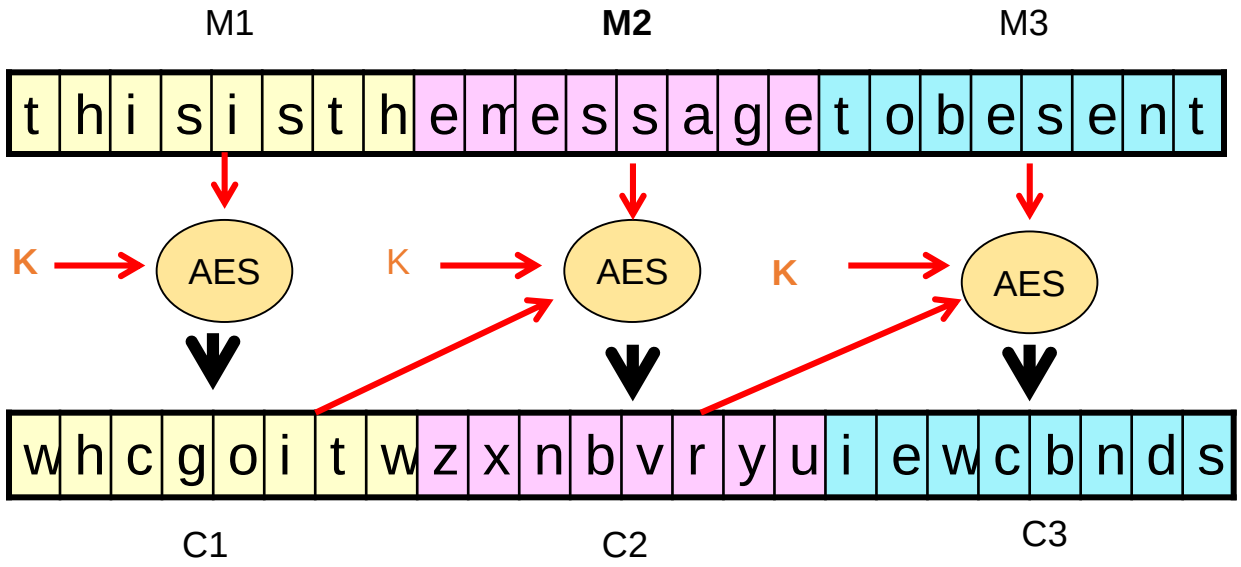
- Lohkosalaus on nopea (>1Gbs), symmetrinen salaus, jossa viesti salataan yleisimmin 128 bitin lohkoissa.
- Salausohjelmistojen se osa, joka tekee suurimman työn eli huolehtii siirrettävän datan salauksesta luotettavasti ja tehokkaasti
- Nykyinen lohkosalausstandardi on AES (2001…)

  (advanced encryption standard)

# Lohkosalauksen historiaa



# Kaavio lohkosalauksen toiminnasta




**Periaate: Viesti M jaetaan lohkoihin m1,m2,m3, … ( nykyisin 128 bittiä)**

**Avain k vähintään 128 bittinen. Salakirjoitus on jono c1, c2, c3, c4**

**Kuvassa on käytettä CBC moodia, jossa edellisen lohkon tulos viedään syötteenä seuraavaan lohkoon**

# Vaatimukset lohkosalaimille

- Vaaditaan, että ne toimivat sekä softana, että kovona (salaussiruina)
- Nopeita ja luotettavia
- Käytetään suurten tietomäärien salaukseen (tietoliikenne ja tiedostot)
- DES:n (1976-2001) toiminta oli tarkoitus pitää salassa, AES (2001-) on julkinen **(Kerckhoffin periaate)**
- **Teoreettinen minimiavainpituus** 80 bittiä , käytännössä minimiavainpituus on **128 bittiä**
- **Salaus on yhtä vahva kuin sen avain:** esim. TLS-yhteyksissä (pankit), istuntoavain generoidaan joka kerta satunnaislukugeneraattorilla

# Diffuusio ja konfuusio

Claude Shannon
Informaatioteorian isä

Lohkolauksen toimina nojaa Claude Shannonin kahteen periaatteeseen: diffuusioon ja konfuusioon.

**Diffuusio:**
Muutettaessa viestin yhtä bittiä, tulisi todennäköisyyden mielivaltaisen salakirjoituslohkon bitin  muuttumiselle olla 50%.

**Konfuusioperiaate:**
Jokaisen salakirjoitetun lohkon bitin tulee riippua useista salausavaimen osista. Jokaisen salakirjoitetun lohkon bitin tulisi riippua kaikista salausavaimen biteistä ja yhdenkin salausavaimen bitin muuttumisen pitäisi muuttaa salakirjoituslohkoa täysin toiseksi.

