Forelesning 3

Konvensjonell kryptering:
Algoritmer
Konfidensialitet

De mest brukte algoritmene idag

- Triple DES
- IDEA
- Blowfish
- RC5
- CAST
- RC2

Kommer:
AES

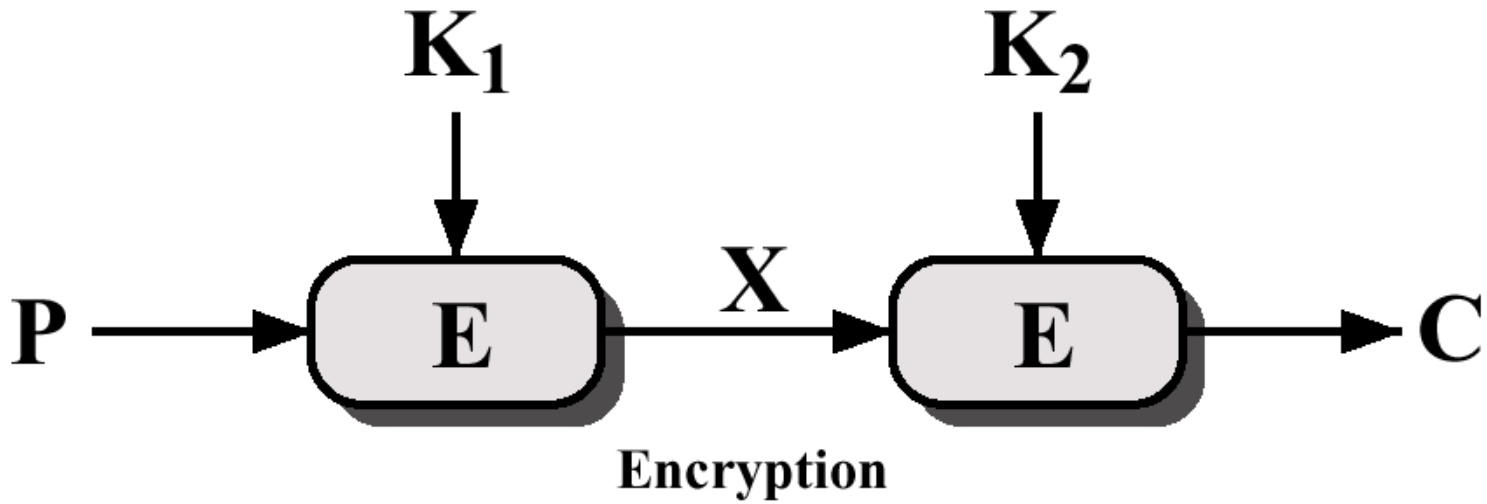
En liten advarsel

- Stallings bruker til enhver tid samme notasjon som designeren av en gitt algoritme har brukt i sin beskrivelse
- Intet forsøk på en enhetlig notasjon i beskrivelsene!
- Caveat studiosis!

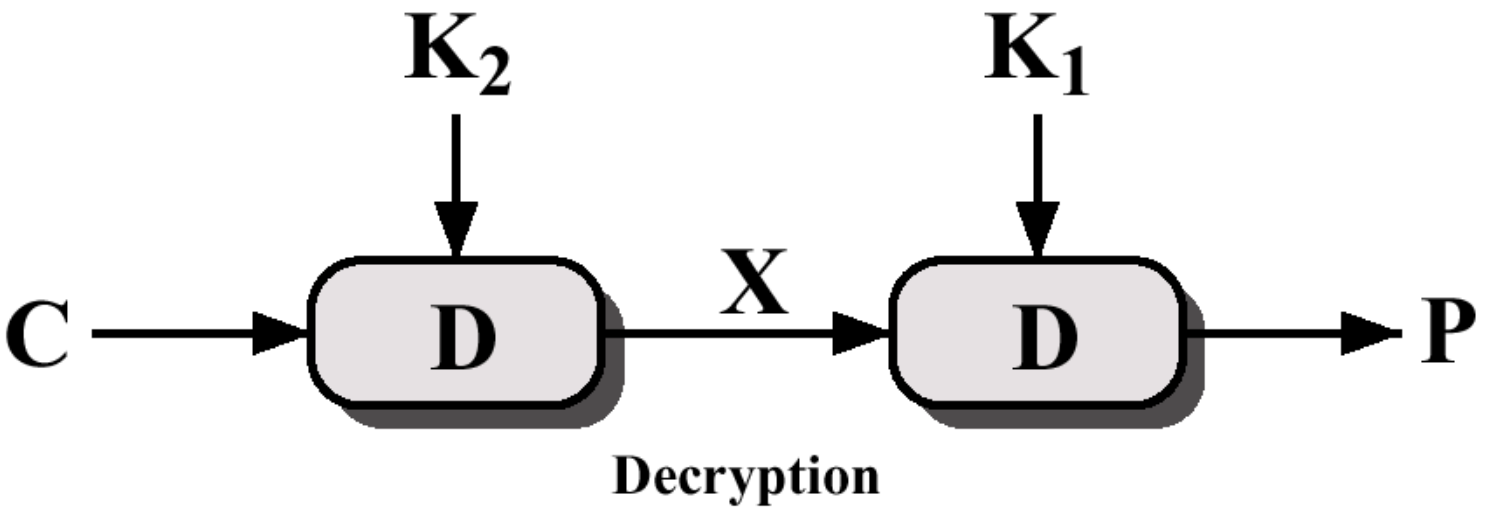
DES har for kort nøkkel!

- Dette har vært akseptert lenge, selv om det er bare i de senere årene at vi har sett at DES kan knekkes i praksis i løpet av et fåtall timer
- Ønskelig å utnytte det som fantes av hardware og infrastruktur for DES til å tilby en algoritme med økt nøkkellengde

Double DES



Encryption



Decryption

Hva med Double DES ?

- $C = E_{K2}[E_{K1}(P)]$
- 56+56 = 112 bits nøkkel
- $\exists K_3 \mid E_{K3}[P] = E_{K1}[E_{K2}(P)]$?
- Alle permutasjoner av 64 bit:
  $(2^{64})! > 10^{10^{20}}$ mappinger
- En mapping pr nøkkel:
  $2^{56} < 10^{17}$ mappinger
- Ingen slik K_3 finnes!

Meet-in-the-middle angrep

- $C = E_{K2}[E_{K1}(P)]$
- $X = E_{K1}[P] = D_{K2}[C]$
- Med et kjent (P,C) par:
  - Krypter P med alle 2^{56} mulige K_1
  - Dekrypter C med alle 2^{56} mulige K_2
  - For hver dekrypterte C, sjekk for match med kryptert P

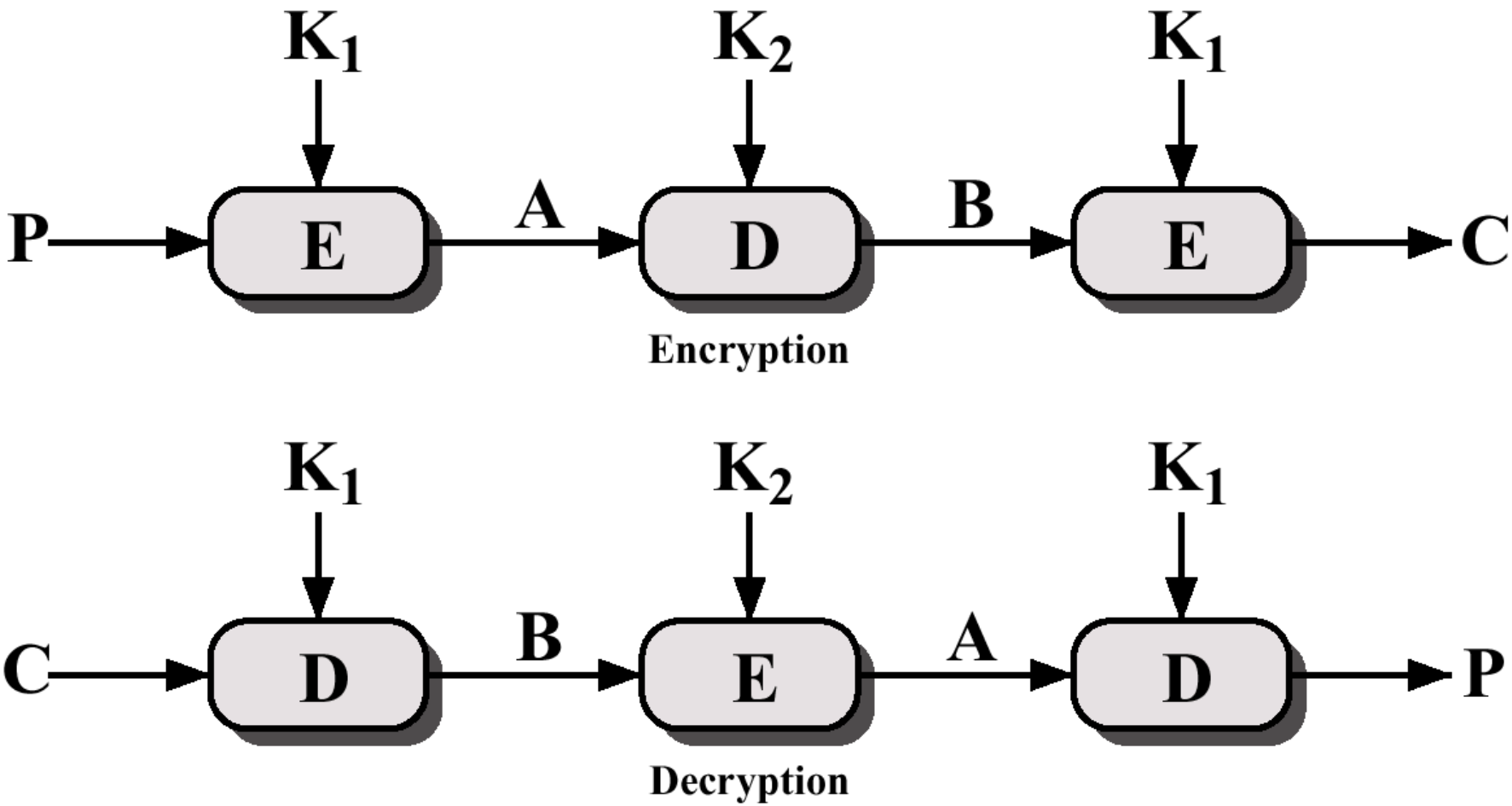
Meet-in-the-middle eksempel

K_1	$E_{K1}[P]$	$D_{K2}[C]$	K_2
1	a	z	i
2	b	h	ii
3	c	p	iii
4	d	q	iv
5	e	b	v
6	f	t	vi
...	...	...	...

Meet-in-the-middle forts.

