AES: oorsprong

- National Institute of Standards and Technology (NIST) publiceert nieuwe standaard in 1999: gebruik 3DES ipv DES
- voordelen:
  - sleutel 168 bits (3x56) vermijdt brute-force attacks
  - onderliggend algoritme = DES
    - sterk geviseerd over lange periode
    - geen aanval via cryptanalyse
    - 3DES dus ook resistent beschouwd tegen cryptanalyse

  ➔ indien alleen veiligheid als criterium: 3DES nog lang gebuikt
- nadelen
  - traag algoritme in software (doel hardware-implementatie)
  - block-size: 64 bits (size>64 is nodig vr efficiëntie en veiligheid)

AES: oorsprong (2)

- AES = Advanced Encryption Standard
- moet op lange termijn DES vervangen
- NIST doet *call for proposals* in 1997
  - conventioneel
  - lengte blok=128
  - sleutel=128, 192, 256 bits
- 21 voorstellen
  - gereduceerd tot 15, later 5
  - volgens volgende criteria
- uiteindelijke keuze en standaard vastgelegd in 2001

Criteria AES

- algemene veiligheid
  - voorstellen getest door cryptografische gemeenschap
  - vooral nadruk op gekende aanvallen
  - uitgevoerde cryptanalyse beperkt in tijd (3 jaar) (vgl DES)
  - gekozen versie van AES wordt wellicht extensiever getest
  - tot nu toe geen gekende geslaagde aanval op Rijndael
- software-implementatie
  - snelheid van uitvoering
  - platformonafhankelijkheid van performantie
  - variatie van de snelheid ifv sleutelomvang
  - bij Rijndael
    - geen verschil tussen encryptie en decryptie
    - goede score op 8-bits tot 64 bits en DSP's
    - langere sleutels verminderen snelheid (# ronden stijgt)
    - laat parallelle uitvoering toe

Criteria AES (2)

- omgevingen met beperkte ruimte
  - vb. smartcards met weinig geheugen (zowel ROM als RAM)
  - opslag van
    - code
    - S-boxen
    - deelsleutels
  - Rijdael scoort goed, ook omwille van slechts of encryptie of decryptie (nooit beide)
- hardware-implementatie
  - snelheid: kan ook geoptimaliseerd worden (net als software)
  - omvang: vertaalt zich wel direct in kostprijs (niet in software)
  - Rijdael
    - snelste doorstroom in feedback mode
    - tweede snelste in niet-feedabck mode

Criteria AES (3)

- aanvallen op implementeringen
  - timing: vb. tijdsverschil tussen verschillende sleutels en PT's
  - verbruikte energie o.a. in smartcard
    - afhankelijk van uitgevoerde instructie
    - afhankelijk van data
    - vb. schrijven van 1 kost meer energie dan van 0
  - Rijndael scoort beter bij bescherming dan concurenten
- encryptie versus decryptie
  - indien verschillend, extra plaats nodig voor 2 algoritmen
  - indien verschillend, is ook benodigde tijd verschillend?
  - bij Rijndael zijn beide verschillend
    - plaats encr+decr = encr + 60%
    - snelheid ongeveer gelijk (deelsleutels decryptie trager)

Criteria AES (4)

- *key agility*
  - snelheid, behendigheid, …
  - refereert naar:
    - wisselen van sleutels: snel, met minimum resources
    - berekenen van deelsleutels of nieuwe sleutels ifv voorgaande
  - van belang indien weinig plaitext met zelfde sleutel wordt geëncrypteerd
  - bij Rijndael
    - encryptie: on-the-fly berekenen van deelsleutels ➔ ok
    - decryptie: alle deelsleutels nodig vóór eerste ronde

Criteria AES (5)

- algemene flexibiliteit
  - mogelijke aanpassing van algoritme aan
    - sleutelomvang, blokgrootte, aantal ronden (gevolg van aanval)
    - mogelijke optimalisatie voor bepaalde omgevingen
  - Rijndael laat blokomvang en sleutelomvang toe van 128, 192 en 256 bits
    - kan inprincipe elk veelvoud van 32 bits aan
- potentieel parallellisme
  - belangrijk indien meerdere processoren of indien processor met ILP (*instruction level parallelism*)
  - Rijndael voldoet hier uitstekend aan

AES = Rijndael

- Rijndael =
  - Joan Daemen
  - Vincent Rijmen
- (ex-)onderzoekers van KUL
- geselecteerd uit 21 kandidaten
- Eigenschappen:
  - weerstand tegen alle gekende aanvallen
  - snelheid
  - compactheid van code
  - op velerlei platformen
  - eenvoud van ontwerp