**Salatun viestin tulisi täyttä tilastollisen satunnaisuuden vaatimukset.**

# Permutaatio – Substituutio verkot

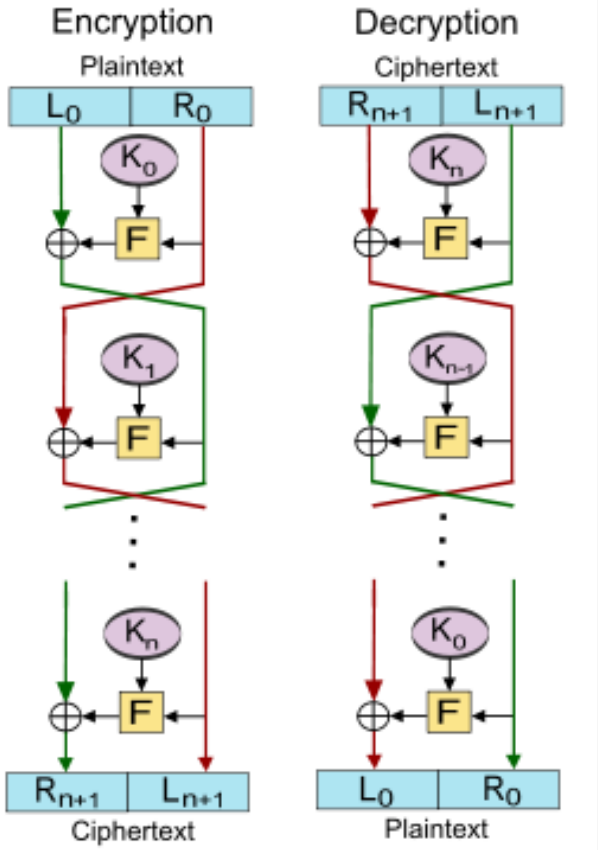
Yleisin tapa toteuttaa diffuusio ja konfuusioperiaatteet, on käyttää permutaatio -substituutioverkkoja.

**Substituutio** = tiettyjen bittilohkojen korvaamista toisilla bittilohkoille, useimmiten korvaustaulukkojen eli SBOX:ien määrittämällä tavalla.

**Permutaatio** = lohkon bittien uudelleen järjestäminen jonkin algoritmin mukaan.

Tuloksena jokainen säännönmukaisuus salattavassa bittijonossa hajoitetaan läpi koko salakirjoituslohkon niin, että salakirjoituksesta on mahdotonta havaita mitään säännönmukaisuutta.

# Feistelin salaus




Many block ciphers, such as DES and Blowfish utilize structures known as *Feistel ciphers*

IBM:n insinööri Feistel toteutti PS verkon 1960 – luvun lopulla käyttäen permutaatioita ja SBOX: eiksi nimitettyjä substituutioita.
Feistelin tuli ensimmäiseen lohkosalausstandardiin DES, myöhemmin sitä on käytetty Blowfish – nimisessä 128 – bittisessä salauksessa ja **nykyään 3G ja 4G -verkkojen 128 bittisessä Kasumi – salauksessa.**

**Feistelin salauksessa** (DES) on 16 kierrosta, joissa käytetään salausavaimesta K rotaatioilla saatuja aliavaimia Ki

Viestilohkon biteille suoritetaan aloituspermutaatio, jonka jälkeen lohko jaetaan kahteen puolikkaaseen.

Kullakin kierroksella tapahtuu seuraavaa:

- Oikea puolikas korvaa vasemman puolikkaan
- Uusi oikea puolikas lasketaan kierroksen aliavaimesta ja vasemmanpuolimmaisesta lohkosta funktiolla, joissa käytetään kahdeksaa substituutiotaulukkoa , SBOX:ia.

Lopuksi tehdään lopetuspermutaatio salakirjoituslohkon biteille. Tuloksena on pseudosatunnainen bittijono.

# Lohkosalauksen toimintamoodit

*Lohkosalaimia voidaan käyttää mm. seuraavissa moodeissa*

**1) ECB  Electronic Codebook**

Lohkot salataan toisistaan riippumatta. Jos sama lohko toistuu, salakirjoituslohko on sama.  Tämä moodi ei ole turvallinen.

**2) CBC Cipher Block Chaining (**TLS-yhteyksissä näihin päiviin saakka **)**

Edellisen lohkon salakirjoitus toimii syötteenä seuraavan lohkon salaukselle.

