Tik-79.159 Tiedon salaus ja suojaus

Teknillinen korkeakoulu

Kevät 1998

Tuomas Aura

Tietoon kohdistuvia uhkia

- Mikä tietoa uhkaa?
  - tahattomat erehdykset
  - uteliaisuus, kokeilunhalu
  - kiusanteko, terrorismi
  - hyötyä tavoitteleva epärehellisyys
- Millainen tieto pitää suojata?
  - liikesalaisuudet, asiakkaan tiedot
  - rahaliikenne ja kaupankäynti
  - yksityisasiat
  - lakisääteiset velvoitteet
  - sotilaalliset salaisuudet
- Hyvä lähtökohta on kaiken tiedon suojaaminen jotenkin.

Tietoturvatavoitteet

salaisuus, luottamuksellisuus	
aitous, todennus, eheys	tunnistaminen
	alkuperän varmistaminen
	oikeuksien tarkistaminen
kiistämättömyys	
vaihdon reiluus	
palvelun saatavuus	

secrecy, confidentiality	
authentication, integrity	*identification*
	= identity authentication
	data origin authentication
	authorization
non-repudiation	
fairness of exchange	
availability	

Kryptografia

- Algoritmeja ja protokollia
- Turvallinen tiedon siirto, talletus ja käsittely epäluotettavissa välineissä
- Luotettava toiminta keskinäisen epäluottamuksen vallitessa

Kurssin tavoite

Kurssin suorittaneella tulisi olla

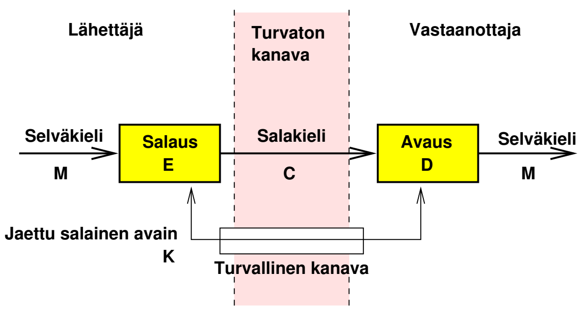
- perustiedot kryptografisista menetelmistä
- mielikuva kryptografian mahdollisuuksista ja rajoituksista
- riittävästi tietoa yleisimpien erehdysten ja huuhaan välttämiseksi
- käsitys alan vaikeudesta ja omien suunnittelutaitojen rajallisuudesta
- teoreettiset perusteet sekä sovelluskursseille että kryptografian matemaattisten perusteiden opiskeluun.

Suunniteltu luento-ohjelma

päivä	aihe
23.1.	aloitusluento: kryptografia, salakirjoitus
30.1.	symmetrinen salaus 1: DES, käyttömoodit
6.2.	symmetrinen salaus 2: kryptoanalyysi
13.2	symmetrinen salaus 3: IDEA, bittivirtasalaus
20.2.	julkinen avain 1: MAC, julkisen avaimen algoritmit
27.2.	julkinen avain 2: RSA
6.3.	julkinen avain 3: Diffie-Hellman, ElGamal
13.3.	kauppa 1: allekirjoitukset, aikaleimat
20.3.	kauppa 2: sähköinen raha
27.3.	avaintenhallinta 1: avaintenjakelu ja autentikointi
3.4.	avaintenhallinta 2: X.509, Kerberos, SSH, SSL
10.4.	pääsisäisloma
17.4.	avaintenhallinta 3: protokollien verifiointi
24.4.	tiedon kätkeminen: steganografia

Salakirjoitus

- Keino salaisuuden (luottamuksellisuuden) säilyttämiseen
- Salausalgoritmeja käytetään komponenttina monissa muissakin tarkoituksissa. **Salaus ei sellaisenaan suojaa eheyttä!**



- Lähettäjä, vastaanottaja, selväkieli, salakieli, salaus, avaus, avain, turvallinen kanava, turvaton kanava

  (*plaintext, ciphertext, encryption, decryption*)

