Securitatea Sistemelor Informatice

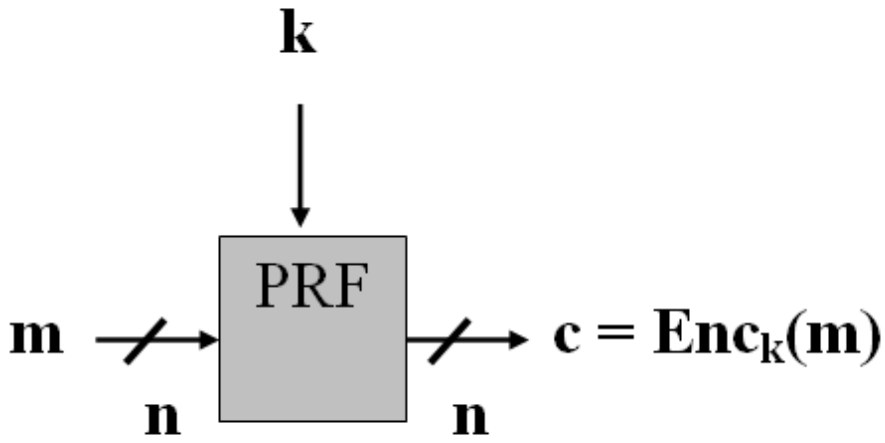
- Curs 6.2 -
Construcții practice PRF

Adela Georgescu

Facultatea de Matematică și Informatică
Universitatea din București
Anul universitar 2022-2023, semestrul I

Sisteme bloc ca PRF

- Am văzut că sistemele de criptare bloc folosesc *PRF*;



Sisteme bloc ca PRF

- În criteriile de evaluare pentru adoptarea AES s-a menționat:

  The security provovided by an algorithm is the most important factor... Algorithms will be judged on the following factors...

  The extent to which the **algorithm output is indistinguishable from a random permutation on the input block**.

Sisteme bloc ca PRF

- În criteriile de evaluare pentru adoptarea AES s-a menționat:

  The security provovided by an algorithm is the most important factor... Algorithms will be judged on the following factors...

  The extent to which the **algorithm output is indistinguishable from a random permutation on the input block**.

- Întrebare: Cum se obțin *PRF* în practică?

Paradigma confuzie-difuzie

- Se construiește funcția F, pe baza mai multor funcții aleatoare f_i de dimensiune mai mică;

Paradigma confuzie-difuzie

- Se construiește funcția F, pe baza mai multor funcții aleatoare f_i de dimensiune mai mică;
- Considerăm F pe 128 biți și 16 funcții aleatoare $f_1, \ldots, f_{16}$ pe câte 8 biți;

Paradigma confuzie-difuzie

- Se construiește funcția F, pe baza mai multor funcții aleatoare f_i de dimensiune mai mică;
- Considerăm F pe 128 biți și 16 funcții aleatoare $f_1, \ldots, f_{16}$ pe câte 8 biți;
- Pentru $x = x_1 || \ldots || x_{16}$, $x \in \{0,1\}^{128}$ $x_i \in \{0,1\}^8$:

$$F_k(x) = f_1(x_1) || \ldots || f_{16}(x_{16})$$

Paradigma confuzie-difuzie

- Se construiește funcția F, pe baza mai multor funcții aleatoare f_i de dimensiune mai mică;
- Considerăm F pe 128 biți și 16 funcții aleatoare $f_1, \ldots, f_{16}$ pe câte 8 biți;
- Pentru $x = x_1 || \ldots || x_{16}$, $x \in \{0,1\}^{128}$ $x_i \in \{0,1\}^8$:

$$F_k(x) = f_1(x_1) || \ldots || f_{16}(x_{16})$$

- Spunem că $\{f_i\}$ introduc confuzie în F.