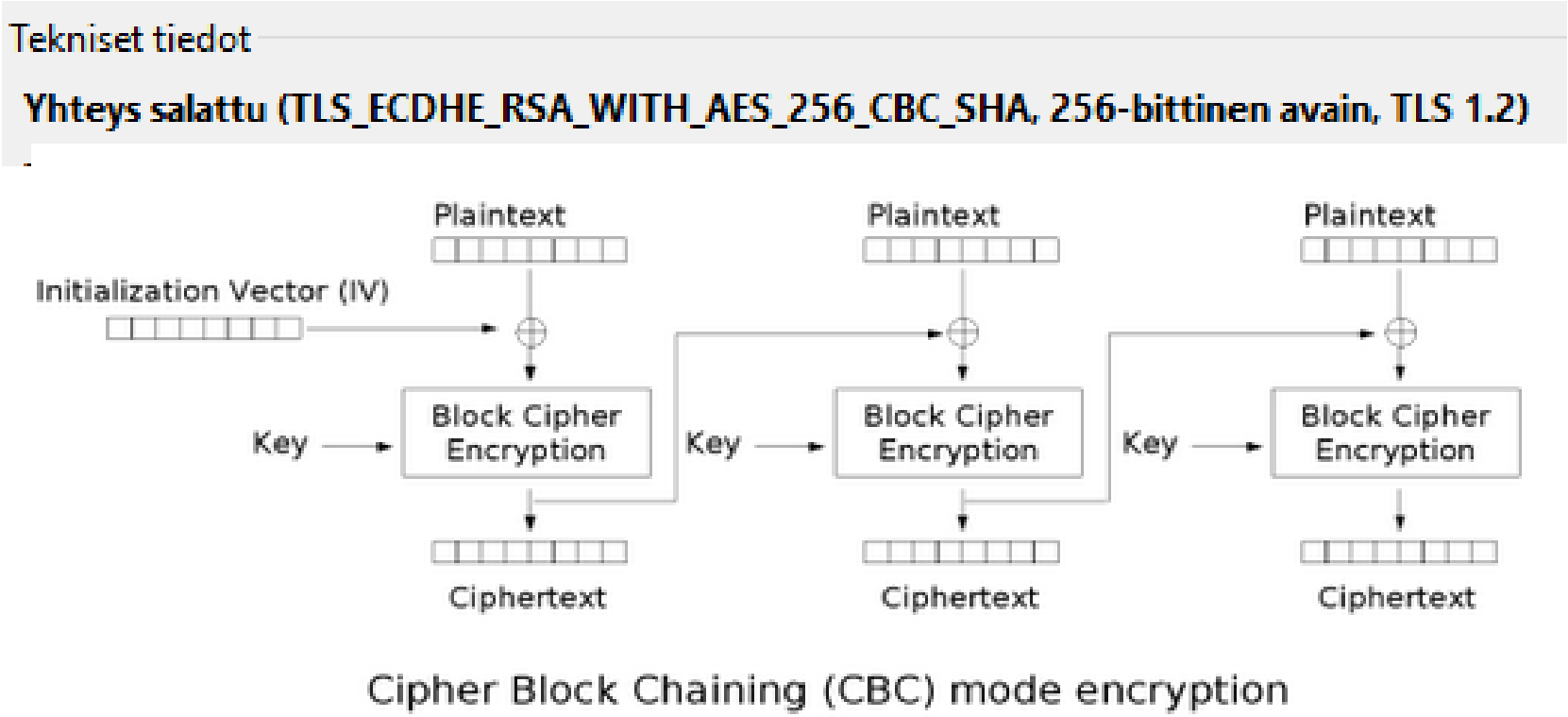
Jokainen lohko siten vailuttaa seuraavien lohkojen salakirjoituksiin Identtisillä lohkoilla on siten eri salakirjoituksen eri kohdissa salakirjoitusta.  TLS – yhteyksissä yleinen tapa on käyttää AES salausta CBC moodissa

3) **GCM  Galois Counter Mode** (ilmestynyt äskettäin TLS-yhteyksiin)

- Otettu käyttöön mm. suom. verkkopankeissa 2017 lopulla. Turvallisin

# AES:n käyttömoodeista

LapinAMK:n sähköposti käyttää AES lohkoasalainta CBC moodissa:




CBC –moodissa salattava data salataan 128 bitin lohkoissa. Salakirjoituslohkon syötteenä on salattava lohko, salausavain, sekä edellinen salakirjoituslohko. Siten jokainen salauslohko vaikuttaa seuraaviin salauslohkoihin, minkä tarkoitus on haitata salauksen murtamista, sekä jonkin lohkon manipulointia tiedonsiirron aikana.

**GCM – moodi** (”Galois Counter Mode”) on korvannut esim. verkkopankeissa v.2017 lopussa CBC moodin. Muutoksella on haluttu lisätä turvatekijöitä, jotka varmistavat viestin muuttumattomuuden ja lähettäjän todentamisen.