- Prosedyren gir 2^{48} falske alarmer med ett kjent (P,C) par
- Med to kjente (P,C) par finner man rett nøkkel med en sannsynlighet på $1-2^{-16} = 0{,}999985$
- Kan knekke Double-DES med kjent klartekst med $\approx 4 * 2^{56}$ operasjoner (mye dårligere enn de 2^{112} vi kanskje kunne forvente)

Triple DES



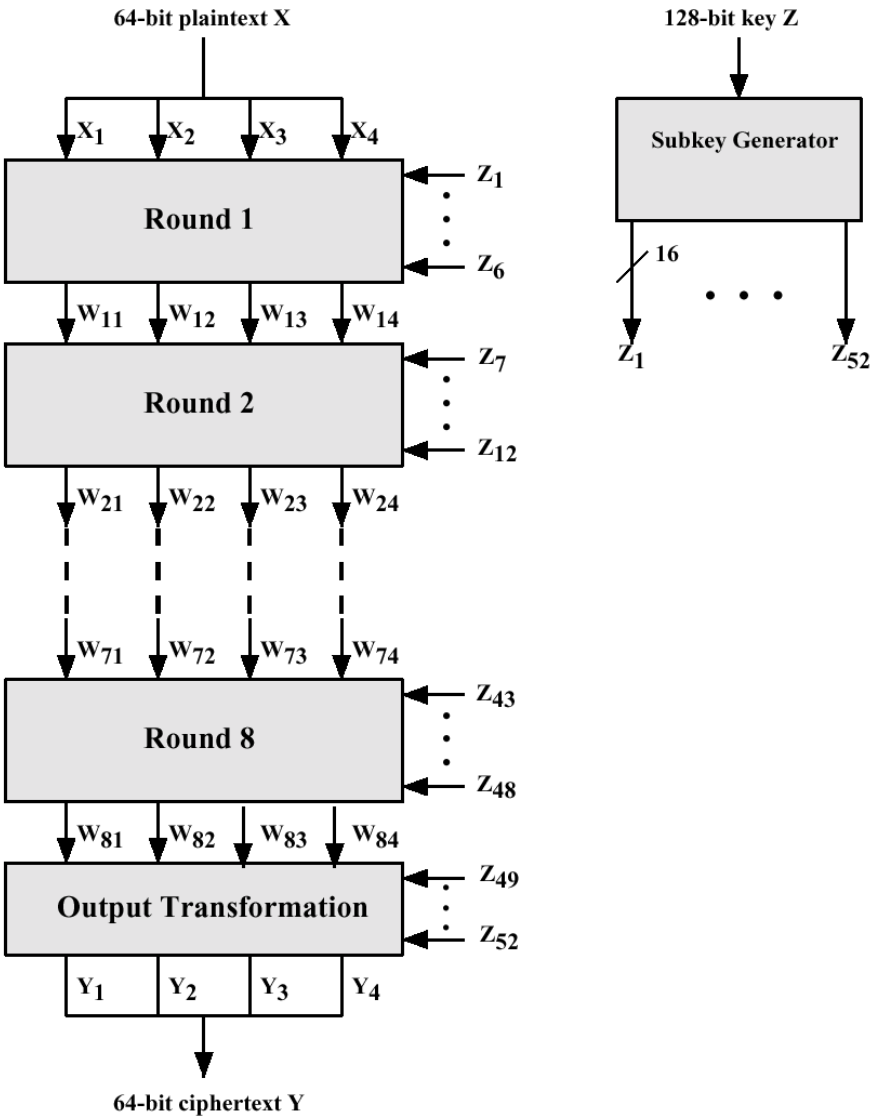
Triple DES

- Tre nøkler ansett som uhåndterlig, to nøkler kompromiss
- Krypter med den første nøkkelen, dekrypter med den andre, og så krypter med den første igjen
  $C = E_{K1}[D_{K2}(E_{K1}[P])]$
- Nøkkellengde 56+56 = 112 bit
- Mye bedre motstand mot m-i-t-m

IDEA

- International Data Encryption Algorithm
- 128 bit nøkkellengde
- 64 bit blokklengde
- 16-bits operasjoner internt
- IKKE et Feistel-chiffer!

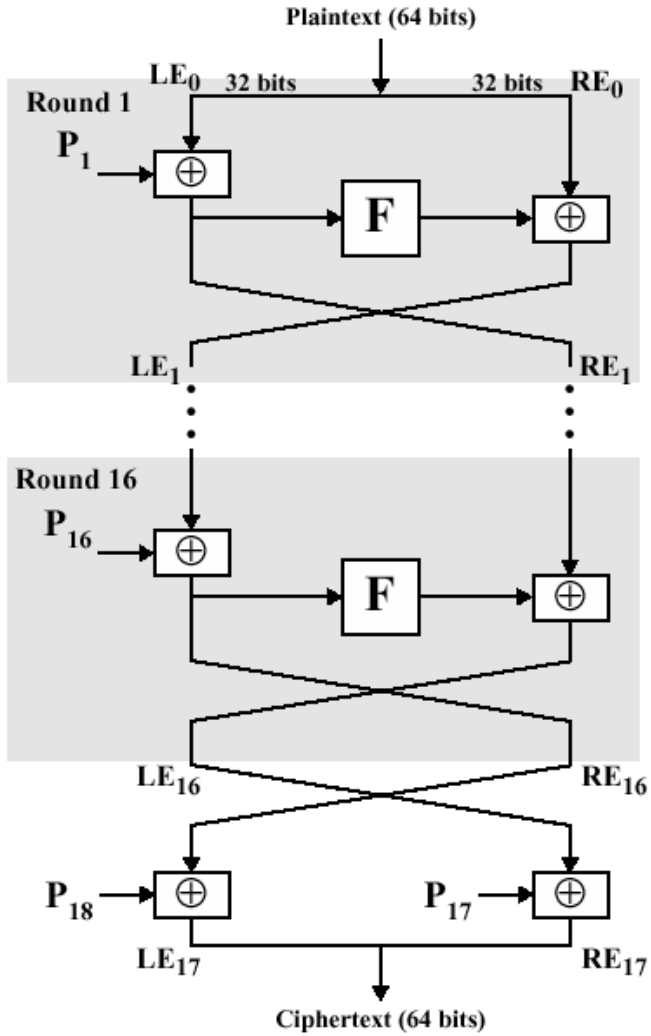
Oversikt over IDEA



Blowfish

- 32 til 448 (=14*32) bit nøkkellengde
- 64 bit blokklengde
- 16 runder
- 32-bits operasjoner internt
- Nøkkelavhengige S-bokser
- ”Feistel-aktig”

Blowfish kryptering



Blowfish betrakninger

- Operasjoner gjøres på begge halvdelene av data i hver runde
- Brute-force gjøres ”umulig” pga. den nøkkelavhengige oppbyggingen av S-boksene
- Etter at P,S er klar (kan beregnes på forhånd for gitt nøkkel) er Blowfish meget rask

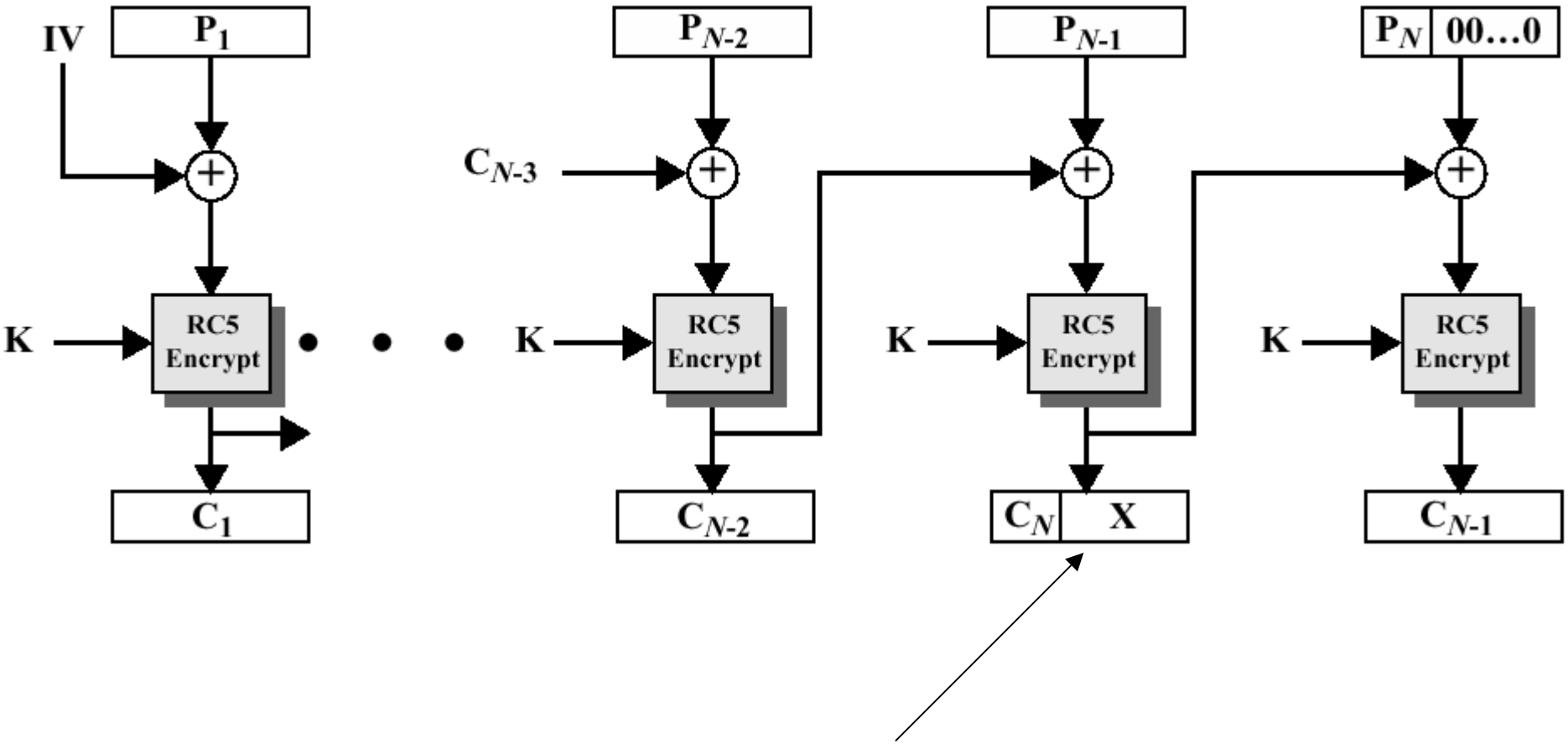
RC5

- 0 til 255 *byte* nøkkel
- 32 til 128 bit blokkstørrelse
- 0 til 255 runder
- Variabel bitstørrelse internt
- IKKE et Feistel-chiffer

