| | |
|---|---|
| Tik-79.159 Tiedon salaus ja suojaus | |
| Tentin 16.1.1998 ratkaisut | Tuomas Aura |

1. (a) CBC (*Cipher Block Chaining*) on lohkosalausalgoritmin käyttötapa, jossa selväkielinen lohko xorrataan edellisen salakielisen lohkon kanssa ennen salaamista algoritmilla. Ensimmäinen lohko xorrataan satunnaisesti valitun alustuvektorin kanssa.

$$C_0 = IV; \quad C_i = E_K(P_i \oplus C_{i-1}), i = 1 \ldots$$

Hyöty on, että sama sanoma kaksi kertaa salattuna näyttää erilaiselta eikä sanomassa toistuvia osia voida havaita. (Kuva olis kiva ylläri.)

(b) Kerckhoffsin periaatteen mukaan salauksen vahvuuden ei pidä riipua algoritmin salaisuudesta. Vain avain on salainen.

(c) Syntymäpäiväparadoksi tarkoittaa sitä, että valittaessa satunnaisesti $k$ alkiota $n$ alkion joukosta (sama saadaan valita useaan kertaan) sattuu valittujen joukkoon yllättävän usein kaksi samaa alkiota. Keksimäärin näin käy, kun $k$ on suuruusluokkaa $\sqrt{n}$. Tätä voidaan käyttää hyväksi esim. etsittäessä törmäyksiä yksisuuntaisille hajautusfunktioille.

(d) Subliminaalinen kanava tarkoittaa viestintätapaa, jota käytettäessä ulkopuolinen taho ei voi edes tietää, että viestejä lähetetään. Sanomat siis piilotetaan. Yleensä sanomien lähetykseen käyttäen jotain menetelmää, jota ei ole alunperin tarkoitettu informaation siirtoon ja siksi sitä ei huomata.

(e) Informaatioteoreettinen menetelmä on turvallinen, koska hyökkääjällä ole riittävästi tietoa hyökkäyksen onnistumiseksi vaikka käytössä olisi rajaton laskentakapasitetti. Siksi hyökkäys on mahdoton. Esim. *one-time-pad* -salattu sanoma ei sisällä riittävästi informaatiota sanoman avaamiseksi ilman avainta.

(f) Message Authentication Code MAC on symmetriseen menetelmään perustuva "allekirjoitus". Sanoma allekirjoitetaan ja sen alkuperä tarkistetaaan saman salaisen avaimen avulla.

2. (a) Hyökkäykset MD5:ttä vastaan vaarantavat *vahvan* törmäyksettömyyden, mutta eivät *heikkoa* törmäyksettömyyttä. Aikaisemmin aikalaimatulle sanomalle ei siksi ole edelleenkään mahdollista löytää paria, jolla olisi sama hajautusarvo. Vaikka olisi mahdollista luoda alunperin sanomapari, joilla on sama hajautusarvo, ei tämä haittaa, sillä molemmat sanomathan olivat tiedossa leimaushetkellä.

(b) Jos allekirjoitettavassa sanomassa on liian vähän redundanssia, eikä sitä mitenkään lisätä, on kokeilemalla mahdollista löytää sanoma-allekirjoituspari. Ääritapauksessa redundanssia ei ole lainkaan, eikä allekirjoitusta tarkistettaessa ole mitenkään mahdollista erottaa aitoa sanomaa väärästä. Useimmiten redundanssia syntyy, kun allekirjoitus liitetään alkuperäisen sanoman perään. RSA:lla allekirjoitettaessa taas täytetään (esim. 1024-bittinen) RSA-lohko vakiobiteillä, joiden säilyminen tarkistetaan allekirjoitusta tarkistettessa. (Huom! Allekirjoitukseen lisätään nimenomaan redundanssia eli vakiobittejä tai toistoa. Salauksessa taas lisätään satunnaisuutta.)

Tehtävän saattoi ymmärtää myös toisin: Kryptografista protokollaa suunnitellessa lisätään sanomiin redundanttia dataa kuten aikaleima, vastaanottajan nimi, palautettava nonssi, sarjanumero jne. Vastaanottaja tarkistaa, että näiden ylimääräisten kenttien arvot ovat odotettuja ennen kuin hyväksyy sanoman. Tällä tavalla estetään useimmat toistohyökkäykset.

(c) Kaksinkertaista salausta vastaan voidaan tehdä ns. *meet-in-the-middle* -hyökkäys. (Tunnetaan selväkieli-salakielipari. Salataan tunnettu selväkielilohko kaikilla mahdollisilla avaimilla ja talletetaan tulos muistiin. Avataan sitten tunnettu salakieli kaikilla mahdollisilla avaimilla ja katsotaan, onko tulos muistissa. Tarkistetaan toisen tunnetun parin avulla, onko löytynyt avainpari oikea.) Hyökkäys vaatii paljon muistia, mutta aikaa vain noin kaksi kertaa niin paljon kuin *brute force* -hyökkäys yksinkertaista salausta vastaan. Niinpä kaksinkertainen salaus ei ole kovin hyvä keino pidentää liian lyhyttä avaimen pituutta.

