ЗАШТИТА ПОДАТАКА

Симетрични алгоритми заштите

блок алгоритми и DES (Data Encryption Standard)

Преглед

- Биће објашњено:
  - принципи дизајна блок алгоритама
  - DES
    - детаљи
    - предности

Савремени блок алгоритми

- најкоришћенији тип криптографских алгоритама (са највећим бројем намена)
- обезбеђује тајност и сервисе за аутентикацију
- биће представљен кроз DES (Data Encryption Standard)

Блок алгоритми наспрам алгоритама тока података (Block vs Stream)

- алгоритми тока података шифрују поруку бит по бит или бајт по бајт
- блок алгоритми шифрују поруку у блоковима, од којих се сваки независно шифрује/дешифрује
- као супституција са великим бројем карактера
  - 64-бита или више
- користећи модове функционисања, може се добити исти ефекат као код алгоритама тока података
- већина алгоритама у употреби данас су блок алгоритми

Принципи блок алгоритама

- блок алгоритми личе на изузетно велике алгоритме замене
- где би била потребна табела од 2^{64} улаза за блок од 64-бита
- За блок од 2 бита, реверзибилно мапирање:
  - 00 01
  - 01 10
  - 10 11
  - 11 00
- За блок од 2 бита, нереверзибилно мапирање:
  - 00 01
  - 01 10
  - 10 00
  - 11 00

Принципи блок алгоритама

- нама је потребно реверзибилно мапирање
- значи имамо $2^n!$ различитих трансформација за блок од n бита
- међутим, ако се користи мала величина блока, систем је еквивалентан са класичним алгоритмима замене
- ови алгоритми су рањиви на нападе статистичком анализом
- уколико је величина блока довољно велика, статистичке карактеристике се елиминишу

Принципи блок алгоритама

- са друге стране за велике блокове, овај начин шифровања није практичан
- ако узмемо да изабрано мапирање представља кључ, за блок од 4 бита, имали би величину кључа од 4b x 16 = 64b
- односно, уопштено за блок од n бита имали би кључ од $n \times 2^n$
- За блок од 64 бита што је минимална величина код које се елиминишу статистичке карактеристике, имали би кључ од 64×2^{64}, што је око 10^{21} бита

Принципи блок алгоритама

- већина симетричних блок алгоритама базирају се на структури **Feistel алгоритма**
- овај принцип предлаже да се уместо на претходни начин блокови шифрују коришћењем мањих градивних елемената, који апроксимирају претходни приступ
- користи се идеја продукционих алгоритама

Claude Shannon и алгоритми замене и пермутације

- 1949. Claude Shannon је представио идеју мрежа замене и пермутације (substitution-permutation (S-P))
  - данашњи продукциони алгоритми замене и транспозиције
- ово је основа данашњих блок алгоритама
- S-P мреже се базирају на две примитивне операције криптографије:
  - *замена (substitution* (S-box))
  - *пермутација (permutation* (P-box))
- омогућавају конфузију (*confusion)* и дифузију (*diffusion)* поруке

