



# Protocoloale de securitate

## Criptarea datelor



# 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 protecția lor).



# Securitate: persoane ce generează probleme

<b>Adversar</b>	<b>Scop</b>
<b>Student</b>	<b>Pentru a se distra furând poșta electronică a celorlalți.</b>
<b>Spărgător</b>	<b>Pentru a testa securitatea sistemului cuiva; pentru a fura date.</b>
<b>Responsabil de vânzări</b>	<b>Pentru a pretinde că reprezintă toată Europa, nu numai Andorra.</b>
<b>Om de afaceri</b>	<b>Pentru a descoperi planul strategic de marketing al competitorului.</b>
<b>Fost funcționar</b>	<b>Pentru a se răzbuna că a fost concediat.</b>
<b>Contabil</b>	<b>Pentru a sustrage bani de la o companie.</b>
<b>Agent de vânzări</b>	<b>Pentru a nega o promisiune făcută clientului prin poștă electronică.</b>
<b>Șarlatan</b>	<b>Pentru a fura numere de cărți de credit și a le vinde.</b>
<b>Spion</b>	<b>Pentru a afla puterea militară a inamicului sau secrete industriale.</b>
<b>Terorist</b>	<b>Pentru a fura secrete legate de conflicte armate.</b>



# Probleme de securitate

- Confidențialitatea:
  - Păstrarea informației din mesaj departe de utilizatorii neautorizați.
- Autentificarea:
  - Determinarea identității persoanei cu care se schimbă mesaje înainte de a dezvălui informații importante.
- Controlul accesului:
  - Protecția împotriva accesului neautorizat.
- Integritatea:
  - Mesajul primit nu a fost modificat sau măsluit în tranzit.
- Non-repudierea:
  - Transmițătorul 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 și hash.
- Securitatea în ierarhia de protocoale:
  - **Fizic**: tuburi de securizare a liniilor de transmisie.
  - **Legătură de date**: legături criptate.
  - **Rețea**: ziduri de protecție (firewalls), IPsec.
  - **Transport**: end-to-end security.
  - **Aplicație**: autentificarea, non-repudierea.

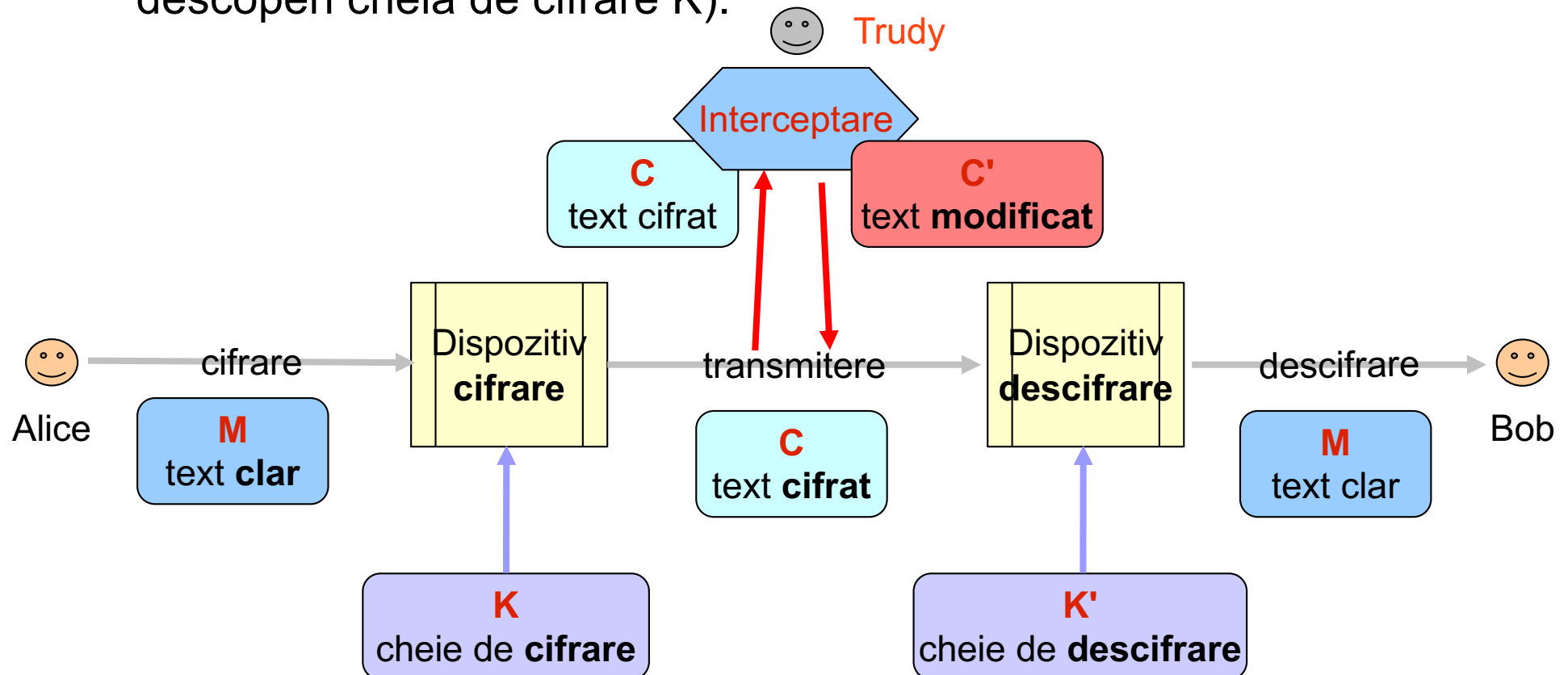


## Alte aspecte

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

# Modelul de bază al criptării

- Transmiterea unui mesaj de la Alice la Bob:
  - Confidențialitatea**: 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  $K$ ).





# Definiții

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

$F : \{M\} \times \{K\} \rightarrow \{C\}$ , unde:

- $\{M\}$  este mulțimea mesajelor;
- $\{K\}$  este mulțimea cheilor;
- $\{C\}$  este mulțimea criptogramelor.
- Operații:
  - Cifrarea:  $C = E_k(M)$ .
  - Descifrarea:  $M = D_{k'}(C)$ .
- Conotație 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 porțiuni de text;
  - Exemplu pentru o bază de date - modificare / efect.



# Caracteristicile sistemelor secrete

- Sistem **necondiționat sigur**:
  - Rezistă la orice atac, indiferent de cantitatea de text cifrat interceptat (exemplu: one time pad).
- **Computațional 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 probabilități apropiate.



# Cerințe criptosisteme cu chei secrete

- Cerințe generale:
  - Cifrare și descifrare eficiente pentru toate cheile.
  - Sistem ușor de folosit (chei de transformare).
  - Securitatea să depindă de chei, nu de algoritm.
- Cerințe specifice pentru **confidențialitate**: să fie imposibil computațional ca un criptanalist să determine sistematic:
  - Transformarea  $D_k$  din  $C$ , chiar dacă ar cunoaște  $M$ .
  - $M$  din  $C$  (fără a cunoaște  $D_k$ ).
- Cerințe specifice pentru **autentificare**: să fie imposibil computațional ca un criptanalist să determine sistematic:
  - Transformarea  $E_k$ , din  $C$ , chiar dacă ar cunoaște  $M$ .
  - Cifrul  $C'$  astfel ca  $D_k(C')$  să fie un mesaj valid (fără a cunoaște  $E_k$ ).