Parameters (bits)

sleutellengte	128	192	256
bloklengte	128	128	128
aantal ronden	10	12	14
rondesleutel	128	128	128
totale uitgebreide sleutel (bytes)	176 B 11x128 / 8	208 B	240 B

AES - IDEA - sleutelverdeling 9

Deelsleutels

- Principes
  - aantal bits totale sleutel:
    sleutellengte (128) x (aantalRonden +1 = 11) = 1408
  - cipher-sleutel wordt geëxpandeerd
  - rondesleutel wordt telkens genomen uit volgende bits van uitgebreide sleutel
- Principes voor de uitbreiding :
  - originele sleutel + 10 streches (stukken) met lengte van originele sleutel
  - in blokken van 32 bits = 4 bytes = woord
  - 44 blokken van 32 bits = 1408 bits of 176 bytes
  - beschouw tabel: *word W[44]*

AES - IDEA - sleutelverdeling 10

Deelsleutels (2): expansion

W[0..3] = key

voor i = 4 tot 43:

`W[i] = W[i-4] xor tmp`

- i geen veelvoud van 4: `tmp = W[i-1]`
- i veelvoud van 4: vertrek van W[i-1] voor tmp
  - left shift over 1 byte
  - pas voor elke byte S-box toe (byte substitutie)
  - 1ste byte xor constante afhankelijk van ronde (RC)
    RC[1]= $(01)_{16}$
    RC[ronde] = RC[ronde-1] ● 2
    1 2 4 8 16 32 64 128 27 54
    ● is de vermenigvuldiging gedefinieerd in veld GF(2^8)

Opbouw S-box

- matrix
  - 16x16 bytes
  - rijen en kolommen genummerd $(0..F)_{16}$

1. initialiseer matrix met waarden $(xy)_{16}$ op plaats x,y
2. vervang elke byte door zijn multiplicatief invers in GF(2^8)
   - x=1/b mod 2^8
   - 00 wordt door zichzelf vervangen
3. elk bit b_i van elke byte in matrix wordt als volgt getransformeerd b'_i:
   - 1 byte = 8 bits voorgesteld door $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$
   - $(63)_{16} = (0110\ 0011)_2$ = c = $c_7, c_6, c_5, c_4, c_3, c_2, c_1, c_0$

$$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

Overzicht

Afkortingen:

ARK	Add Round Key
BSB	Byte SuBstitute
SR	Shift Row
MC	Mix Columns

Encryptie:

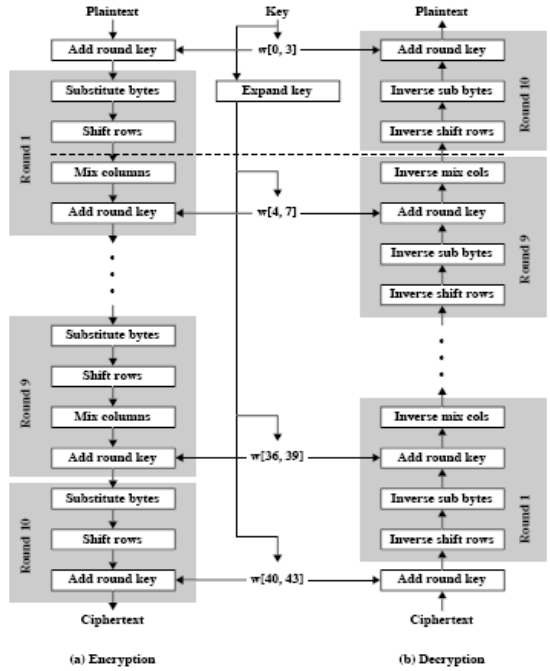
ARK

9 x (BSB – SR – MC – ARK)

BSB – SR – ARK

Decryptie: “omgekeerde volgorde”

AES - IDEA - sleutelverdeling 13



14

ARK - BSB

ARK 4 blokken van geëxpandeerde sleutel
xor
128 bits getransformeerde plaintext

BSB door middel van S-box
16 x 16 byte waarden, alle wrden van 8 bits (256)
elke invoerbyte = 4 + 4 bits
eerste 4 = rij
tweede 4 = kolom
uitvoer = element[rij,kolom]

Shift Rows

1 blok = 1byte, genummerd van 1 tot 16 (=128 bits)

– organisatie in tabel

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

– elke byte-rij n keer geroteerd (n=rijnummer)