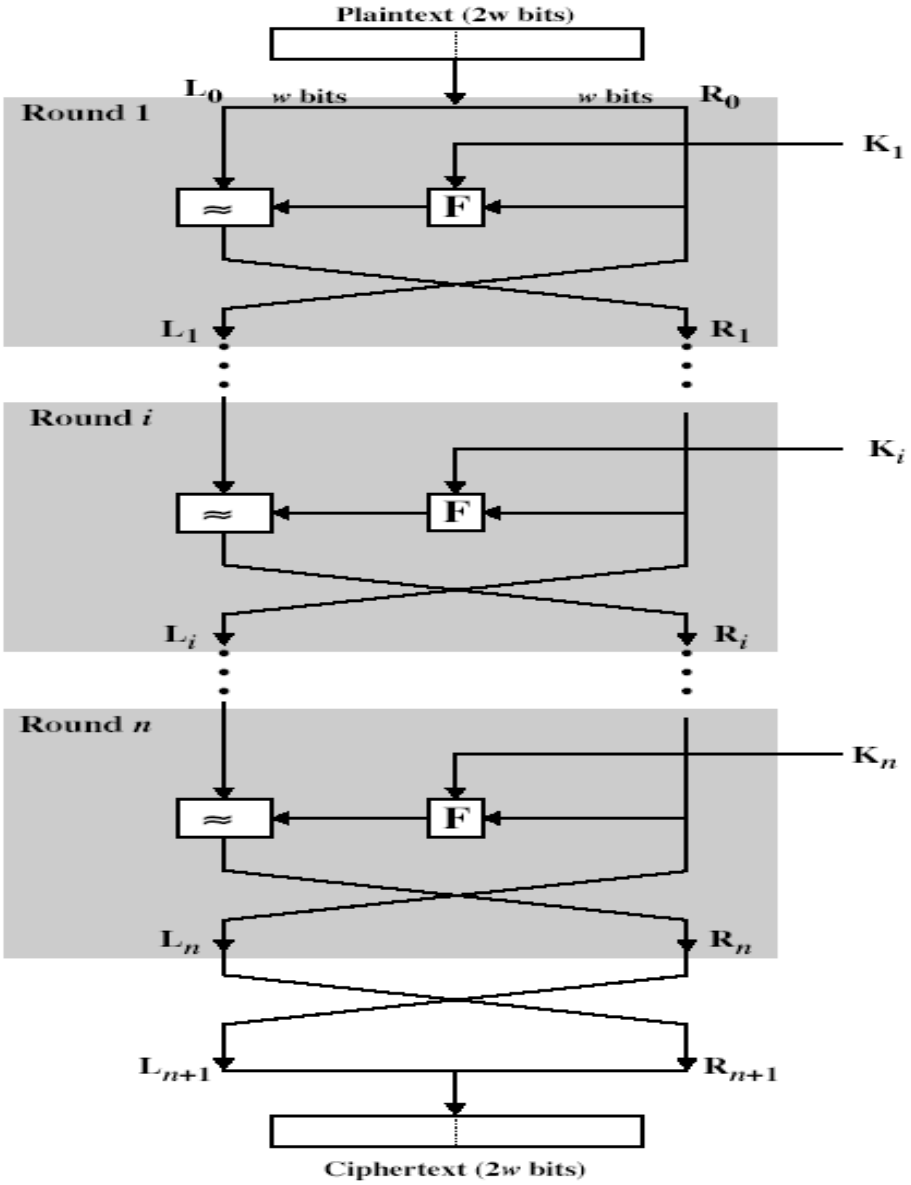
Конфузија и дифузија

- шифра би требало да буде потпуно лишена било каквих статистичких својстава оригиналне поруке
- one-time pad алгоритам ово постиже
- али практичније решење је дао Shannon који је предложио да се комбинују:
- **дифузија**
  - неутралише статистичке структуре оригиналне поруке у шифрованој поруци
  - постиже се тако што сваки бит (слово) оригиналне поруке утиче на много бита (слова) шифроване поруке
  - употреба неколико пермутација за редом, уз примену одређене функције на пермутацију након сваке пермутације
- **конфузија**
  - ствара што компликованију везу између шифроване поруке и кључа коришћеног за шифровање
  - ово се постиже коришћењем сложеног алгоритма замене

Структура Feistel алгоритма

- Horst Feistel је осмислио **feistel алгоритам**
  - базиран на концепту инвертабилних продукционих алгоритама
- дели улазни блок на две половине
  - обрађује их у више итерација које
  - изврше алгоритам замене за леву половину података
  - базирано на функцији итерације десне половине и поткључа
  - и на крају пермутацијом замени половине
- примењује Shannon-ов концепт S-P мрежа