- Salauksen matemaattinen esitys:

  sanoma-avaruus $\mathfrak{M}$, esim. 64-bittiset kokonaisluvut

  salakieliavaruus $\mathfrak{C}$, esim. 64-bittiset kokonaisluvut

  avainavaruus $\mathfrak{K}$, esim. 128-bittiset kokonaisluvut

$$k \in \mathfrak{K}$$
$$M \in \mathfrak{M}$$

  salaus- ja avausfunktiot

$$E_k : \mathfrak{M} \longrightarrow \mathfrak{C}$$
$$D_k : \mathfrak{C} \longrightarrow \mathfrak{M}$$

  symmetrisen salakirjoituksen perusominaisuus

$$D_k(E_k(M)) = M$$

- **Kerkhoffin periaate**: julkinen algoritmi, salainen avain
  - algoritmi koeteltu, viat paljastuvat
  - oivalluksesta ei apua ratkaisussa
  - toisaalta salausmenetelmä paljastuu muidenkin käyttöön

Hyökkäykset

- Salakirjoituksen ratkaiseminen
  - selväkielisen tekstin selvittäminen
  - avaimen selvittäminen
- Hyökkäykseen tarvittava materiaali
  - tunnetun salakielen hyökkäys
  - tunnetun selväkielen hyökkäys
  - valitun selväkielen hyökkäys
  - adaptiivinen valitun selväkielen hyökkäys
- Hyökkäyksen vaativuus
  - *brute-force* ja kryptoanalyyttinen hyökkäys
  - hyökkäyksen vaatima laskenta- ja muistikapasiteetti
  - voidaanko laskenta tehdä etu- tai jälkikäteen
  - passiivinen ja aktiivinen hyökkäys
  - kuka voi hyökätä (esim. tietoverkossa)

Salauksen vahvuus

- **Vahva ja heikko salaus**

  Vahvaa salausta ei kukaan pysty nykyisllä tai näköpiirissä olevilla resursseilla ratkaisemaan.

  Jos salaus ratkeaa tänään parhailla supertietokoneilla, huomenna sen aukeaa kotimikrolla. Heikennetty salaus (esim. lyhyt avain) ei ole salaus ollenkaan.

- **Avaimen pituus**

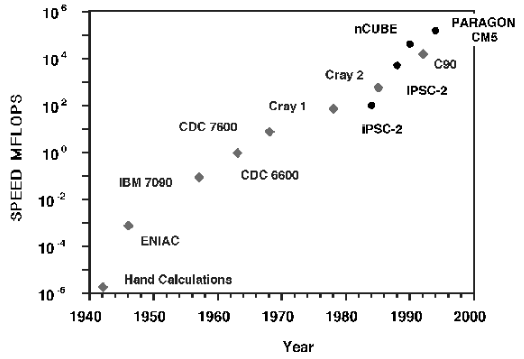
  Jos avainavaruus on liian pieni (esim. 64-bittiset kokonaisluvut), löytyy avain kokeilemalla (*brute-force*-hyökkäys).

  Nykytekniikalla 80-bittisiä avaimia ei pystytä löytämään kokeilemalla.

  Avaimen pitää olla riittävän pitkä ($\geq$ 112 bittiä). Muuten tekniikan kehitys vaarantaa salauksen turvallisuuden. Ei ole mitään hyvää syytä käyttää lyhyempiä avaimia

  Pitkä avain ei kuitenkaan takaa algoritmin luotettavuutta.

- **Mooren lain** mukaan tietokoneiden kapasiteetti kaksinkertaistuu kahdessa vuodessa. Kahdessa vuodessa tekniikan kehitys siis kuluttaa yhden bitin avainpituudesta.



- Vahvoja, luotettavia salausalgoritmeja on olemassa. Salakirjoitus on tietoturvallisuuden parhaiten tunnettu ja taidettu osa-alue.

  Vain *one-time-pad* on matemaattisesti todistettu turvalliseksi.

- Luotettavaksi tunnettuja algoritmeja käytettäessä turvallisuusaukot eivät yleensä löydy algoritmista vaan
  - fyysisessä turvallisuudesta
  - protokollista
  - käyttäjistä
  - poliitikoista.

