Lohkosalauksen moodit

Roope Salminen, Tieto- ja viestintätekniikan koulutus, TLTITVT21SV

AT00BY44-3004 Tutkimusseminaari

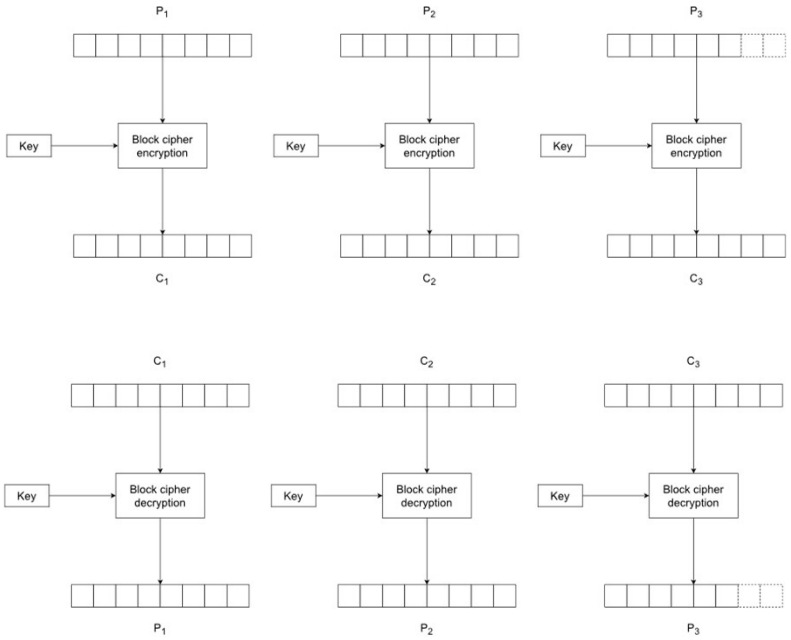
Johdanto ja tutkimuksen tavoite

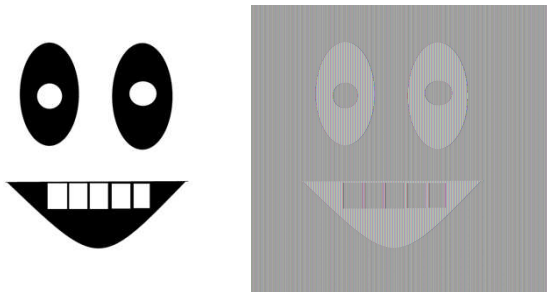
- Internet on julkinen väylä, jossa liikkuu paljon yksityiseksi tarkoitettua dataa
- Suurien datamäärien siirrossa käytetään usein symmetristä salausta lohkosalaimen ja jonkin toimintamoodin yhdistelmänä
- Tavoitteena oli tutustua lohkosalauksen neljän tunnetun toimintamoodin toimintaan, kartoittaa niiden ominaisuuksia ja heikkouksia ja näiden tietojen perusteella vertailla moodeja keskenään

Lohkosalaus yleisesti

- Lohkosalain on salausalgoritmi, joka ottaa syötteenä salausavaimen ja vakiokokoisen selväkielisen lohkon ja tuottaa vakiokokoisen salalohkon
- Lohkosalain on deterministinen ja symmetrinen
  - Deterministisyys tarkoittaa, että sama syöte (avain, selväkielinen lohko) tuottaa aina saman salalohkon
  - Symmetrisyys tarkoittaa, että salaukselle on olemassa käänteinen operaatio, jolla alkuperäinen selväkielinen lohko saadaan palautettua samaa avainta käyttämällä
  - $\forall P: D_K\big(E_K(P)\big) = P$
- Lohkosalaimen tärkeimpiä ominaisuuksia ovat lohkon ja salausavaimen koko (bittien määrä)
  - Avain täytyy olla riittävän iso, jotta väsytyshyökkäys (brute-force) ei ole varteenotettava vaihtoehto
  - Lohkon täytyy olla riittävän iso jotta törmäystodennäköisyys pysyy riittävän pienenä huolimatta käytettävästä moodista. Törmäys tässä yhteydessä tapahtuma, jossa moodin toiminta aiheuttaa sen, että kaksi salalohkoa sattumalta identtiset.
  - Birthday bound ⇒ törmäys odotettavissa ”suurella” tn. noin $\sqrt{2^n} = 2^{n/2}$ lohkon salauksen jälkeen, missä n on lohkon koko. Eli avain vaihdettava hyvissä ajoin ennen tätä
  - Isompi lohko suurentaa myös salattavan tiedoston maksimikokoa
- Käytännössä tarvitaan aina jokin toimintamoodi, joka määrittää kuinka lohkosalainta sovelletaan lohkon koon ylittäviin viesteihin