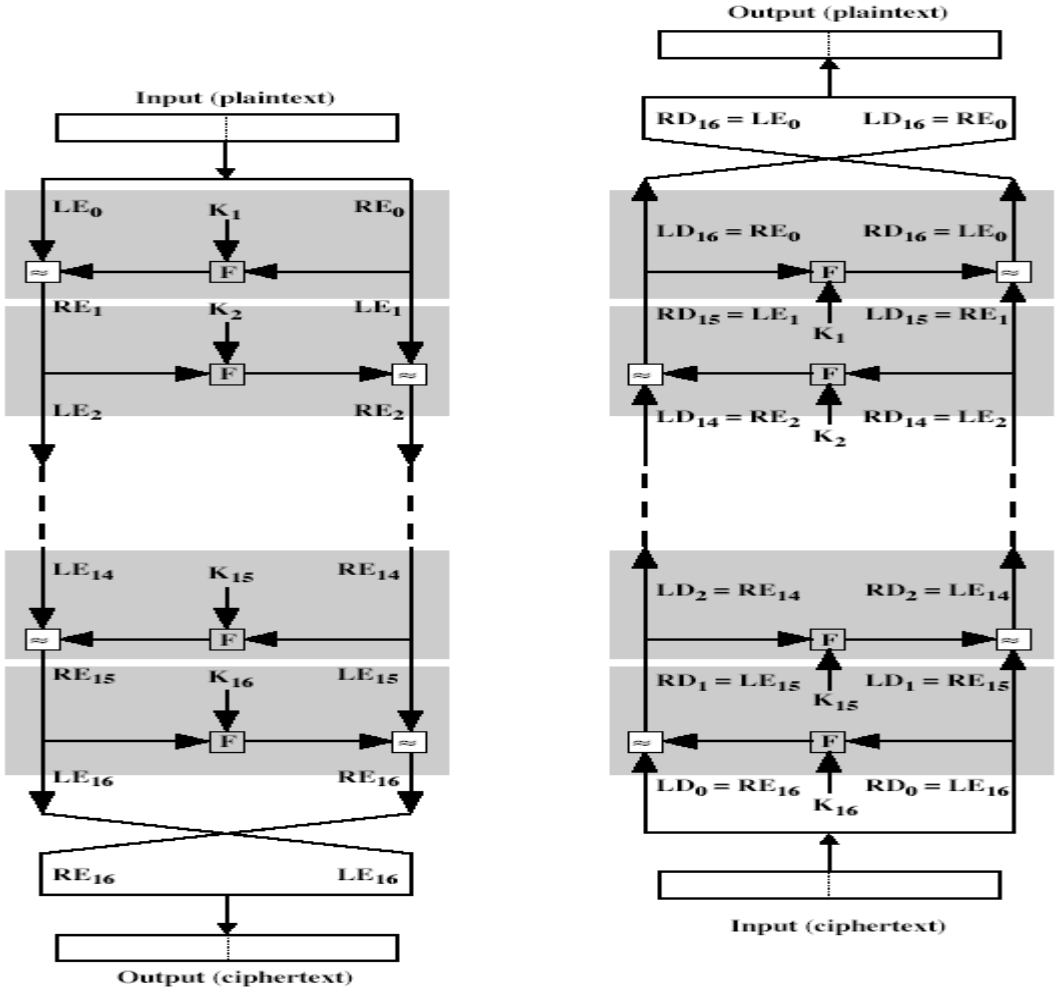
Структура Feistel алгоритма



Принципи дизајна Feistel алгоритма

- **величина блокова**
  - повећањем величине повећава се сигурност, али се успорава алгоритам
- **величина кључа**
  - повећање величине кључа побољшава безбедност, отежавајући претраживање свих кључева, али може да успори алгоритам
- **број итерација**
  - повећање броја, побољшава безбедност, али успорава алгоритам
- **генерисање поткључа**
  - већа комплексност може да отежа анализу, али успорава алгоритам
- **функција итерације**
  - већа комплексност може да отежа анализу, али успорава алгоритам