Matemaattisia peruskäsitteitä

- bijektio
- permutaatio
- algebrallinen ryhmä
- binääriluvut, $0 \ldots 2^{64} - 1 = 8$ tavua
- XOR $\oplus$
- moduloaritmetiikka

Algebrallinen ryhmä

- **Ryhmä** $< G, * >$ koostuu joukosta alkioita G ja binäärisestä laskutoimituksesta $*$, joilla on seuraavat ominaisuudet:
  1. Joukko G on suljettu laskutoimituksen $*$ suhteen,
  $$\forall a, b \in G : a * b \in G.$$
  2. Operaattori $*$ on assosiatiivinen,
  $$\forall a, b, c \in G : a * (b * c) = (a * b) * c.$$
  3. On olemassa yksikköalkio e, jolle pätee
  $$\forall a \in G : e * a = a * e = a.$$
  4. Jokaisella alkiolla on käänteisalkio,
  $$\forall a \in G \exists a' \in G : a * a' = a' * a = e.$$
- Ryhman muodostavat esimerkiksi
  - kokonaisluvut, laskutoimituksena $+$,
  - N-bittiset kokonaisluvut, laskutoimituksena $+ \pmod{2^N}$,
  - N-bittiset kokonaisluvut, laskutoimituksena $\oplus$,
  - mielivaltaisen joukon permutaatiot, laskutoimituksena funktioiden yhdistäminen.

Informaatioteoriaa

- Shannon
- **Täydellinen (informaatioteoreettinen) salaus** tarkoittaa, ettei salakielinen sanoma (ilman avainta) anna mitään tietoa alkuperäisestä sanomasta. Vastustajalla oletetaan olevan rajattomat laskennalliset resurssit.

  Tämä on mahdollista vain, jos salausavaimia on yhtä monta kuin mahdollisia sanomia eli avain on yhtä monta bittiä pitkä kuin sanoma.
- **One-time-pad**-salauksessa sanoma XORataan lähettäjän ja vastaanottajan etukäteen vaihtaman **satunnaisen** avaimen kanssa,
  $$C = K \oplus P.$$
  Huom! Avaimen pitää todella olla satunnainen, ei algoritmin generoima. (Vrt. Elementrix Power One Time Pad.)

  Huom! Avainta saa käyttää vain yhden kerran. Muuten paljastuu kahden selväkielisen tekstin XOR.
- Suositeltava menetelmä lyhyille, tärkeille tai harvoin lähetettäville viesteille. Salaisuus säilyy ikuisesti tekniikan kehityksestä huolimatta.

- Sanoman sisältämän informaation mitta on **entropia**. Sanoman keskimääräinen entropia bitteinä:
  $$H(M) = -\sum P(M_i) \log_2 P(M_i).$$
  Kielen entropia (bittiä/merkki):
  $$r = \lim_{n\to\infty} H(M)/n,$$
  missä n on sanoman M pituus.

  Kielen **redundanssi** (bittiä/merkki):
  $$D = R - r,$$
  missä R on merkin koodaukseen käytetty tila.
- **Entropia** $\sim$ **epäjärjestys** $\sim$ **satunnaisuus**
  esim. satunnaisluku
- **Redundanssi** $\sim$ **järjestys** $\sim$ **ennustettava osa**
  esim. vakio-otsake

  Redundanssi mahdollistaa oikean selväkielen tunnistamisen avainta kokeillessa ja siten salauksen ratkaisemisen.
- Suurten englanninkielisten tekstitiedostojen entropia on noin 2.1 bittiä/merkki. Kun merkin koodaamiseen käytettään 8 bittiä, on redundanssi $8 - 2.1 = 5.9$ bittiä/merkki.

  Esim. two-time-pad ja kirjasalakirjoitus

- Esim.

koodi	sanoma	TN	informaatio / bittiä
0	aurinko paistaa	0.1	$-\log_2 0.1 = 3.32$
1	puolipilvistä	0.5	$-\log_2 0.5 = 1$
2	pilvipoutaa	0.3	$-\log_2 0.3 = 1.74$
3	sataa	0.1	$-\log_2 0.1 = 3.32$

Oletetaan perättäiset sanomat toisistaan riippumattomiksi. Sanoman keskimääräinen entropia $H = 0.1 * 3.32 + 0.5 * 1 + 0.3 * 1.73 + 0.1 * 3.32 = 1.7$ bittiä. Jos sanoma koodaamiseen käytetään 2 bittiä, on redundanssi $2 - 1.7 = 0.3$ bittiä/sanoma.

Sääasemalla salakirjoitetaan päivittäiset sanomat korvaamalla merkit toisilla. Kolmen viikon aikana kuulemme seuraavat salatut sanomat:

ma	ti	ke	to	pe	la	su
0	3	3	0	1	0	0
ma	ti	ke	to	pe	la	su
1	1	1	0	0	0	0
ma	ti	ke	to	pe	la	su
0	3	0	3	3	2	3

Millainen sää oli?

