ЗАШТИТА ПОДАТАКА

Симетрични алгоритми заштите

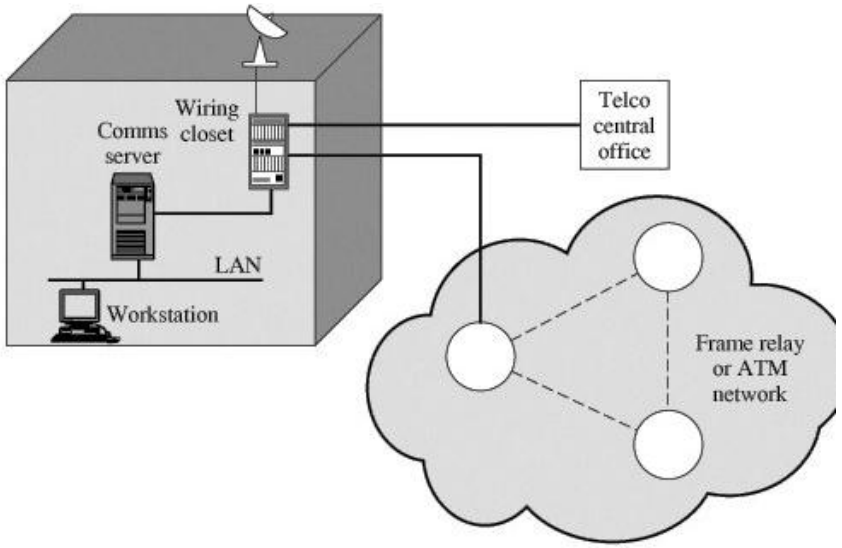
поверљивост симетричних алгоритама

Преглед

- биће објашњено:
  - коришћење симетричних алгоритама заштите како би се заштитила поверљивост
  - потреба за добрим системом размене кључева
  - генерисање случајних бројева

Поверљивост симетричних алгоритама заштите

- традиционално симетрични алгоритми заштите се користе да обезбеде поверљивост поруке
- размотримо типичан случај
  - радне станице у LAN-у приступају другим радним станицама и серверима у истом LAN-у
  - LAN је исповезиван помоћу свичева и рутера
  - и има неке екстерне линије приступа



Поверљивост симетричних алгоритама заштите

- размотримо могуће нападе у оваквом случају
  - њушкање (snooping) са друге радне станице
  - коришћење dial-in приступа LAN-у или серверу за њушкање
  - коришћење спољашњег линка за упад и њушкање
  - надгледање и/или модификовање саобраћаја на екстерном линку

Поверљивост симетричних алгоритама заштите

- постоје два начина коришћења симетричних алгоритама за заштиту поверљивости
- **шифровање на линку**
  - шифровање се врши независно по сваком линку
  - подразумева да се мора дешифровати саобраћај између линкова
  - захтева велики број уређаја, али су кључеви упарени
- **шифровање са краја на крај (end-to-end)**
  - шифровање се догађа између оригиналног извора и финалне дестинације
  - потребни су уређаји са дељеним кључевима на сваком крају

Анализа саобраћаја

- када се користи шифровање са краја на крај морају се оставити оригинална заглавља
  - да би подаци били коректно рутирани кроз мрежу
- тако да иако је садржај заштићен, остављен је простор за праћење тока саобраћаја
- идеално би било када бисмо имали оба одједном
  - шифровање са краја на крај би штитило податке током читаве путање и пружало аутентикацију
  - шифровање на линку би штитило ток саобраћаја од надгледања