Rețele de substituție - permutare

F încă nu este PRF dar

- F se transformă în PRF în 2 pași:

Rețele de substituție - permutare

F încă nu este PRF dar

- F se transformă în PRF în 2 pași:
  - **Pasul 1**: se introduce difuzie prin amestecarea (permutarea) biților de ieșire;

Rețele de substituție - permutare

F încă nu este PRF dar

- F se transformă în PRF în 2 pași:
  - **Pasul 1**: se introduce difuzie prin amestecarea (permutarea) biților de ieșire;
  - **Pasul 2**: se repetă o rundă (care presupune *confuzie* și *difuzie*) de mai multe ori;

Rețele de substituție - permutare

F încă nu este PRF dar

- F se transformă în PRF în 2 pași:
  - **Pasul 1**: se introduce difuzie prin amestecarea (permutarea) biților de ieșire;
  - **Pasul 2**: se repetă o rundă (care presupune *confuzie* și *difuzie*) de mai multe ori;
- Repetarea *confuziei* și *difuziei* face ca modificarea unui singur bit de intrare să fie propagată asupra tuturor biților de ieșire;

Rețele de substituție - permutare

F încă nu este PRF dar

- F se transformă în PRF în 2 pași:
  - **Pasul 1**: se introduce difuzie prin amestecarea (permutarea) biților de ieșire;
  - **Pasul 2**: se repetă o rundă (care presupune *confuzie* și *difuzie*) de mai multe ori;
- Repetarea *confuziei* și *difuziei* face ca modificarea unui singur bit de intrare să fie propagată asupra tuturor biților de ieșire;

Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;

Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;
- $\{f_i\}$ se numesc S-boxes (Substitution-boxes);

Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;
- $\{f_i\}$ se numesc S-boxes (Substitution-boxes);
- Cum nu mai depind de cheie, aceasta este utilizată în alt scop;

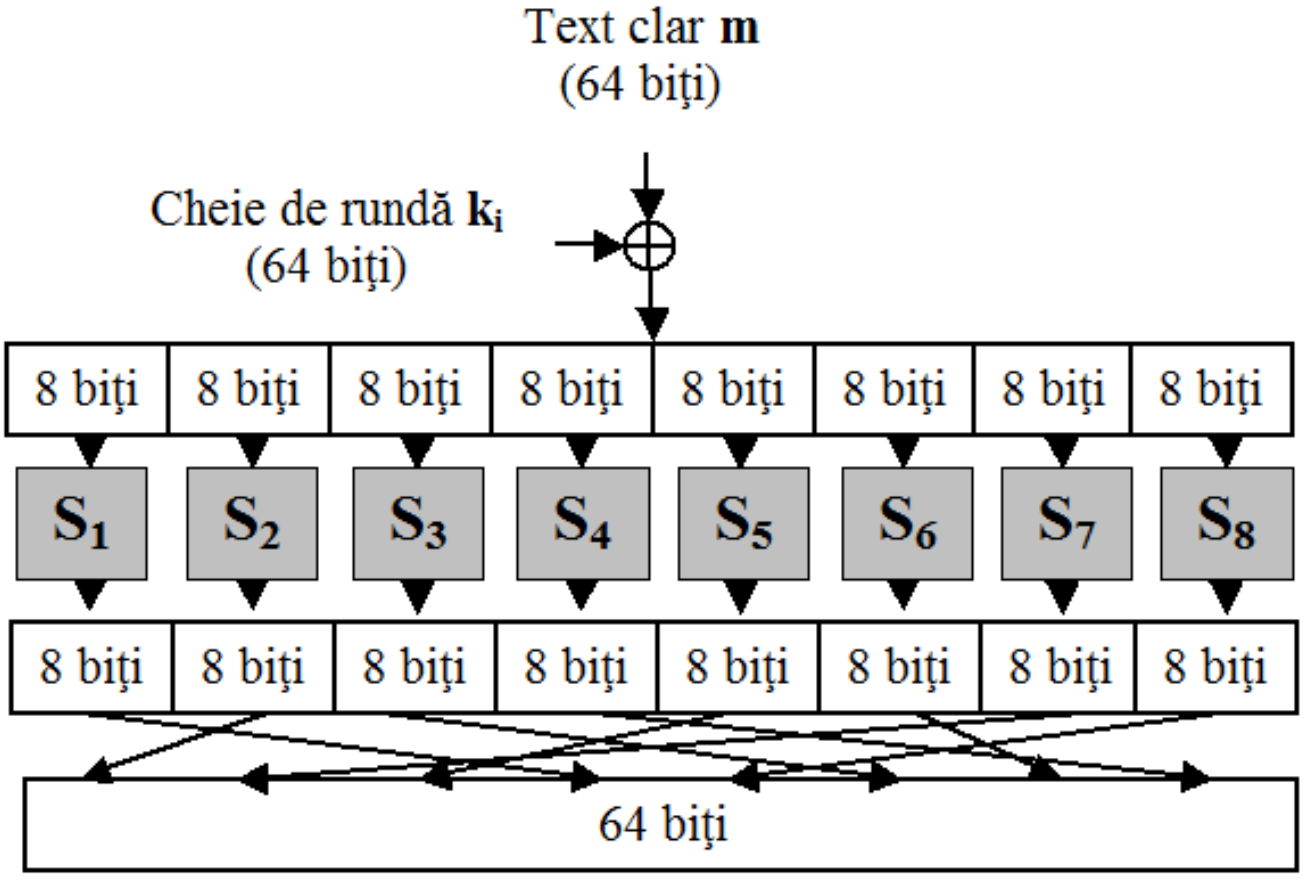
Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;
- $\{f_i\}$ se numesc S-boxes (Substitution-boxes);
- Cum nu mai depind de cheie, aceasta este utilizată în alt scop;
- Din cheie se obțin mai multe **chei de rundă** (*sub-chei*) în urma unui proces de derivare a cheilor (*key schedule*);

Rețele de substituție - permutare

- O rețea de substituție-permutare este o implementare a construcției anterioare de *confuzie-difuzie* în care funcțiile $\{f_i\}$ sunt **fixe** (i.e. nu depind de cheie) si se numesc permutări;
- $\{f_i\}$ se numesc S-boxes (Substitution-boxes);
- Cum nu mai depind de cheie, aceasta este utilizată în alt scop;
- Din cheie se obțin mai multe **chei de rundă** (*sub-chei*) în urma unui proces de derivare a cheilor (*key schedule*);
- Fiecare cheie de rundă este XOR-ată cu valorile intermediare din fiecare rundă.

Rețele de substituție - permutare



Rețele de substituție - permutare

- Există 2 principii de bază în proiectarea rețelelor de substituție - permutare:

Rețele de substituție - permutare

- Există 2 principii de bază în proiectarea rețelelor de substituție - permutare:
  - **Principiul 1**: Inversabilitatea S-box-urilor;
    - dacă toate S-box-urile sunt inversabile, atunci rețeaua este inversabilă;
    - necesitate funcțională (pentru decriptare)

Rețele de substituție - permutare

- Există 2 principii de bază în proiectarea rețelelor de substituție - permutare:
  - **Principiul 1**: Inversabilitatea S-box-urilor;
    - dacă toate S-box-urile sunt inversabile, atunci rețeaua este inversabilă;
    - necesitate funcțională (pentru decriptare)
  - **Principiul 2**: Efectul de avalanșă
    - Un singur bit modificat la intrare **trebuie** să afecteze toți biții din secvența de ieșire;
    - necesitate de securitate.

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;
- septembrie 1997 - 15 propuneri: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish;

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;
- septembrie 1997 - 15 propuneri: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish;
- 1998, 1999 - au loc 2 workshop-uri în urma carora ramân 5 finaliști: MARS, RC6, Rijndael, Serpent, Twofish;

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;
- septembrie 1997 - 15 propuneri: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish;
- 1998, 1999 - au loc 2 workshop-uri în urma carora ramân 5 finaliști: MARS, RC6, Rijndael, Serpent, Twofish;
- octombrie 2000 - după un al treilea workshop se anunță câștigătorul: Rijndael.

Exemplu: AES - Advanced Encryption Standard

- ianuarie 1997 - NIST anunță competiția pentru selecția unui nou sistem de criptare bloc care să înlocuiască DES;
- septembrie 1997 - 15 propuneri: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish;
- 1998, 1999 - au loc 2 workshop-uri în urma carora ramân 5 finaliști: MARS, RC6, Rijndael, Serpent, Twofish;
- octombrie 2000 - după un al treilea workshop se anunță câștigătorul: Rijndael.
- AES este folosit in multe standarde comerciale: IPsec, TLS, IEEE 802.11i (WPA2), SSH, Skype, etc.

AES - Advanced Encryption Standard

[Google Scholar - User profiles]

[`http://keccak.noekeon.org/team.html`]

Rijndael = **Rij**men + **Dae**men

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

- Folosește o matrice de octeți 4×4 numită **stare**;

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

- Folosește o matrice de octeți 4×4 numită **stare**;
- Starea inițială este mesajul clar ($4 \times 4 \times 8 = 128$);

Descriere AES

- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

- Folosește o matrice de octeți 4×4 numită **stare**;
- Starea inițială este mesajul clar ($4 \times 4 \times 8 = 128$);
- Starea este modificată pe parcursul rundelor prin 4 tipuri de operații: *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*;

Descriere AES

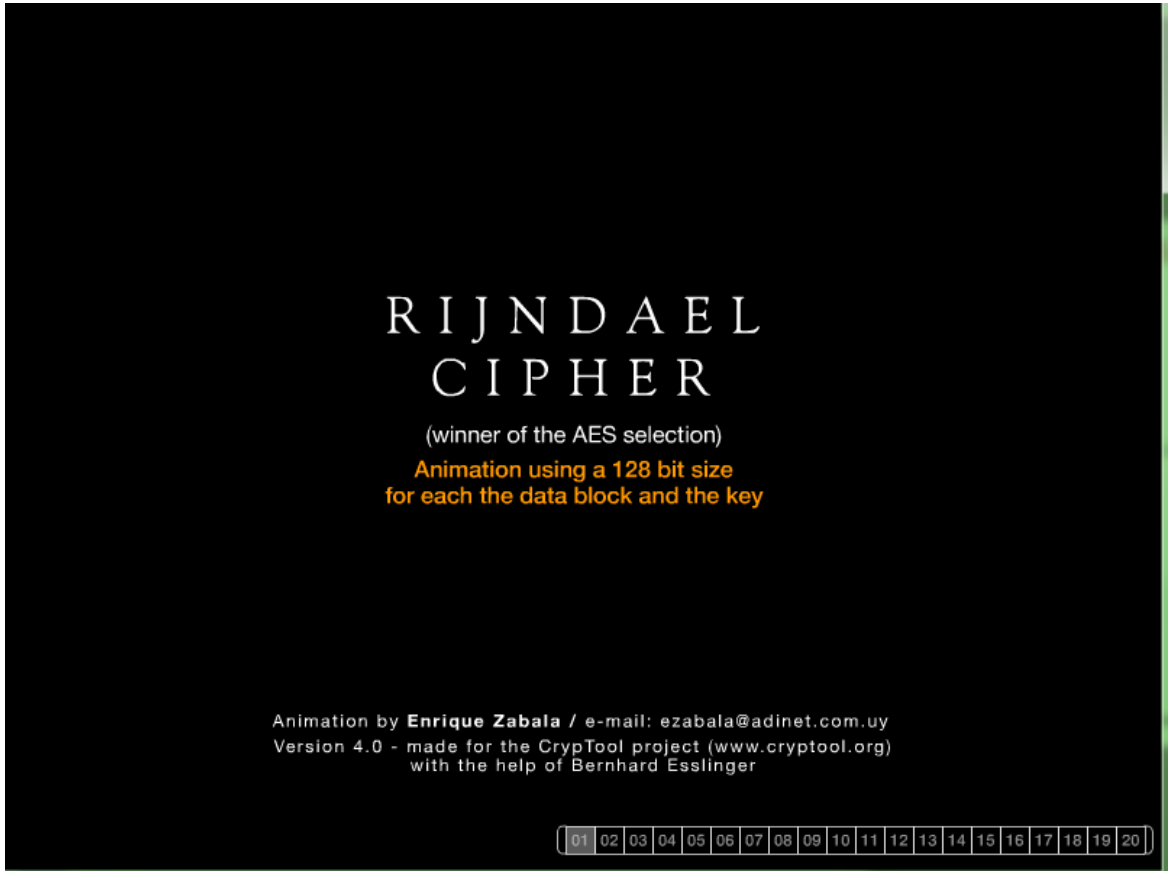
- AES este o rețea de substituție - permutare pe 128 biți care poate folosi chei de 128, 192 sau 256 biți;
- Lungimea cheii determină numărul de runde:

Lungime cheie (biți)	128	192	256
Număr runde	10	12	14

- Folosește o matrice de octeți 4×4 numită **stare**;
- Starea inițială este mesajul clar ($4 \times 4 \times 8 = 128$);
- Starea este modificată pe parcursul rundelor prin 4 tipuri de operații: *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*;
- Ieșirea din ultima rundă este textul criptat.

Descriere AES

- Rijndael Animation - CrypTool Project:



[http://www.cryptool.org/en/]

Securitatea sistemului AES

- Singurele atacuri netriviale sunt asupra AES cu număr redus de runde:
  - AES-128 cu 6 runde: necesită 2^{72} criptări;
  - AES-192 cu 8 runde: necesită 2^{188} criptări;
  - AES-256 cu 8 runde: necesită 2^{204} criptări.

Securitatea sistemului AES

- Singurele atacuri netriviale sunt asupra AES cu număr redus de runde:
  - AES-128 cu 6 runde: necesită 2^{72} criptări;
  - AES-192 cu 8 runde: necesită 2^{188} criptări;
  - AES-256 cu 8 runde: necesită 2^{204} criptări.
- Nu există un atac mai eficient decât căutarea exhaustivă pentru AES cu număr complet de runde.

Securitatea sistemului AES

- Singurele atacuri netriviale sunt asupra AES cu număr redus de runde:
  - AES-128 cu 6 runde: necesită 2^{72} criptări;
  - AES-192 cu 8 runde: necesită 2^{188} criptări;
  - AES-256 cu 8 runde: necesită 2^{204} criptări.
- Nu există un atac mai eficient decât căutarea exhaustivă pentru AES cu număr complet de runde.

"It is free, standardized, efficient, and highly secure."

(J.Katz, Y.Lindell, *Introduction to Modern Cryptography*)

Rețele Feistel

- Se aseamănă rețelelor de substituție-permutare în sensul că păstrează aceleași elementele componente: S-box, permutare, procesul de derivare a cheii, runde;

Rețele Feistel

- Se aseamănă rețelelor de substituție-permutare în sensul că păstrează aceleași elementele componente: S-box, permutare, procesul de derivare a cheii, runde;
- Se diferențiază de rețelele de substituție-permutare prin proiectarea de nivel înalt;

Rețele Feistel

- Se aseamănă rețelelor de substituție-permutare în sensul că păstrează aceleași elementele componente: S-box, permutare, procesul de derivare a cheii, runde;
- Se diferențiază de rețelele de substituție-permutare prin proiectarea de nivel înalt;
- Introduc avantajul major că S-box-urile NU trebuie să fie inversabile;

Rețele Feistel

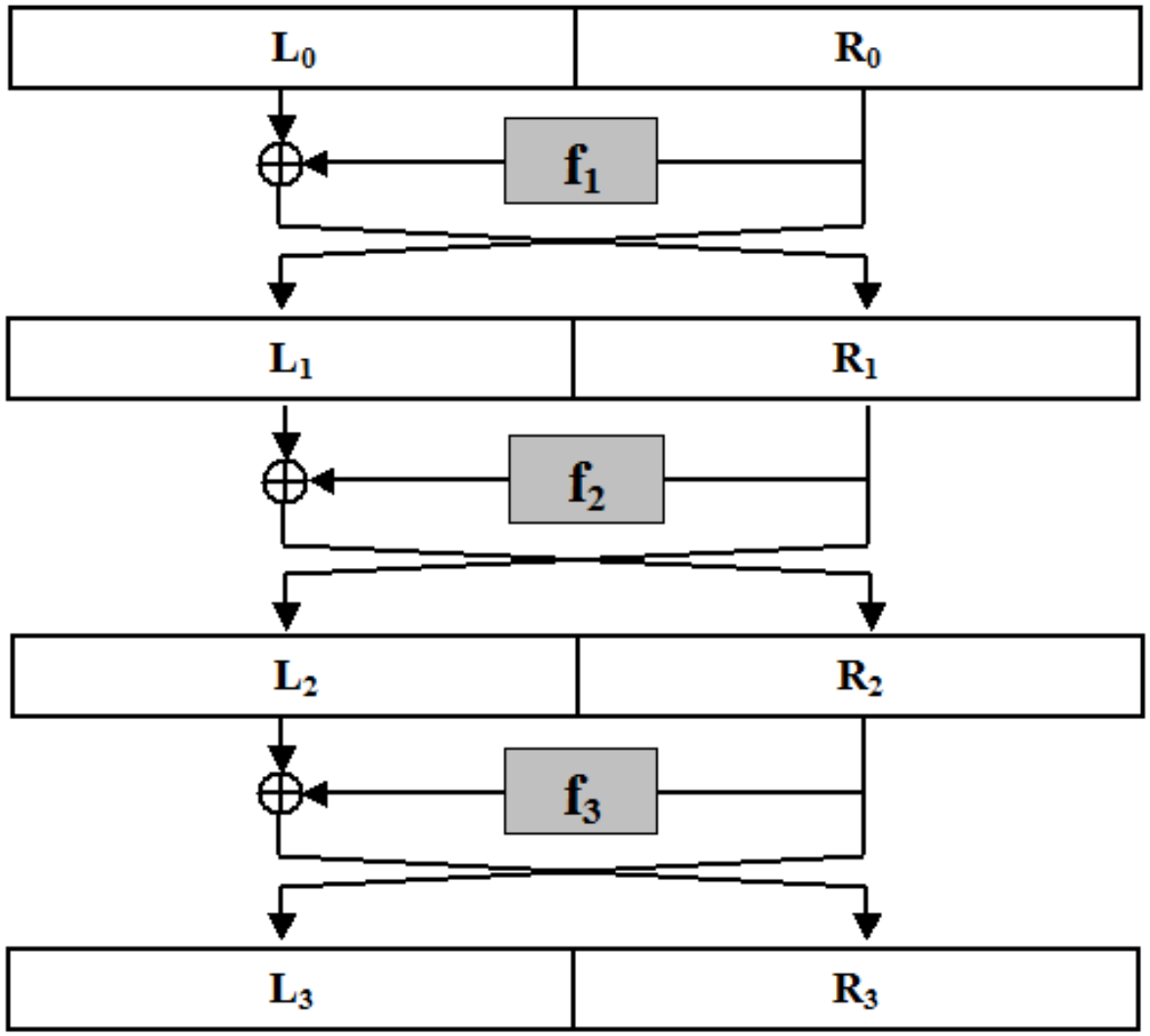
- Se aseamănă rețelelor de substituție-permutare în sensul că păstrează aceleași elementele componente: S-box, permutare, procesul de derivare a cheii, runde;
- Se diferențiază de rețelele de substituție-permutare prin proiectarea de nivel înalt;
- Introduc avantajul major că S-box-urile NU trebuie să fie inversabile;
- Permit așadar obținerea unei structuri *inversabile* folosind elemente *neinversabile*.

Horst Feistel (1915 - 1990)

[Wikipedia]

- Structurile simetrice utilizate în construcția sistemelor bloc poartă numele lui Feistel;
- Munca sa de cercetare la IBM a condus la sistemul de criptare Lucifer și mai târziu la DES.

Rețele Feistel



Rețele Feistel

- Intrarea în runda i se împarte în 2 jumătăți: L_{i-1} și R_{i-1} (i.e. *Left* și *Right*);

Rețele Feistel

- Intrarea în runda i se împarte în 2 jumătăți: L_{i-1} și R_{i-1} (i.e. *Left* și *Right*);
- Ieșirile din runda i sunt:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f_i(R_{i-1})$$

Rețele Feistel

- Intrarea în runda i se împarte în 2 jumătăți: L_{i-1} și R_{i-1} (i.e. *Left* și *Right*);
- Ieșirile din runda i sunt:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f_i(R_{i-1})$$

- Funcțiile f_i depind de cheia de rundă, derivând dintr-o funcție publică $\widehat{f_i}$:

$$f_i(R) = \widehat{f_i}(k_i, R)$$

Rețele Feistel

- Rețelele Feistel sunt inversabile indiferent dacă funcțiile f_i sunt inversabile sau nu;

Rețele Feistel

- Rețelele Feistel sunt inversabile indiferent dacă funcțiile f_i sunt inversabile sau nu;
- Fie (L_i, R_i) ieșirile din runda i;

Rețele Feistel

- Rețelele Feistel sunt inversabile indiferent dacă funcțiile f_i sunt inversabile sau nu;
- Fie (L_i, R_i) ieșirile din runda i;
- Intrările (L_{i-1}, R_{i-1}) în runda i sunt:

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f_i(R_{i-1})$$