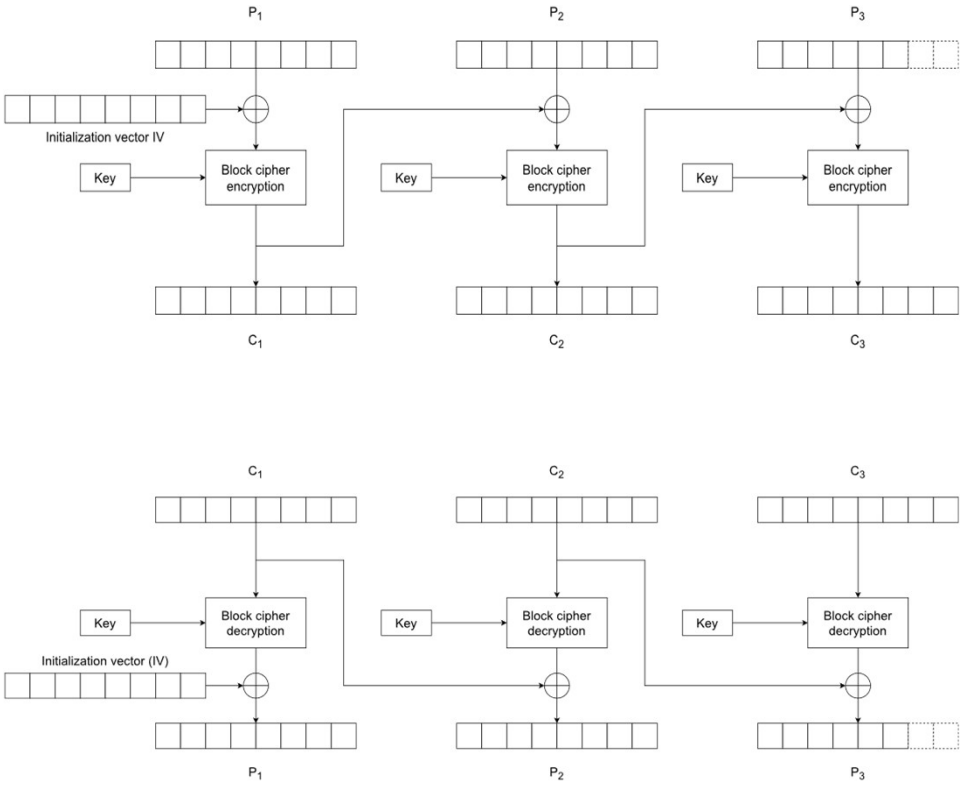
ECB – Electronic Code Book





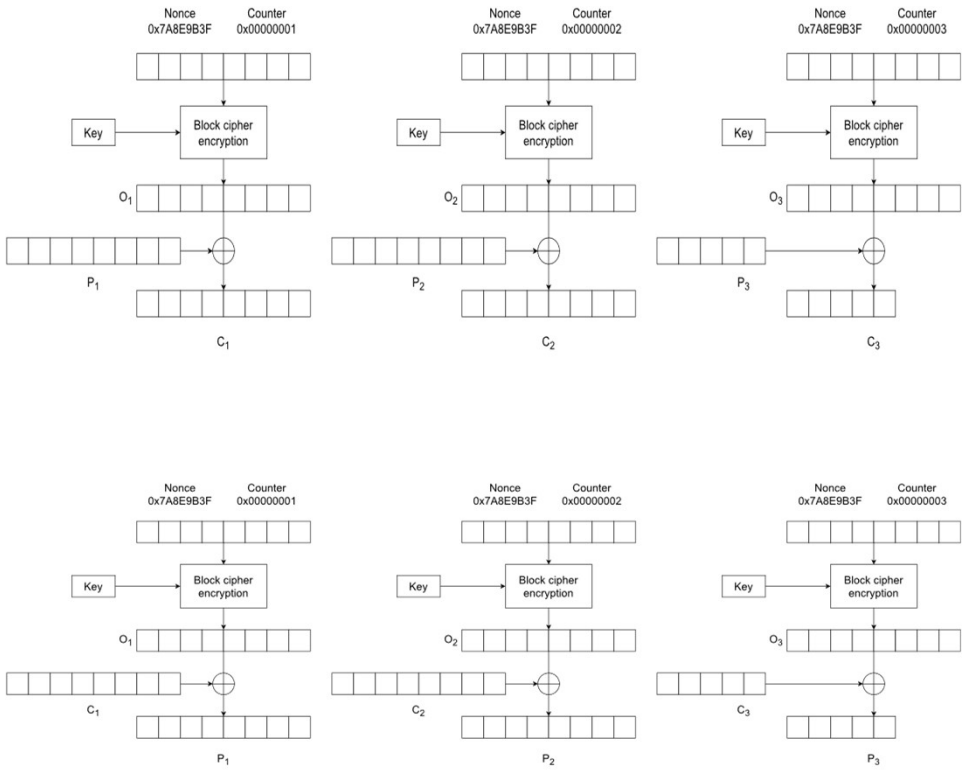
- Yksinkertainen moodi, jossa salalohkot tuotetaan yksinkertaisesti ajamalla selvälohkot suoraan lohkosalaimen läpi
- Viimeinen selvälohko vaatii päddäystä, jos se ei ole täysi lohko (lohkosalain käsittelee ainoastaan täysiä lohkoja)
- Ei diffuusiota. Jos käytettään samaa avainta, identtiset selväkieliset tekstit tuottavat identtiset salatekstit. Myös toistuvat fraasit saman tekstin sisällä tuottavat toistoa salatekstin sisällä
- Ei kannata käyttää missään käytännön sovelluksessa, salateksti paljastaa liikaa selväkielisestä tekstistä
- Klassikkodemonstraatio on kuvan salaaminen ECB-moodilla. Kuvasta näkee helposti mitä salaamaton kuva esittää, vaikka itse lohkosalain olisikin turvallinen (esim. AES)
- $C_i = E_K(P_i), i = 1,2, \dots, n$