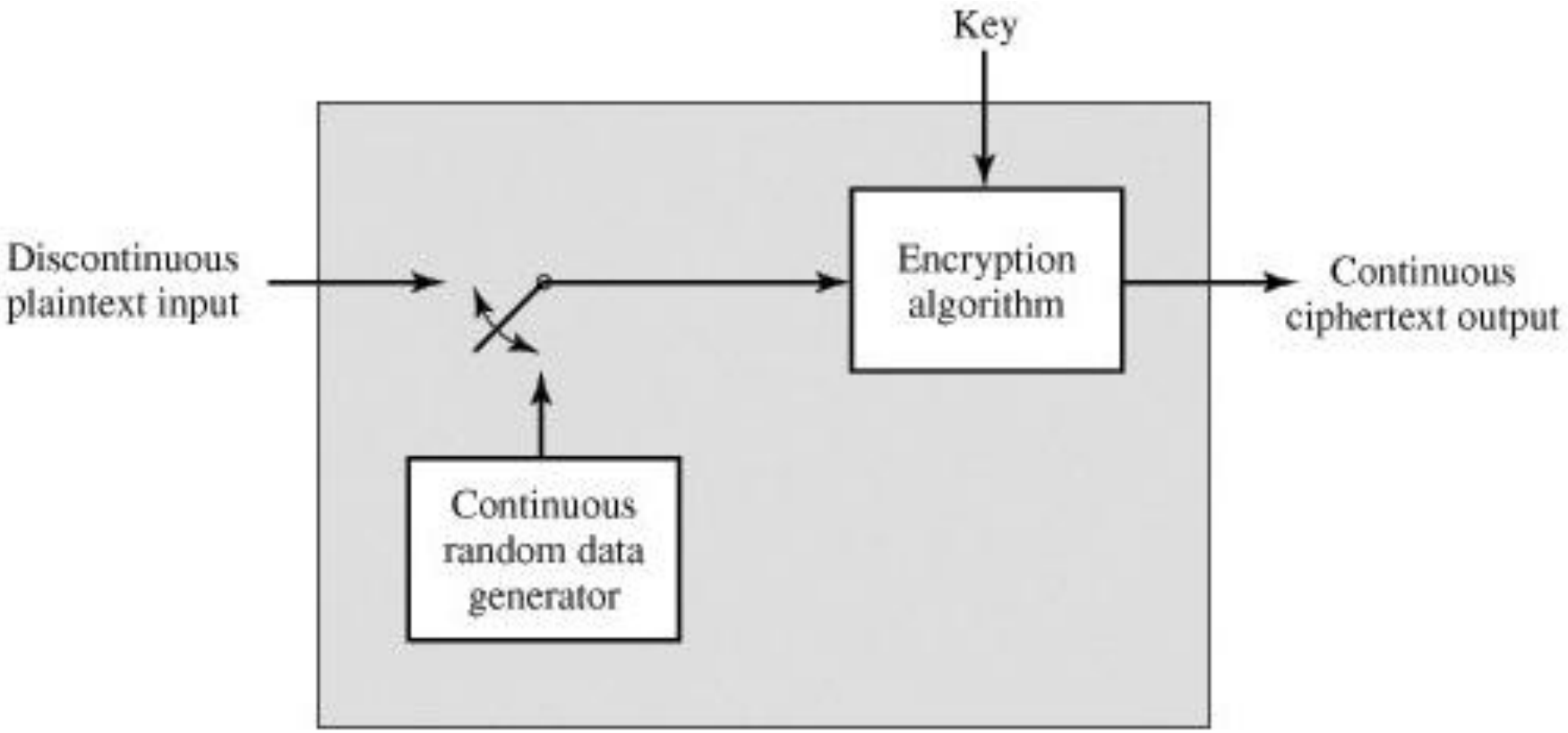
Ниво шифровања

- шифровање се може извршити на различитим слојевима OSI референтног модела
  - шифровање на линку се може вршити на слојевима 1 или 2
  - шифровање са краја на крај се може вршити на слојевима 3, 4, 6, 7
  - што је шифровање на вишем слоју, мање података се шифрује, али је и заштита бољa, мада комплекснија и са више ентитета и кључева

Анализа саобраћаја

- представља надгледање токова комуникације између учесника у комуникацији
  - користи се за шпијунажу
- шифровање на линку сакрива детаље заглавља
  - мада укупан обим саобраћаја у мрежи на крајњим тачкама је ипак видљив
- додавање саобраћаја (traffic padding) може додатно да замаскира токове саобраћаја
  - али по цену непрекидног саобраћаја

Анализа саобраћаја



Дистрибуција кључа

- симетрични алгоритми заштите захтевају да обе стране деле заједнички тајни кључ
- поставља се питање како на сигуран начин дистибуирати тај кључ
- често сигурносни системи отказују због упада у систем за размену кључа

Дистрибуција кључа

- неке стране A и B имају различите могућности **дистрибуције кључа**:
  1. A може да изабере кључ и физички га достави B
  2. нека трећа страна може да изабере кључ и достави га и A и B
  3. ако су A и B комуницирали у прошлости могу да искористе претходни кључ да направе нови кључ
  4. ако A и B имају сигурну комуникацију са трећом страном C, C може да пренесе кључ између A и B

Сценарио дистрибуције кључа

- Претпоставља се да сваки корисник дели јединствени мастер кључ са **KDC**.
- Кораци:

  1. A шаље захтев KDC-у за кључем сесије за комуникацију са B.

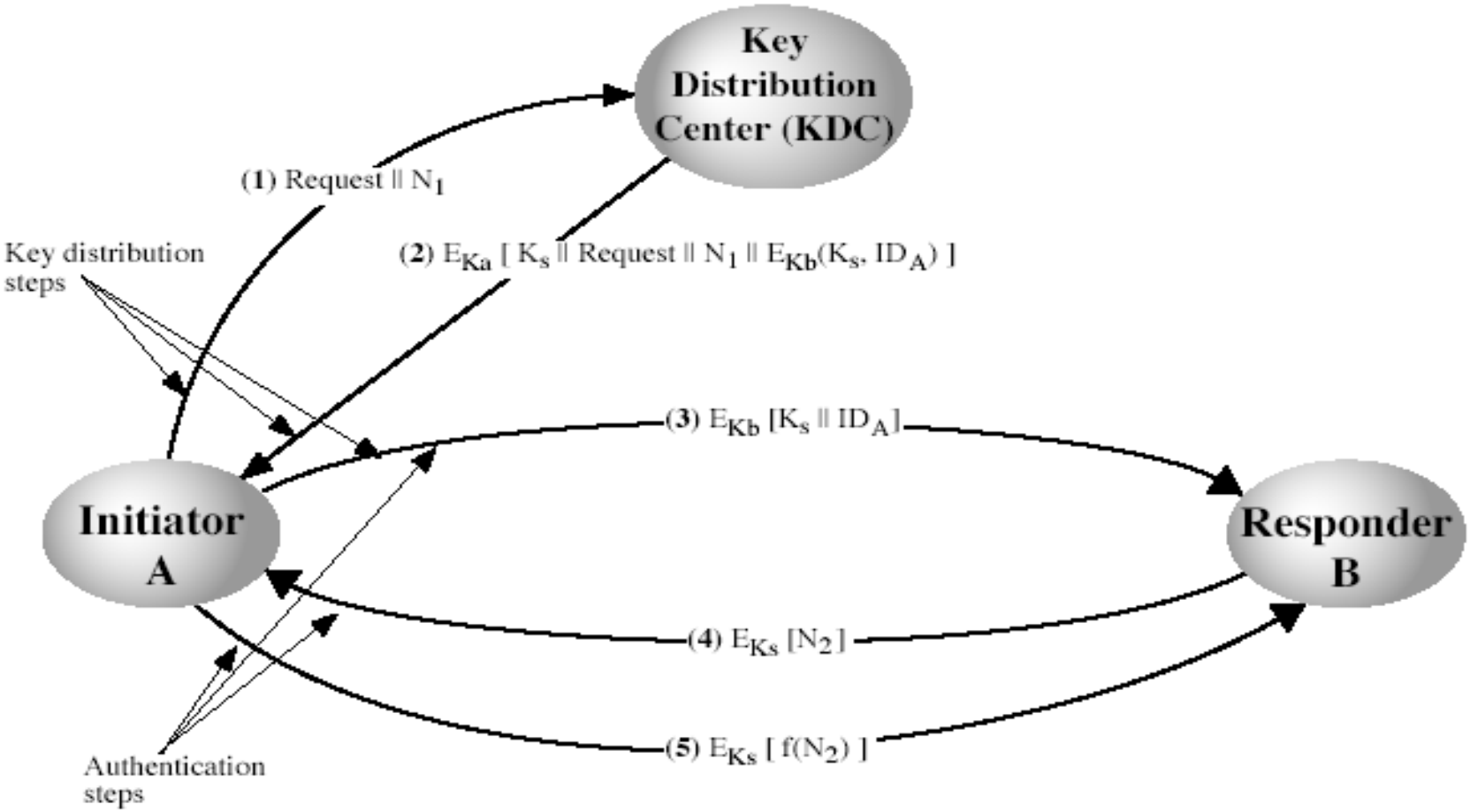
  2. KDC одговара поруком која је шифрована са Ka и која садржи:
  - кључ сесије Ks
  - оригиналну поруку од A
  - податке шифроване по кључу Kb

  3. A чува кључ сесије за комуникацију са B и шаље B-у податке шифроване са Kb

  4. по кључу сесије, B шаље N2 за A.

