Tik-79.159 Tiedon salaus ja suojaus
Tentin 18.11.1997 ratkaisut Tuomas Aura

1. (a) Salauksessa konfuusion tehtävä on häivyttää havaittavat yhteydet selväkielisen ja salakielisen tekstin välillä ja estää selväkielisen sanoman redundanssin havaitseminen salakielisessä. Lohkosalauksessa konfuusio saadaan useimmiten aikaan substituutiolla.

   (b) Steganografia on tiedon kätkemistä. Kätkemällä tietoa toisen tiedon joukkoon voidaan salata tiedon olemassaolo (vrt. sisällön salakirjoittaminen) tai estää sen poistaminen (esim. alkuperää osoittavat vesileimat).

   (c) ECB-moodissa salattava teksti jaetaan lohkoihin ja lohkot salataan lohkosalausalgoritmilla lohko kerrallaan toisistaan riippumatta.

   (d) Sertifikaatti on allekirjoitettu todistus tai valtakirja. Sertifikaattipolku on jono sertifikaatteja, joista edellinen aina varmentaa seuraavan sertifikaatin allekirjoittajan luotettavuuden. (määritelmän sanamuoto Tuomas Pekkalan vastauksesta)

   (e) Pseudosatunnaisluku on algoritmisesti tuotettu satunnaista (arvottua) bittijonoa muistuttava luku. Koska se on tuotettu algoritmisesti, ei se tietenkään ole oikeasti satunnainen. Vahvoja pseudosatunnaislukuja on nykyisillä tai näköpiirissä olevilla resursseilla kenenkään mahdotonta erottaa todellisesta satunnaisluvusta tilastollisilla tai muillakaan testeillä.

   (f) Hybridisalauksessa itse sanoma salataan symmetrisellä salausmenetelmällä ja satunnaisella istuntoavaimella. Istuntoavain salataan vastaanottajan julkisella avaimella ja lähetetään vastaanottajalle erikseen tai sanoman mukana.

2. (a) *One-time-pad* -salauksen avain on yhtä pitkä kuin sanoma, joten salakielestä saadaan mikä tahansa selväkieli avaamalla se sopivalla avaimella. Niinpä salakieli ei ilman avainta sisällä mitään tietoa selväkielestä, eikä *brute-force* -hyökkäyskään ole mahdollinen. Koska avainta käytetään vain kerran, ei yhden sanoman tai avaimen tunteminen myöskään auta muiden murtamisessa.

   (b) 2 kierroksen Feistel-salaus:

$$
\begin{aligned}
L, R &= P \\
L' &= L \oplus f(R, K_1) \\
R' &= R \oplus f(L', K_2) \\
C &= L', R'
\end{aligned}
$$

   Tunnetaan $C = L', R'$ ja osataan arvata selväkielen toinen puolisko R. Tehdään valitun selväkielen hyökkäys: salataan $0, R$ ja saadaan salakieli jonka vasen puolisko on $f(R, K_1)$. Lasketaan $L = L' \oplus f(R, K_1)$. Saatiin siis selville myös vasen puolisko, kun oikea osattiin arvata. (Vaikka 3 kierroksen salaus olisi periaatteessa ok, jos f olisi hyvin vahva, tarvitaan todellisissa salauksissa enemmän kierroksia.)

   (c) Yksisuuntaisen hajautusfunktion heikko törmäyksettömyys tarkoittaa, että annetulle sanomalle ei ole mahdollista löytää paria, joka tuottaa saman hajautusfunktion arvon. Vahva törmäyksettömyys tarkoittaa, että ei ole mahdollista löytää mitään kahta sanomaa, jotka tuottaisivat saman hajautusarvon. Allekirjoituksessa sanomasta lasketaan hajautusarvo joka sitten allekirjoitetaan allekirjoitusalgoritmilla (esim. RSA). Allekirjoituksessa on tärkeää, että allekirjoitetun sanoman sisältö on yksikäsitteinen, eikä siitä voi myöhemmin syntyä kiistaa. Jos on mahdollista löytää sanomapari, jolle allekirjoituksessa käytetyn yksisuuntaisen funktion arvo on sama, ja joku tahallaan allekirjoittaa toisen sanomista, voi hän myöhemmin väittää allekirjoittaneensa sen toisen. (Huom. Tämä liittyy syntymäpäivähyökkäykseen ainoastaan siten, että syntymäpäivähyökkäys on keino löytää törmäyksiä.)

3. (a) $n = pq = 59 \cdot 37 = 2183$. $\phi(n) = 58 \cdot 36 = 2088$. Julkinen eksponentti on $e = 17$. Salainen eksponentti on $d = 17^{-1} \bmod 2088$. Ratkaistaan $d \cdot 17 + k \cdot 2088 = 1$ Eukleideen algoritmilla:

$$\begin{aligned} 2088 &= 122 \cdot 17 + 14 \qquad & 14 &= 2088 - 122 \cdot 17 \\ 17 &= 14 + 3 & 3 &= 17 - 14 \\ 14 &= 4 \cdot 3 + 2 & 2 &= 14 - 4 \cdot 3 \\ 3 &= 2 + 1 & 1 &= 3 - 2 \end{aligned}$$

$$\begin{aligned} 1 &= 3 - 2 = 3 - (14 - 4 \cdot 3) \\ &= -14 + 5 \cdot 3 = -14 + 5 \cdot (17 - 14) \\ &= 5 \cdot 17 - 6 \cdot 14 = 5 \cdot 17 - 6 \cdot (2088 - 122 \cdot 17) \\ &= -6 \cdot 2088 + \mathbf{737} \cdot 17 \end{aligned}$$

Salainen avain $d = \mathbf{737}$.