- $P_i = D_K(C_i), i = 1,2, \dots, n$

CBC – Cipher Block Chaining



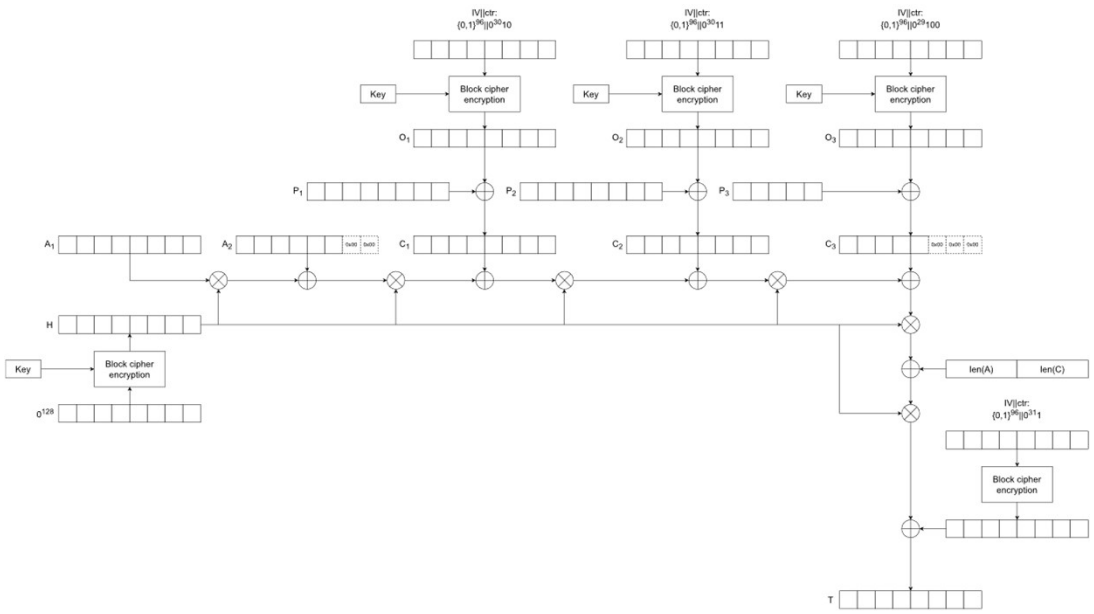
- Salalohkojen ketjutus: jokainen salalohko on riippuvainen kaikista sitä edeltävistä salalohkoista. Ensimmäinen salalohko on riippuvainen alkuarvosta (Initialization vector, IV)
- Ketjutus tapahtuu XOR-operaation avulla
- Vaihtuvan IV:n ansiosta täysin samat selväkieliset tekstit tuottavat täysin erilaiset salatekstit myös saman avaimen alla (lumivyöryefekti)
- IV täytyy generoida huolella. Se ei saa olla ennalta-arvattava, muutoin hyökkääjä voi arvata selkokielisen tekstin sisältöä ja pystyy tarkistamaan arvauksensa oikeellisuuden
- Vaatii päddäystä, joka myöskin voi johtaa tietoturvariskeihin (Padding oracle attack)
- $C_i = E_K(C_{i-1} \oplus P_i), i = 1, 2, \ldots, n, C_0 = IV$
- $P_i = D_K(C_i) \oplus C_{i-1}, i = 1, 2, \ldots, n, C_0 = IV$