1	5	9	13
6	10	14	2
11	15	3	7
16	4	8	12

Mix Columns

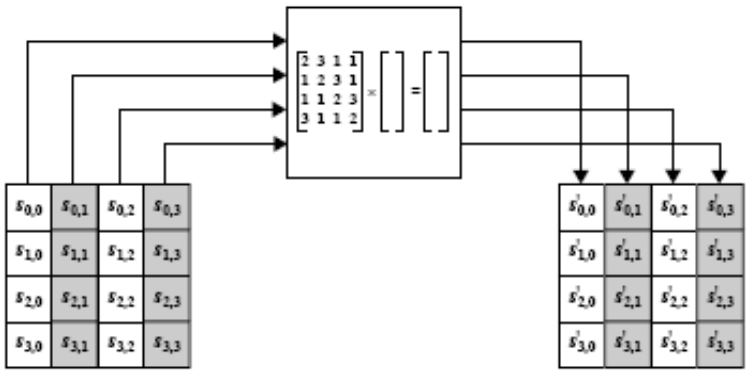
Elke kolom in SR wordt vermenigvuldigd met matrix
(figuur en uitwerking: zie volgende slide)
Elk nieuw byte is functie van 4 bytes uit kolom
x = vermenigvuldiging in GF(2^8)
+ = xor = optelling in GF(2^8)

Byte wordt bekeken als **polynoom**, niet als getal
(cfr CRC)
als extra bits (meer dan 8):

- niet weggelaten !!
- resultaat `xor 1 0001 1011`

Mix Columns (2)

Elke kolom in SR wordt vermenigvuldigd met matrix



Decryptie

Encryptiesequentie:

A BSMA BSMA … BSMA BSA

Decryptiesequentie is omgekeerde encryptievolgorde (z.fig.)

A SBAM SBAM … SBAM SBA

Omwisselen van BSB en SR verandert resultaat niet ! (encr)

A BSAM BSAM … BSAM BSA

Gelijkaardig behalve ronde-afbakening

MA wordt AM

moet niet worden omgewisseld omwille van distributiviteit van matrixvermenigvuldiging

Decryptie (2)

dus decryptievolgorde = encryptievolgorde

ARK = xor , blijft zelfde

MC: geïnverteerde matrix

BSB: inverse S-box

SR: inverse shift-rows

Bovendien:

deelsleutels in omgekeerde volgorde nodig

elke deelsleutel ondergaat nog een extra inverse MC

IDEA

- International Data Encryption Algorithm
- ontwikkeld in Zürich, 1990
- block-cipher
- sleutellengte 128 bits
- datablokken 64 bits
- 2 sleutelwoorden:
  - confusion
    CT ingewikkeld afhankelijk van PT en key
    statistische afhankelijkheid van CT en PT moeilijk te bepalen
  - diffusion
    elk bit PT en in key beïnvloedt alle bits in CT
- ooit voorgesteld als vervanger van DES, nu nog in PGP

IDEA: confusion

anders dan bij DES
- XOR
- kleine S-Boxen

IDEA: mix van 3 operaties op 2 inputs van 16 bits
- XOR
- optelling modulo 2^{16} 
- vermenigvuldiging van integers modulo $(2^{16}+1)$ 
- geen distributiviteit
- geen associativiteit

IDEA: vermenigvuldiging ⊙

vermenigvuldiging van integers modulo $(2^{16}+1)$

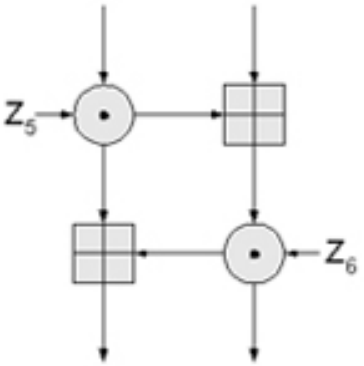
- invoer en uitvoer zijn 16-bits integers zonder teken
  - uitgezonderd blok van allemaal nullen
    0000 0000 0000 0000
    behandeld alsof = 2^{16}
- stel 2 bits ipv 16 ➔ modulo 5 (= 2^2+1)
- 00 ⊙ 10 = 11 (4 x 2 = 8 mod 5 = 3)