3. (a) $n = pq = 47 \cdot 41 = 1927$. $\phi(n) = 46 \cdot 40 = 1840$. Julkinen eksponentti on $e = 17$. Salainen eksponentti on $d = 17^{-1} \bmod 1840$. Ratkaistaan $d \cdot 17 + k \cdot 1840 = 1$ Eukleideen algoritmilla:

$$\begin{aligned} 1840 &= 108 \cdot 17 + 4 & \qquad 4 &= 1840 - 108 \cdot 17 \\ 17 &= 4 \cdot 4 + 1 & \qquad 1 &= 17 - 4 \cdot 4 \end{aligned}$$

$$\begin{aligned} 1 &= 17 - 4 \cdot 4 = 17 - 4 \cdot (1840 - 108 \cdot 17) \\ &= 433 \cdot 17 - 4 \cdot 1840 \end{aligned}$$

Salainen avain $d = \mathbf{433}$.

(b) $h(M) = 123$. $d = 433 = 110110001_2$.

| | | | |
|---|---|---|---|
| 1 | $123^1 \bmod 1927$ | | $= \mathbf{123}$ |
| 0 | $123^2 \bmod 1927$ | $= 123^2 \bmod 1927$ | $= 1640$ |
| 0 | $123^4 \bmod 1927$ | $= 1640^2 \bmod 1927$ | $= 1435$ |
| 0 | $123^8 \bmod 1927$ | $= 1435^2 \bmod 1927$ | $= 1189$ |
| 1 | $123^{16} \bmod 1927$ | $= 1189^2 \bmod 1927$ | $= \mathbf{1230}$ |
| 1 | $123^{32} \bmod 1927$ | $= 1230^2 \bmod 1927$ | $= \mathbf{205}$ |
| 0 | $123^{64} \bmod 1927$ | $= 205^2 \bmod 1927$ | $= 1558$ |
| 1 | $123^{128} \bmod 1927$ | $= 1558^2 \bmod 1927$ | $= \mathbf{1271}$ |
| 1 | $123^{256} \bmod 1927$ | $= 1271^2 \bmod 1927$ | $= \mathbf{615}$ |

$$\begin{aligned} S = h(M)^d \bmod n &= 123^{433} \bmod 1927 \\ &= 123^{256} \cdot 123^{128} \cdot 123^{32} \cdot 123^{16} \cdot 123 \bmod 1927 \\ &= (615 \cdot 1271) \cdot (205 \cdot 1230 \cdot 123) \bmod 1927 \\ &= 1230 \cdot 1312 \bmod 1927 = \mathbf{861}. \end{aligned}$$

Allekirjoitettu sanoma on kokonaisuudessaan ("Moi, mitä kuuluu?",861).

4. Laaditaan LaGrange-(3,5)-kynnysrakenne, jossa laskutoimitukset tehdään modulo 65537 (alkuluku ja suurempi kuin suurin mahdollinen salaisuus). Arvotaan toisen asteen polynomin kertoimet väliltä $0 \ldots 65536$.

$$P(x) = (4212 \cdot x^2 + 11834 \cdot x + 1998) \bmod 65537.$$

Lasketaan viisi varjoa:

$P(1) = 4212 \cdot 1 + 11834 \cdot 1 + 1998 \bmod 65537 = 18044.$
$P(2) = 4212 \cdot 4 + 11834 \cdot 2 + 1998 \bmod 65537 = 42514.$
$P(3) = 4212 \cdot 9 + 11834 \cdot 3 + 1998 \bmod 65537 = 9871.$
$P(4) = 4212 \cdot 16 + 11834 \cdot 4 + 1998 \bmod 65537 = 51189.$
$P(5) = 4212 \cdot 25 + 11834 \cdot 5 + 1998 \bmod 65537 = 35394.$

Varjot ovat siis (1,18043), (2,42513), (3,9870), (4,5118), (5,3539). Varjojen lisäksi pitää tietää myös moduli 65537.

LaGrange-kynnysrakennetta laatiessa tulee muistaa seuraavat asiat: modulin (jos sellaista käytetään) pitää olla suurempi kuin suurin mahdollinen salaisuus. Siis jos tehtävässä valitaan moduli luokkaa $\sim 2000$, pitää perustella, ettei salaisuus millään voi olla tuota suurempi. Polynomin kertoimet pitää valita satunnaisesti (modulia $n$ käytettäessä väliltä $0 \ldots n-1$), joten esim. 3 ja 10 eivät ole uskottava valinta, ellei erikseen kerrota miten ne on arvottu. $x$:n arvot 1,2,3,4 ja 5 kelpaavat hyvin. Ei ole mitään syytä valita satunnaisia suuria lukuja.

Salaisuus selviää kolmesta varjosta muodostamalla lineaarinen yhtälöryhmä ja ratkaiselmalla polynomin kertoimet. Kun polynomi on selvillä, salaisuus on $P(0)$. Varjojen lisääminen tapahtuu laskemalla polynomilta uusia pisteitä.

Alicelle ja Bobille voidaan antaa oikeus yhdessä ratkaista salaisuus esimerkiksi antamalle heille erillinen (2,2)-kynnnysrakenne (Jaskalle satunnaisluku $r$ ja Reiskalle $r \oplus 1997$), tai luomalla (6,10)-kynnysrakenne, josta Jaskalle ja Reiskalle annetaan 3 varjoa ja muille 2.

On huomattava, että Jaskalle ei voida antaa kahta (5,3)-rakenteen varjoa, koska hän pystyisi tuolloin avaamaan salaisuuden kenen tahansa muunkin kuin Reiskan kanssa. Varjon puolittaminen ei myöskään onnistu antamalle toiselle $x$ ja toiselle $P(x)$, sillä samaa $x$:n arvoa ei käytetä kahta kertaa ja $x$:n epävarmuus on siten pienempi kuin $P(X)$:n.

5. (a) Jos sanomasta ei käy ilmi vastaanottajan nimeä, on hyökkääjän mahdollista kopioida sanoma toisille vastaanottajille. Joskus tästä on haittaa, ei aina.

   (b) Lisätään sanomiin aikalaima:

   | | | |
   |---|---|---|
   | 1. | $A \longrightarrow B$ | $\text{data}, B, T_A,\ S_A(\text{data}, B, T_A)$. |

   Voidaan myös lisätä nonssi, jonka vastaanottaja muistaa aikaleiman voimassaoloajan. Näin sanomaa ei voi toistaa edes silloin.

   (c) Jos lähettäjän kello on jäljessä tai vastaanottajan kello edellä, hylätään aidot sanomat turhaan niin kauan kuin kellon on väärässä. Se ei yleensä ole suuri turvallisuusongelma.

   Jos vastaanottajan kello on jäljessä, hyväksyy hän vanhoja sanomia. Kun kello taas asetetaan oikeaan aikaan, ei hyökkäys enää onnistu.

   Jos lähettäjän kello on edellä, voi hyökkääjä nauhoittaa tulevaisuuteen päivättyjä sanomia ja toistaa niitä sitten kun kyseinen päivä koittaa. Tätä ei estä edes kellon asettaminen takaisin oikeaan aikaan! (Vastaanottaja tietysti hylkää paitsi liian vanhat myös tulevaisuuteen päivätyt sanomat, mutta sekään ei ratkaise ongelmaa kokonaan.)

   (d) Tuoreus pitää osoittaa nonssilla, mikä vaatii ylimääräisen sanoman:

   | | | |
   |---|---|---|
   | 1. | $B \longrightarrow A$ | $N_B$ |
   | 2. | $A \longrightarrow B$ | $\text{data}, B, N_B\ S_A(\text{data}, B, N_B)$. |

6. Salauslohkon pituuden tulee olla niin suuri, että ei ole mahdollista kerätä sanakirjaa tunnetuista selväkieli-salakielipareista. 32-bittinen lohko on liian lyhyt, sillä on mahdollista tallettaa $2^{32} \approx 10^9$ tietoalkiota. 64 bittinen lohko riittää erinomaisesti nykyistä ja odotettavissa olevaa tallennuskapasiteettia ajatellen. 128-bittiä riittää varmasti. Lohkon pitää kuitenkin olla melko lyhyt, jotta lohkon sisällä syntyy nopeasti täydellinen diffuusio.

   Avaimenpituuden tulee olle niin suuri, ettei *brute force* -hyökkäys ole mahdollinen. 64-bittinen avain on nykytekniikalla mahdollista löytää kokeilemalla, mutta yli 80-bittistä ei. 128-bittinen avain riittää nykyisellä tietokoneiden suorituskyvyn kasvuvauhdilla pitkälle tulevaisuuteen. Toisaalta jos avain on liian pitkä, eivät sen bitit ehdi sekoittua salauslohkoon riittävästi. Tällöin avaimen osan tunteminen saattaa helpottaa loppuosan selvittämistä. Tästä seuraa helposti ongelmia, jos salausalgoritmia käytetään tai avaimia käsitellään jollain erikoisella tavalla.

   Tietokoneissa kaikki parametrit valitaan mieluiten 2:n potensseiksi. Uudet tietokoneet pystyvät tehokkaasti käsittelemään 64-bittisiä lukuja joten 128-bittisillä lohkoilla koneiden kapasiteetti tulee ainakin Feistel-salauksessa (64-bittiset puolilohkot) hyödynnettyä paremmin. Avaimen pituuden ei ole niin tärkeää olla 2:n potenssi.

   Yleensä turvallisuus on tärkeämpää kuin tehokkuus ja halutaan varautua tekniikan yllättäviin edistysaskeliin. Joissain sovelluksissa (esim. videokuvan siirto) nopeus kuitenkin on niin tärkeää, että salausta ei voida tehdä varman päälle. Tällöin verrataan salauksen ratkaisemisesta saatavaa hyötyä ratkaisemiskustannuksiin. Siksi on hyvä, jos salauksen taso ja kustannukset voidaan valita tilanteen mukaan. Tähän päästään esim. parametrisoimalla avaimen pituus. Turvalliseksi uskotuista salauksista yleiseen käyttöön tulee joka tapauksessa nopein ja halvimmalla toteutettava.