RC5 modi

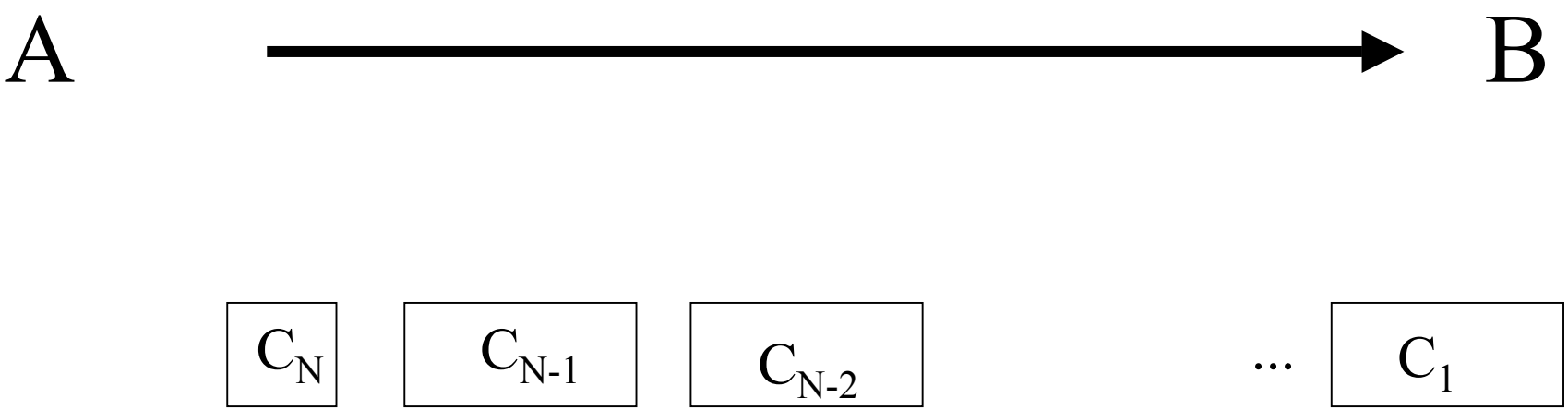
- RC5 Block – ECB
- RC5-CBC
- RC5-CBC-Pad – CBC med padding
- RC5-CTS

Ciphertext Stealing Mode



Kastes! (finnes igjen vha XOR 00...0)

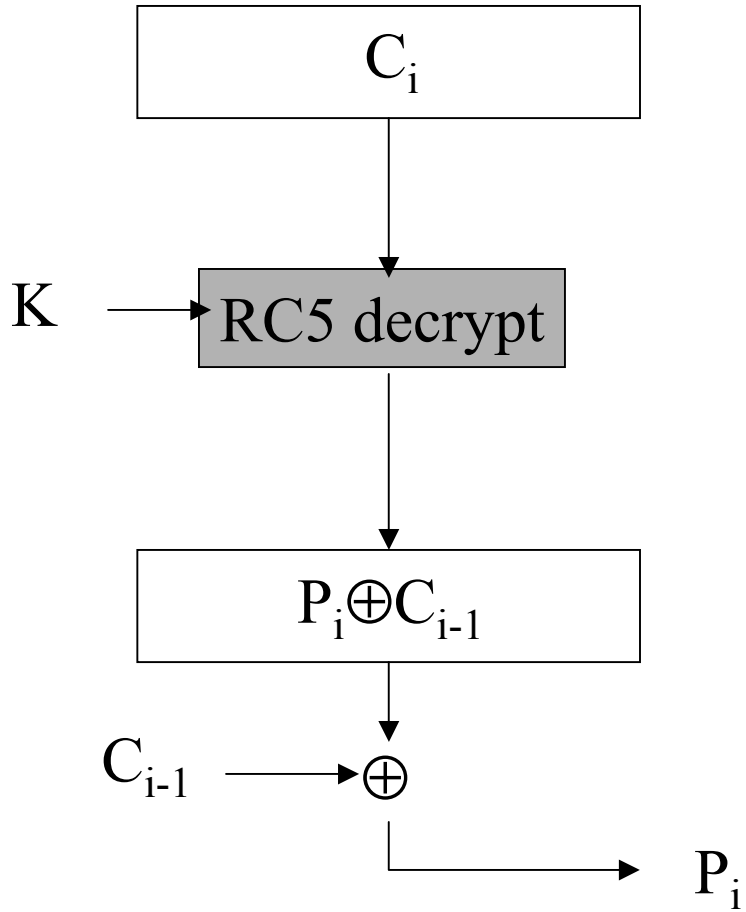
Pakkestrøm



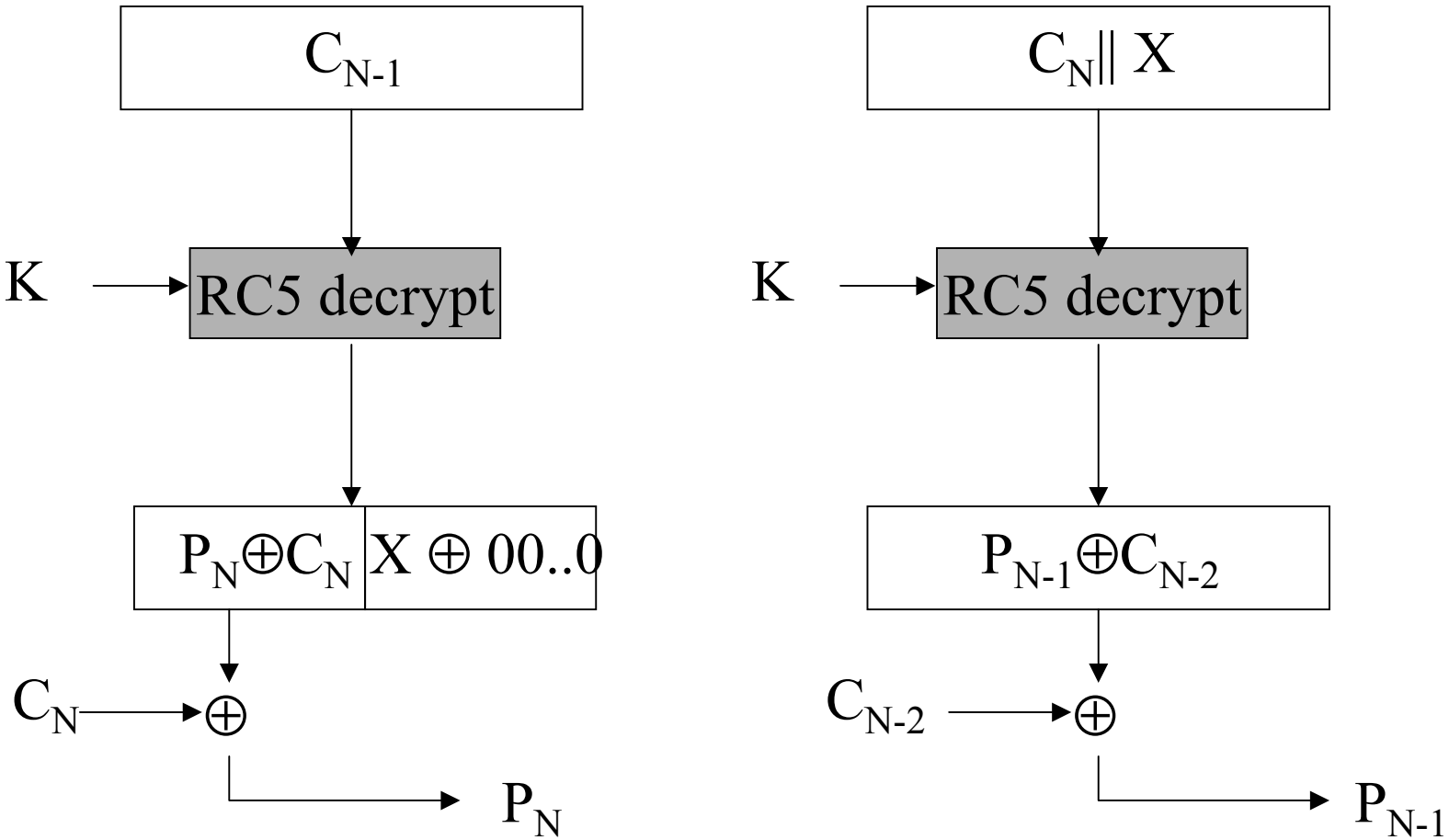
Dekryptering

- B må vite at ciphertext stealing er utført, og må derfor stoppe dekryptering når han kommer til den NEST siste blokken
- De to siste blokkene må behandles spesielt

Vanlig CBC dekryptering



Dekryptering, 2 siste blokker



Hva er poenget?

- Ved å bruke Ciphertext Stealing Mode slipper man å sende flere bit enn man faktisk har i meldingen ut på nettet

CAST-128

- 40 til 128 bit nøkkellengde
- 64 bit blokklengde
- 16 runder
- 32 bit operasjoner internt
- Feistel-struktur
- Runde-avhengig F

Blokkchiffer oppsummering

	Nøkkel	Blokk	Runder	Internt	Spesial
3-DES	112 (168)	64	48	32	EDE
IDEA	128	64	8	16	Ikke Feistel
Blowfish	32-448	64	16	32	Nøkkelavh. S-boks
RC5	0-255*8	32-128	0-255	Var.	Ikke Feistel
CAST-128	40-128	64	16	32	Rundeavh.operatorer
RC2	8-1024	64	18	16	Absolutt ikke Feistel

Hastighet Algoritmer (Pentium)

Algoritme	Klokkesykler per runde	Antall runder	Klokkesykler per byte kryptert
Blowfish	9	16	18
RC5	12	16	23
DES	18	16	45
IDEA	50	8	50
Triple-DES	18	48	108

Karakteristikker

Følgende karakteristikker gjelder i større eller mindre grad for moderne blokkchifre:

- Variabel nøkkellengde
- Flere primitive operatorer
- Data-avhengig rotasjon
- Nøkkel-avhengig rotasjon
- Nøkkel-avhengige S-bokser

Karakteristikker, forts.