IDEA: diffusion

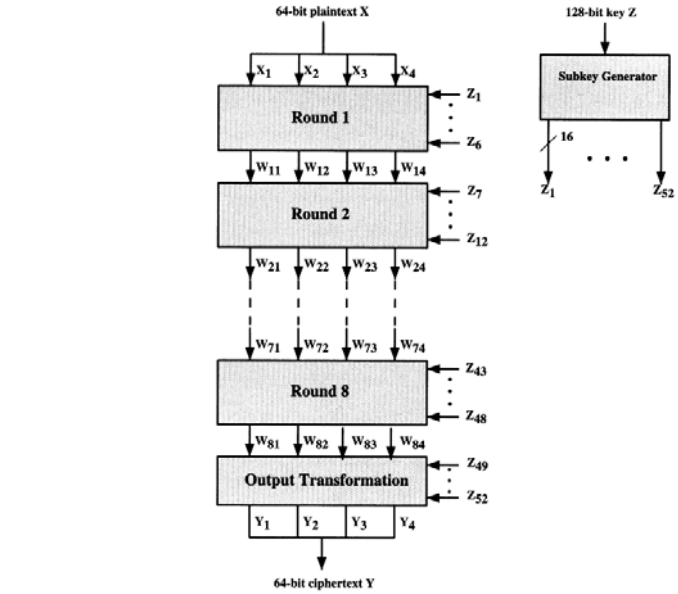
bouwsteen met

- 2x16 bits invoer
- 2 deelsleutels

wordt 8x toegepast

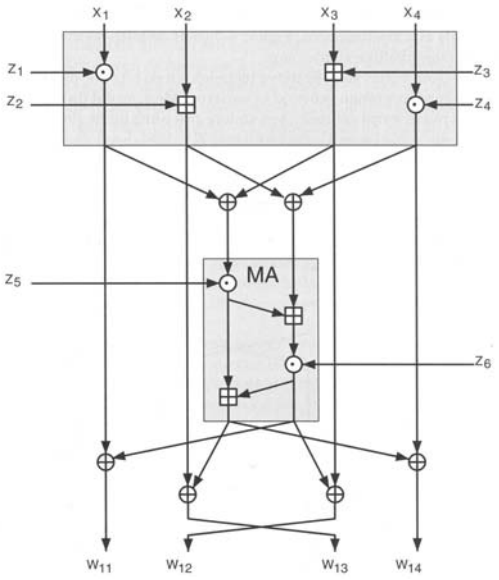


IDEA overzicht



25

IDEA: 1 ronde



26

IDEA overzicht

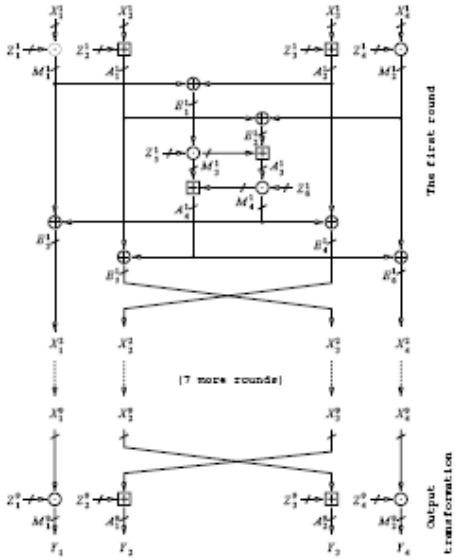


Fig. 1. The schematics of IDEA

27

Andere conventionele algoritmen

- RC5
  Ronald Rivest 1994
- RC2
  RSA 1996
- RC4
  stream cipher vergelijkbaar met RC2