Huom! Koska lähekkäisten päivien säät riippuvat toisistaan, on sääaseman käyttämän kielen redundanssi todellisuudessa suurempi kuin laskettu 0.3 bittiä/sanoma.

- Halutaan käyttää lyhyitä avaimia. Sallitaan vastustajalle jotain tietoa viestistä, kunhan epävarmuus sisällöstä on riittävän suuri.
- Avaimen entropia (monenko bitin avain):

  $$H(K) = log_2|K|.$$

  Oletus: avain satunnaisesti valittu.

  Salakirjoituksen mahdollisten selväkielisten ratkaisujen määrä:

  $$2^{H(K)-nD},$$

  missä n on sanoman M pituus ja D selväkielen redundanssi.

  Varoitus! Kaava on likimääräinen, eikä sen perusteella voi osoittaa salauksen turvallisuutta.
- Redundanssittoman (esim. täydellisesti pakatun tai satunnaisen) tiedon salaukseen riittää periaatteessa yksinkertainen menetelmä. Epävarmuus ratkaisusta on aina yhtä suuri, jos sallitaan vain tunnetun salakielisen tekstin hyökkäys.
- Yleensä käytännön salakirjoituksilla on yksi helposti tunnistettava ratkaisu.

- Esim.

arpakuution lukema	TN	informaatio / bittiä
1	0.17	2.58
2	0.17	2.58
3	0.17	2.58
4	0.17	2.58
5	0.17	2.58
6	0.17	2.58

Perättäiset nopanheitot ovat toisistaan riippumattomia. Sanoman keskimääräinen entropia $H = 2.58$ bittiä. Jos kielessä on 6 merkkiä eli sanoma koodataan 2.58 bitillä, ei kielessä ole redundanssia lainkaan.

Mitä seuraavista korvausmenetelmällä salakirjoitetuista nopanheiton tuloksista voidaan sanoa?

3	6	6	1	2	4	4
2	4	1	3	1	1	1
2	4	5	3	6	5	4

Koska avain (korvaustaulu) on lyhyempi kuin viesti, paljastuu välttämättä jonkin verran tietoa. Täydellinen ratkaiseminen ei ole mahdollista eli mahdollisia selväkieliä on useita.

- Käytännössä halutaan käyttää samaa lyhyttä avainta monta kertaa. Yritetään tehdä salakirjoituksen ratkaiseminen mahdollisimman vaikeaksi, vaikka se onkin mahdollista.

  Modernit symmetriset lohkosalakirjoitusmenetelmät perustuvat Shannonin ideoihin:
- **Konfuusio**

  Häivytetään yhtäläisyydet selväkielen ja salakielen välillä. Kun salakieli näyttää satunnaiselta, ei redundanssia havaita, eikä sitä voida käyttää hyväksi.

  Menetelmä: **korvaus** eli **substituutio**
- **Diffuusio**

  Hajotetaan redundanssi laajalle alueelle salakieliseen tekstiin. Sanoman (lohkon) jokaisen bitin pitää riippua jokaisesta selväkielen bitistä.

  Menetelmä: **permutaatio**
- Klassisissa salauksissa joko permutaatio tai substituutio. Moderneissa salauksissa molempia vuorotellen ja monta kertaa. (*product cipher*)

Kompleksisuusteoriaa

- Käsitteitä:
  - **P, NP, PSPACE, EXPTIME**
  - *Bounded-Propability Polynomial time* (**BPP**): Polynomisessa ajassa tietyllä äärellisellä todennäköisyydellä antaa oikean tuloksen.
- Kun salakirjoituksen ratkaiseminen on mahdollista, **haluttaisiin** sen olevan laskennallisesti liian kallista.

  Alarajakompleksisuus! *Worst case* tai keskimääräinen vaikeus ei kiinnosta.

  Alarajoista tiedetään kompleksisuusteoriassa varsin vähän. Siksi salausjärjestelmiä ei todisteta oikeaksi.

  Parhaassa tapauksessa salauksen ratkaiseminen osoitetaan yhtä vaikeaksi kuin jokin tunnettu vaikea ongelma, esim. tekijöihin jako.