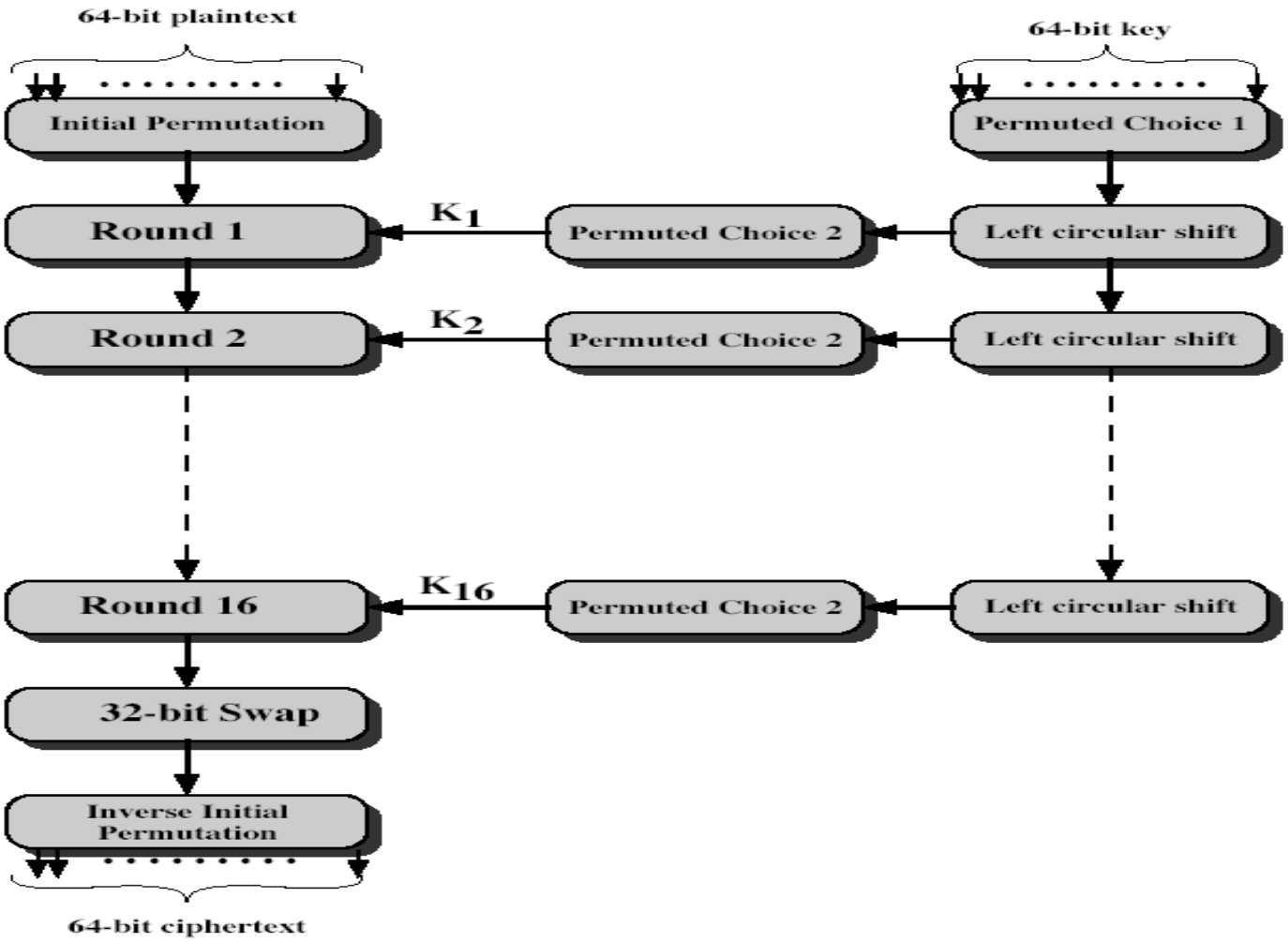
Дешифровање Feistel алгоритма



Data Encryption Standard (DES)