- CAST-128
- Blowfish

AES - IDEA - sleutelverdeling 28

Gebruiksmodi voor block ciphers

Gelden voor ALLE conventionele algoritmen

2 block modes

- ECB electronic code book
- CBC cipher block chaining

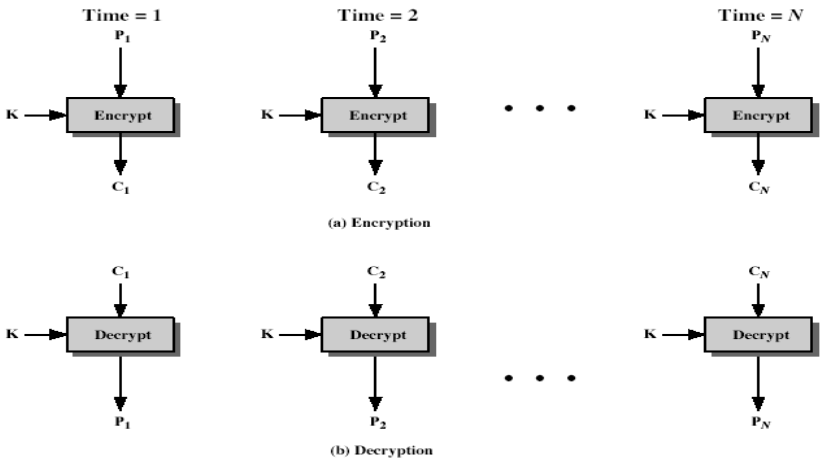
2 stream modes

- CFB cipher feedback
- OFB output feedback

1 speciale modes

- CTR counter mode

ECB: electronic code book



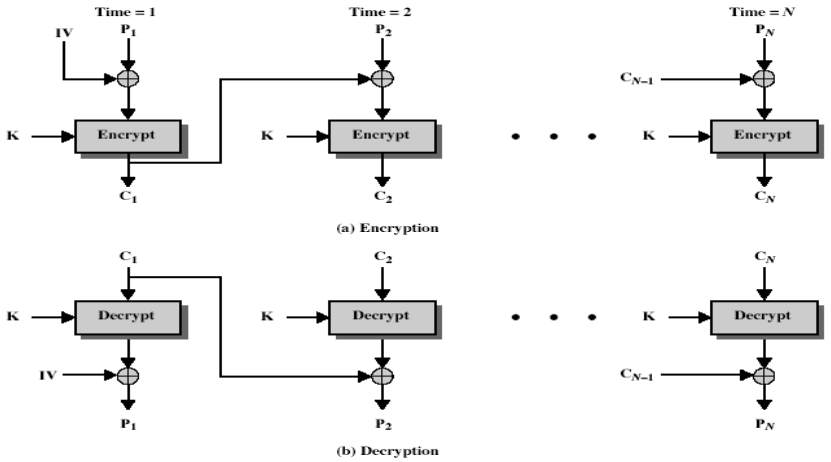
ECB: electronic code book

- boodschap wordt opgesplitst in onafhankelijke blokken die geëncrypteerd worden
- elk blok = waarde die gesubstitueerd wordt, zoals in een codeboek
- elk blok wordt onafhankelijk van de andere geëncrypteerd
  - $C_i = Encr_K (P_i)$

ECB: voor- en nadelen

- gelijke blokken PT ➔ identieke blokken CT
  - lange berichten met repetitieve elementen
  - door cryptanalyse sleutel te achterhalen
- gebruik: beveiligde overdracht van enkele waarden, vb. encryptiesleutel
- goed voor overdracht via kanaal met ruis:
  - 1 foutief blok blijft beperkt tot dat ene blok

CBC: cipher block chaining



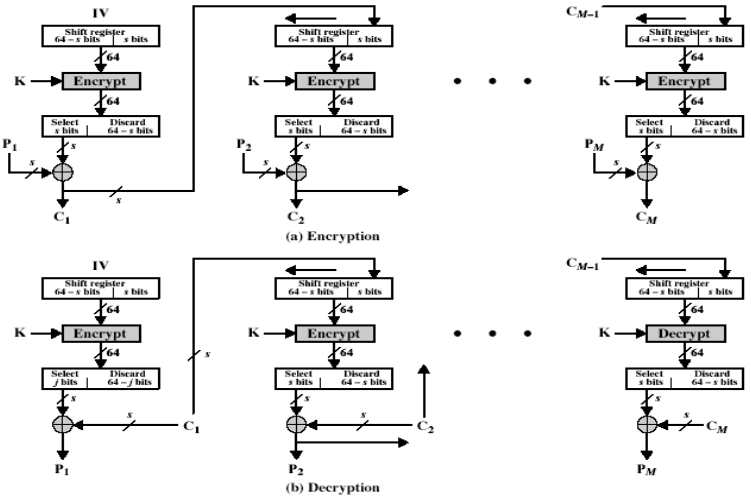
CBC: cipher block chaining

- boodschap wordt in blokken verdeeld
- elk vorig CT-blok verbonden met nieuwe PT-blok
  - vandaar de naam
- gebruik van Initiële Vector, Initial Value (IV) om proces te starten
  - $C_i = Encr_K(P_i \text{ xor } C_{i-1})$
  - $C_{-1} = IV$
- gebruikt bij
  - grote hoeveelheden gegevens
  - authenticatie

CBC: voor- en nadelen

- elk blok CT hangt af van alle blokken PT
- gelijke blokken PT ➔ verschillende blokken CT
- verandering in boodschap verandert de volledige cipherthext
- Initial Value (IV) moet gekend zijn bij zender en ontvanger
  - ofwel vaste waarde
  - ofwel verzonden in ECB voor de rest van de boodschap
- als tekort blok op einde bericht
  - aanvullen met gekende waarde (vb. nullen)
  - aanvullen met aantal opgevulde bytes
    vb. [b1 b2 b3 0 0 0 0 5] → 3 data bytes, dan 5 bytes pad+aantal