CTR – Counter mode



- Tuotetaan avainketju salaamalla kertakäyttöluvun (nonce) ja nousevan laskurin konkatenaatio toistuvasti lohkosalaimella
- Salalohko saadaan XORaamalla selväkielinen lohko vastaavalla avainjonon lohkolla
- Uusi nonce jokaiselle viestille, laskuri kasvaa lohkojen mukana.
  - Huolehdittava, että noncet generoidaan siten, ettei saman avaimen alla käytetä samaa noncea useammin kuin kerran
  - Huolehdittava myös, ettei laskuri kiepsahda ympäri ja ala alusta saman viestin sisällä
- Lohkot käsitellään erikseen, joten rinnakkaislaskenta sekä salauksessa että salauksen purkamisessa mahdollista. Myös avainketju voidaan laskea etukäteen jos laskurin toiminta ja nonce ovat tiedossa
- $C_i = P_i \oplus O_i$, kun $i < n$ and $C_n = P_n \oplus MSB_l(O_n)$
- Tarvitaan ainoastaan lohkosalaimen salaussuunta. XOR-operaation ominaisuuden ansiosta purku tapahtuu käyttämällä samaa avainketjua mutta vaihtamalla selvälohkon ja salalohkon paikkaa. Esim. $C_1 = O_1 \oplus P_1 \Leftrightarrow P_1 = O_1 \oplus C_1$

GCM – Galois/Counter mode



- Yleisesti käytössä esim. HTTPS:ssä
- Salaus ja purku kuten CTR-moodissa
- Lisänä viestien autentikointi
  - Tapahtuu laskemalla autentikointitagi salatuista lohkoista ja mahdollisesta salaamattomasta autentikoitavasta datasta
  - Laskenta perustuu polynomin $x^{128} + x^7 + x^2 + x + 1$ määrittämään äärelliseen (Galois'n) kuntaan. Kaikki 128 bittiset bittijonot voidaan nähdä kunnan alkiona, ja niiden tulot voidaan redusoida korkeintaan 127 asteen polynomeiksi, jolloin tulot ovat myös aina esitettävissä 128 bitin jonoina
  - Autentikoitava data ja salalohkot kerrotaan vakiolla H, tulos XORataan seuraavan lohkon kanssa, tulos kerrotaan H:lla, XORataan seuraavan lohkon kanssa …
- Vastaanottaja voi laskea salalohkoista tagin ja verrata sitä vastaanottamaansa tagiin
- Suunniteltu nimenomaan 128 bitin lohkosalaimille
- Suositus: nonce 96 bittiä, counter 32 bittiä ⇒ salattavan tiedoston maks. koko 64 GiB (n. 68,7 GB)

Moodien vertailu lyhyesti

	ECB	CBC	CTR	GCM
Diffusion	No	Yes	Yes	Yes
Requires IV	No	Yes	Yes	Yes
IV must be unpredictable	-	Yes	No	No
Encryption parallelizable	Yes	No	Yes	Yes
Decryption parallelizable	Yes	Yes	Yes	Yes
Requires padding	Yes	Yes	No	No
Built-in authentication	No	No	No	Yes
Needs inverse of underlying block cipher	Yes	Yes	No	No
Preprocessing possible	No	No	Yes	Yes