(b) $C = 406$. $d = 737 = 1011100001_2$.

1	$406^1 \bmod 2183$		$= \mathbf{406}$
0	$406^2 \bmod 2183$	$= 406^2 \bmod 2183$	$= 1111$
0	$406^4 \bmod 2183$	$= 1111^2 \bmod 2183$	$= 926$
0	$406^8 \bmod 2183$	$= 926^2 \bmod 2183$	$= 1740$
0	$406^{16} \bmod 2183$	$= 1740^2 \bmod 2183$	$= 1962$
1	$406^{32} \bmod 2183$	$= 1962^2 \bmod 2183$	$= \mathbf{815}$
1	$406^{64} \bmod 2183$	$= 815^2 \bmod 2183$	$= \mathbf{593}$
1	$406^{128} \bmod 2183$	$= 593^2 \bmod 2183$	$= \mathbf{186}$
0	$406^{256} \bmod 2183$	$= 186^2 \bmod 2183$	$= 1851$
1	$406^{512} \bmod 2183$	$= 1851^2 \bmod 2183$	$= \mathbf{1074}$

$$\begin{aligned} P = C^d \bmod n &= 406^{737} \bmod 2183 \\ &= 406^{512} \cdot 406^{128} \cdot 406^{64} \cdot 406^{32} \cdot 406 \bmod 2183 \\ &= 1074 \cdot 186 \cdot 593 \cdot 815 \cdot 406 \bmod 2183 \\ &= 998 \cdot 1111 \bmod 2183 = \mathbf{1997}. \end{aligned}$$

Selväkielinen sanoma on siin $P = 1997$.

4. (a) Jos moduli on pieni, on se helppo jakaa tekijöihin. Jos yksityinen eksponentti eli salainen avain on pieni, on se helppo löytää kokeilemalla. Ratkaisu: valitaan riittävän suuria lukuja.

(b) Jos julkinen eksponentti on pieni, eivät pienet sanomat tule kunnolla salattua, koska ne eivät redusoidu modulo n. Esim. $M^d =$

$2^3 = 8 << n$. Ratkaisu: lisätään salattavaan sanomaan satunnainen osa (suola) niin, ettei luku koskaan ole pieni. Hybridisalauksessa satunnaisuus tulisi lisättyä automaattisesti satunnaisen istuntoavaimen muodossa. Voidaan myös käyttää suurempaa eksponenttia.

(c) Jos selväkielisiä sanomia on vähän, voi hyökkääjä kokeilla salata niitä julkisella avaimella ja toivoa saavansa tulokseksi murrettavan salakielisen tekstin (*forward search attack*). Ratkaisu: lisätään salattavaan sanomaan satunnainen osa, jolloin sanomia on paljon erilaisia.

(d) Jos selväkielinen sanoma on muiden selväkielien tulo, on myös salakielinen sanoma vastaavien salakielien tulo (RSA:n multiplikatiivisuus). Tämän voi huomata selväkieltä katselemalla. Ratkaisu: lisätään sanoman satunnainen osa, jolloin selväkielien erikoinen riippuvuus peittyy.

5. (a) Salausfunktion E pitää olla julkisen avaimen salaus ja sen pitää myös suojata sanomien eheys niin, ettei salatun sanoman osia voi muuttaa hallitusti. (Koko sanoma tietysti voidaan salata uudestaan, mutta nonssia N_A tuntematon ei esim. voi vaihtaa lähettäjän A nimeä sanomassa 1 muuttamatta nonssia.) Tämä voidaan toteuttaa joko salaamalla sopivalla menetelmällä (esim. RSA) yhdenä lohkona tai käyttämällä MAC:ia hybridisalauksen kanssa.

(b) Kun A avaa yhteyden B:hen, B pystyy esiintymään C:lle A:na.

$$
\begin{array}{lll}
A \longrightarrow B: & E_B(N_A, A) & \text{(1)} \\
B(A) \longrightarrow C: & E_C(N_A, A) & \text{(1')} \\
C \longrightarrow B(A): & E_A(N_A, N_C) & \text{(2')} \\
B \longrightarrow A: & E_A(N_A, N_C) & \text{(2)} \\
A \longrightarrow B: & E_B(N_C) & \text{(3)} \\
B \longrightarrow C: & E_C(N_C) & \text{(3')}
\end{array}
$$

(c) Esimerkiksi lisätään sanomaan 2 lähettäjän nimi:

$$\boxed{2. \quad B \longrightarrow A \quad E_A(N_A, N_B, B)}$$

(d) Tunnistus on eksplisiittinen molempiin suuntiin. A näkee sanoman 2 nonssista N_A, että B on vastaanottanut sanoman 1 ja avannut sen julkisella avaimellaan. Vastaavasti nonssi N_B voi olla sanomassa 3 vain, jos A on kopioinut sen sinne.

6. Moninkertaista salausta käytetään joko luotettavaksi tunnetun menetelmän avaimen pituuden kasvattamiseen (esim. 3DES) tai turvallisuuden lisäämiseen kierroslukua kasvattamalla tai riippumattomia salauksia yhdistelemällä.

Erityisesti pitää kuvata *meet-in-the-middle* -hyökkäys ja siitä johtuva kaksinkertaisen salauksen yllättävä heikkous kolminkertaiseen verrattuna. Kolminkertainen salaus kahdella avaimella EDE-moodissa on myös periaatteessa heikompi kuin aito kolmen avaimen salaus.

Oppikirjasta, luennoilta ja laskuharjoituksista sekä kirjallisuudesta löytyy paljon muutakin asiaa, esim. ryhmäominaisuuden merkitys.