  5. користећи Ks, A одговара са f(N2)

Сценарио дистрибуције кључа



Проблеми дистрибуције кључа

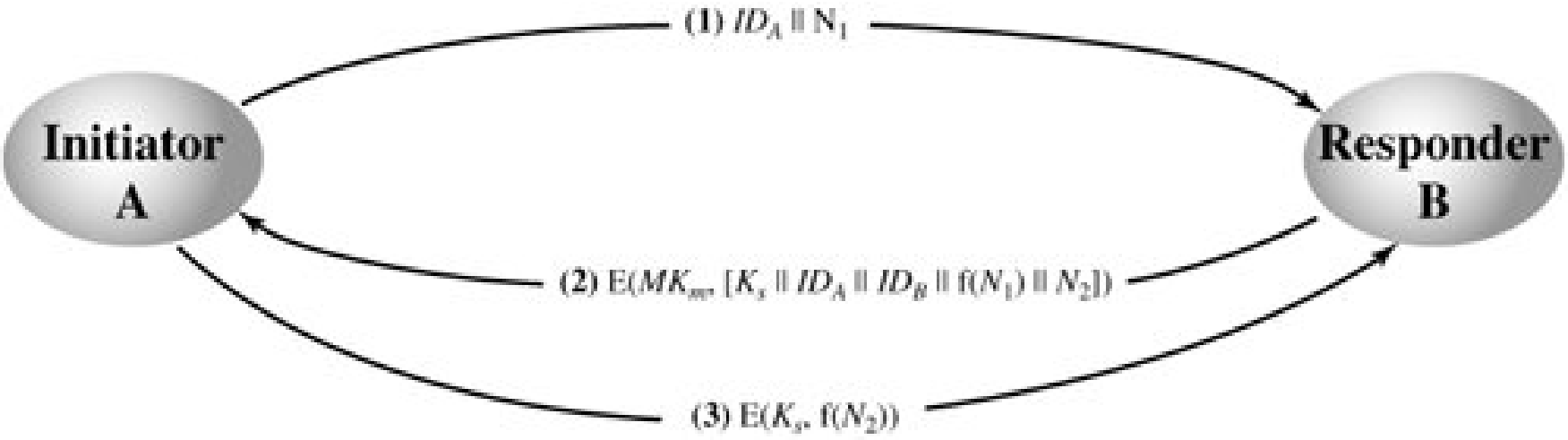
- за велике мреже потребна је хијерархија KDC-ова, који имају сигурну комуникацију
- кључеви сесије би морали имати ограничен век трајања ради веће безбедности
- може се користити аутоматска дистрибуција кључа у име корисника, али систем мора бити од поверења
- децентрализована дистрибуција кључа

Децентрализована дистрибуција кључа

1. А шаље захтев В за кључ сесије и укључује N1

2. В одговара поруком шифрованим дељеним мастер кључем, која садржи кључ сесије, идентификатор В, вредност f(N1) и N2.

3. користећи нови кључ сесије, А шаље f(N2) за В.



Случајни бројеви

- многе су употребе **случајних бројева** у криптографији
  - у протоколима за аутентикацију да се спрече replay напади
  - кључеви сесије
  - генерисање јавних кључева
  - кључ за one-time pad алгоритам
- у свим случајевима је критично да ове вредности буду
  - статистички случајне
    - са униформном расподелом
  - непредвидиве

Природно случајни бројеви

- најбољи извор је природна случајност у реалном свету
- пронаћи прави, али случајан догађај и посматрати
- углавном је потребан специјалан хардвер да би се ово урадило
  - нпр. бројач радијације, радио сигнали, аудио сигнали, итд.
- почиње да се појављује такав хардвер у новим процесорима
- проблеми неравномерне расподеле у сигналима
  - ово мора да се компензује у случају коришћења
  - најбоље користити неколико најизраженијих бита из сваког узорка

Публиковани извори

- постоји неколико публикованих извора колекција случајних бројева
- Rand Co, је 1955. године, објавио 1 милион бројева
  - генерисани су коришћењем електронског точка за рулет
  - коришћени су у неким дизајнима алгоритама заштите
- проблеми су ти да:
  - ограничен је број
  - сувише добро су познати за већину употреба

Псеудослучајни генератори бројева

- базирано на алгоритмима за стварање случајних бројева
  - иако нису заиста случајни
  - могу проћи многе тестове случајности

Линеарни конгруентни генератор

- позната итеративна техника која користи:

  $X_{n+1} = (aX_n + c) \bmod m$

- ако се користе одговарајуће вредности параметара могу се створити дуге секвенце наизглед случајних бројева
- погодан критеријум је:
  - функција генерише читав период
  - генерисана секвенца би требало да изгледа случајно
  - ефикасна имплементација са 32-битном аритметиком
- нападач може да реконструише секвенцу на основу малог броја познатих вредности

Криптографски генерисани случајни бројеви

- може се користити блок алгоритам за генерисање бројева
- у Counter моду функционисања

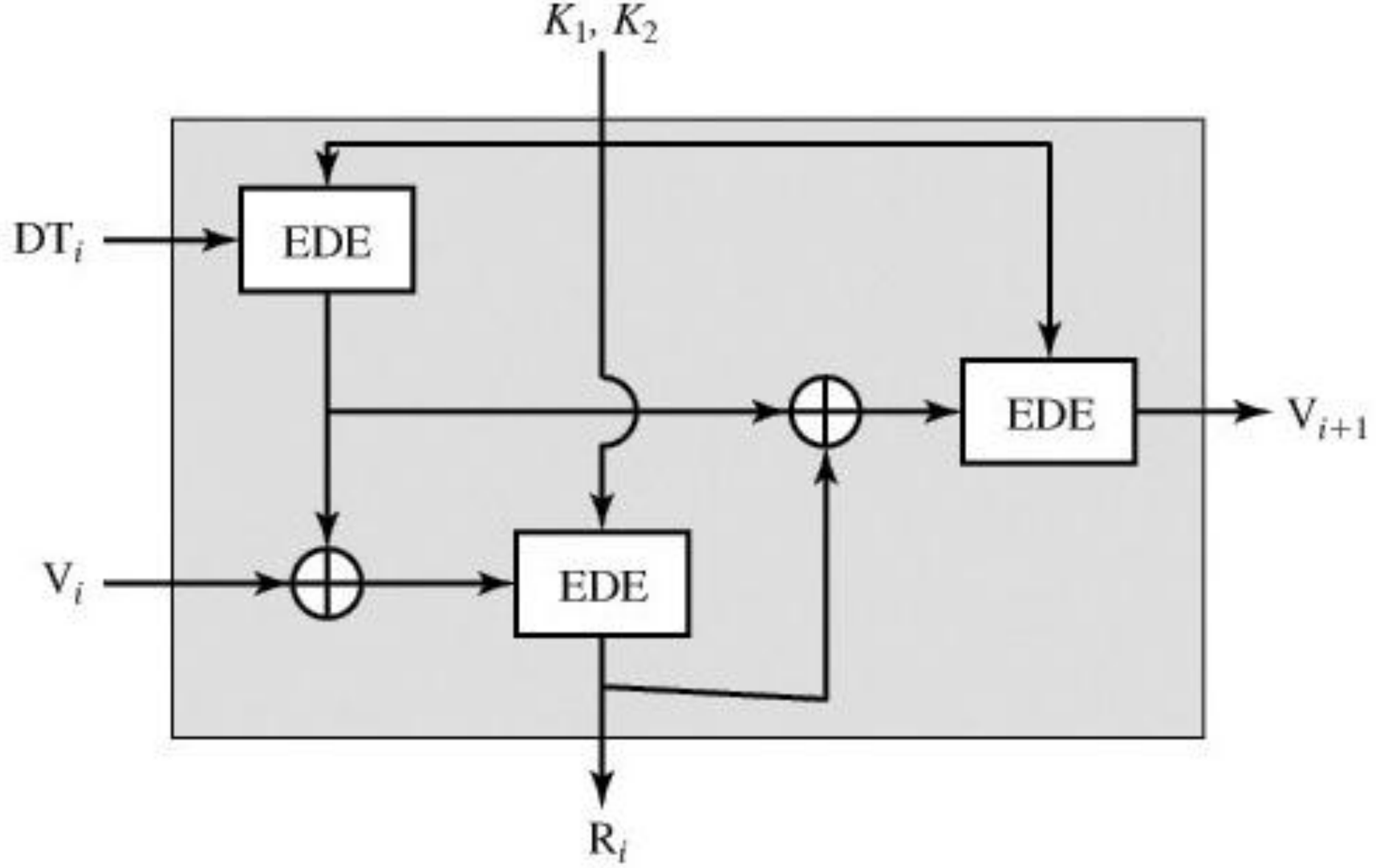
  $X_i = E_{Km}[i]$

- у Output Feedback моду функционисања

  $X_i = E_{Km}[X_{i-1}]$

- ANSI X9.17 PRNG
  - користи датум и време + улазне податке и 3 triple-DES шифровања да направи нове бројеве

Криптографски генерисани случајни бројеви



Blum Blum Shub генератор

- базира се на алгоритмима са јавним кључем
- користи се најмање значајан бит из итеративне једначине или бит парности:
  - $x_{i+1} = x_i^2 \bmod n$
  - где је $n=p \times q$, а велики прости бројеви $p,q=3 \pmod 4$
- непредвидив
- спор, јер се морају користити јако велики бројеви
- сувише спор за коришћење у шифровању, добар за генерисање кључа