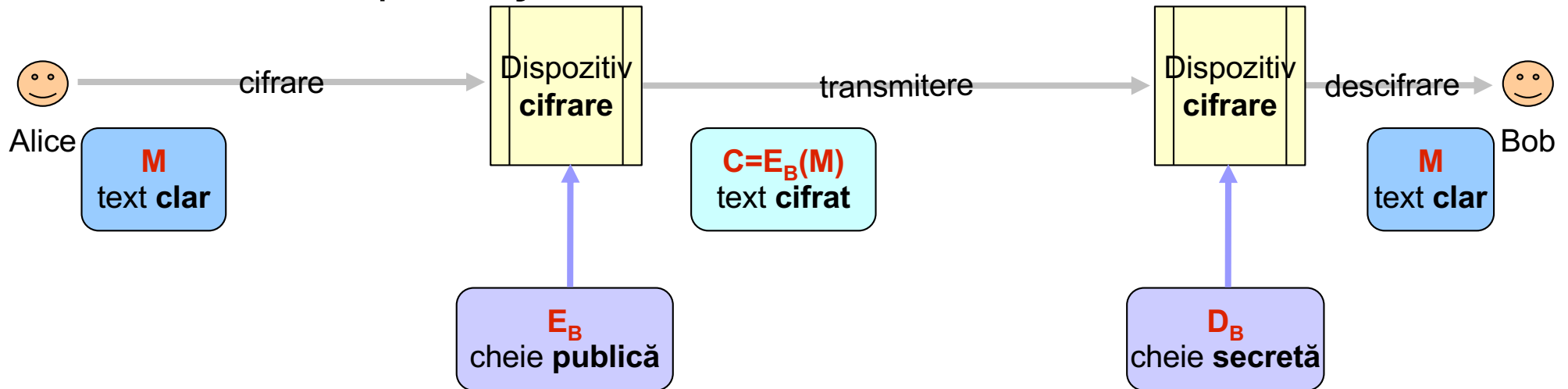


# Modelul criptografic cu chei publice

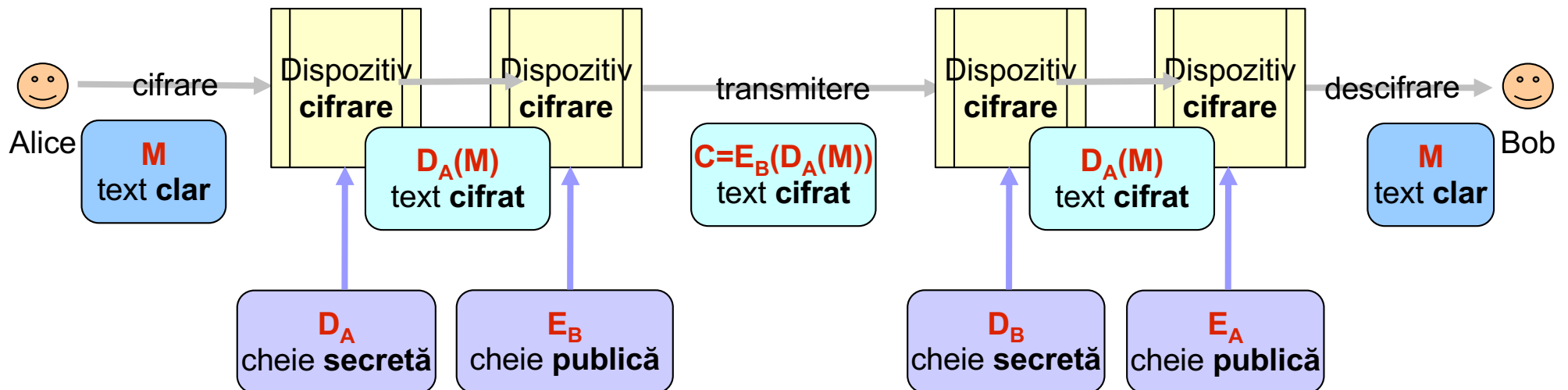
- Sistemele criptografice:
  - Simetrice.
  - Asimetrice:
    - Propuse de Diffie și Hellman în 1976.
    - Chei diferite de cifrare  $E$  și descifrare  $D$ .
    - Nu se pot deduce (ușor) una din alta, mai precis:
      - $D(E(M)) = M$ ;
      - Este extrem de greu să se deducă  $D$  din  $E$ ;
      - $E$  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:
  - Condiția necesară este ca transformările  $E_a$  și  $D_a$  să comute, adică  $E_a(D_a(M)) = D_a(E_a(M)) = M$ .

# Scheme folosite

- Schema de protecție:



- Schema de autentificare:

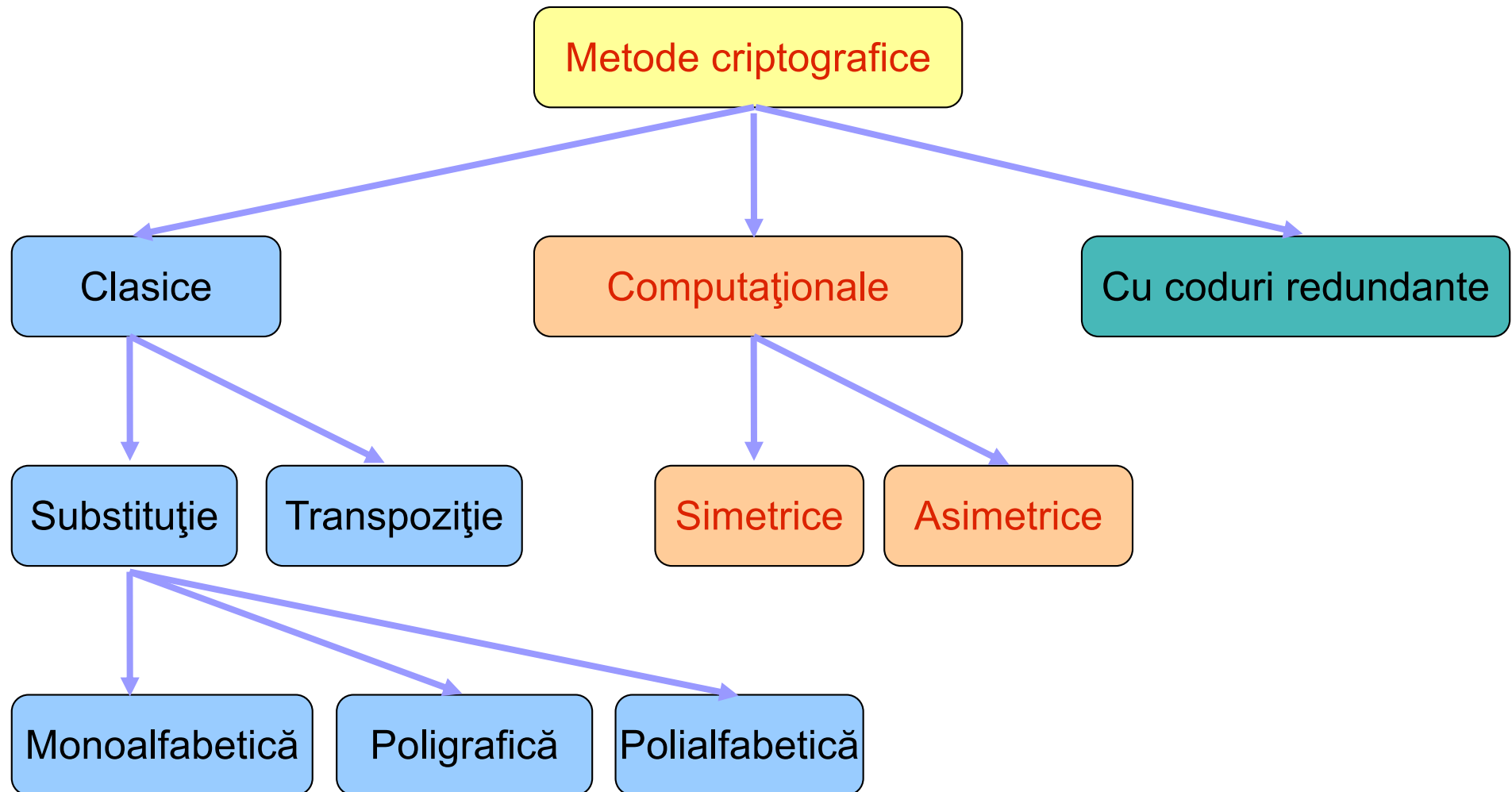




# Scheme folosite

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

# Clasificare generală





# Utilizarea TI în evaluarea algoritmilor criptografici

- **Analogia** între transmisie și confidențialitate:
  - perturbarea  $\Leftrightarrow$  cifrarea mesajului;
  - mesajul recepționat  $\Leftrightarrow$  text cifrat.
- Cantitatea de informație = **entropie**.
  - $X_1, \dots, X_n$ : mesajele unei surse;
  - $p(X_1), \dots, p(X_n)$ : probabilitățile ( $\sum_{i=1,n} p(X_i) = 1$ ).
- **Entropia** unui mesaj:  $H(X) = -\sum_{i=1,n} p(X_i) \log p(X_i)$ 
  - Intuitiv:  $\log(1/p(X))$  = numărul de biți folosiți pentru codificarea optimă a lui X.
- Entropia măsoară și **incertitudinea**:
  - $H(X)$  maxim când  $p(X_1) = p(X_2) = \dots = p(X_n) = 1/n$ .
  - $H(X)$  descrește când distribuția mesajelor se restrânge.
  - $H(X) = 0$  când  $p(X_i) = 1$  pentru un mesaj i.





# Echivocitatea

- Dat fiind  $Y$  din mulțimea mesajelor  $Y_1, \dots, Y_n$  cu  $\sum_{i=1,n} p(Y_i) = 1$ , fie:
  - $p_Y(X)$  probabilitatea mesajului  $X$  condiționat de  $Y$ .
  - $p(X, Y)$  probabilitatea mesajelor  $X$  și  $Y$  luate împreună:
    - $p(X, Y) = p_Y(X) \cdot p(Y)$ .
- **Echivocitatea** este entropia lui  $X$  condiționat de  $Y$ :
  - $H_Y(X) = -\sum_{X,Y} p(X, Y) \cdot \log p_Y(X)$
  - $H_Y(X) = \sum_{X,Y} p_Y(X) \cdot p(Y) \log (1/p_Y(X)) = \sum_Y p(Y) \sum_X p_Y(X) \cdot \log (1/p_Y(X))$ .
- Exemplu:
  - $n = 4$  și  $p(X) = 1/4$  pentru fiecare  $X \Rightarrow H(X) = \log 4 = 2$ .
  - Fie  $m=4$  și  $p(Y) = 1/4$  pentru fiecare  $Y$ .
  - Presupunem că fiecare  $Y$  restrânge  $X$ :
    - $Y1: X1 \text{ sau } X2, Y2: X3 \text{ sau } X4, Y3: X2 \text{ sau } X3, Y4: X4 \text{ sau } X1$ .
  - Echivocitatea este:  $H_Y(X) = 4 \left( \left( \frac{1}{4} \right)^2 \left( \frac{1}{2} \right) \log 2 \right) = \log 2 = 1$ .
    - Adică: **cunoașterea lui  $Y$  reduce incertitudinea lui  $X$  la un bit.**