- најраспрострањенији блок алгоритам коришћен широм света
- усвојен 1977.
- шифрира 64-битни податак користећи 56-битни кључ
- има широк дијапазон намена

DES шифровање



Почетна пермутација (Initial Permutation IP)

- први корак у алгоритму
- IP прераспореди улазни низ битова
- парни битови иду у леву половину, непарни битови иду у десну половину, према дефинисаној таблици

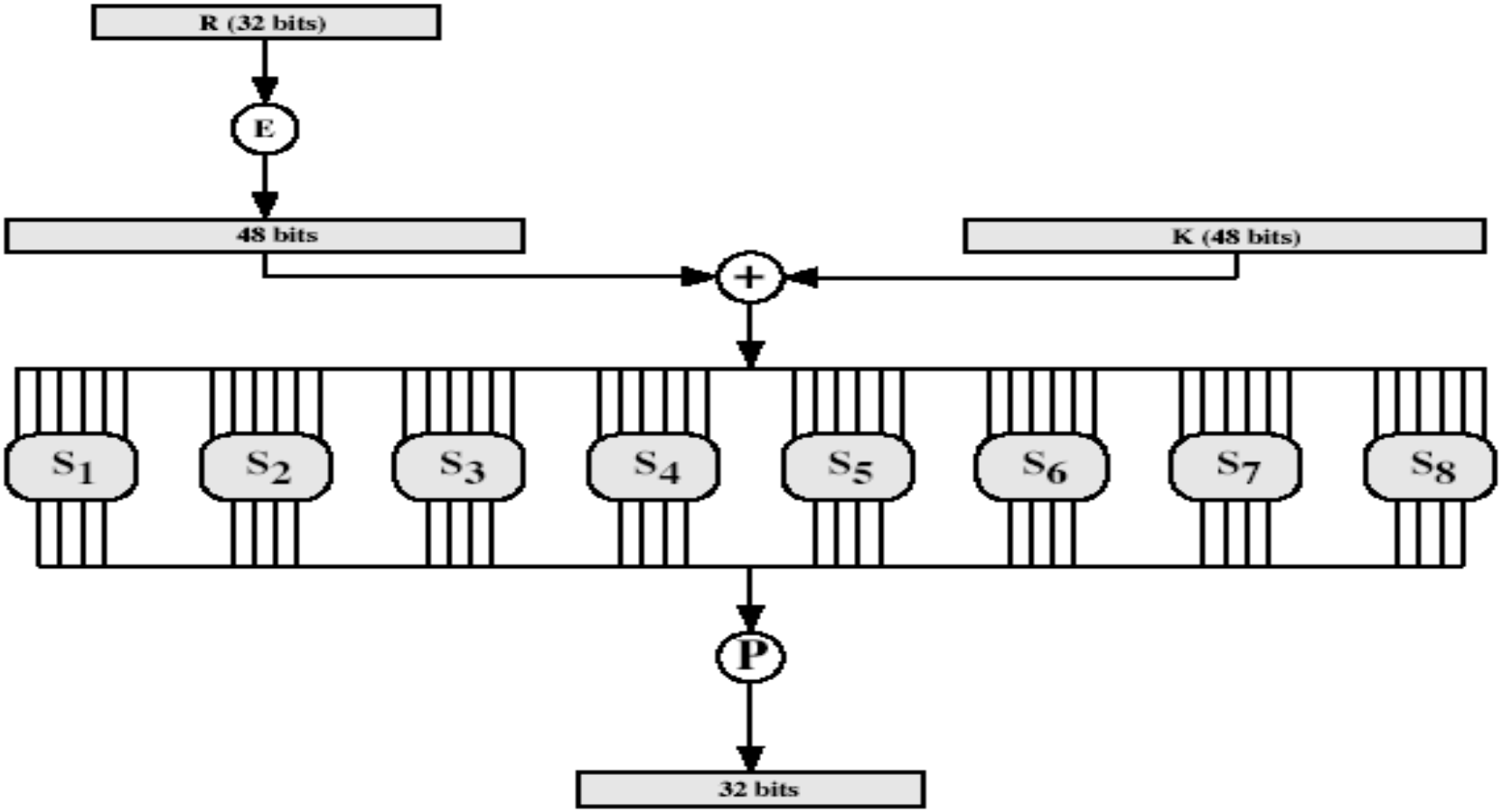
DES структура итерације

- користи две 32-битне половине (L & R)
- као код било ког Feistel алгоритма описане су са:

  $L_i = R_{i-1}$

  $R_i = L_{i-1}$ xor F(R_{i-1}, K_i)
- узима 32-бита десне (R) половине и 48-бита поткључа и:
  - проширује R на 48-битова користећи пермутацију E
  - додаје их поткључу
  - пролази кроз 8 замена (S-boxes) да би се добио 32-битни резултат
  - коначно пермутује ово користећи 32-битну пермутацију P

DES структура итерације



Алгоритми замене (Substitution Boxes S)

- постоји 8 алгоритама замене (S-boxes) који мапирају 6 у 4 бита
- сваки алгоритам замене (S-box) је у ствари 4 мала 4 битна алгоритма
  - спољни битови 1 & 6 (**row** bits) изаберу један ред
  - унутрашњи битови 2-5 (**col** bits) се мењају
  - резултат је 8 група од по 4 бита, тј. 32 бита
- избор реда зависи од података и од кључа истовремено

DES генерисање кључева

- Креће се од кључа дужине 64 бита, али сваки осми бит се игнорише, због тога кључ има 56 бита
- користи се за формирање поткључева за коришћење у свакој итерацији
- састоји се од:
  - почетне пермутације кључа (PC1) која одабира 56-битова у две 28-битне половине
  - 16 нивоа који се састоје од:
    - избора 24-бита из сваке половине
    - пермутује их помоћу PC2 за коришћење у f,