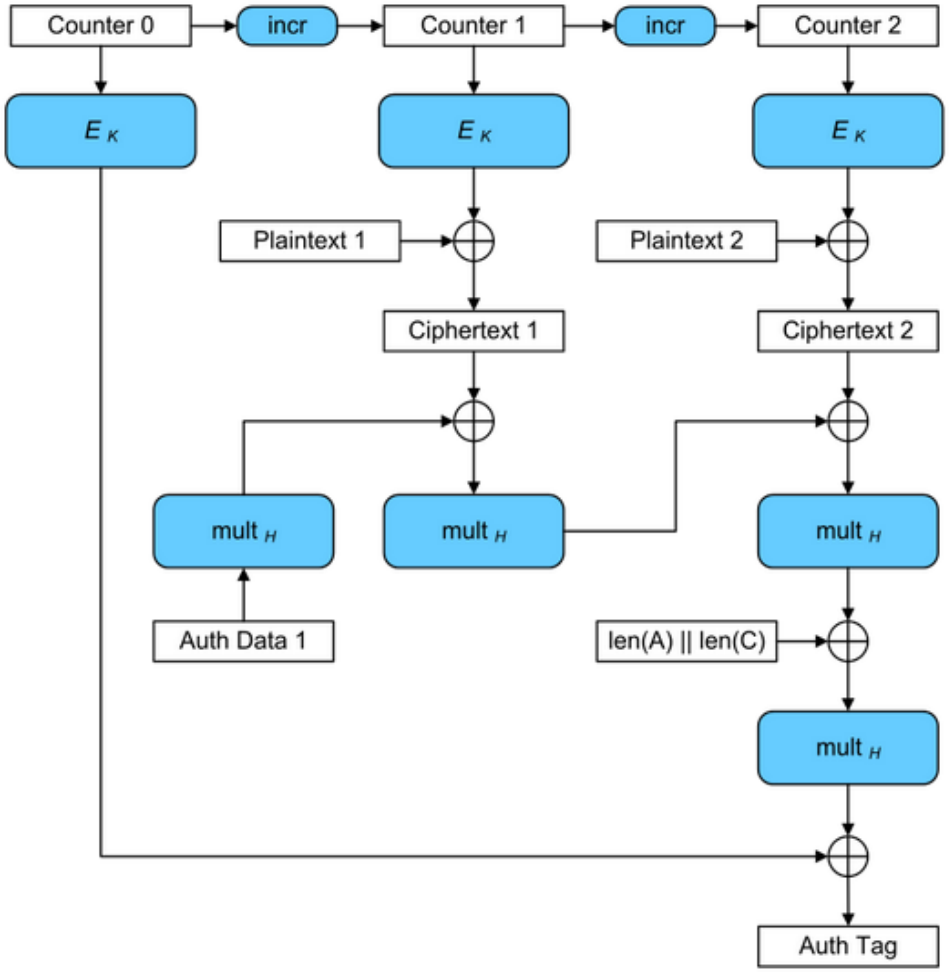
Tekniset tiedot

**Yhteys salattu (TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, 256-bittinen avain, TLS 1.2)**

GCM- moodissa AES:n syötteenä on laskurin arvo (1,2,...) sekä salausavain K. Tuloksena saatu 128 bittiä lisätään XOR – yhteenlaskua käyttäen salattavaan viestilohkoon.

Uutena piirteenä on, että salaus muodostaa tiedonsiirron loppuun 384 bitin tiivisteen, joka syötteenä ovat kaikki salakirjoituslohkot, lähettäjän autentikointidata (esim. henkilötiedot), viestin pituus, sekä salausavain.

Tarkoituksena on, että taitavinkaan hyökkääjä ei pystyisi väärentämään lähettäjää tai muuttamaan viestiä paljastumatta.



**”Suomen valtion salauskäytännöt” asiakirja v. 2012, joka löytyy valtion tietoturvasivustolta www.vm.fi/vahti, antaa seuraavat suosituksen lohkosalainten käytöstä**

Luottamukselliset dokumentit
**AES128**

*) numero = avaimen pituus bitteinä

Erittäin salainen materiaali :
**AES258**

Huom! Ym. asiakirja ei ole enää voimassa. Uudet suositukset löytyvät päivitetystä dokumentista, joka on Moodlessa tämän pdf:n perässä. Uudessakin versiossa turvaluokan II asiakirjat salataan AES258:lla, turvaluokan I asiakirjoissa tämäkään ei ole riittävä

Tehtävään 12 voi vastata päivitettyjen ohjeiden mukaisesti.

# EU:n salaustyöryhmän arviot riittävistä avainpituuksista

| Avainpituus bitteinä | Kuvaus |
|---|---|
| 72 | Voidaan murtaa perustekniikoilla |
| 80 | Teoriassa kestää murtoyrityksiä |
| 96 | Yleisesti pidetään ehdottomana miniminä |
| **112** | **Riittävä minimitaso** |
| **128** | **Riittävä, lukuun ottamatta erittäin salaisia tiedostoja** |
| **256** | **Riittävä myös erittäin salaisille tiedostoille** |