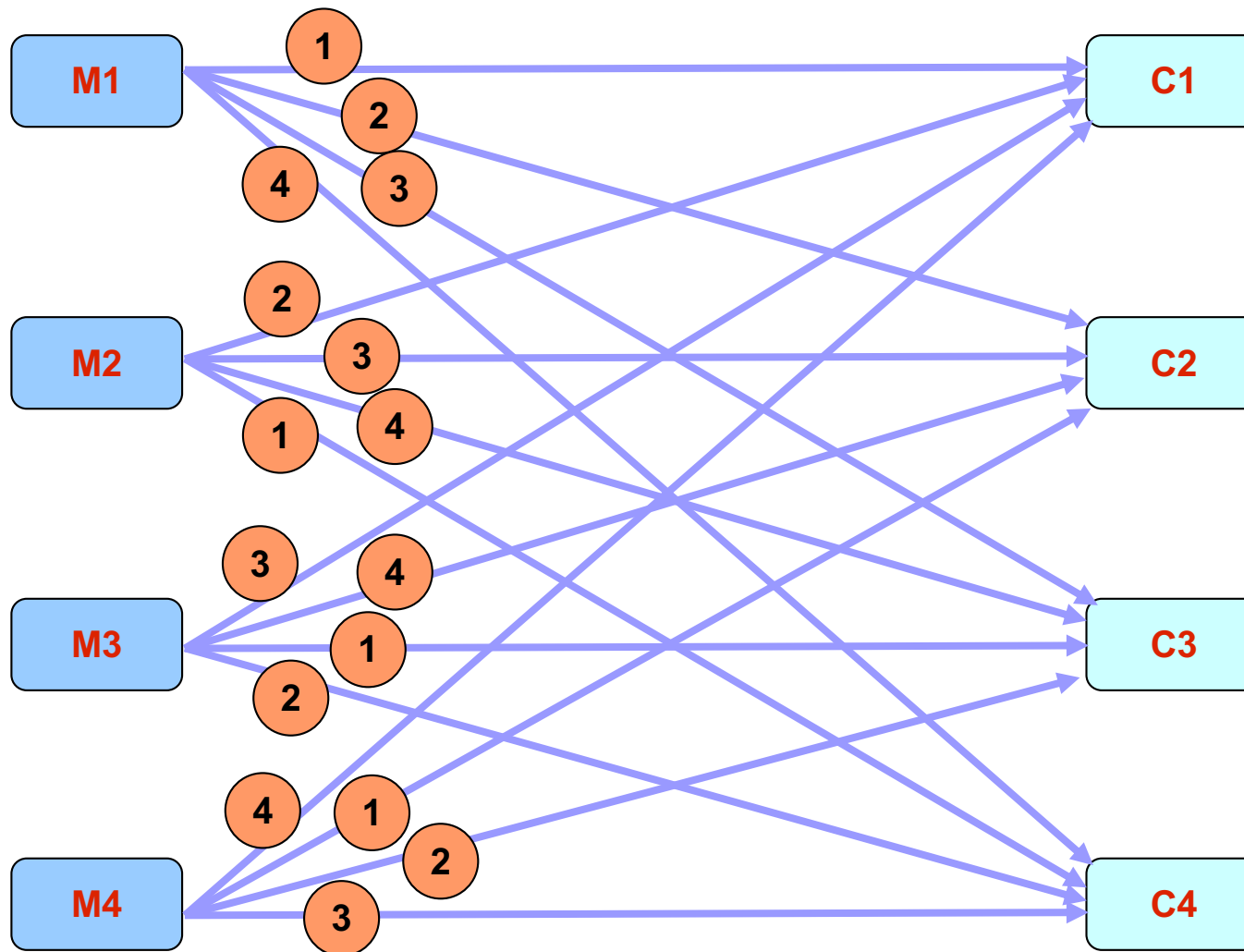


# Confidențialitatea perfectă

- Fie:
  - M texte clare cu probabilitatea  $p(M)$ ,  $\sum_M p(M) = 1$ .
  - C criptograme, cu probabilitatea  $p(C)$ ,  $\sum_C p(C) = 1$ .
  - K chei cu probabilitatea  $p(K)$ ,  $\sum_K p(K) = 1$ .
  - $p_C(M)$  probabilitatea să se fi transmis M când se recepționează C.
- **Confidențialitatea perfectă**  $\Leftrightarrow p_C(M) = p(M)$ .
- $p_M(C)$  probabilitatea să se recepționeze C când s-a transmis M:
  - $p_M(C) = \sum_{k, E_k(M)=C} p(k)$ .
- Confidențialitatea perfectă:
  - $p_M(C) = p(C)$ , pentru toate M și orice C.

# Confidențialitatea perfectă

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





# Distanța de unicitate

- **Confidențialitatea:**
  - Cantitatea de incertitudine în  $K$ , dat fiind  $C$ , adică:
    - $H_C(K) = \sum_C p(C) \sum_K p_C(K) \log (1/p_C (K))$
- Dacă  $H_C(K)=0$  nu există incertitudine și cifrul se poate sparge.
- Când crește lungimea  $N$  a textelor cifrate echivocitatea scade.
- **Distanța 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 distanță unicitate

- Notății:
  - Pentru un limbaj, luăm mulțimea mesajelor de lungime N.
  - **Rata limbajului:**
    - $r = H(X) / N$
    - $r = 1 \dots 1.5$  pentru limba engleză.
  - **Rata absolută a limbajului** (pentru L simboluri):
    - $R = \log L = -\sum_{i=1,L} (1/L) \log (1/L)$
    - $R = \log 26 = 4.7$  biți pe literă pentru limba engleză.
  - **Redundanțele** apar din structura limbajului: distribuția frecvențelor literelor, digramelor, trigramelor, etc.):
    - $D = R - r$
    - $D = 3.2 \dots 3.7$  în limba engleză.



# Calcul aproximativ distanță unicitate

- Ipoteze:
  - Sunt  $2^{rN}$  mesaje posibile de lungime  $N$ , din care  $2^{rN}$  au sens.
  - Toate mesajele cu sens au aceeași probabilitate,  $1/2^{rN}$ .
  - Toate mesajele fără sens au probabilitate 0.
  - Sunt  $2^{H(K)}$  chei cu probabilități egale.
  - Cifrul este aleator:
    - Pentru fiecare  $k$  și  $C$ , 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 aceeași probabilitate  $q$  de a da soluție falsă:
    - Același  $C$  se obține criptând un alt mesaj  $M'$  cu înțeles, cu altă cheie  $K'$ 
      - $q = 2^{rN} / 2^{RN} = 2^{-DN}$  ( $D = R - r$  este redundanța limbajului).
    - Numărul de soluții 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 substituție

- **Cifrul lui Cezar** (substituție **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

– Relația de calcul:  $c[i] = (m[i] + 3) \bmod 26$ .

- Textul clar: **CRIPTOGRAFIE**
- Textul cifrat: **FULSWRJUDILH**

– În general:  $c[i] = (a * m[i] + b) \bmod n$ .

- Substituția **polialfabetică** (Vigenere):

– Folosește 36 de cifruri Cezar și o cheie de cifrare de lungime l.

– Exemplu: cheia POLIGRAF:

**POLIGRAF**POLIGRAG**POLIGRAF**POLIGRAF**POLI**  
**AFOSTODATACANPOVESTIAFOSTCANICIODATA**  
**PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI**



# Cifrarea prin substituție

- Cifrul **Beaufort**:
  - Cifrare:  $c[i] = (k[i] - m[i]) \bmod n$ .
  - Descifrare:  $m[i] = (k[i] - c[i]) \bmod n$
- Substituția **poligrafică**:
  - Un grup de  $n$  litere este înlocuit cu un alt grup de  $n$  litere.





# Analiza cifrării prin substituție

- Substituție **monoalfabetică**:
  - $N = H(K) / D = \log n! / D$
  - Pentru limba engleză:
    - $N = \log 26! / 3.2 = 27.6$
- Substituție **periodică** cu perioada  $d$ :
  - Sunt  $s^d$  chei posibile pentru fiecare substituție 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 transpoziție