CFB: cipher feedback



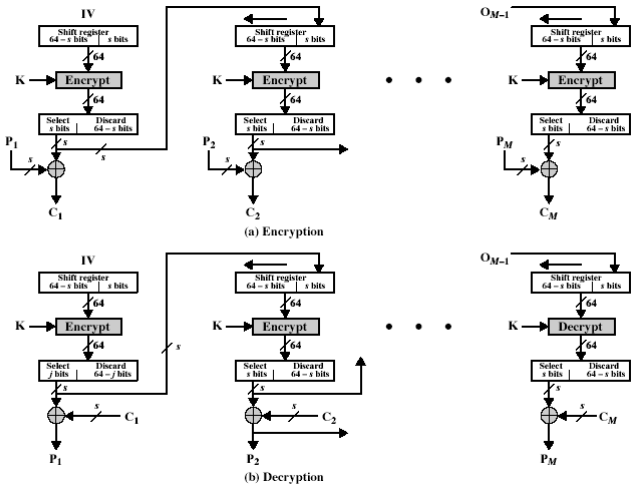
CFB: cipher feedback

- boodschap is stroom van bits
- wordt opgeteld (xor) bij output van block cipher
- resultaat teruggekoppeld bij volgende stap (vandaar de naam)
- de standaard laat willekeurig aantal bits toe
  - vb. 1,8, 64, … dat kan teruggekoppeld worden
  - genoteerd als CFB-1, CFB-8, CFB-64, etc.
- meest efficiënt is gebruik van alle 64 bits (CFB-64)
  - $C_i = P_i$ xor $Encr_K(C_{i-1})$
  - $C_{-1} = IV$
- gebruik: encryptie van stroom data, authenticatie

CFB: voor- en nadelen

- geschikt indien gegevens arriveren in bits of bytes
- meest gebruikte *stream mode*
- block cipher altijd in encryptie-mode gebruikt !
- fouten propageren

OFB: output feedback



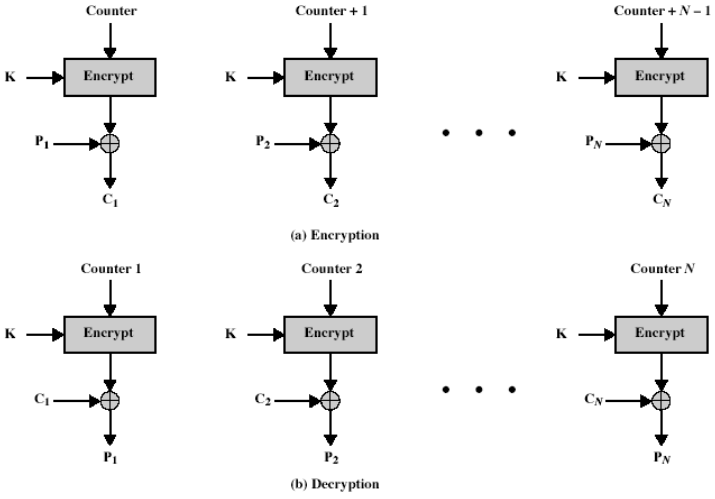
OFB: output feedback

- boodschap is stroom van bits
- wordt opgeteld (xor) bij output van block cipher
- output is teruggekoppeld bij volgende stap (vandaar de naam)
- de standaard laat willekeurig aantal bits toe
- feedback is onafhankelijk van de boodschap
- kan vooraf berekend worden
  - $C_i = P_i$ xor O_i
  - $O_i = Encr_K(O_{i-1})$
  - $O_{-1} = IV$
- gebruik: encryptie van stroom data over kanaal met ruis

OFB: voor- en nadelen

- gebruikt waar encryptie moet gebeuren vooraleer boodschap beschikbaar is
- oppervlakkig gelijkaardig aan CFB, maar
  - feedback komt van cipher output
  - feedback is onafhankelijk van boodschap
- zenden en ontvangen moet gesynchroniseerd gebeuren, eventueel voorzien in herstelprocedure
- research heeft aangetoond dat best uitsluitend **OFB-64** zou gebruikt worden

CTR: counter



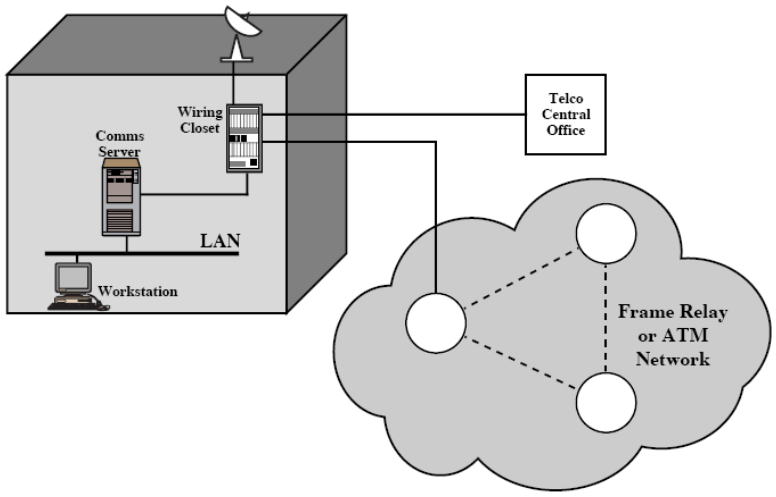
CTR: counter

- nieuwe methode, hoewel reeds vroeger voorgesteld
- gelijkaardig aan OFB, maar tellerwaarde wordt geëncrypteerd i.p.v. een feedback waarde
- moet gebruik maken van een verschillende tellerwaarde voor elke blok PT
  - $C_i = P_i$ xor O_i
  - $O_i = Encr_K(i)$
- gebruik: encryptie op hogesnelheid netwerklijnen zoals ATM en IPSec

CTR: voor- en nadelen

- efficiëntie
  - er kan parallel geëncrypteerd worden
  - eventueel vooraf indien nodig
  - goed voor high speed links die in burst data vervoeren
- CT data blokken zijn random toegankelijk
- even goed als de andere modes
  - geen hergebruik van tellerwaarden voor zelfde blokken PT

Waar encrypteren?