- ECB on käyttökelvoton oikeissa sovelluksissa
- CBC on teoriassa ihan kelpo moodi. Implementointi vaatii huolellisuutta IV:n ja päddäyksen vuoksi. Salaus ei rinnakkainlaskettavissa
- CTR toimii hyvin, kunhan laskurin ei anneta mennä ympäri ja noncea ei käytetä uudelleen saman avaimen kanssa. Rinnakkaislaskennan ja esiprosessoinnin mahdollisuus tekee siitä nopean
- GCM on näistä neljästä selvästi paras. Se yhdistää CTR:n hyviin ominaisuuksiin autentikoinnin

Yhteenveto

- Kannattaa aina olettaa, että kaikki ulkopuoliset tahot ovat mahdollisesti pahantahtoisia, fiksuja ja suurella laskentateholla varustettuja
- Kryptografia kannattaa ottaa vakavasti, eikä kannata koodata itse salausmenetelmiä. Käytä aina julkisia yleisesti hyväksyttyjä kirjastoja ja pysy ajan hermolla
- Dataliikenne tulee kasvamaan ja yhä useampi yksityiselämän ja yhteiskunnan toiminta digitalisoituu
- Turvallinen salaus tulee olemaan aina vain merkittävämmässä roolissa

Kiitos!

Työssä käytetyt lähteet:

Bhargavan, K., Leurent, G. 2016. On the Practical (In-)Security of 64-bit Block Ciphers. Referenced 11.10.2023. Available at https://sweet32.info/SWEET32_CCS16.pdf

Brady, S. 2021. Authenticated Encryption in .NET with AES-GCM. Referenced 14.10.2023. Available at https://www.scottbrady91.com/c-sharp/aes-gcm-dotnet

Cook, J. Finite fields. Referenced 14.10.2023. Available at https://www.johndcook.com/blog/finite-fields/

Dazell, T. 2012. Many Time Pad Attack - Crib Drag. Referenced 16.10.2023. Available at http://travisdazell.blogspot.com/2012/11/many-time-pad-attack-crib-drag.html

Dworkin, M. 2001. Recommendation for Block Cipher Modes of Operation: Methods and Techniques. NIST. Referenced 3.11.2023. Available at https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

Heaton, R. 2013. The Padding Oracle Attack. Referenced 14.10.2023. Available at https://robertheaton.com/2013/07/29/padding-oracle-attack/

Hornsby, T. 2013. Encryption - CBC Mode IV: Secret or Not? Referenced 15.10.2023. Available at https://defuse.ca/cbcmodeiv.htm

Janssen, M., Lindsey, M. 2019. Rings with Inquiry. Referenced 14.10.2023. Available at https://ringswithinquiry.org/rwi/SubSec-Fields.html

McGrew, D., Viega, J. 2005. The Galois/Counter Mode of Operation (GCM). Referenced 28.10.2023. Available at https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf

NIST. 2001. Announcing the Advanced Encryption Standard (AES). Referenced 3.11.2023. Available at https://csrc.nist.gov/files/pubs/fips/197/final/docs/fips-197.pdf

NIST. a. Computer security resource center. Glossary. Mode of operation. Referenced 3.11.2023. Available at https://csrc.nist.gov/glossary/term/mode_of_operation

NIST. b. Computer security resource center. Glossary. Block cipher. Referenced 3.11.2023. Available at https://csrc.nist.gov/glossary/term/block_cipher

Rogaway, P. 2011. Evaluation of Some Blockcipher Modes of Operation. Referenced 3.11.2023. Available at https://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf

Saarinen, M-J. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. Referenced 3.11.2023. Available at https://eprint.iacr.org/2011/202.pdf

Unicode.org. Frequently asked questions. UTF-8, UTF-16, UTF-32 & BOM. Referenced 11.10.2023. Available at https://unicode.org/faq/utf_bom.html

Villanueva, J. An Introduction to Stream Ciphers vs. Block Ciphers. Referenced 11.10.2023. Available at https://www.jscape.com/blog/stream-cipher-vs-block-cipher

Wikipedia. Block cipher mode of operation. Referenced 5.11.2023. Available at https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Wells, A. 2021. Cryptography - PKCS#7 Padding. Referenced 14.10.2023. Available at https://node-security.com/posts/cryptography-pkcs-7-padding/