- 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**

<b>POLIGRAF</b>							
A	F	O	S	T	O	D	A
T	A	C	A	N	P	O	V
E	S	T	I	A	F	O	S
T	C	A	N	I	C	I	O

- text cifrat: **DOOIAVSOTNAISAINOCTAFASCATETOPFC**

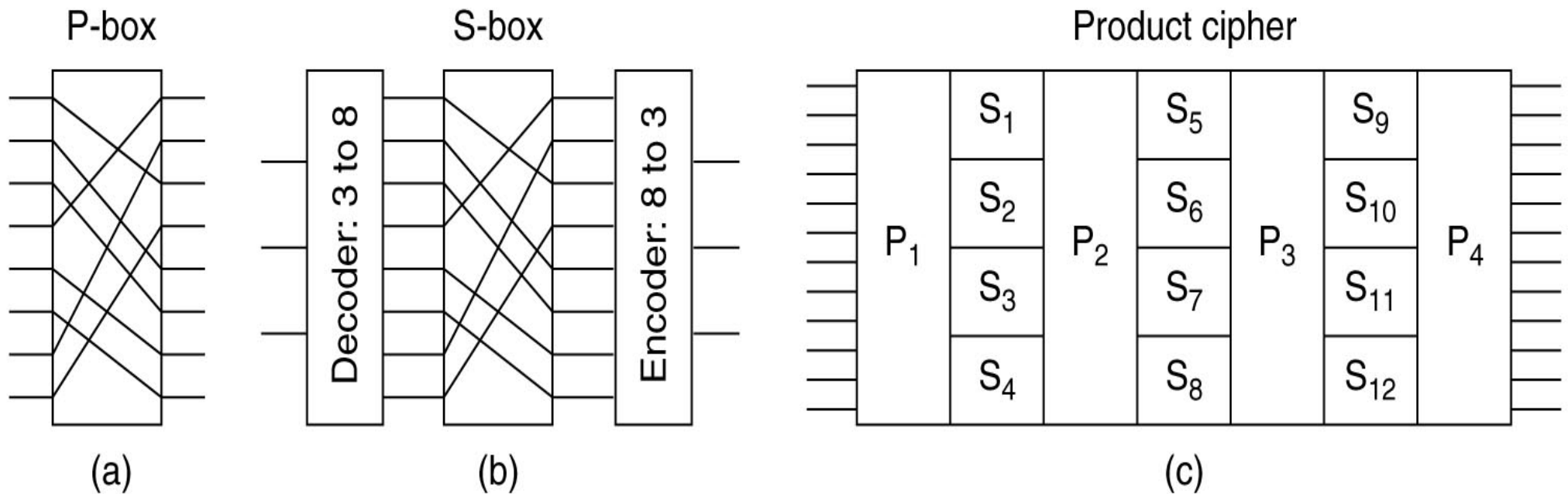


# Analiza cifrării prin transpoziție

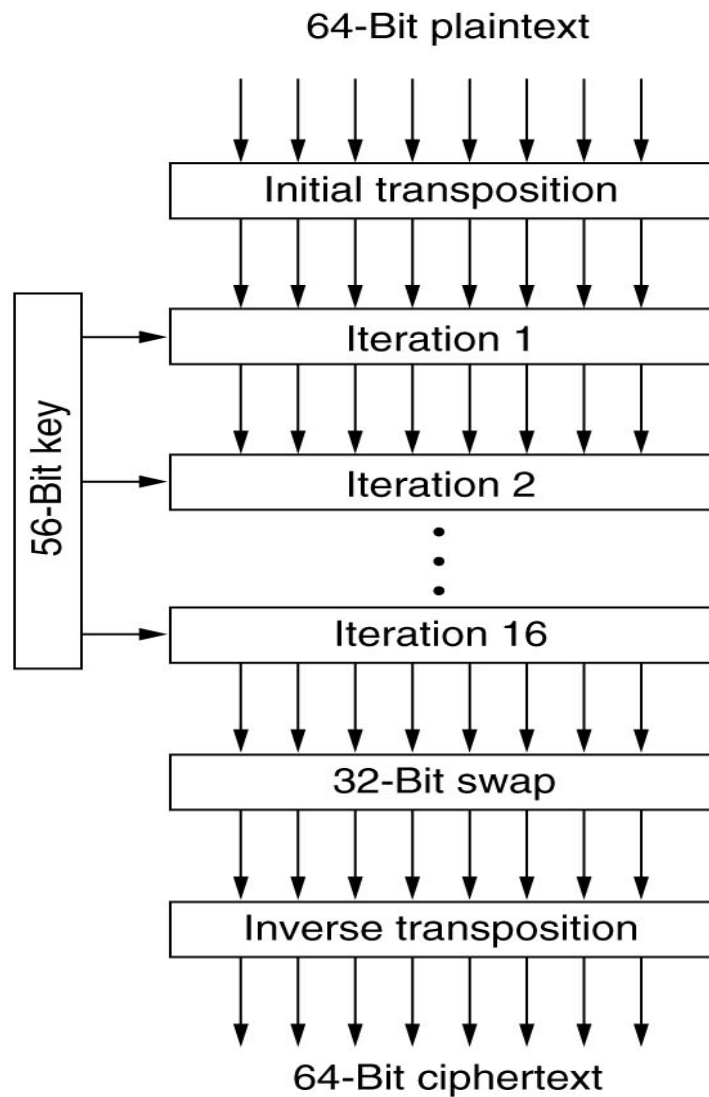
- Pentru spargerea cifrului:
  - Cifrul permută caracterele cu o perioadă fixă  $d$ .
  - Sunt  $d!$  permutări posibile.
  - Toate sunt echiprobabile.
- $H(K) = \log d!$ 
  - $N = H(K) / D = \log d! / D$
  - $N = d \log (d/e) / D$
- Pentru  $d = 2.7$  și  $D = 3.2$  rezultă:
  - $N = 27$

# Cifruri produs

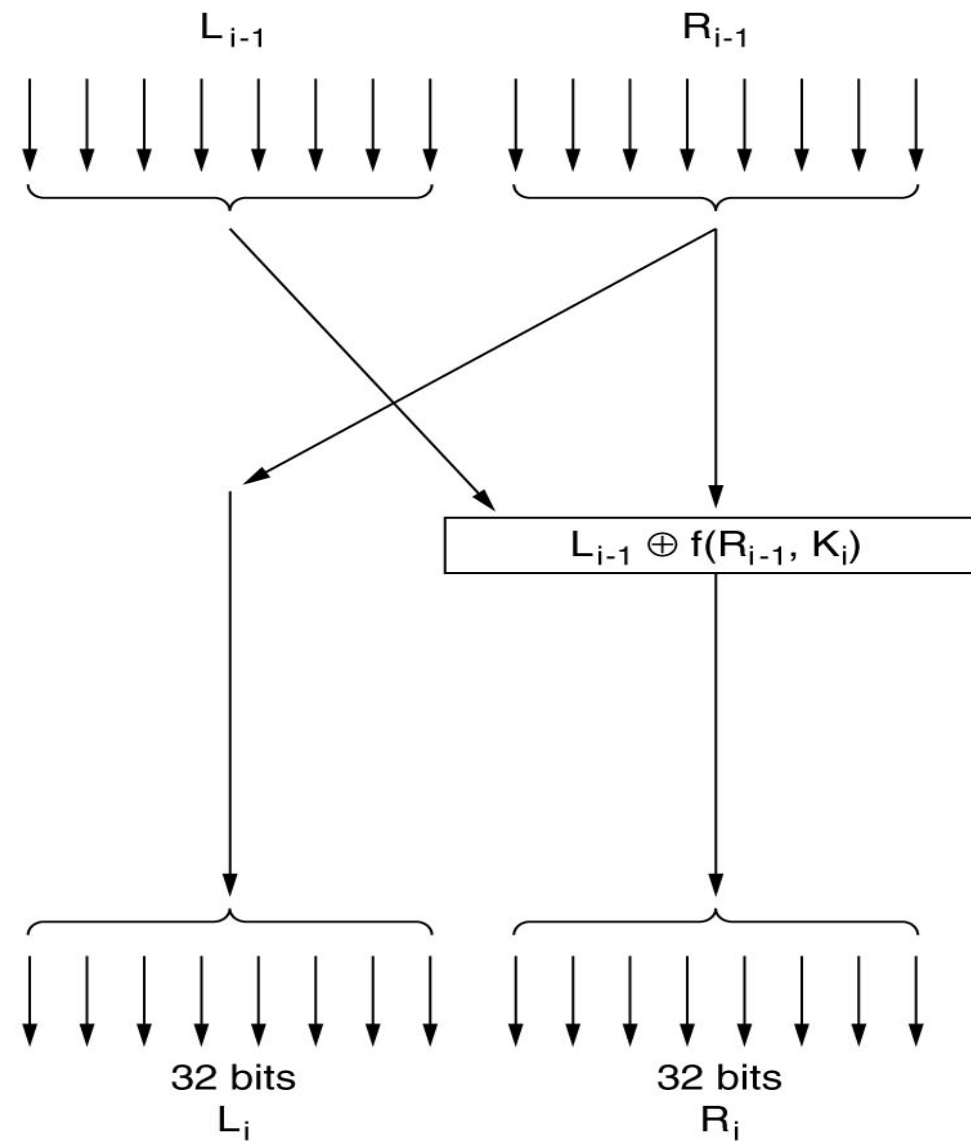
- Folosesc succesiuni de permutări și substituții:



# DES (Data Encryption Standard)

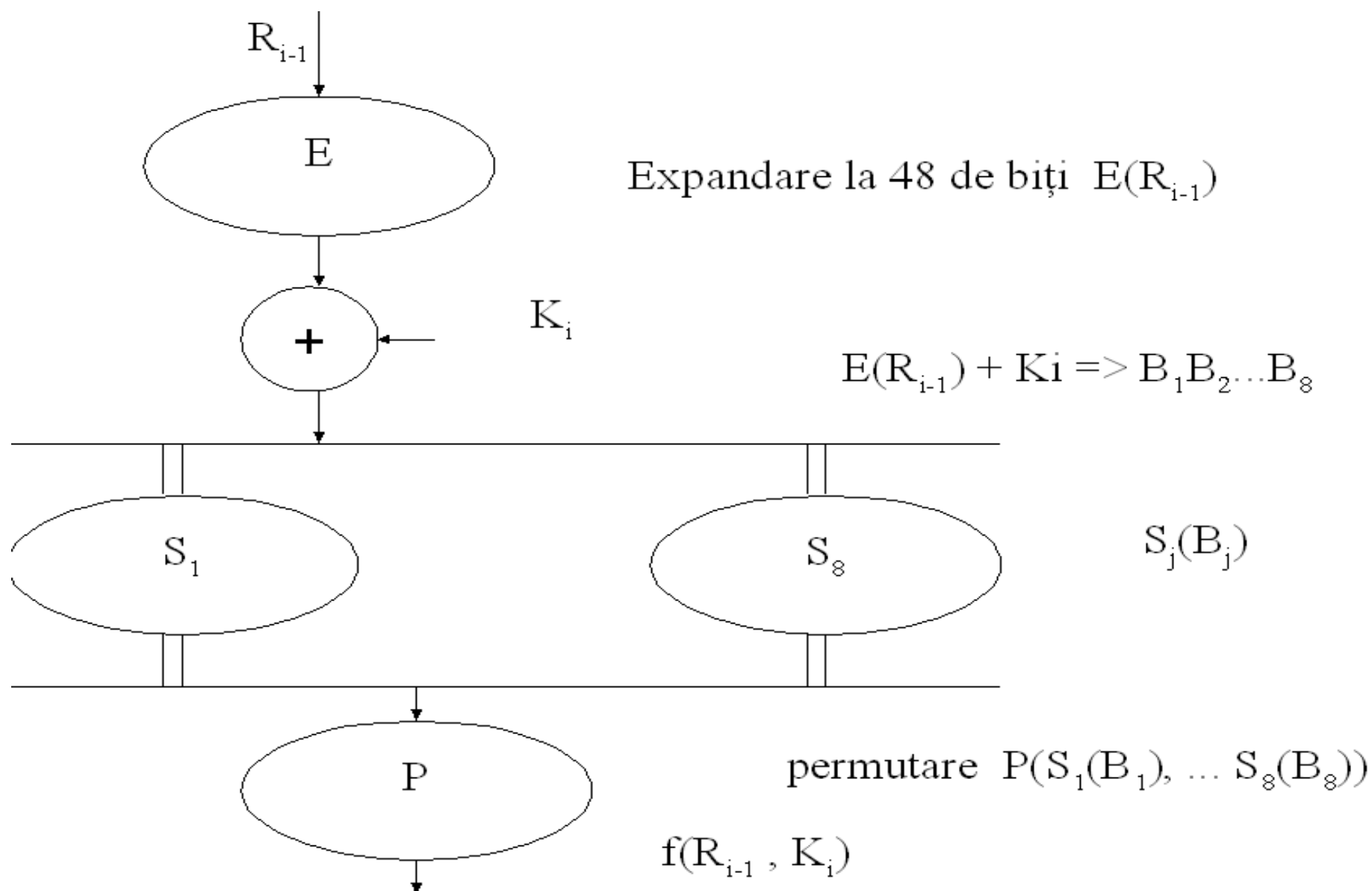


(a)

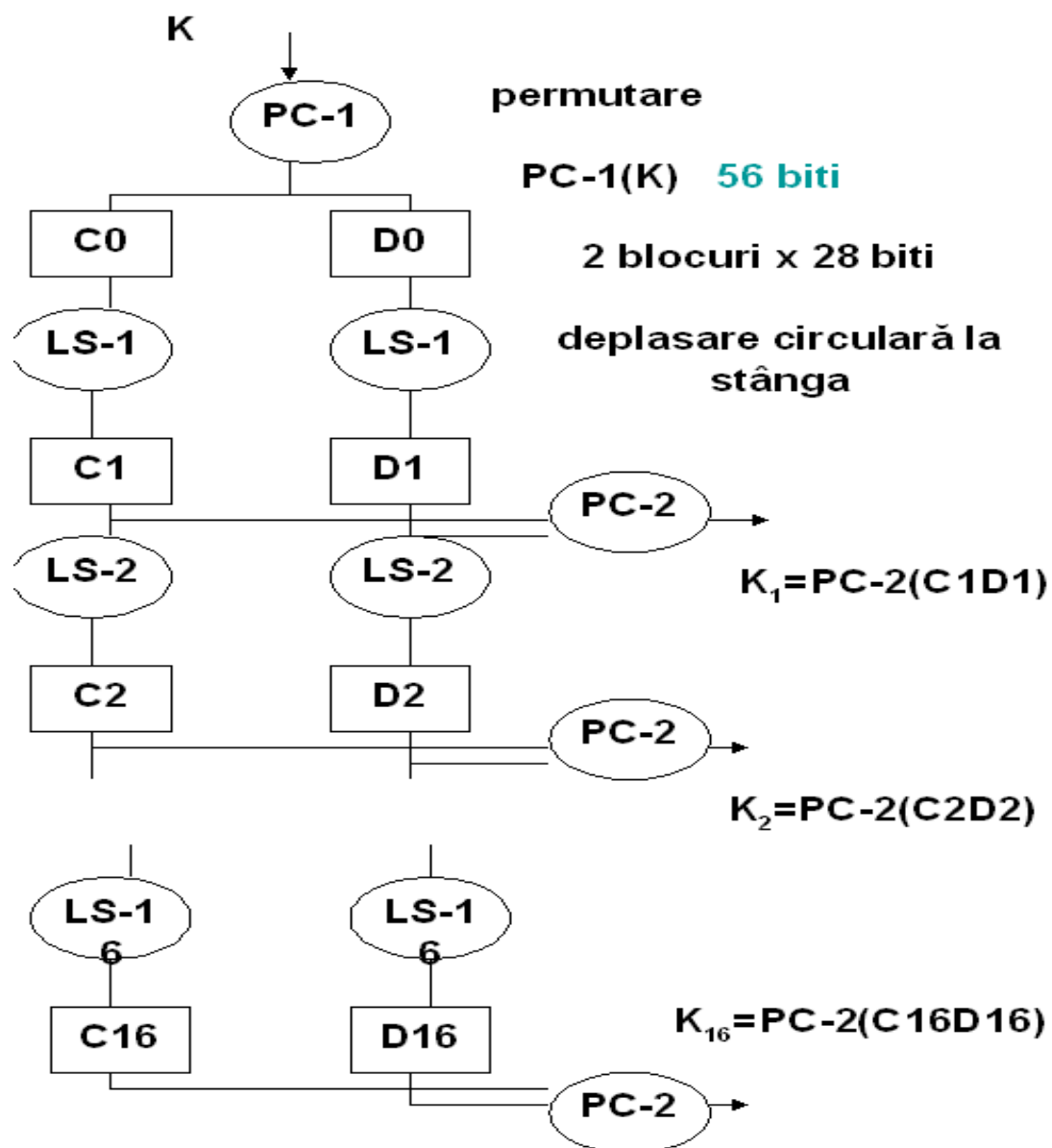


(b)

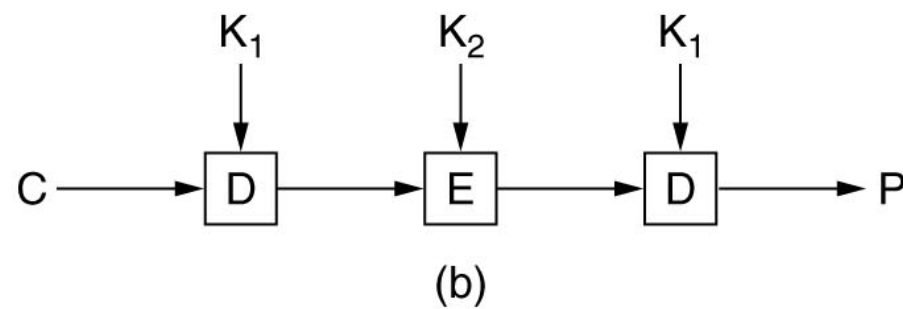
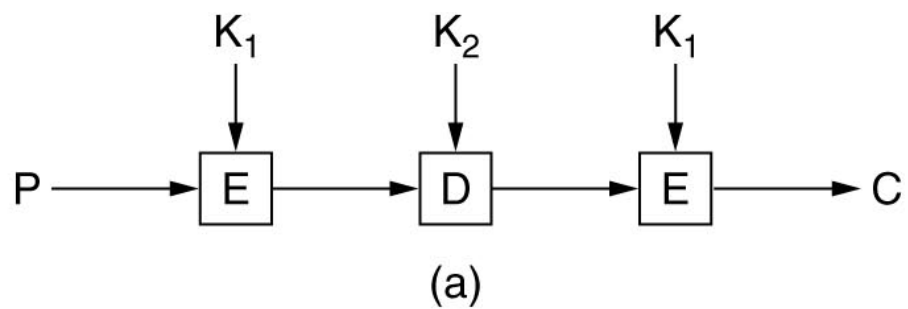
# Calculul lui $f$



# Calculul cheii



# Triplu DES





# AES

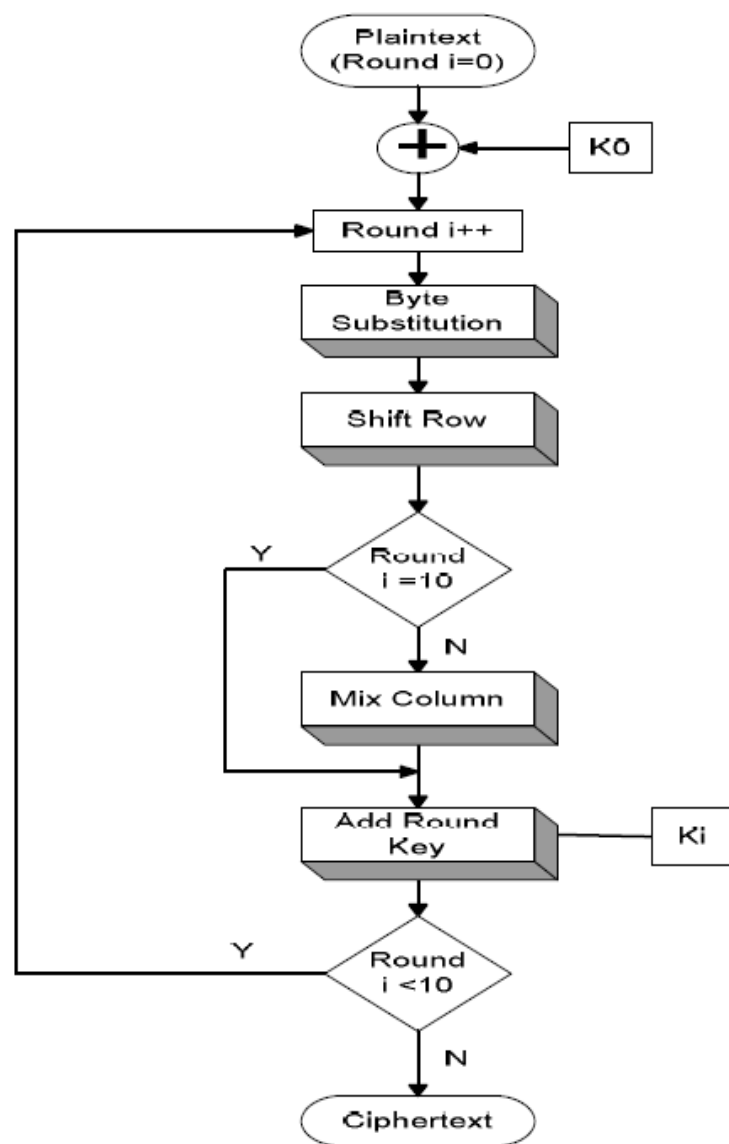
- Regulile concursului organizat de NIST (ianuarie 1997) erau:
  - 1. Algoritmul trebuie să fie un cifru bloc simetric.
  - 2. Tot proiectul trebuie să fie public.
  - 3. Trebuie să fie suportate chei de 128, 192, și de 256 biți.
  - 4. Trebuie să fie posibile implementări hardware și software.
  - 5. Algoritmul trebuie să fie public sau cu licență nediscriminatorie.
- Finaliștii și scorurile lor au fost următoarele:
  - 1. Rijndael (din partea lui Joan Daemen și Vincent Rijmen, 86 voturi);
  - 2. Serpent (din partea lui Ross Anderson, Eli Biham și 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>



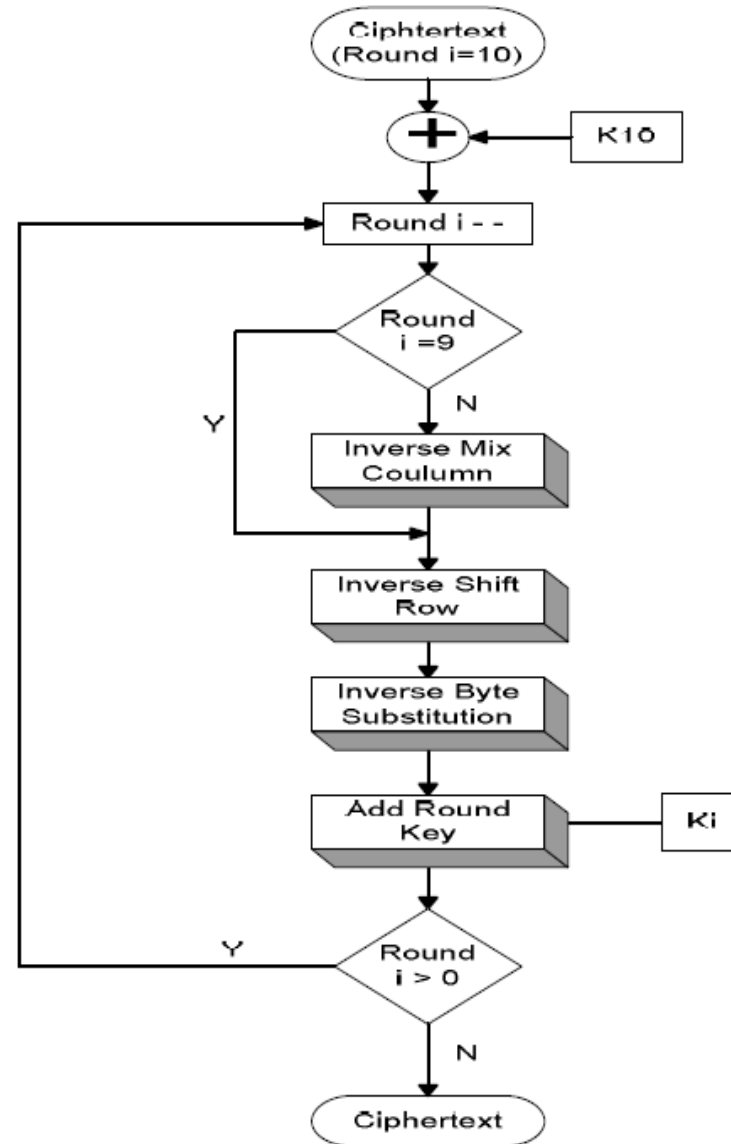
# AES

- 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 */

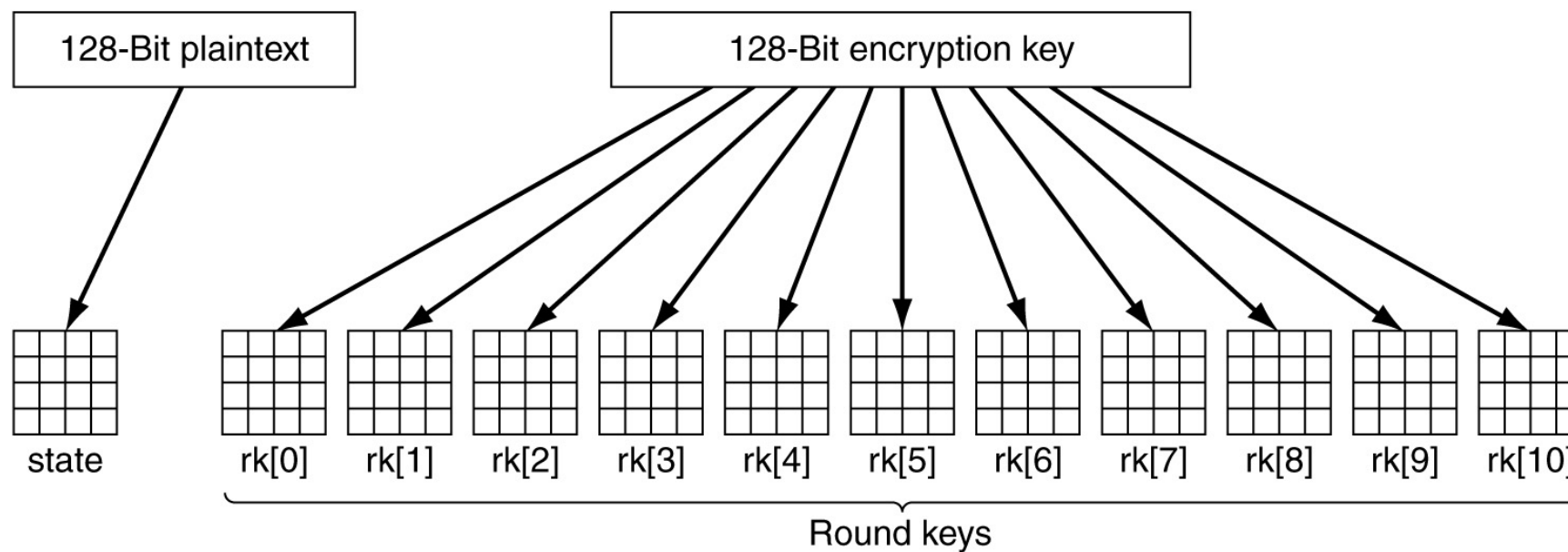
    expand_key(key, rk);                         /* construct the round keys */
    copy_plaintext_to_state(state, plaintext);   /* init current state */
    xor_roundkey_into_state(state, rk[0]);       /* XOR key into state */

    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 și rk:





# Cifrarea prin funcții greu inversabile

- Funcții greu inversabile:
  - Cunoscând  $x$  este ușor 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 intractabilă** doar pentru criptanalist;
  - Calculul lui  $x$  din  $f(x)$  trebuie să fie o **problemă tractabilă** pentru destinatarul autorizat, care dispune de o trapă ce face problema ușor de rezolvat.
- Problemă intratabilă:
  - Nu există un algoritm de rezolvare în timp polinomial.
- Metode:
  - Algoritmi exponențiali.
  - Problema rucsacului.



# Algoritmi exponențiali

- În metodele:
  - PH (Pohling și Hellman).
  - RSA (Rivest, Shamir și 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 și  $n-1$ ).
- **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,  $p$  și  $q$ , (de obicei de 1024 biți).
  - 2. Se calculează  $n = p \times q$  și  $z = (p - 1) \times (q - 1)$ .
  - 3. Se alege un număr relativ prim cu  $z$  și este notat cu  $d$ .
  - 4. Se găsește  $e$  astfel încât  $e \times d = 1 \pmod{z}$ .
- Exemplu:
  - Alegem  $p = 3$  și  $q = 11$ , rezultând  $n = 33$  și  $z = 20$ .
  - Alegem  $d = 7$  ( $7$  și  $20$  nu au factori comuni).
  - $e$  poate fi găsit din  $7e = 1 \pmod{20}$ , care dă  $e = 3$ .

Plaintext (P)		Ciphertext (C)			After decryption	
Symbolic	Numeric	$P^3$	$P^3 \pmod{33}$	$C^7$	$C^7 \pmod{33}$	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

- Funcția lui Euler:
  - $\Phi(n)$  = numărul de întregi pozitivi  $< n$  relativ și primi cu  $n$ .
  - Dacă  $p$  prim atunci  $\Phi(p) = p-1$ .
- Teoremă
  - Pentru  $n = p \cdot q$  cu  $p, q$  prime avem:
  - $\Phi(n) = \Phi(p) \cdot \Phi(q) = (p-1)(q-1)$
- 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  $a$  și  $n$  cu  $(a,n) = 1$  avem:
    - $a^{\Phi(n)} \bmod n = 1$ .



# Fundamentare

- Teoremă (cifrare):
  - Date fiind  $e$  și  $d$  care satisfac  $ed \bmod \Phi(n) = 1$  și un mesaj  $M \in [0, n-1]$  astfel că  $(M, n) = 1$ , avem  $(M^e \bmod n)^d \bmod n = M$ .
- Demonstrație:
  - $ed \bmod \Phi(n) = 1 \Rightarrow ed = t \Phi(n) + 1$  pentru un anumit  $t$ .
  - Calculăm  $(M^e \bmod n)^d \bmod n$ 

$$\begin{aligned}
 &= M^{ed} \bmod n \\
 &= M^{t \Phi(n) + 1} \bmod n \\
 &= M * M^{t \Phi(n)} \bmod n \\
 &= M (M^{t \Phi(n)} \bmod n) \bmod n \\
 &= M ((M^{\Phi(n)} \bmod n)^t \bmod n) \bmod n \\
 &= M ((1)^t \bmod n) \bmod n \\
 &= (M * 1) \bmod n \\
 &= M
 \end{aligned}$$
- Prin simetrie, cifrarea și descifrarea sunt comutative și mutual inverse:  $(M^e \bmod n)^d \bmod n = M$ , prin urmare
  - RSA poate fi utilizat pentru confidențialitate și autentificare.



# Metoda MH (Merkle și Hellman)

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



# Metoda MH (Merkle și 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' și A în A'.
  - Regăsirea lui M din C' și A' este o problemă "rucsac simplă".



# Sumar

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