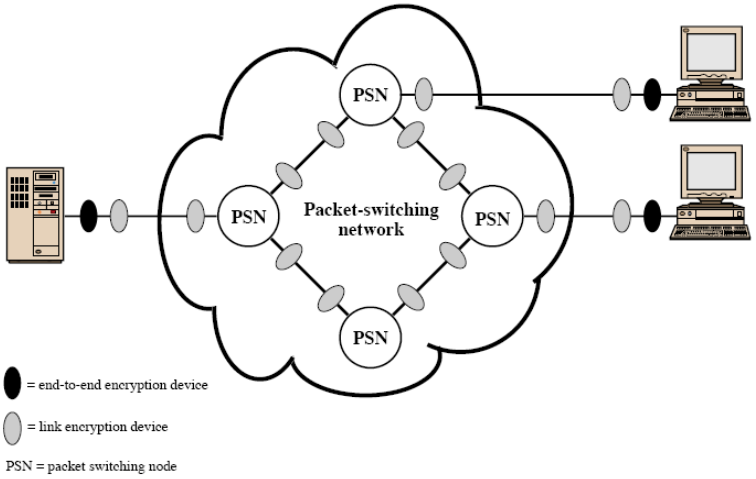
AES - IDEA - sleutelverdeling 45

Kwetsbare plaatsen

- typisch scenario
  - werkstations op LANs hebben toegang tot andere werkstations & servers op het netwerk
  - netwerken zijn
    - onderling verbonden door switches/routers
    - extern verbonden via radio of satelliet
- mogelijke aanvallen in dit scenario
  - afluisteren van werkstations met gebruik van
    - server
    - router/switch
  - monitoren en/of wijzigen van berichten

AES - IDEA - sleutelverdeling 46

Link ←→ end-to-end encryptie



AES - IDEA - sleutelverdeling 47

Link-encryptie

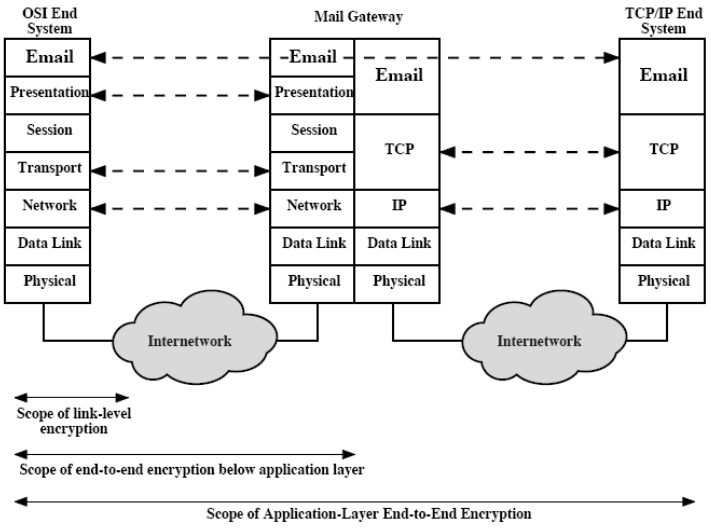
- encryptie gebeurt onafhankelijk op elke verbinding
- decryptie van verkeer op knooppunten
  adres in pakkethoofding
- vereist vele devices, maar met elk een sleutelpaar
- elk tussenstation moet deelnemen
- op laag 1 of 2 (fysische of datalink laag)

AES - IDEA - sleutelverdeling 48

end-to-end encryptie

- encryptie tussen originele bron en eindbestemming
- devices op elk uiteinde met elk een gedeelde sleutel
- welke gegevens coderen?
  - hoofdingen niet → adressen moeten zichtbaar blijven
  - trafiekpatroon is niet beveiligd
- garandeert zekere vorm van authentisering
- mogelijk op lagen 3, 4, 6, 7
  netwerklaag (3, TCP, X25) … applicatielaag
- hoe hoger, hoe minder info beveiligd
- mail: alleen mogelijk op applicatieniveau (hoofdingen moeten zichtbaar blijven)