    - ротира **сваку половину** посебно за 1 или 2 места у зависности од **распореда ротације кључа** K

DES дешифровање

- пролазе се кораци уназад у односу на шифровање
- помоћу Feistel алгоритма, понови се шифровање
- коришћењем поткључева у обрнутом редоследу (SK16 … SK1)
- обратити пажњу да IP сакрива коначно решење
- први пролаз са SK16 открива 16-ту итерацију шифровања
- ….
- 16-ти пролаз са SK1 открива 1-ву итерацију шифровања
- на крају FP (final permutation) открива почетну поруку

Ефекат лавине

- пожељно својство криптографског алгоритма
- промена једног бита у улазној поруци или кључу изазива промену приближно половине бита на излазу
- ово чини откривање шифре, погађањем кључа практично немогућим
- DES представља јаку лавину

Сигурност DES – величина кључа

- 56-битни кључеви имају $2^{56} = 7.2 \times 10^{16}$ вредности
- brute force претрага је тешка
- међутим, могућа са напретком технологије
  - 1997. за пар месеци
  - 1998. за неколико дана
  - 1999. за 22 сата
- мада, и даље је потребно препознати поруку када се до ње стигне

Сигурност DES – напади на основу дужине трајања

- напад на имплементацију алгоритма
- користи се познавање детаља имплементације да се претпоставе неки или сви битови поткључа
- нарочито узимајући у обзир чињеницу да калкулације могу да трају различито времена у зависности од улазних вредности

Сигурност DES – аналитички напади

- било је неколико аналитичких напада на DES
- користе структуру алгоритма у дубину
  - сакупљају податке о шифровању
  - може довести до откривања неких бита или целих поткључева
  - ако је потребно може се испробавањем доћи до осталих бита поткључа
- уопштено, ово су статистички напади
- укључују
  - диференцијалну криптоанализу
  - линеарну криптоанализу
  - повезане нападе на кључ