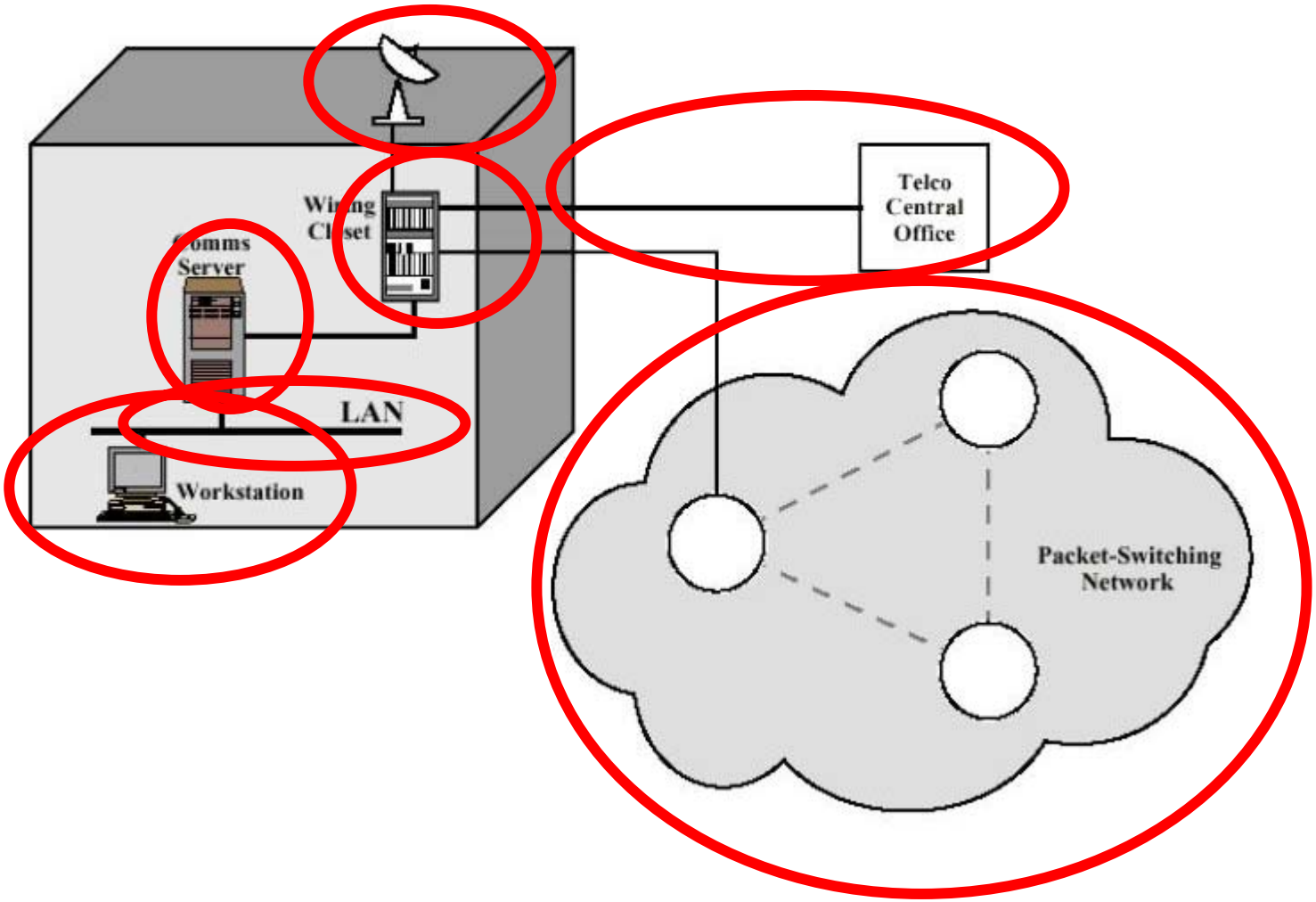
- Lang ”key schedule” algoritme
- Variabel F-funksjon
- Variabel blokklengde
- Variabelt antall runder
- Operasjon på begge halvpartene av data i hver runde

Konfidensialitet ved hjelp av konvensjonell kryptering

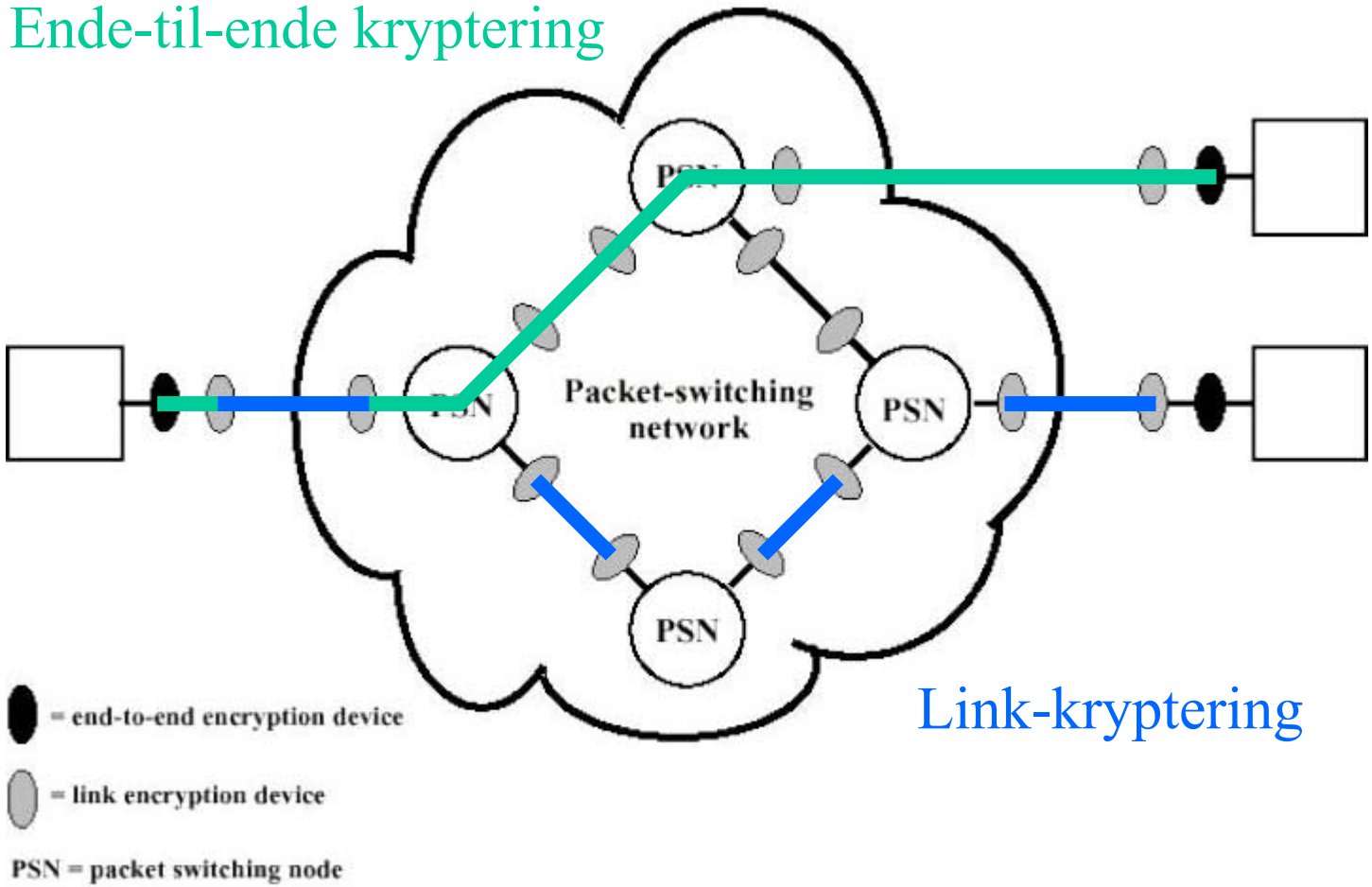
Hvor finnes truslene?

- Lokal maskin
- Lokalt nettverk (LAN)
- Oppringt samband
- Koblingsskap
- Eksternt nettverk

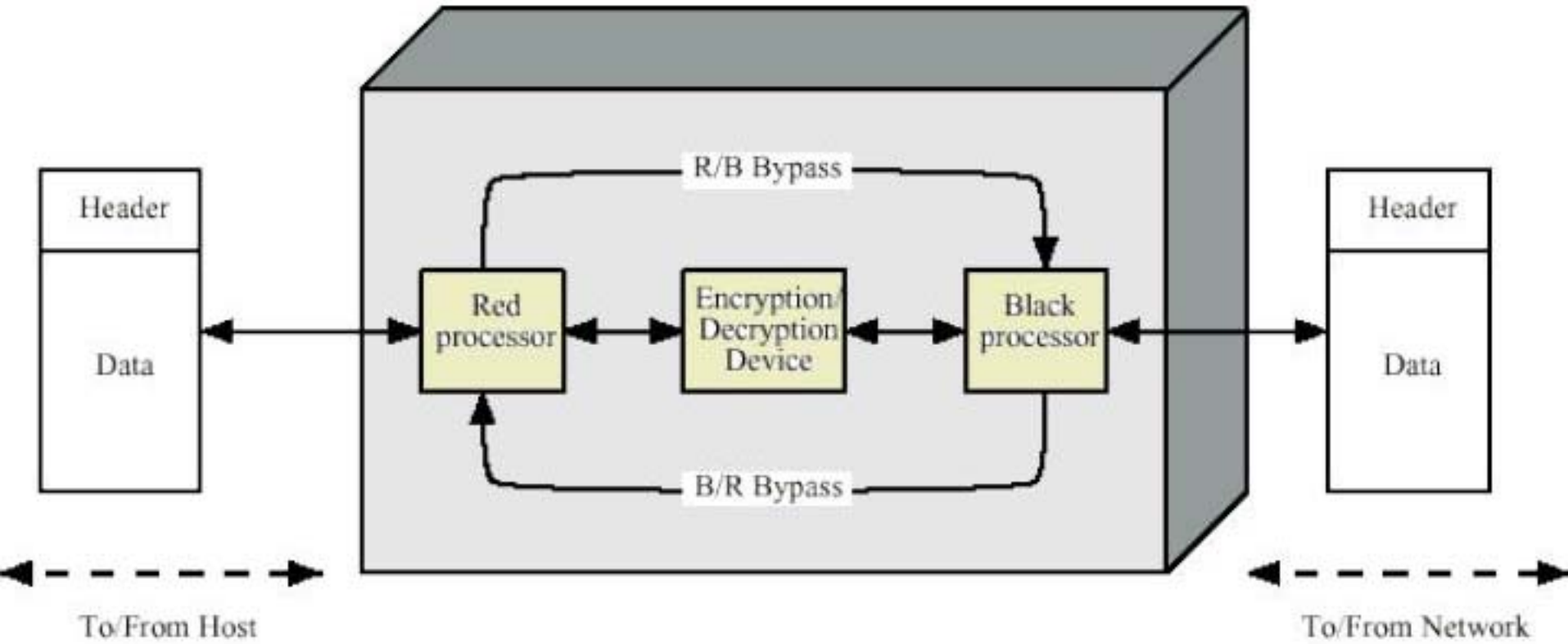
Svake punkter



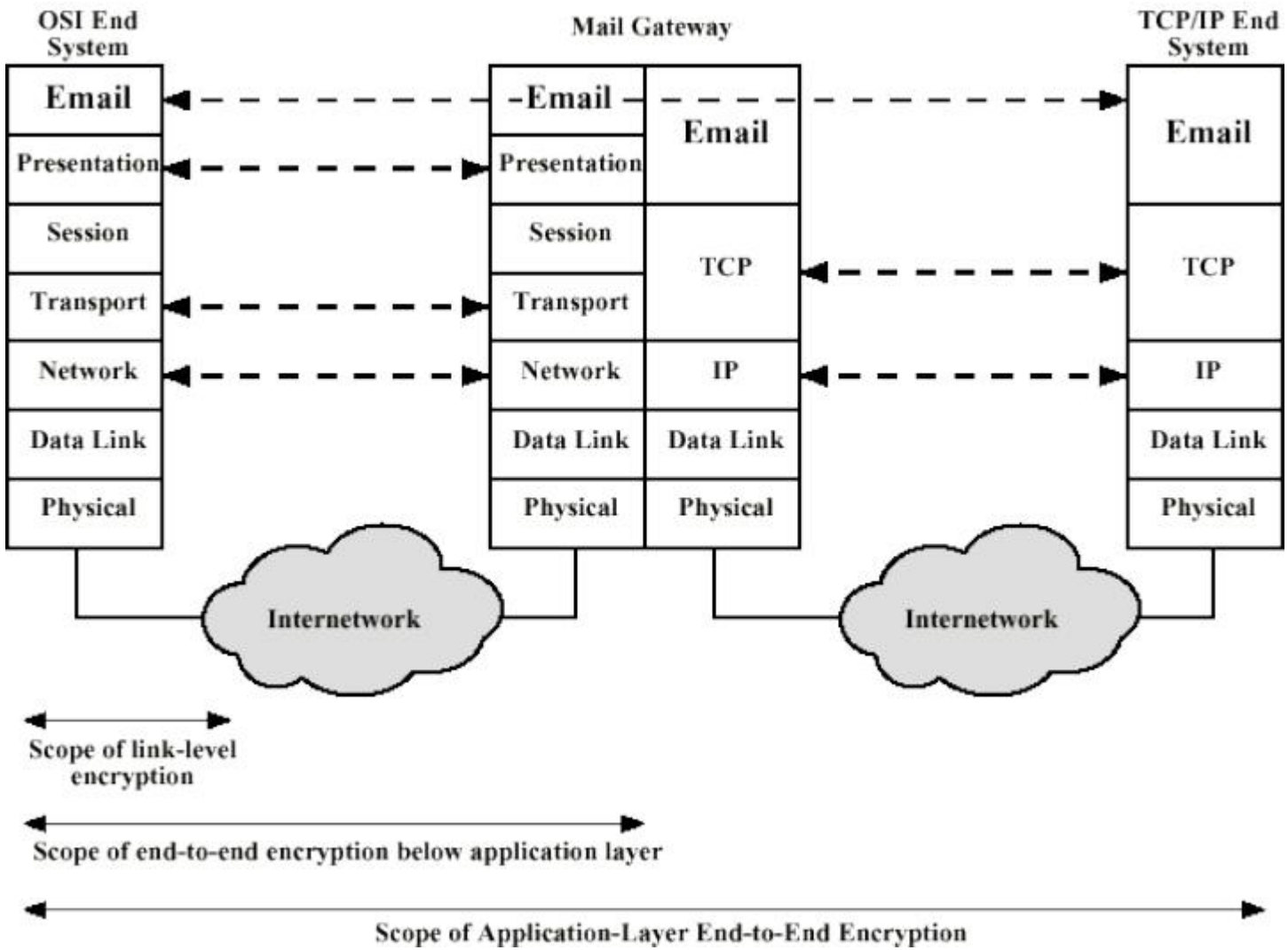
Link vs Ende-til-ende kryptering



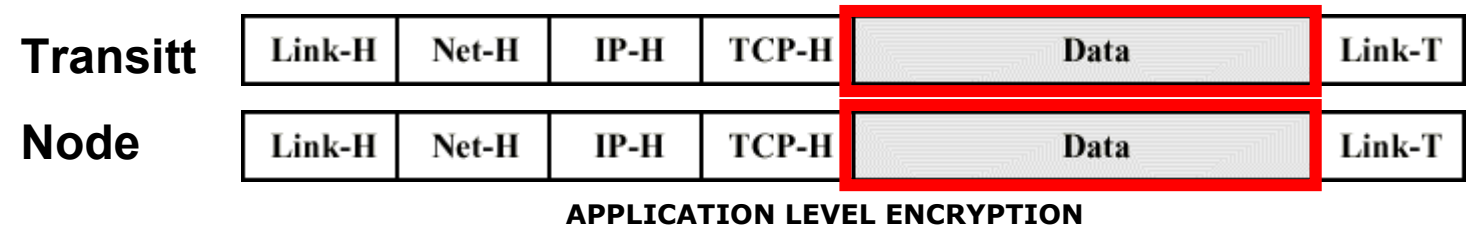
Front End Processor (FEP)



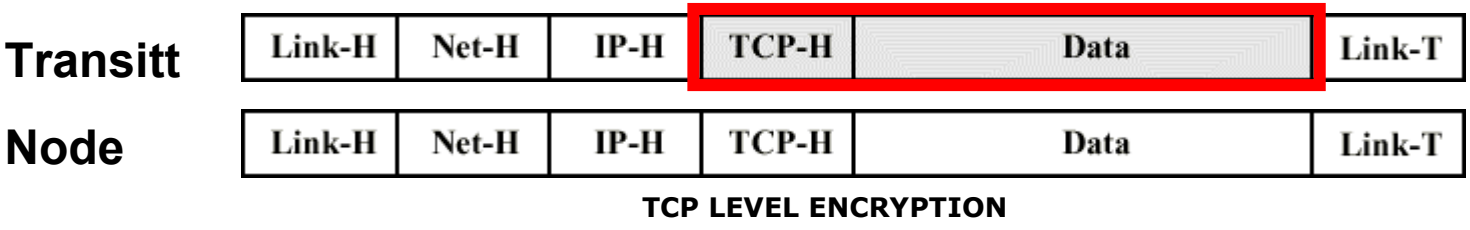
Kryptering på ulike nivåer



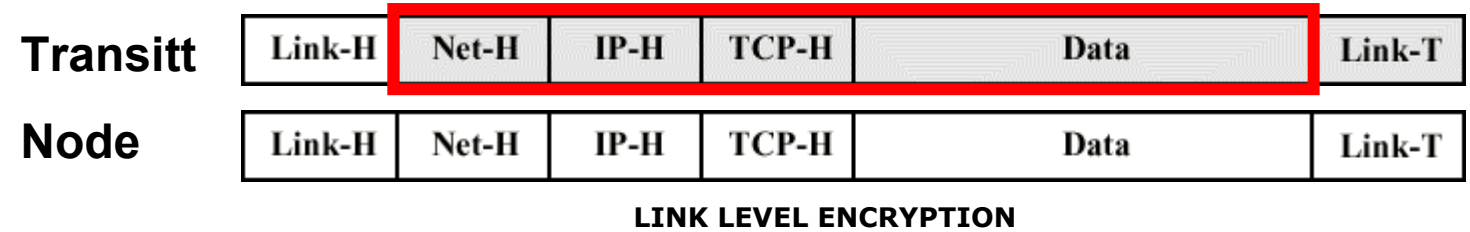
Trusselevaluering



APPLICATION LEVEL ENCRYPTION



TCP LEVEL ENCRYPTION



LINK LEVEL ENCRYPTION

Trafikkanalyse

- Identitet til partnere
- Hvor ofte kommuniserer partnere?
- Trafikkmønster, meldingslengder eller meldingsmengde kan indikere viktighet
- Begivenheter som korrelerer med spesielle konversasjoner mellom spesielle partnere

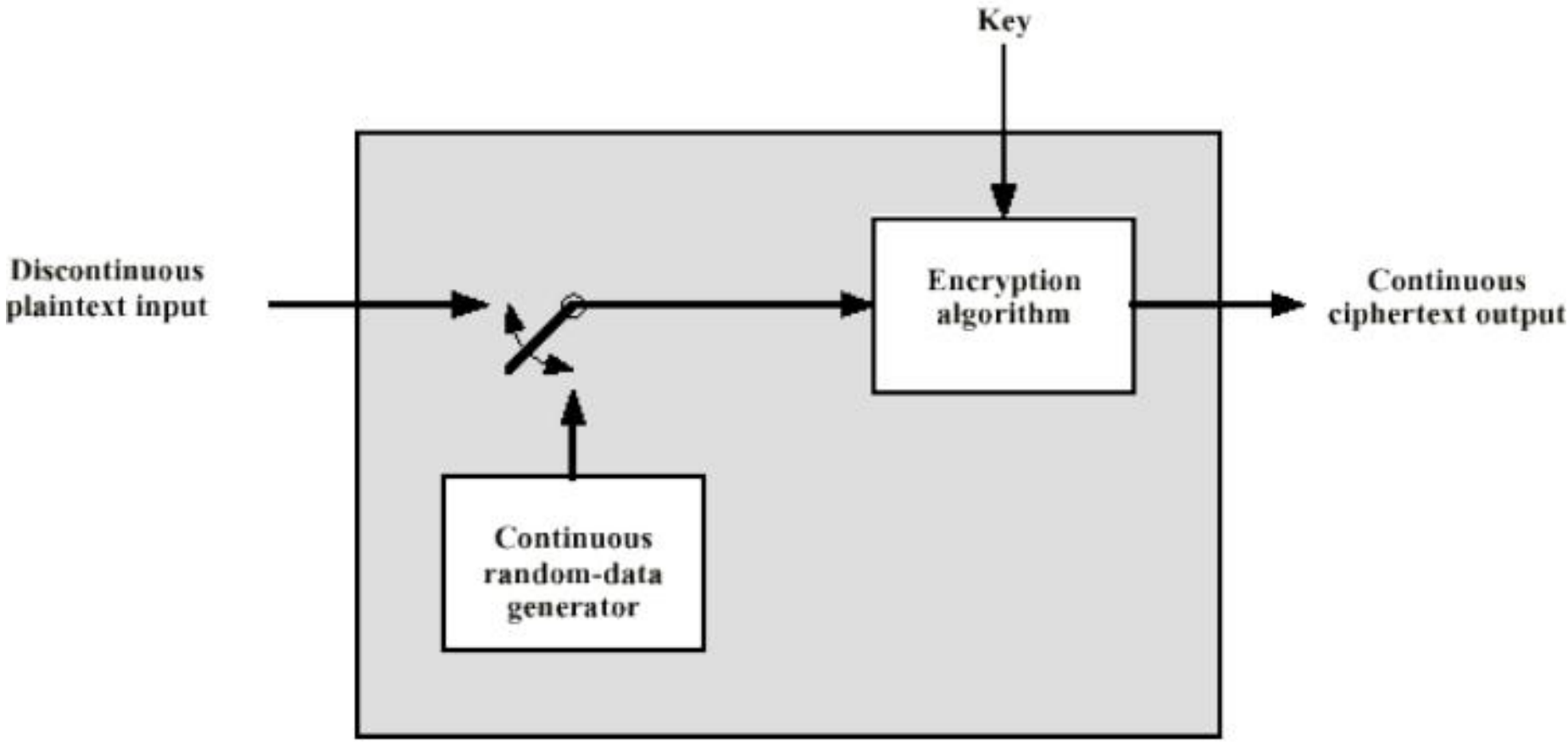
Covert Channel

- Bruk av en kommunikasjonskanal på en måte som ikke var tiltenkt av designerne
- F.eks. definere at meldinger fra A til B større enn 20kB betyr ”1” og meldinger mindre enn 20kB betyr ”0”
- Overvåkning av dataene mellom A og B vil ikke avsløre den skjulte dataflyten
- Også: Variable forsinkelser, etc.

Padding

- Kan brukes for å forhindre trafikkanalyse
- Mest aktuelt for link-kryptering
- Medfører konstant datarate mellom alle noder
- Kan i gitte tilfeller også forhindre ”covert channels”

Traffic Padding



Nøkkeldistribusjon

- For at to parter skal kunne kommunisere vha. konvensjonell kryptering, må begge ha den samme nøkkelen
- Hvordan oppnår man dette?

Nøkkeldistribusjon

- A velger en nøkkel, og leverer den manuelt til B
- En tredjepart velger en nøkkel for A og B, og leverer den manuelt til dem
- A kan velge en nøkkel og sende den til B kryptert med en tidligere nøkkel
- Hvis A og B begge har en kryptert forbindelse til C, kan C sende en kryptert nøkkel til hver av dem

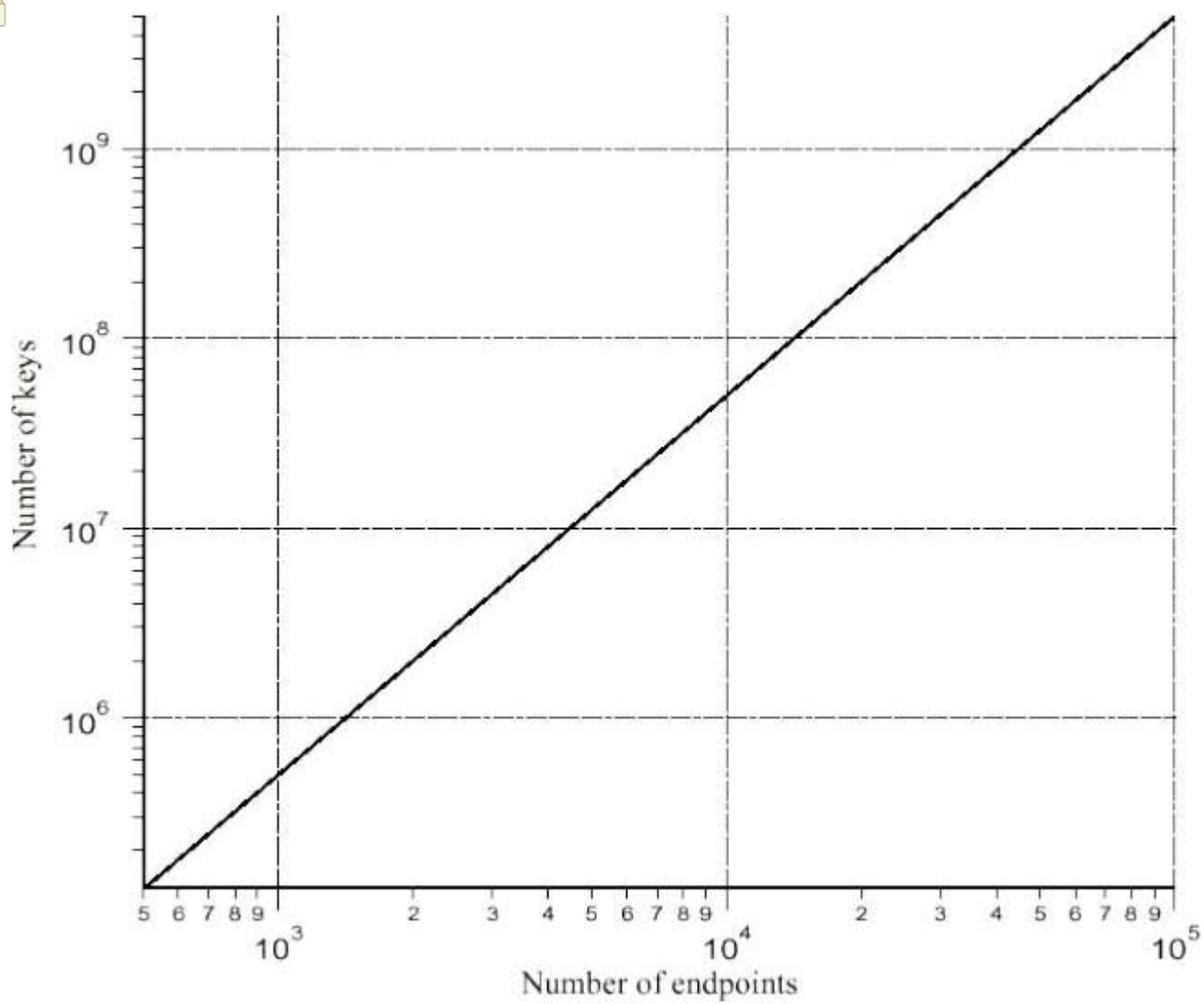
Nøkkelantall

- Hvis N maskiner skal kommunisere med hverandre, trenger man

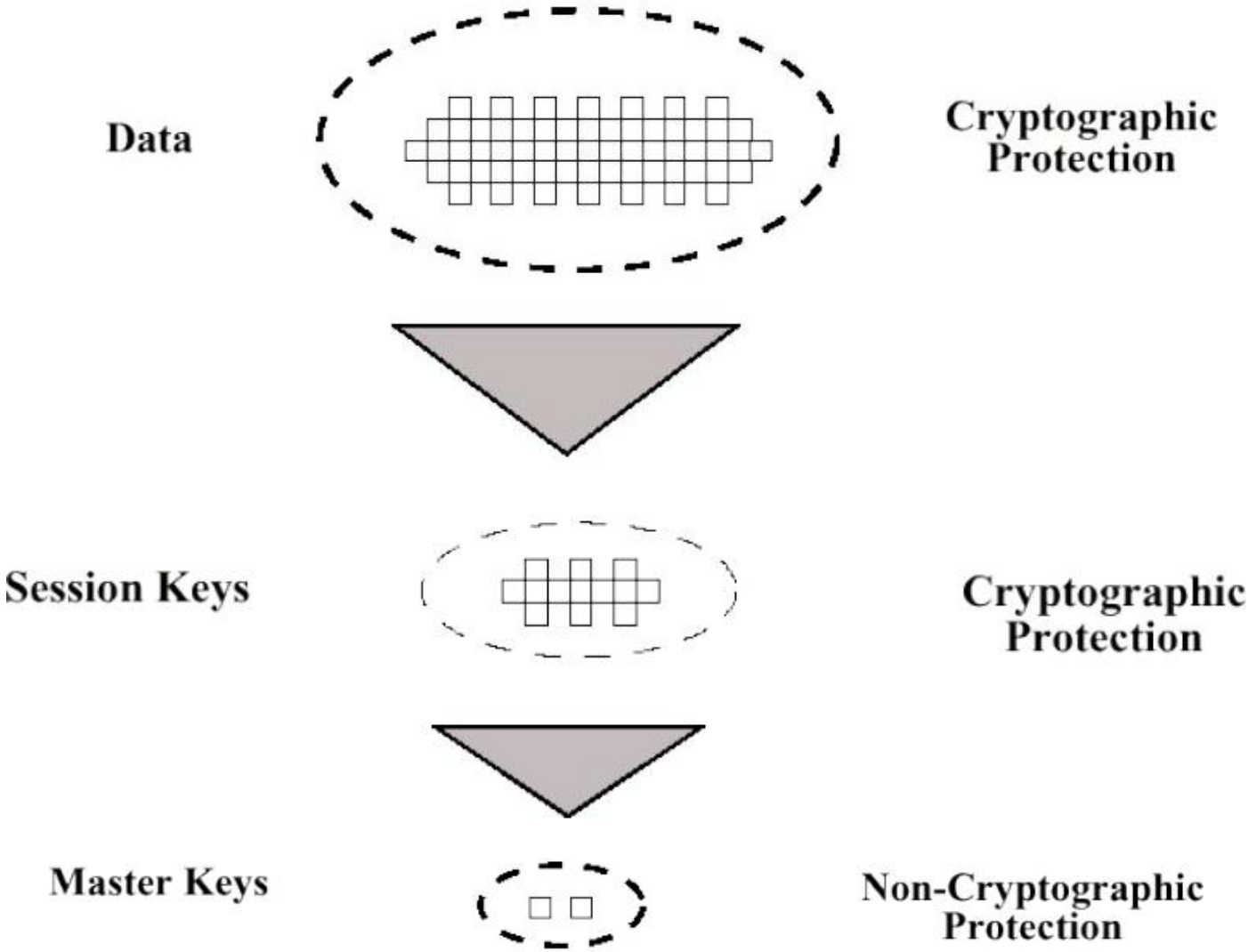
$$\frac{N(N-1)}{2}$$

nøkler

Nøkkelantall illustrert



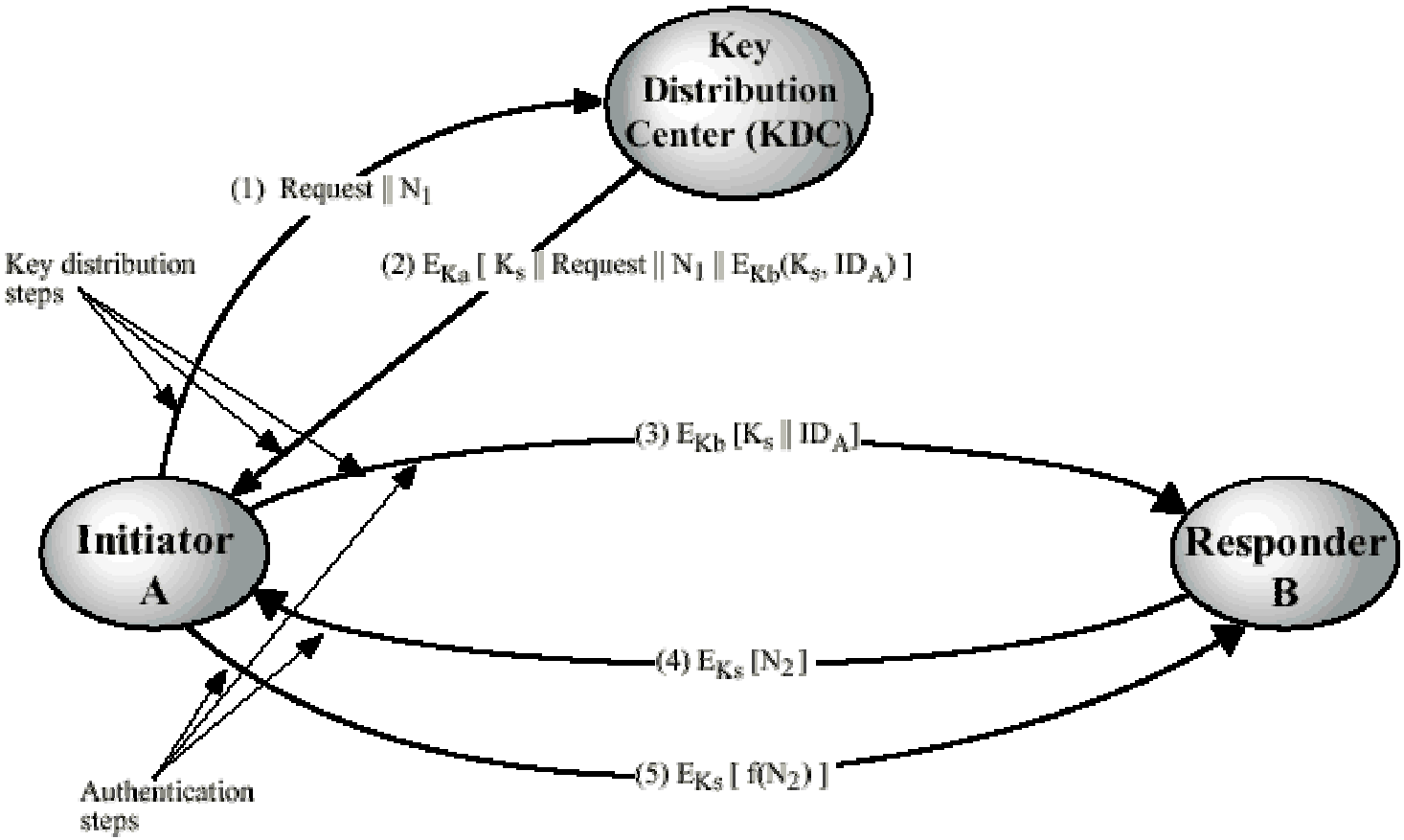
Nøkkelhierarki



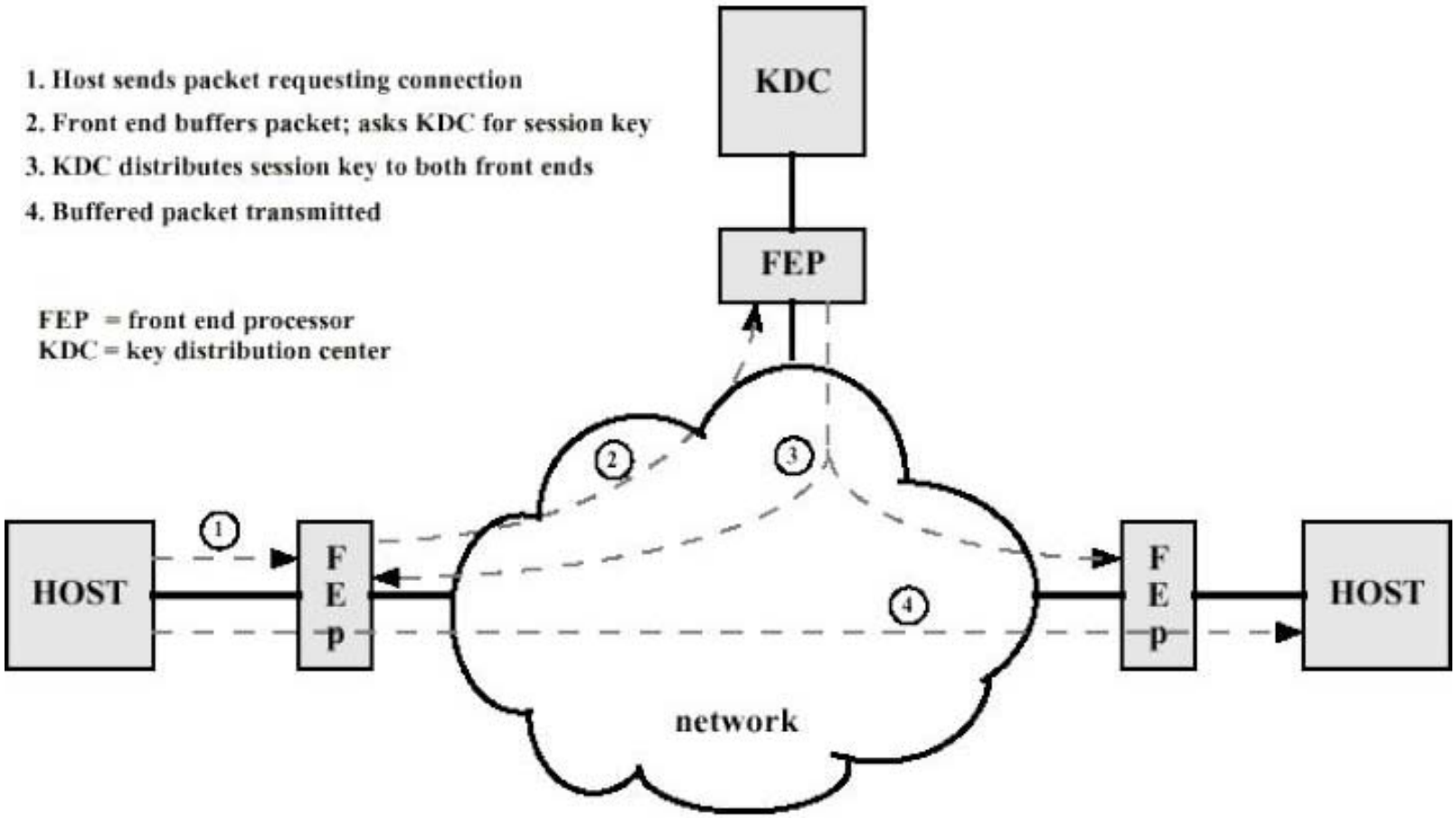
Nøkkeldistribusjon vha. KDC

- A deler en nøkkel K_a med KDC
- B deler en nøkkel K_b med KDC
- KDC velger sesjonsnøkkel K_s
- A velger tilfeldig tall (nonce) N_1
- B velger tilfeldig tall (nonce) N_2

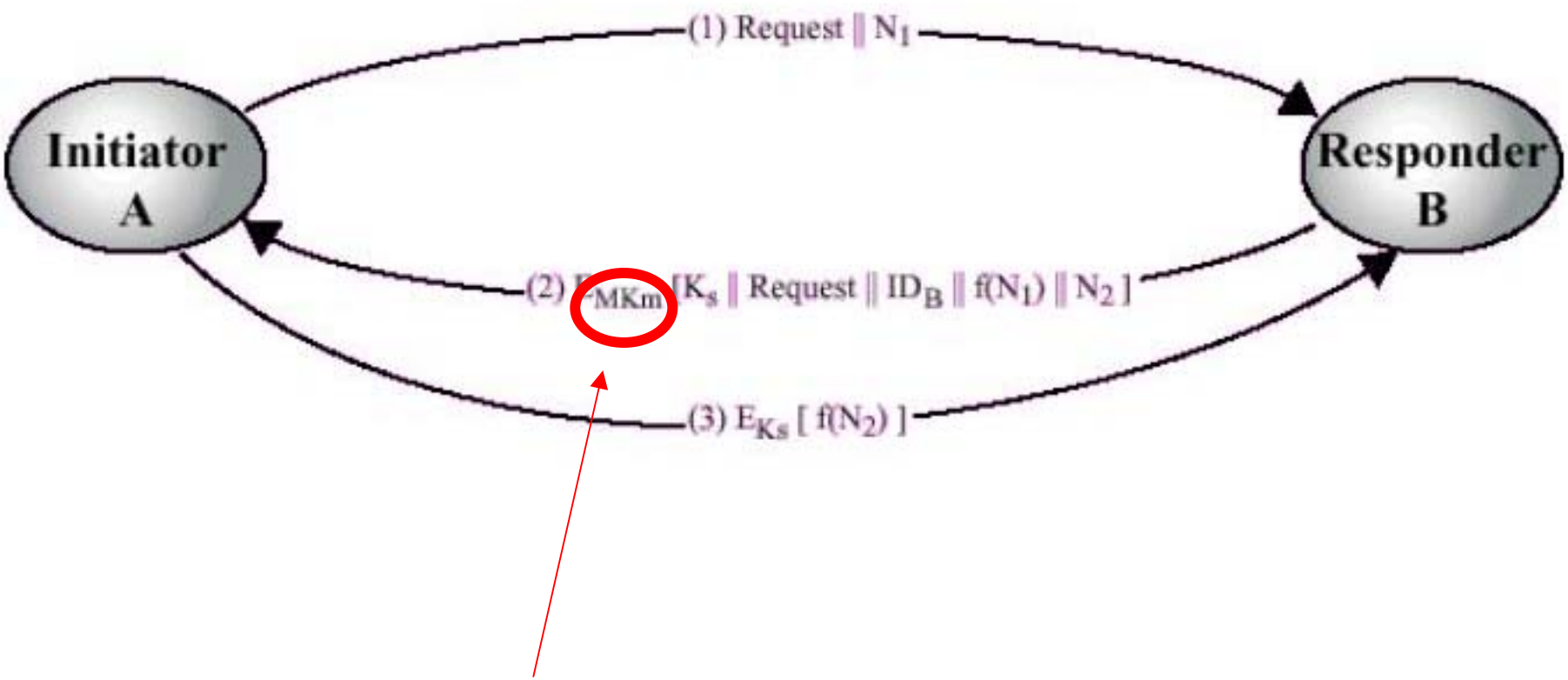
Nøkkeldistribusjon-scenario



Transparent nøkkeldistribusjon



Desentralisert nøkkeldistribusjon

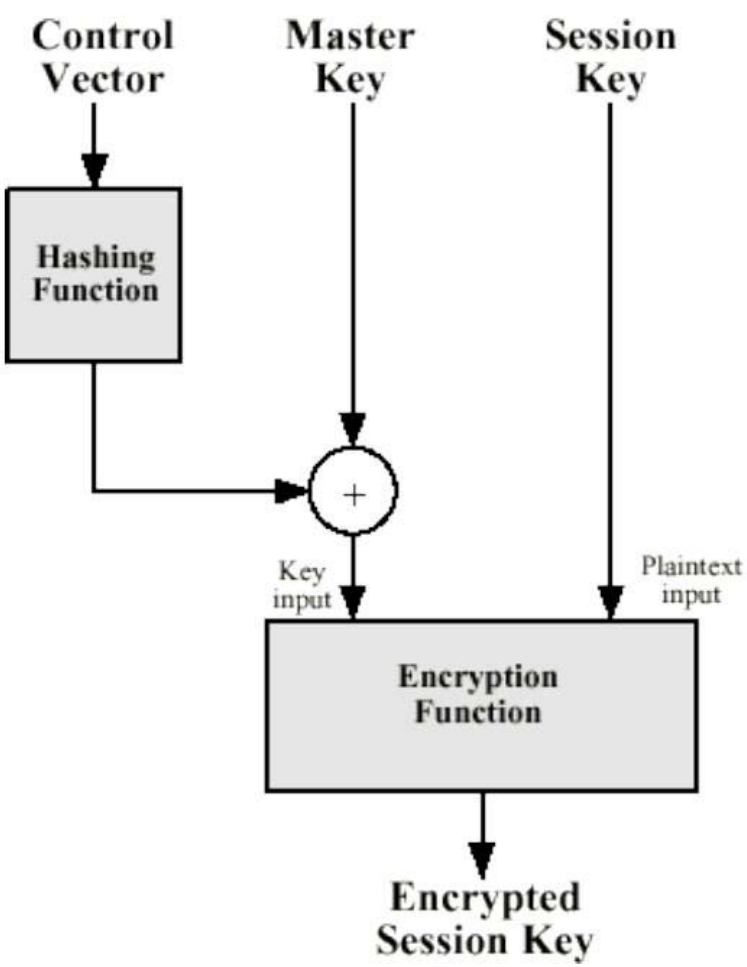


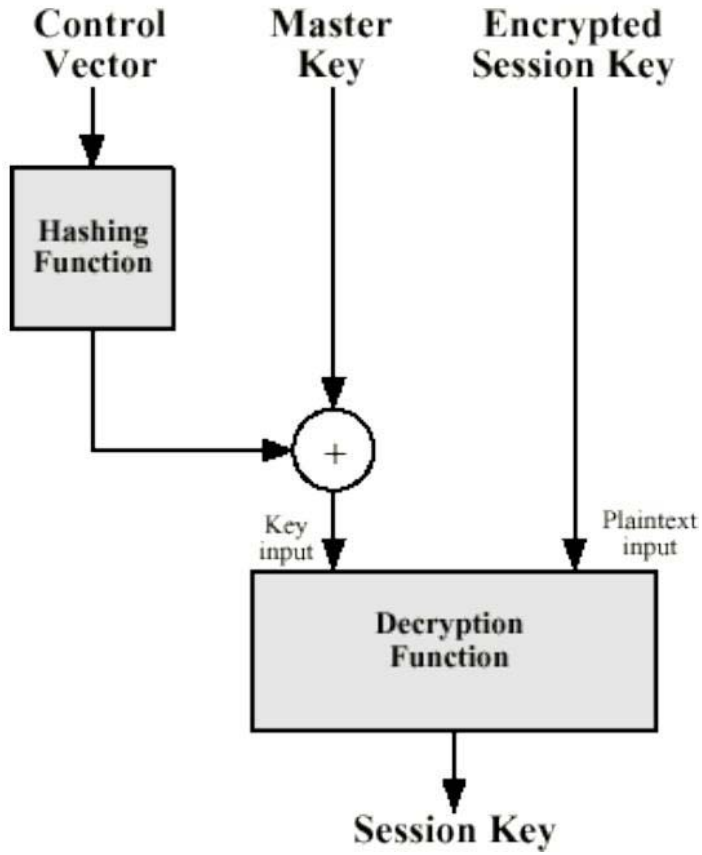
Må ha en delt Master Key med alle kommunikasjonspartnere!

Rett nøkkel på rett sted

- Et nøkkelhierarki forutsetter at nøklene brukes som forventet
- Hvis f.eks. en ”master key” brukes til å kryptere en større datastrøm, vil den være betraktelig mer sårbar for kryptoanalyse, noe som truer hele hierarkiet
- Kan være ønskelig å ”merke” nøkler med forventet bruksområde

Control Vector Coupling





Tilfeldige tall

- Autentisering basert på nonce (forhindrer replay)
- Sesjonsnøkkel-generering
- Generering av RSA-nøkler

Krav til tilfeldige tall

- ”Randomitet”
  - Uniform fordeling
  - Uavhengighet
- Uforutsigbarhet