Bereik van encryptie



Sleutelverdeling

- conventionele encryptieschema's vereisen dat beide partijen (zender en ontvanger) over zelfde sleutel beschikken
- hoe kan deze sleutel veilig verdeeld worden?
- vaak wordt op een veilig systeem ingebroken door een inbraak in het schema van de sleutelverdeling

Sleutelverdeling: algemeen

Mogelijkheden:

- A kiest een sleutel en levert hem **fysiek** af aan B
- derde partij C kiest een sleutel en levert hem **fysiek** af aan A & B
- indien A & B een sleutel delen (voordien uitgewisseld):
  - deze sleutel wordt gebruikt om nieuwe sleutel geëncrypteerd uit te wisselen
- indien A & B beveiligd verbonden zijn met derde partij C:
  - C kan nieuwe sleutel geëncrypteerd doorsturen naar A en B

Sleutelverdeling: wanneer / wat?

- optie 1 en 2 vereisen manuele tussenkomst
  - redelijk bij link-encryptie
  - niet mogelijk bij end-to-end encryptie
    - N hosts ➔ N(N-1)/2 sleutels
    - 10000 ➔ 50.000.000
- optie 3 kan voor beide soorten encryptie
  - evenwel gevaar indien 1 sleutel ontdekt ➔ alle volgende sleutels gekend
- ➔ beter scenario?

KDC

Key Distribution Center

- variant van optie 4
- verspreidt sleutels voor paren gebruikers (hosts, processen, toepassingen)
- elke gebruiker deelt unieke sleutel met KDC
  - master key
  - N hosts ➔ N sleutels
  - eerste is fysiek uitgewisseld
- KDC deelt *sessiesleutels* uit
  - beperkt tot de duur van de logische connectie
  - worden geëncrypteerd met de *master key*

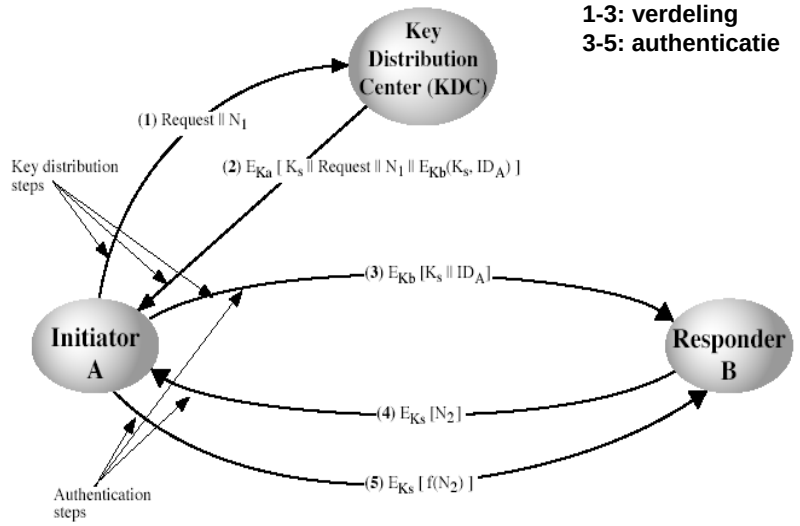
KDC scenario (zie figuur)

A wil verbinding met B

- A vraagt sessiesleutel K_S aan KDC
  - bericht bevat nonce N_1 (“voor de gelegenheid”, random getal)
  - voor elke aanvraag andere waarde
- bericht van KDC naar A
  - geëncrypteerd met K_A
  - bevat sessiesleutel, N_1 + aanvraag (controle door A), bericht voor B
- bericht voor B
  - geëncrypteerd met K_B
  - bevat sessiesleutel, ID_A
- A stuurt dit bericht door naar B
- B en A wisselen nonce uit geëncrypteerd met K_S

Key Distribution Scenario

1-3: verdeling
3-5: authenticatie



Sleutelverdeling: opmerkingen

- verdeling moet niet beperkt worden tot 1 KDC
  - in groter netwerk: hiërarchie van KDC's
  - lokale KDC staat in voor lokale verbindingen
  - verbinding tussen 2 domeinen:
    - 2 lokale en 1 globale KDC
    - kunnen elk sleutel kiezen
    - moeten mekaar vertoruwen
- levensduur sessiesleutels moet beperkt zijn
  - hoe korter, hoe veiliger
  - vb. duur van de connectie
  - bij transactieprotocol: best wisselen bij elke uitwisseling of na bepaald aantal uitwisselingen
- KDC moet te vertrouwen zijn!!

Gedecentraliseerde uitwisseling

- vereist eenmalige uitwisseling van N(N-1)/2 master keys tussen N hosts
- sessiesleutel kan als volgt bekomen worden: