Nivelul prezentare

Nivelul prezentare

- Rolul nivelului prezentare:
  - Reprezentarea datelor:
    - Date cu tip;
    - Sintaxa de transfer;
    - Conversia (Exemplu: ASN.1).
  - Compresia datelor (pentru reducerea volumului);
  - Criptarea datelor (pentru protecţia lor).

Securitate: persoane ce generează probleme

Adversar	Scop
Student	Pentru a se distra furând poşta electronică a celorlalţi.
Spărgător	Pentru a testa securitatea sistemului cuiva; pentru a fura date.
Responsabil de vânzări	Pentru a pretinde că reprezintă toată Europa, nu numai Andorra.
Om de afaceri	Pentru a descoperi planul strategic de marketing al competitorului.
Fost funcţionar	Pentru a se răzbuna că a fost concediat.
Contabil	Pentru a sustrage bani de la o companie.
Agent de vânzări	Pentru a nega o promisiune făcută clientului prin poştă electronică.
Şarlatan	Pentru a fura numere de cărţi de credit şi a le vinde.
Spion	Pentru a afla puterea militară a inamicului sau secrete industriale.
Terorist	Pentru a fura secrete legate de conflicte armate.

Probleme de securitate

- Confidenţialitatea:
  - Păstrarea informaţiei din mesaj departe de utilizatorii neautorizaţi.
- Autentificarea:
  - Determinarea identităţii persoanei cu care se schimbă mesaje înainte de a dezvălui informaţii importante.
- Controlul accesului:
  - Protecţia împotriva accesului neautorizat.
- Integritatea:
  - Mesajul primit nu a fost modificat sau măsluit în tranzit.
- Non-repudierea:
  - Transmiţătorul nu poate nega transmiterea unui mesaj pe care un receptor l-a primit.

Metode de rezolvare

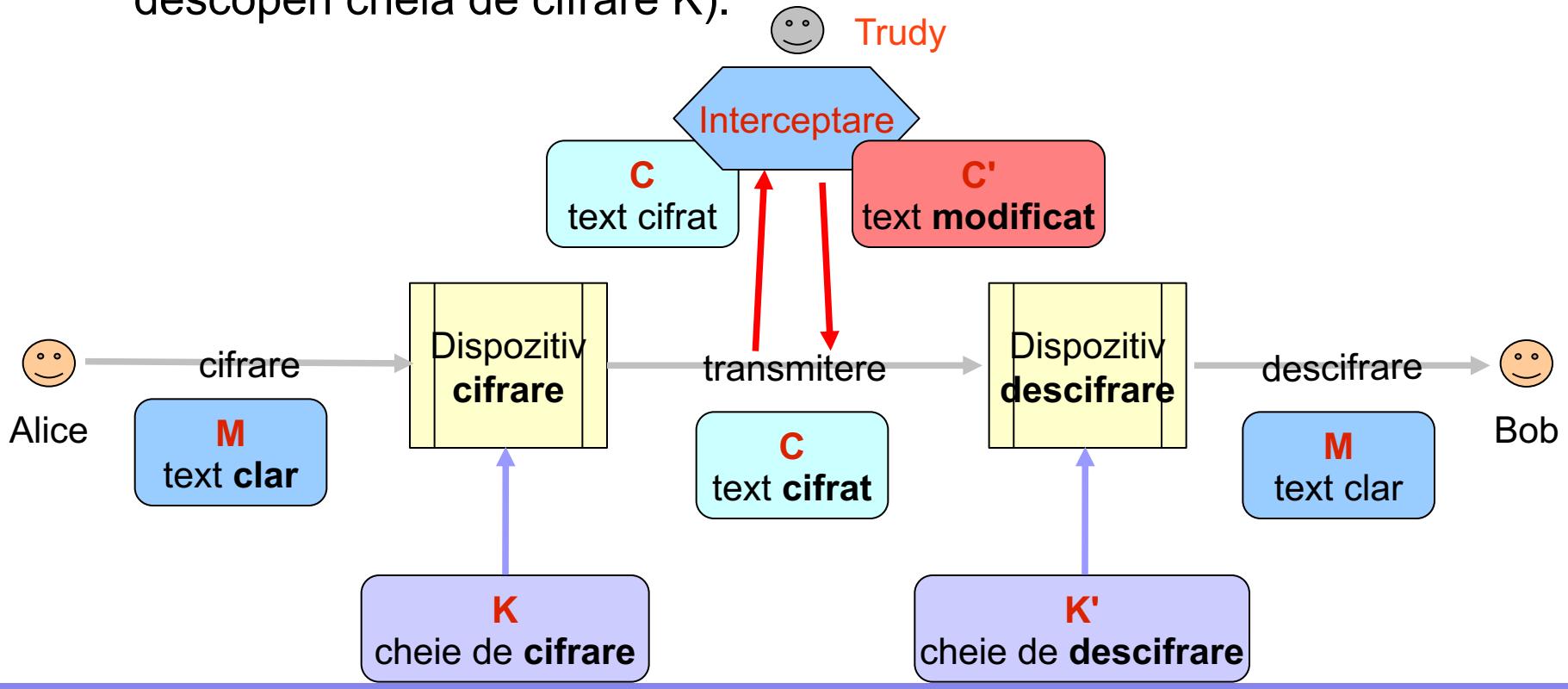
- Organizare:
  - Servicii (protocoale) de securitate.
  - Mecanisme de securitate:
    - Criptare.
    - Rezumare (hash).
    - Semnătură digitală.
  - Algoritmi de criptare şi hash.
- Securitatea în ierarhia de protocoale:
  - Fizic: tuburi de securizare a liniilor de transmisie.
  - Legătură de date: legături criptate.
  - Reţea: ziduri de protecţie (firewalls), IPsec.
  - Transport: end-to-end security.
  - Aplicaţie: autentificarea, non-repudierea.

Alte aspecte

- Politici de securitate.
- Control software (antivirus).
- Control hardware:
  - Cartele inteligente;
  - Biometrie.
- Control fizic (protecţie).
- Educaţie.
- Măsuri legale.

Modelul de bază al criptării

- Transmiterea unui mesaj de la Alice la Bob:
  - Confidenţialitatea: intrusul (Trudy) să nu poată reconstitui M din C (adică să nu poată descoperi cheia de descifrare K‘).
  - Autentificarea: intrusul să nu poată introduce un text cifrat C', fără ca acest lucru să fie detectat de destinatar (adică să nu poată descoperi cheia de cifrare K).



Definiţii

- Spargerea cifrurilor = criptanaliză.
- Proiectarea cifrurilor = criptografie.
- Ambele sunt subdomenii ale criptologiei.
- Transformarea F realizată la cifrarea unui mesaj:

F : {M} x {K} -> {C}, unde:

  - {M} este mulţimea mesajelor;
  - {K} este mulţimea cheilor;
  - {C} este mulţimea criptogramelor.

- Operaţii:
  - Cifrarea: $C = E_k(M)$.
  - Descifrarea: $M = D_{k'}(C)$.
- Conotaţie de ordin practic!

Problema criptanalistului

- Criptanaliză cu **text cifrat cunoscut**; se cunosc:
  - Un text cifrat;
  - Metoda de criptare;
  - Limbajul textului clar;
  - Subiectul;
  - Anumite cuvinte din text.
- Criptanaliză cu **text clar cunoscut**; se cunosc:
  - Un text clar;
  - Textul cifrat corespunzător;
  - Anumite cuvinte cheie (login).
- Criptanaliză cu **text clar ales**; se cunosc:
  - Mod cifrare anumite porţiuni de text;
  - Exemplu pentru o bază de date - modificare / efect.

Caracteristicile sistemelor secrete

- Sistem **necondiţionat sigur**:
  - Rezistă la orice atac, indiferent de cantitatea de text cifrat interceptat (exemplu: one time pad).
- **Computaţional sigur** sau **tare**:
  - Nu poate fi spart printr-o analiză sistematică cu resursele disponibile.
- **Sistem ideal**:
  - Indiferent de volumul textului cifrat care este interceptat, o criptogramă nu are o rezolvare unică, ci mai multe, cu probabilităţi apropiate.

Cerinţe criptosisteme cu chei secrete

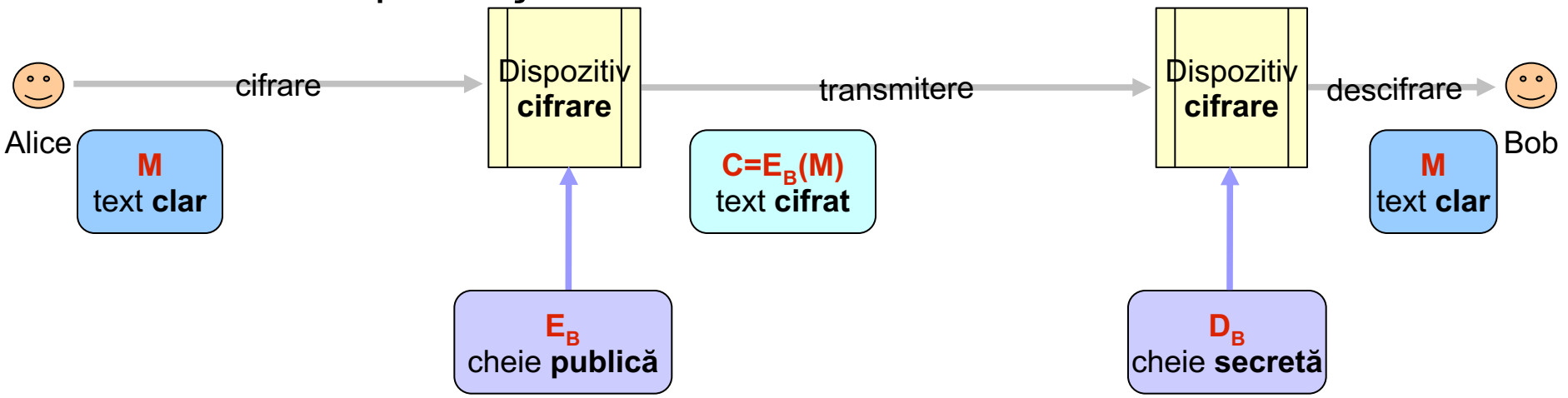
- Cerinţe generale:
  - Cifrare şi descifrare eficiente pentru toate cheile.
  - Sistem uşor de folosit (chei de transformare).
  - Securitatea să depindă de chei, nu de algoritm.
- Cerinţe specifice pentru confidenţialitate: să fie imposibil computaţional ca un criptanalist să determine sistematic:
  - Transformarea D_k din C, chiar dacă ar cunoaşte M.
  - M din C (fără a cunoaste D_k).
- Cerinţe specifice pentru autentificare: să fie imposibil computaţional ca un criptanalist să determine sistematic:
  - Transformarea E_k, din C, chiar dacă ar cunoaşte M.
  - Cifrul C' astfel ca $D_k(C')$ să fie un mesaj valid (fără a cunoaşte E_k).

Modelul criptografic cu chei publice

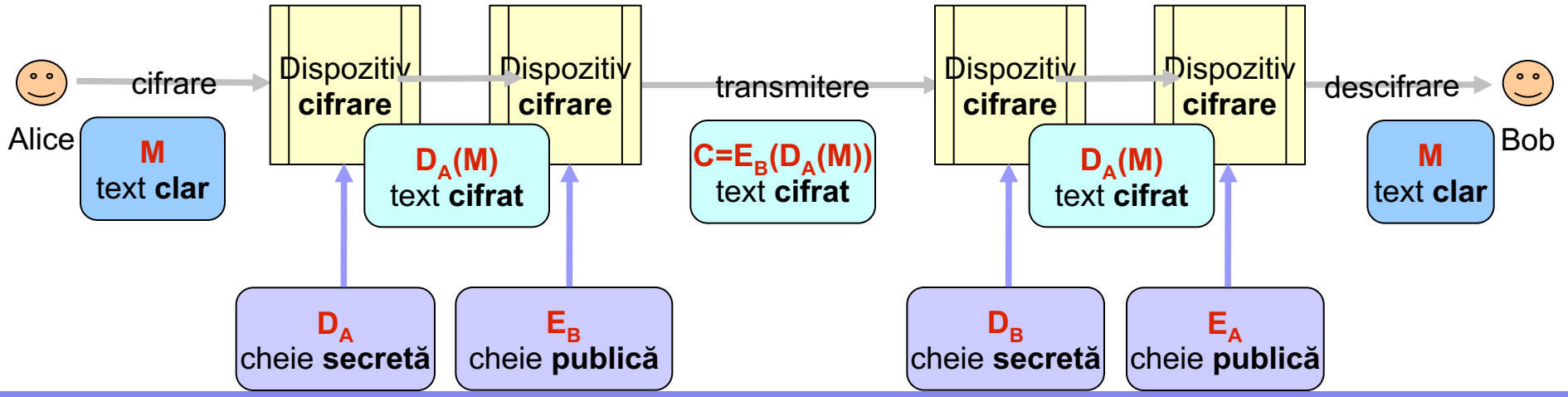
- Sistemele criptografice:
  - Simetrice.
  - Asimetrice:
    - Propuse de Diffie şi Hellman în 1976.
    - Chei diferite de cifrare E şi descifrare D.
    - Nu se pot deduce (uşor) una din alta, mai precis:
      - D(E(M)) = M;
      - Este extrem de greu să se deducă D din E;
      - E nu poate fi "spart" prin criptanaliză cu text clar ales.
- Într-un sistem asimetric, fiecare utilizator U:
  - Face publică cheia (transformarea) E_u de cifrare.
  - Păstrează secretă cheia (transformarea) D_u de descifrare.
- Schema de autentificare:
  - Condiţia necesară este ca transformările E_a şi D_a să comute, adică $E_a(D_a(M)) = D_a(E_a(M)) = M$.

Scheme folosite

- Schema de protecţie:



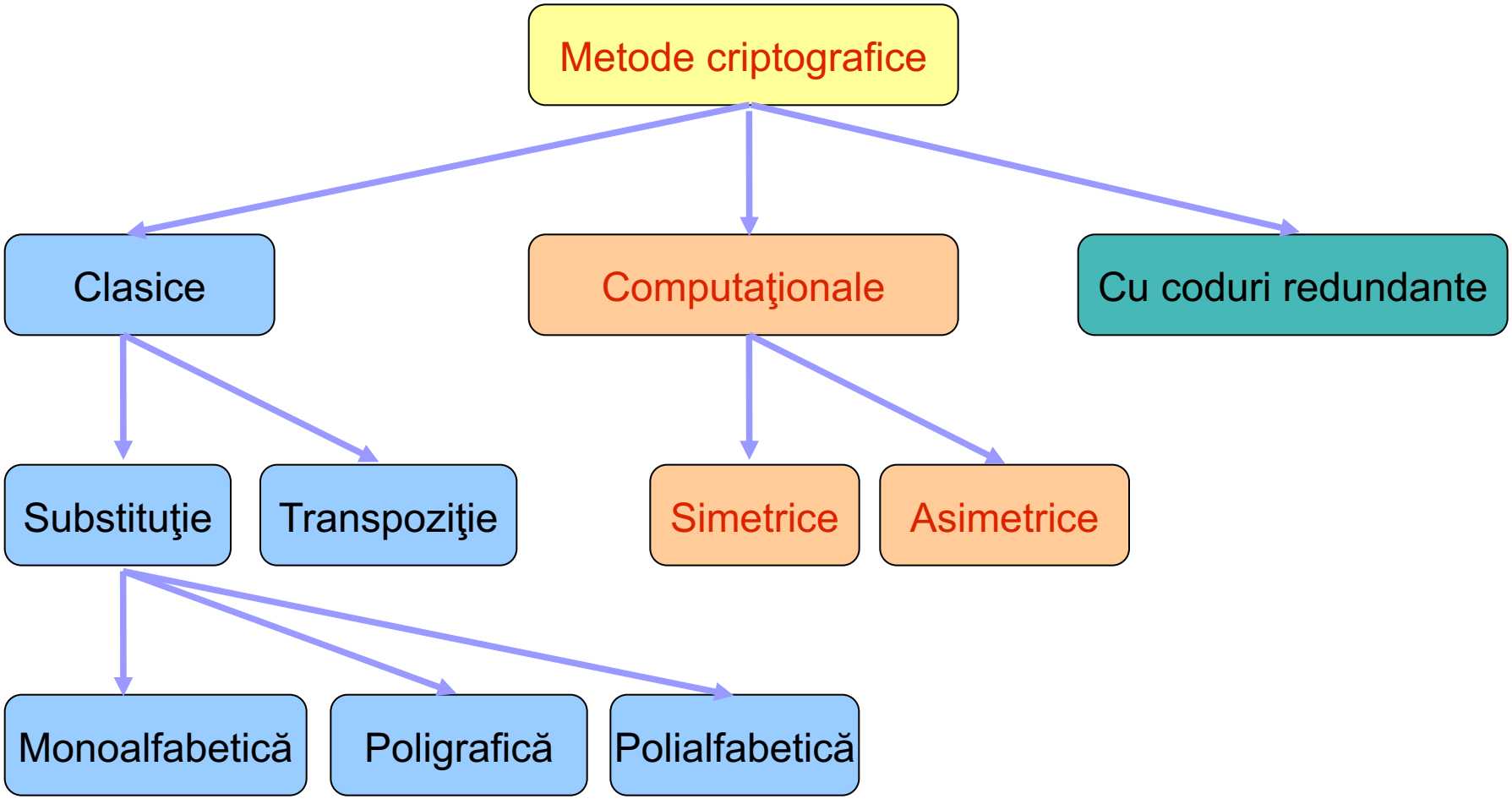
- Schema de autentificare:



Scheme folosite

- Se asigură:
  - Confidenţialitate: nu se trimite nimic în clar.
  - Autentificare: Bob are garanţia că Alice este sursa mesajului.
    - Semnătură digitală: se poate semna doar rezumatul mesajului.
  - Non-repudiere: folosind perechea $D_A(M)$ şi M.

Clasificare generală



Utilizarea TI în evaluarea algoritmilor criptografici

- Analogia între transmisie şi confidenţialitate:
  - perturbarea <=> cifrarea mesajului;
  - mesajul recepţionat <=> text cifrat.
- Cantitatea de informaţie = entropie.
  - X_1,..., X_n: mesajele unei surse;
  - $p(X_1)$, ..., $p(X_n)$: probabilităţile ($\Sigma_{i=1,n}\ p(X_i) = 1$).
- Entropia unui mesaj: $H(X) = -\Sigma_{i=1,n}\ p(X_i)*\log p(X_i)$
  - Intuitiv: log(1/p(X)) = numărul de biţi folosiţi pentru codificarea optimă a lui X.
- Entropia măsoară şi incertitudinea:
  - H(X) maxim când $p(X_1) = p(X_2) = ... = p(X_n) = 1/n$.
  - H(X) descreşte când distribuţia mesajelor se restrânge.
  - H(X) = 0 când $p(X_i) = 1$ pentru un mesaj i.

Echivocitatea

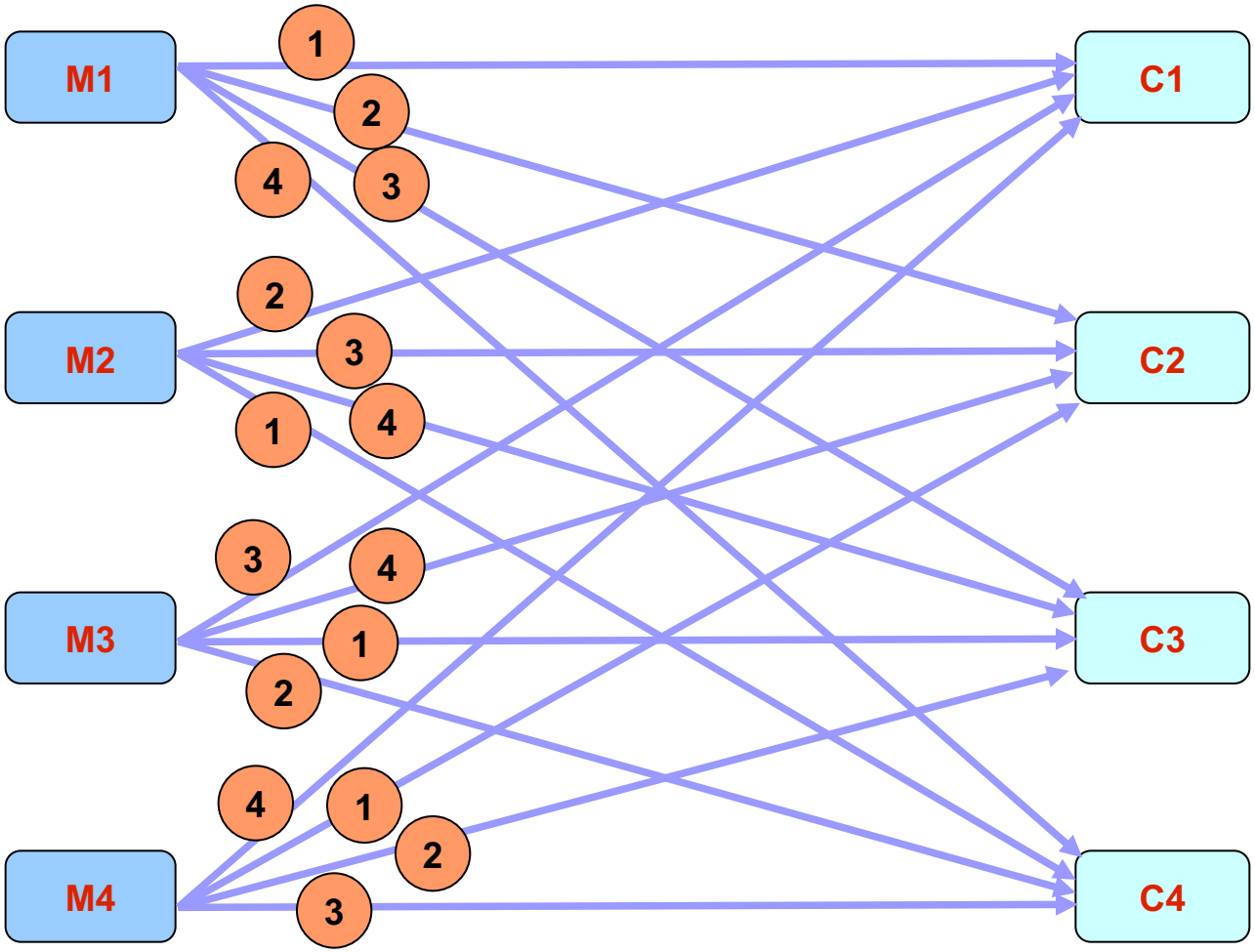
- Dat fiind Y din mulţimea mesajelor $Y_1,..., Y_n$ cu $\Sigma_{i=1,n}\ p(Y_i) = 1$, fie:
  - $p_Y(X)$ probabilitatea mesajului X condiţionat de Y.
  - p(X,Y) probabilitatea mesajelor X şi Y luate împreună:
    - $p(X,Y) = p_Y(X)*p(Y)$.
- Echivocitatea este entropia lui X condiţionat de Y:
  - $H_Y(X) = -\Sigma_{X,Y}\ p(X,Y)*\log\ p_Y(X)$
  - $H_Y(X) = \Sigma_{X,Y}\ p_Y(X)*p(Y)\log\ (1/p_Y(X)) = \Sigma_Y\ p(Y)\Sigma_X\ p_Y(X)*\log\ (1/p_Y(X))$.
- Exemplu:
  - n = 4 şi p(X) = 1/4 pentru fiecare X => H(X) = log 4 = 2.
  - Fie m=4 şi p(Y) = 1/4 pentru fiecare Y.
  - Presupunem că fiecare Y restrânge X:
    - Y1: X1 sau X2, Y2: X3 sau X4, Y3: X2 sau X3, Y4: X4 sau X1.
  - Echivocitatea este: $H_Y(X) = 4\ (\ (1/4)\ 2\ (1/2)\ \log\ 2) = \log\ 2 = 1$.
    - Adică: cunoaşterea lui Y reduce incertitudinea lui X la un bit.

Confidenţialitatea perfectă

- Fie:
  - M texte clare cu probabilitatea p(M), Σ_M p(M) = 1.
  - C criptograme, cu probabilitatea p(C), Σ_C p(C) = 1.
  - K chei cu probabilitatea p(K), Σ_K p(K) = 1.
  - p_C(M) probabiltiatea să se fi transmis M când se recepţionează C.
- Confidenţialitatea perfectă <=> p_C(M) = p(M).
- p_M(C) probabilitatea să se recepţioneze C când s-a transmis M:
  - $p_M(C) = \Sigma_{k,\ Ek(M)=C}\ p(k)$.
- Confidenţialitatea perfectă:
  - p_M(C) = p(C), pentru toate M şi orice C.

Confidenţialitatea perfectă

- Confidenţialitatea perfectă este posibilă dacă se folosesc chei la fel de lungi ca mesajele codificate.



Distanţa de unicitate

- Confidenţialitatea:
  - Cantitatea de incertitudine în K, dat fiind C, adică:
    - $H_C(K) = \Sigma_C\, p(C) \Sigma_K\, p_C(K) \log (1/p_C(K))$
- Dacă $H_C(K)=0$ nu există incertitudine şi cifrul se poate sparge.
- Când creşte lungimea N a textelor cifrate echivocitatea scade.
- Distanţa de unicitate:
  - Cel mai mic N pentru care $H_C(K)$ este foarte apropiat de 0.
- Cifru neconditionat sigur:
  - $H_C(K)$ nu se apropie niciodată de 0.

Calcul aproximativ distanţă unicitate

- Notaţii:
  - Pentru un limbaj, luăm mulţimea mesajelor de lungime N.
  - Rata limbajului:
    - r = H(X) / N
    - r = 1 ... 1.5 pentru limba engleză.
  - Rata absolută a limbajului (pentru L simboluri):
    - $R = \log L = -\sum_{i=1,L} (1/L) \log (1/L)$
    - R = log 26 = 4.7 biţi pe literă pentru limba engleză.
  - Redundanţele apar din structura limbajului: distribuţia frecvenţelor literelor, digramelor, trigramelor, etc.):
    - D = R - r
    - D = 3.2 ... 3.7 în limba engleză.

Calcul aproximativ distanţă unicitate

- Ipoteze:
  - Sunt 2^{RN} mesaje posibile de lungime N, din care 2^{rN} au sens.
  - Toate mesajele cu sens au aceeaşi probabilitate, $1/2^{rN}$.
  - Toate mesajele fără sens au probabilitate 0.
  - Sunt $2^{H(K)}$ chei cu probabilităţi egale.
  - Cifrul este aleator:
    - Pentru fiecare k şi C, descifrarea $D_K(C)$ este variabilă aleatoare independentă uniform distribuită pe toate mesajele, cu sau fără sens.
- Fie cifrarea $C = E_K(M)$.
  - Criptanalistul are de ales între $2^{H(K)}$ chei, doar una este corectă.
  - Rămân $2^{H(K)}-1$ chei cu aceeaşi probabilitate q de a da soluţie falsă:
    - Acelaşi C se obţine criptând un alt mesaj M' cu înţeles, cu altă cheie K'
      - $q = 2^{rN} / 2^{RN} = 2^{-DN}$ (D = R-r este redundanţa limbajului).
    - Numărul de soluţii false N:
      - $F = (2^{H(K)} -1)q = (2^{H(K)} -1)\, 2^{-DN} \approx 2^{H(K)-DN}$
      - $\log F = H(K)-DN = 0$
      - $N = H(K) / D$

Cifrarea prin substituţie

- Cifrul lui Cezar (substituţie monoalfabetică):

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

  - Relaţia de calcul: c[i] = (m[i] + 3) mod 26.
    - Textul clar: **CRIPTOGRAFIE**
    - Textul cifrat: **FULSWRJUDILH**
  - În general: c[i] = (a*m[i] +b) mod n.
- Substituţia polialfabetică (Vigenere):
  - Foloseşte 36 de cifruri Cezar şi o cheie de cifrare de lungime l.
  - Exemplu: cheia POLIGRAF:

POLIGRAFPOLIGRAGPOLIGRAFPOLIGRAFPOLI
AFOSTODATACANPOVESTIAFOSTCANICIODATA
PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI

Cifrarea prin substituţie

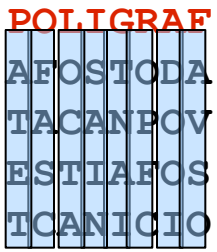
- Cifrul Beaufort:
  - Cifrare: c[i] = (k[i] - m[i]) mod n.
  - Descifrare: m[i] = (k[i] - c[i]) mod n
- Substituţia poligrafică:
  - Un grup de n litere este înlocuit cu un alt grup de n litere.

Analiza cifrării prin substituţie

- Substituţie monoalfabetică:
  - N = H(K) / D = log n! / D
  - Pentru limba engleză:
    - N = log 26! / 3.2 = 27.6
- Substituţie periodică cu perioada d:
  - Sunt s^d chei posibile pentru fiecare substiţutie simplă:
    - N = H(K) / D = log s^d / D = (d*log s) / D
- Pentru cifrul Vigenere s = 26:
  - N = d * 4.7 / 3.2 = 1.5 d

Cifrarea prin transpoziţie

- Modifică ordinea caracterelor; uzual:
  - Textul clar dispus în liniile succesive ale unei matrice;
  - Parcurgerea acesteia după o anumită regulă pentru stabilirea noii succesiuni de caractere.
- Exemplu:
  - Caracterele dispuse pe linii sunt citite pe coloane;
  - Ordinea coloanelor este dată de ordinea alfabetică a literelor unei chei:
    - Cheie: **POLIGRAF**
    - Ordine: **76543812**
    - text clar: **AFOSTODATACANPOVESTIAFOSTCANICIO**



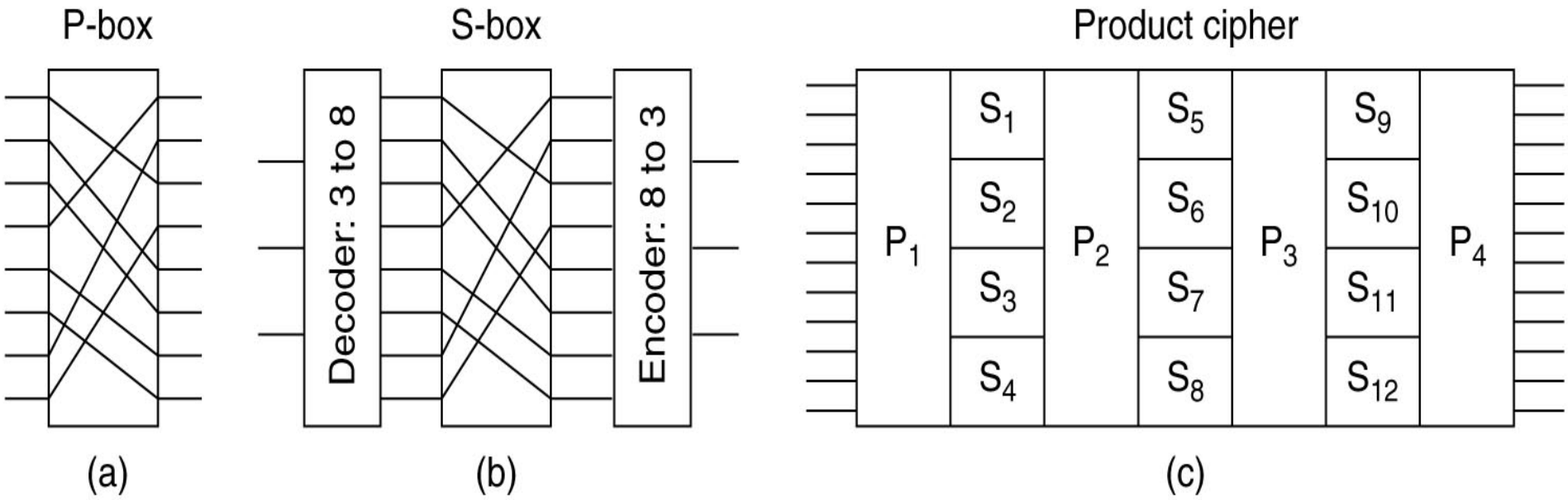
    - text cifrat: **DOOIAVSOTNAISAINOCTAFASCATETOPFC**

Analiza cifrării prin transpoziţie

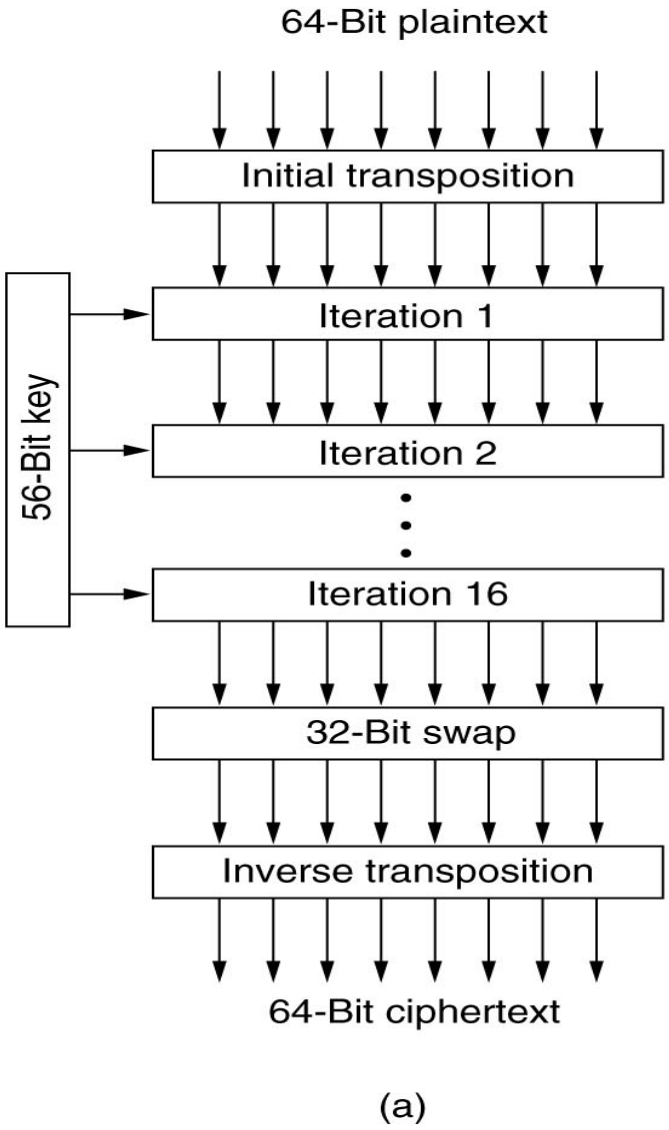
- Pentru spargerea cifrului:
  - Cifrul permută caracterele cu o perioadă fixă d.
  - Sunt d! permutări posibile.
  - Toate sunt echiprobabile.
- H(K) = log d!
  - N = H(K) / D = log d! / D
  - N = d log (d/e) / D
- Pentru d = 2.7 şi D = 3.2 rezultă:
  - N = 27

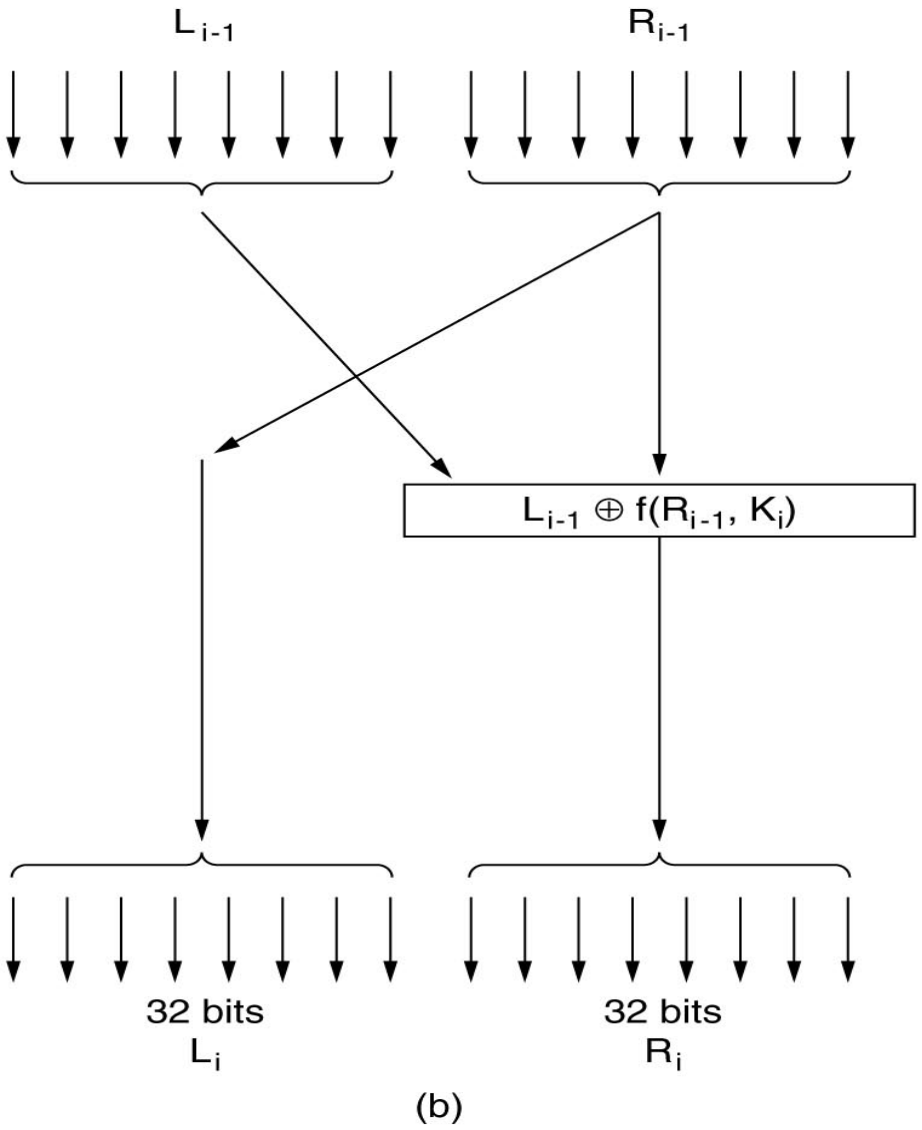
Cifruri produs

- Folosesc succesiuni de permutări şi substituţii:

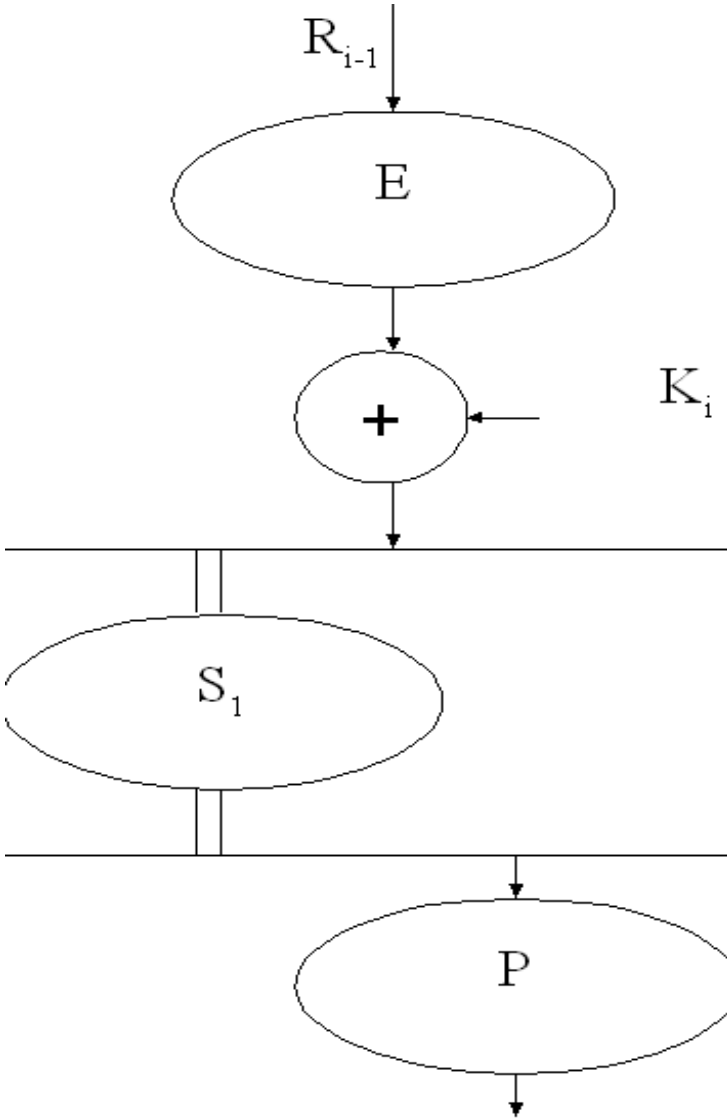


DES (Data Encryption Standard)





Calculul lui f



Expandare la 48 de biți $E(R_{i-1})$

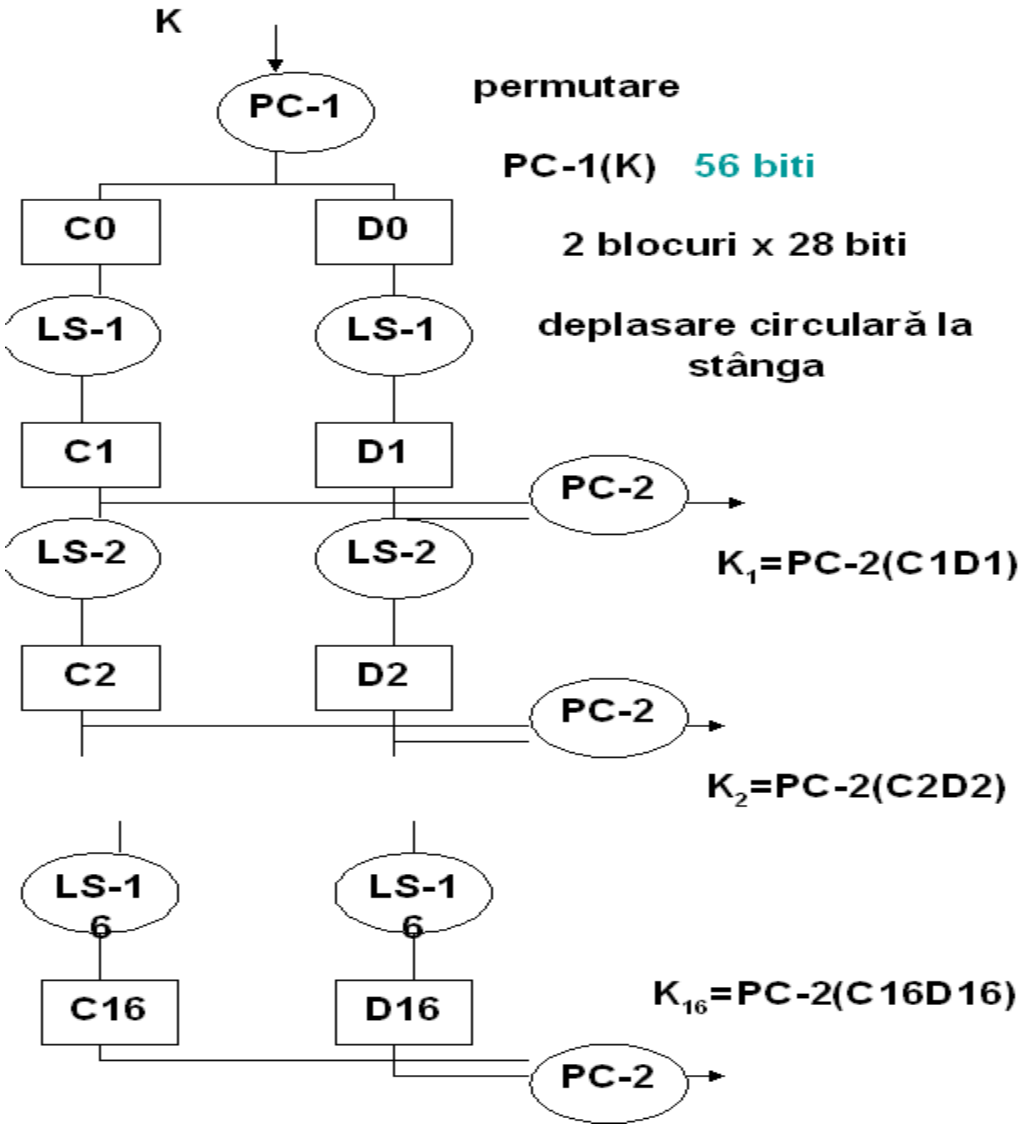
$E(R_{i-1}) + Ki => B_1B_2...B_8$

$S_j(B_j)$

permutare $P(S_1(B_1), \ldots S_8(B_8))$

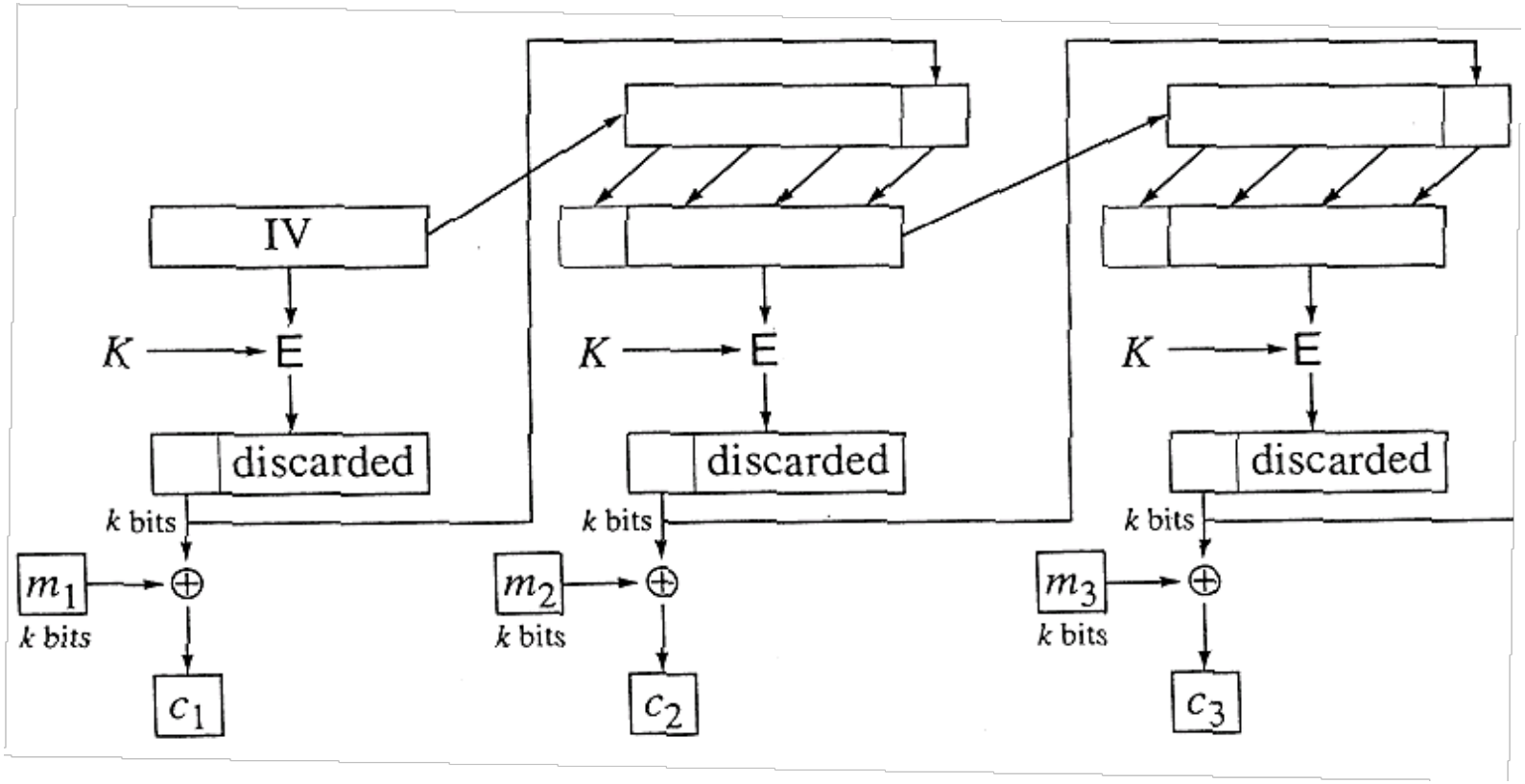
$f(R_{i-1}, K_i)$

Calculul cheii

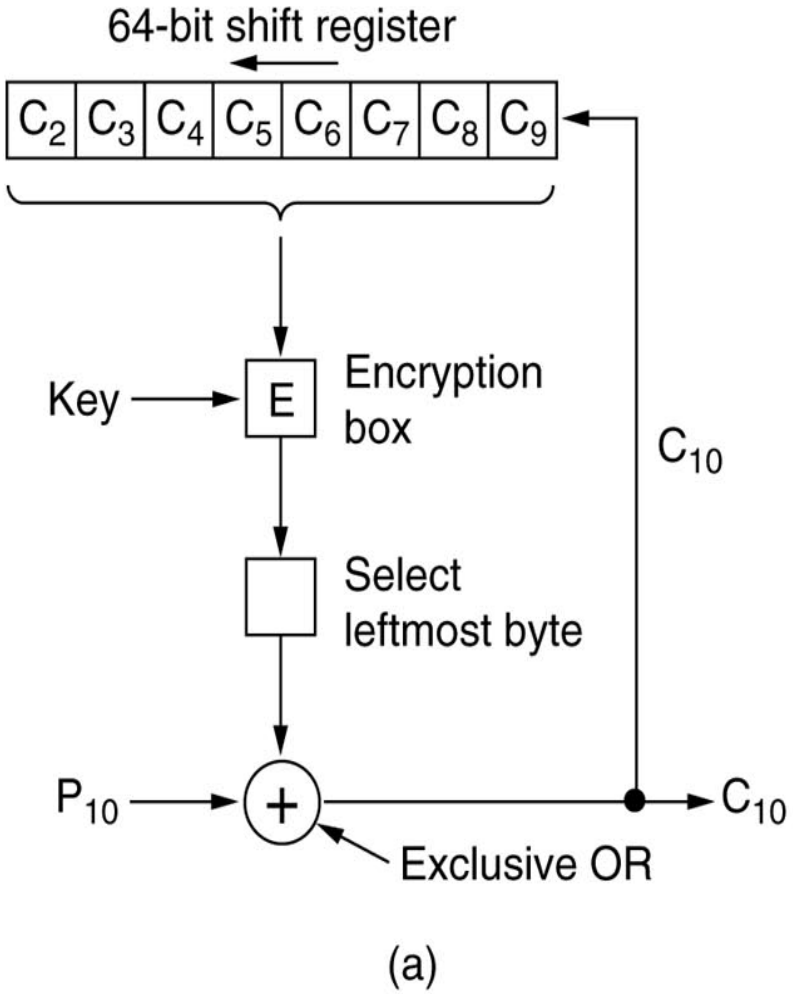


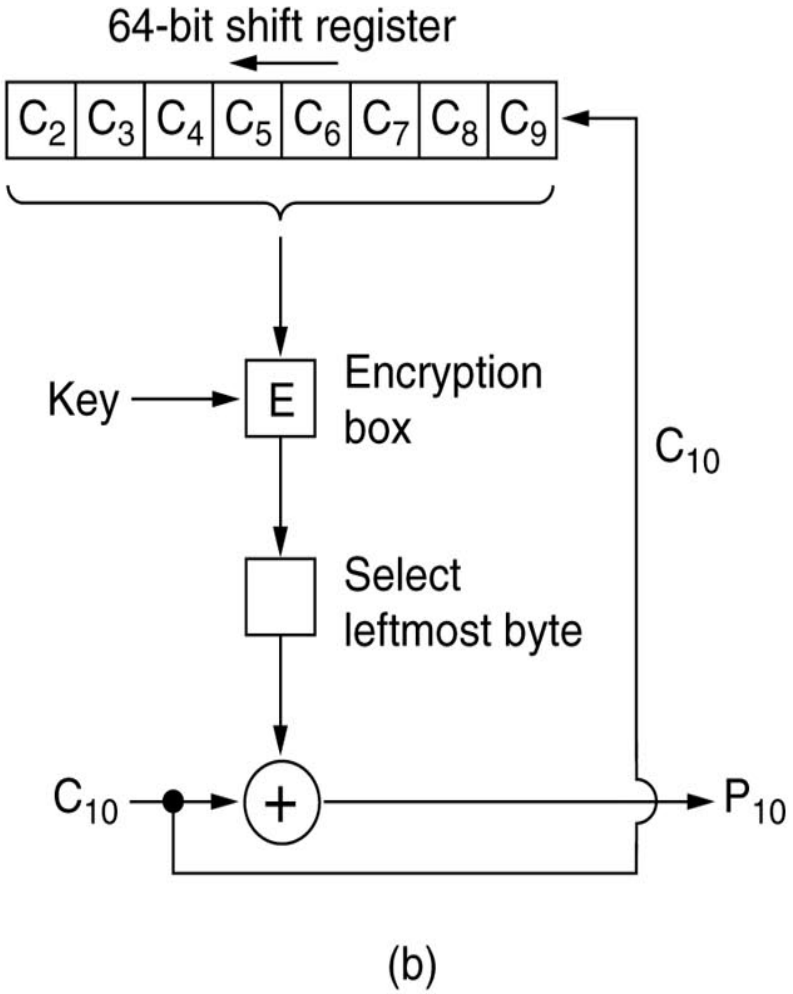
Cifrarea secvenţială

- Sistem secvenţial sincron cu reacţie bloc (OFB - Output Feedback):

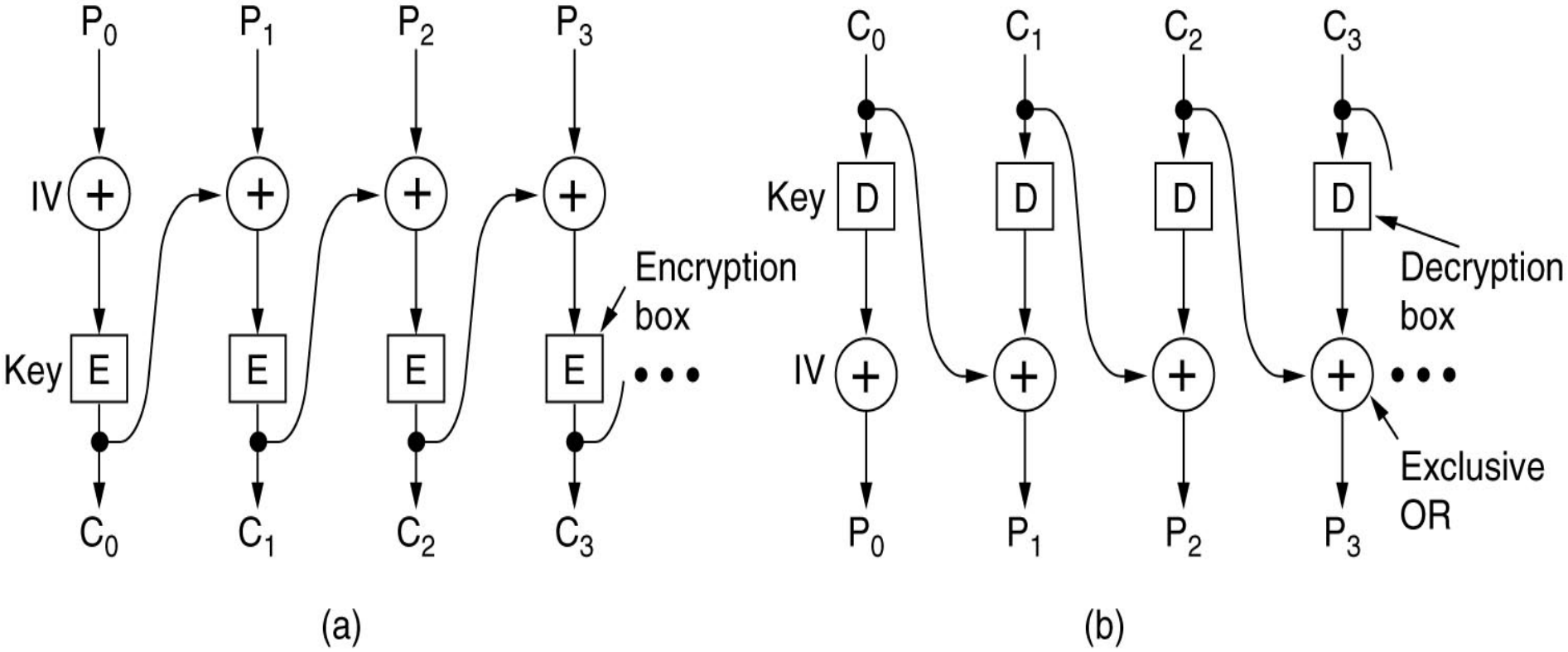


K-bit Cipher Feedback (CFB)

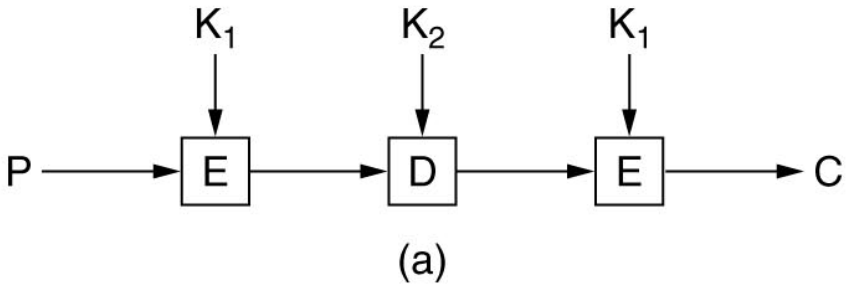




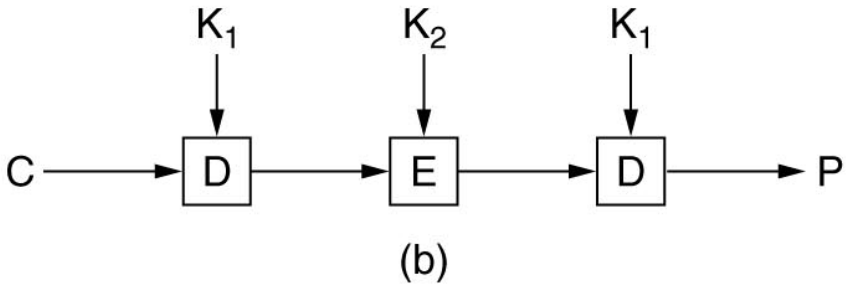
Cipher Block Chaining (CBC)



Triplu DES



(a)



(b)

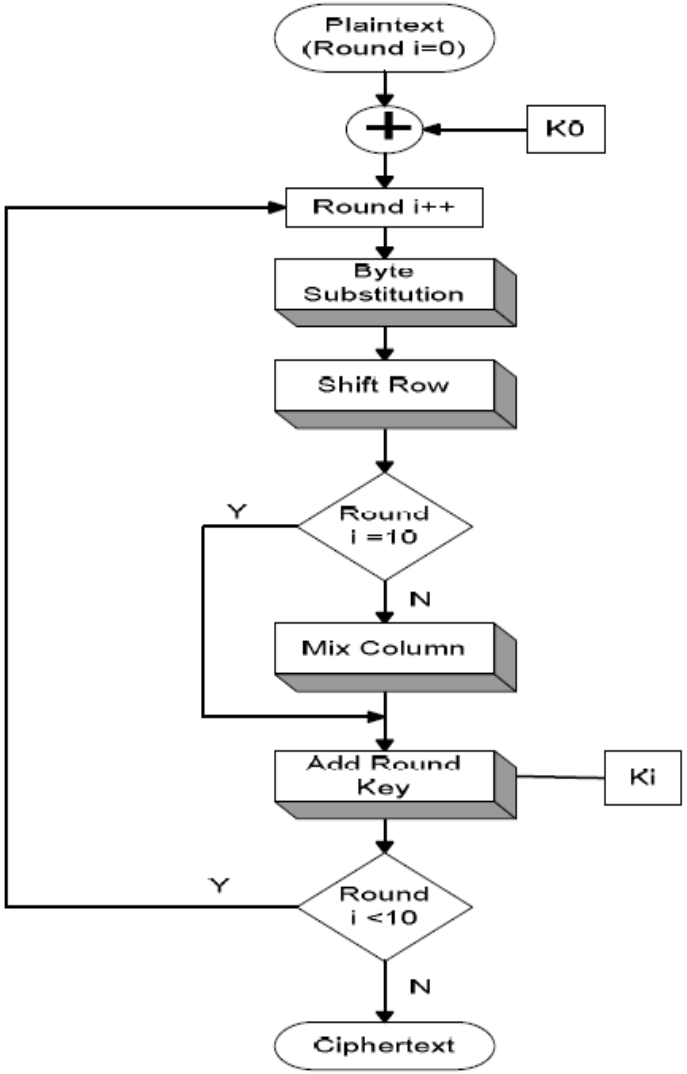
AES

- Regulile concursului organizat de NIST (ianuarie 1997) erau:
  - 1. Algoritmul trebuie să fie un cifru bloc simetric.
  - 2. Tot proiectul trebuie sa fie public.
  - 3. Trebuie să fie suportate chei de 128, 192, şi de 256 biţi.
  - 4. Trebuie să fie posibile implementări hardware şi software.
  - 5. Algoritmul trebuie să fie public sau cu licenţă nediscriminatorie.
- Finaliştii şi scorurile lor au fost următoarele:
  - 1. Rijndael (din partea lui Joan Daemen şi Vincent Rijmen, 86 voturi);
  - 2. Serpent (din partea lui Ross Anderson, Eli Biham şi Lars Knudsen, 59 voturi);
  - 3. Twofish (din partea unei echipe condusă de Bruce Schneier, 31 voturi);
  - 4. RC6 (din partea RSA Laboratories, 23 voturi);
  - 5. MARS (din partea IBM, 13 voturi).
- http://csrc.nist.gov/archive/aes/index.html

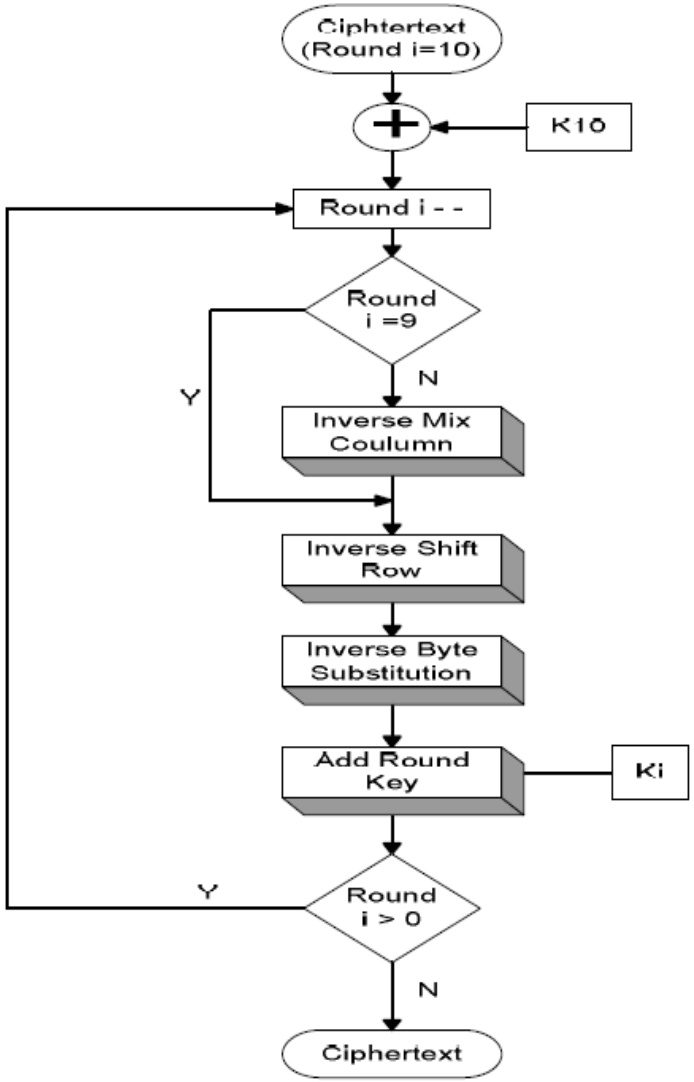
AES

- Rijndael (Galois Fields Theory).
- Dimensiune bloc, Nb: (128), 160, 192, 224, 256.
- Dimensiune cheie, Nk: (128), 160, (192), 224, (256).
- Număr runde, Nr: Nr = 6 + max (Nb,Nk).

AES



(a) Encryption



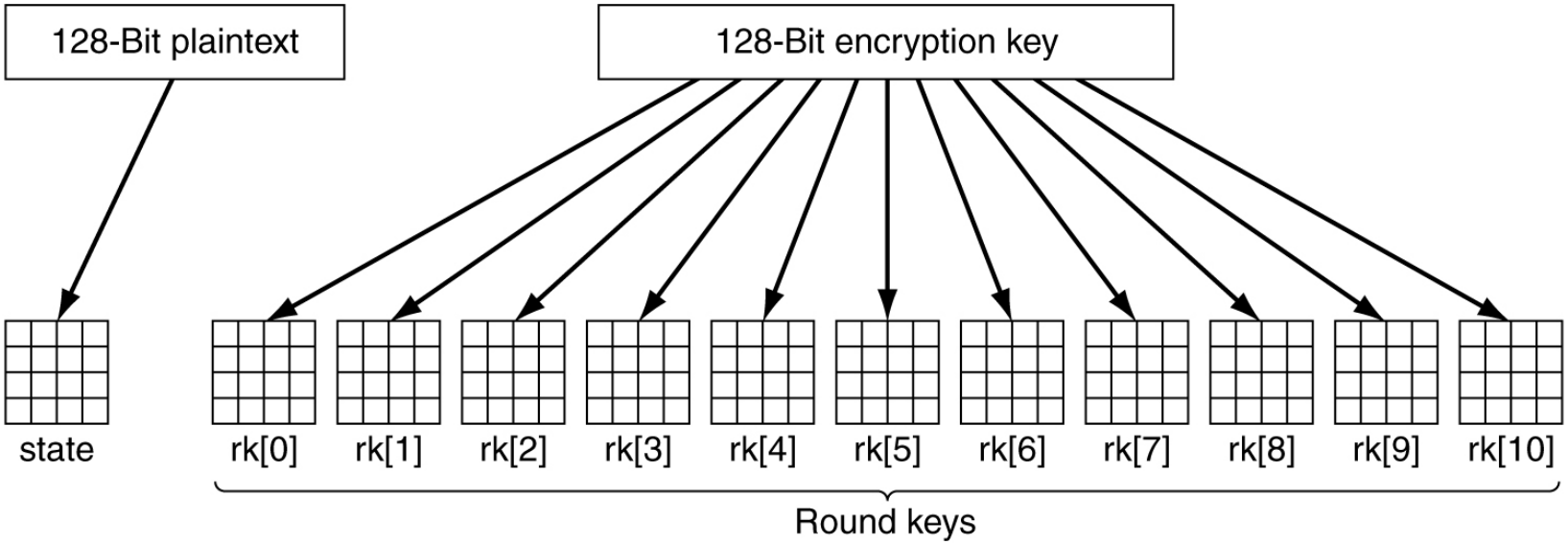
(b)Decryption

AES

```
#define LENGTH 16                               /* # bytes in data block or key */
#define NROWS 4                                 /* number of rows in state */
#define NCOLS 4                                 /* number of columns in state */
#define ROUNDS 10                               /* number of iterations */
typedef unsigned char byte;                     /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
  int r;                                        /* loop index */
  byte state[NROWS][NCOLS];                     /* current state */
  struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];      /* round keys */

  expand_key(key, rk);                          /* construct the round keys */
  copy_plaintext_to_state(state, plaintext);    /* init current state */
  xor_roundkey_into_state(state, rk[0]);        /* XOR key into state */

  for (r = 1; r <= ROUNDS; r++) {
        substitute(state);                      /* apply S-box to each byte */
        rotate_rows(state);                     /* rotate row i by i bytes */
        if (r < ROUNDS) mix_columns(state);     /* mix function */
        xor_roundkey_into_state(state, rk[r]);  /* XOR key into state */
  }
  copy_state_to_ciphertext(ciphertext, state);  /* return result */
}
```

AES

- Crearea matricilor state şi rk:



Cifrarea prin funcţii greu inversabile

- Funcţii greu inversabile:
  - Cunoscând x este uşor de calculat f(x);
  - Calculul lui x din f(x) este foarte dificil.
- Adaptare:
  - Calculul lui x din f(x) trebuie să fie o problemă intratabilă doar pentru criptanalist;
  - Calculul lui x din f(x) trebuie să fie o problemă tratabilă pentru destinatarul autorizat, care dispune de o trapă ce face problema uşor de rezolvat.
- Problemă intratabilă:
  - Nu există un algoritm de rezolvare în timp polinomial.
- Metode:
  - Algoritmi exponenţiali.
  - Problema rucsacului.

Algoritmi exponenţiali

- În metodele:
  - PH (Pohling şi Hellman).
  - RSA (Rivest, Shamir şi Adleman).
- Cifrarea se face prin calculul $C = (M^e) \bmod n$, unde
  - (e, n) reprezintă cheia de cifrare.
  - M este un bloc de mesaj (valoare întreagă între 0 şi n-1).
- Descifrarea se face prin calculul $M = (C^d) \bmod n$, unde
  - (d, n) reprezintă cheia de descifrare.

RSA

- Metoda:
  - 1. Se aleg două numere prime, p şi q, (de obicei de 1024 biţi).
  - 2. Se calculează n = p×q şi z = (p - 1)×(q-1).
  - 3. Se alege un număr relativ prim cu z şi este notat cu d.
  - 4. Se găseşte e astfel încât e×d = 1 mod z.
- Exemplu:
  - Alegem p = 3 şi q = 11, rezultând n = 33 şi z = 20.
  - Alegem d = 7 (7 şi 20 nu au factori comuni).
  - e poate fi găsit din 7e = 1 (mod 20), care dă e = 3.

Plaintext (P)		Ciphertext (C)			After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation — Receiver's computation

Fundamentare

- Funcţia lui Euler:
  - $\Phi(n)$ = numărul de întregi pozitivi <n relativ şi primi cu n.
  - Dacă p prim atunci $\Phi(p) = p-1$.
- Teoremă
  - Pentru n = p*q cu p, q prime avem:
  - $\Phi(n) = \Phi(p) * \Phi(q) = (p-1)\ (q-1)$
- Teorema Fermat
  - Fie p un număr prim. Atunci, pentru orice a cu (a,p)=1 avem:
    - $a^{p-1} \bmod p = 1$.
- Teorema Euler
  - Pentru orice a şi n cu (a,n) = 1 avem:
    - $a^{\Phi(n)} \bmod n = 1$.

Fundamentare

- Teoremă (cifrare):
  - Date fiind e şi d care satisfac ed mod $\Phi(n) = 1$ şi un mesaj $M \in [0, n-1]$ astfel că $(M,n) = 1$, avem $(M^e \bmod n)^d \bmod n = M$.
- Demonstraţie:
  - ed mod $\Phi(n) = 1$ => $ed = t\,\Phi(n)+1$ pentru un anumit t.
  - Calculăm $(M^e \bmod n)^d \bmod n$
    $= M^{ed} \bmod n$
    $= M^{t\,\Phi(n) + 1} \bmod n$
    $= M*M^{t\,\Phi(n)} \bmod n$
    $= M(M^{t\,\Phi(n)} \bmod n) \bmod n$
    $= M((M^{\Phi(n)} \bmod n)^t \bmod n) \bmod n$
    $= M((1)^t \bmod n) \bmod n$
    $= (M*1) \bmod n$
    $= M$
- Prin simetrie, cifrarea şi descrifrarea sunt comutative şi mutual inverse: $(M^e \bmod n)^d \bmod n = M$, prin urmare
  - RSA poate fi utilizat pentru confidenţialitate şi autentificare.

Metoda MH (Merkle şi Hellman)

- Problema rucsacului:
  - Se cere determinarea lui X = (x[1],x[2],...x[n]) cu elemente binare, a.i. C = $\Sigma_{i=1,n}$ x[i] * a[i]
  - O soluţie x poate fi verificată prin cel mult n operaţii de adunare.
  - Găsirea unei solutii implică un număr de operaţii care creşte exponential cu n.
- Trapa:
  - Dacă A satisface proprietatea de dominanţă, adică a[i] > $\Sigma_{j=1,i-1}$ a[j]
  - Atunci problema poate fi rezolvată în timp liniar.

Metoda MH (Merkle şi Hellman)

- Criptare:
  - M, în reprezentare binară este cifrat prin $C = A * M^{(T)}$
    - Unde A este un vector "rucsac greu"
  - Regăsirea lui M din C este o problemă netratabilă pentru un criptanalist.
- Decriptare:
  - Receptorul dispune de o trapă cu care transformă C în C' şi A în A'.
  - Regăsirea lui M din C'şi A' este o problemă “rucsac simplă”.

Sumar

- Rolul nivelului prezentare:
  - Reprezentarea datelor;
  - Compresia datelor (pentru reducerea volumului);
  - Criptarea datelor (pentru protecţia lor).
- Probleme de securitate
  - Confidenţialitatea;
  - Autentificarea;
  - Controlul accesului;
  - Integritatea;
  - Non-repudierea.
- Modelul de bază al criptării.
- Problema criptanalistului.
- Caracteristicile sistemelor secrete.
- Cifrarea prin substituţie.
- Cifrarea prin transpoziţie.
- Cifruri produs:
  - DES (Data Encryption Standard).
  - Triplu DES.
  - Advanced Encryption Standard (AES).
- Cifrarea prin funcţii greu inversabile:
  - RSA;
  - MH.