- Oftest sier man seg tilfreds med tall som er ”pseudo-random”

The Linear Congruential Method

- Velg m | m>0
- Velg a | $0 \leq a < m$
- Velg c | $0 \leq c < m$
- Velg X_0 | $0 \leq X_0 < m$
- $X_{n+1} = (aX_n + c) \bmod m$

- Ikke uforutsigbar!

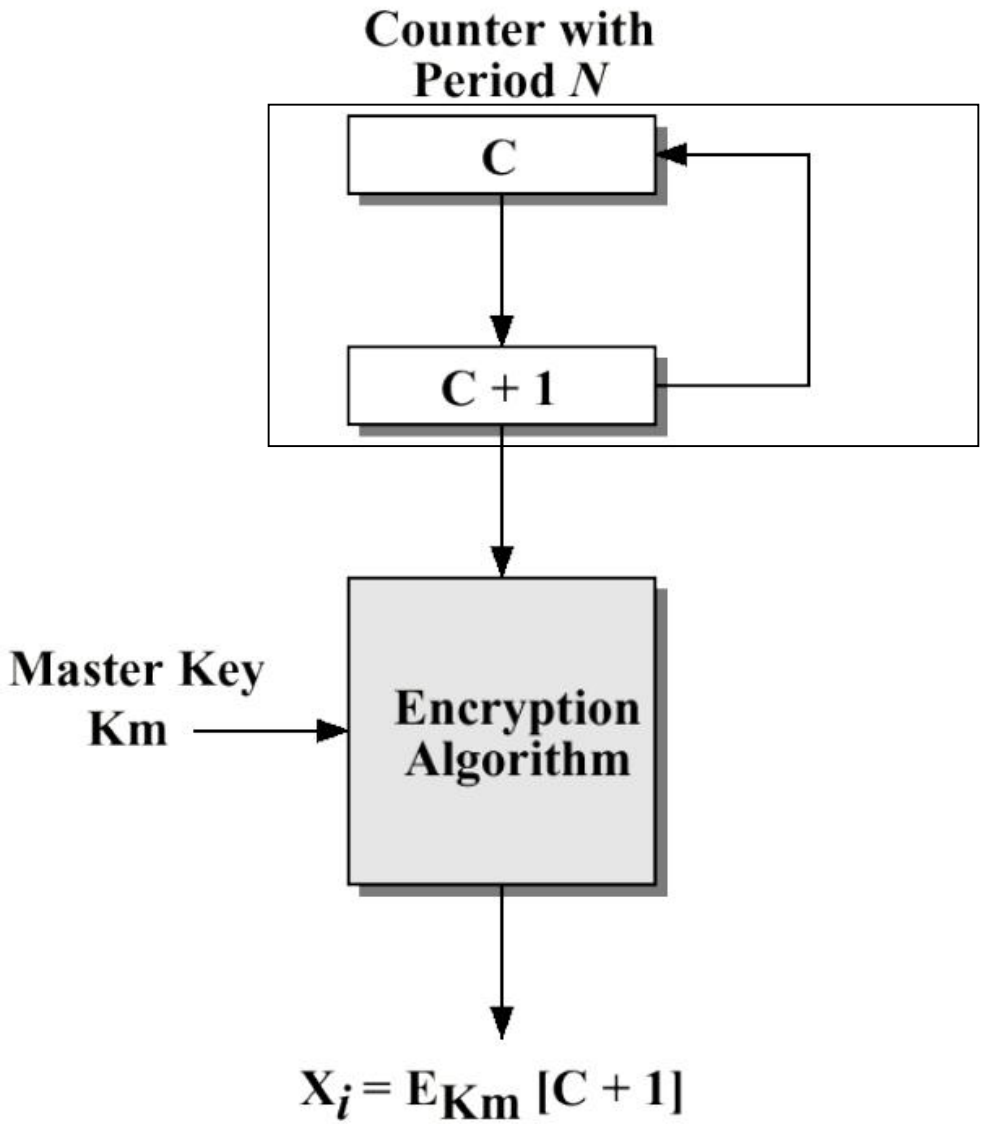
Kriterier for random-funksjoner

- Funksjonen må generere en full periode
- Sekvensen må se tilfeldig ut
- Funksjonen må kunne implementeres effektivt

Kryptografisk generering

- Syklisk kryptering
- DES OFB
- ANSI X9.17

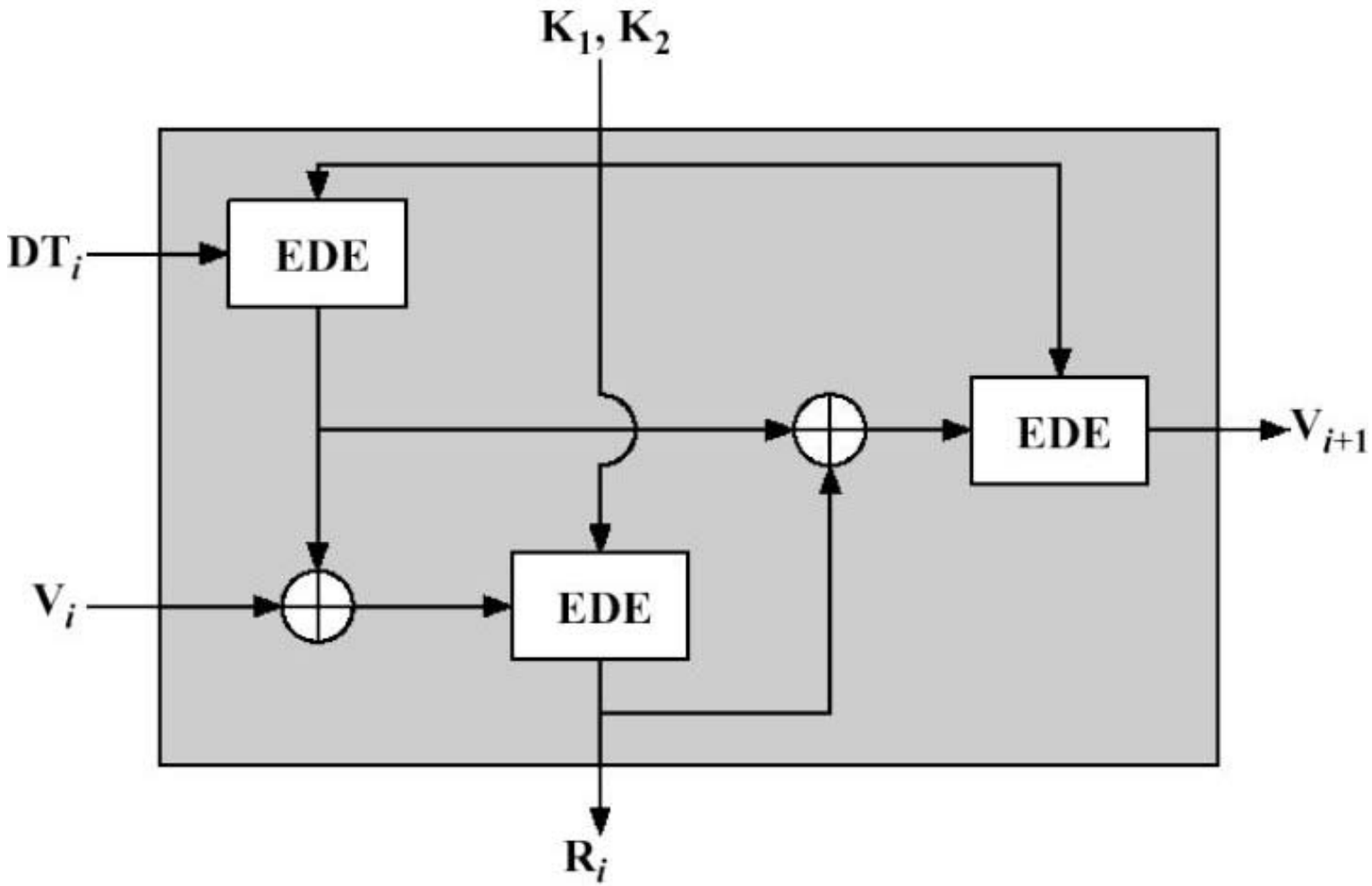
Syklisk kryptering



ANSI X9.17

- Input
  - DT_i – 64-bits representasjon av dato og tid
  - V_i – 64 bits ”seed”
  - K_1, K_2 – Triple-DES-nøkler
- Output
  - R_i – 64-bits pseudorandom tall
  - V_{i+1} – 64-bits ”seed” til *neste* runde

ANSI X9.17



Dagens website

- http://www.securityfocus.com