



Nivelul Prezentare

Rolul nivelului prezentare
reprezentarea datelor cu tip, sintaxa de
transfer, conversia (Ex. ASN.1)
compresia datelor
criptarea



Scopul securității

- **confidentialitatea**
 - informația este disponibilă doar utilizatorilor autorizați
- **integritatea**
 - informația poate fi modificată doar de utilizatorii autorizați sau în modalitatea autorizată (mesajul primit nu a fost modificat în tranzit sau măsluit)
- **disponibilitatea**
 - accesul la informație a utilizatorilor autorizați nu este îngrădit (opusul este **denial of service**)

Probleme derivate

- **autentificarea**
 - determinarea identității persoanei cu care schimbi mesaje înainte de a dezvălui informații importante
- **controlul accesului**
 - protecția împotriva accesului ne-autorizat
- **non-repudierea**
 - transmitatorul nu poate nega transmiterea unui mesaj pe care un receptor l-a primit



Vulnerabilitati si atacuri

- Pregatirea atacurilor
 - scanarea porturilor
 - ingineria sociala
- Interceptarea traficului
- Impersonarea (mascarada, man-in-the-middle)
- DoS- Denial of Service
- Cal Troian – inclus in alt software.
- Virus – se reproduce prin alte fisiere executabile.
- Worm – se auto-reproduc.
- Logic Bomb – ramane inactiv pana la producerea unui eveniment (data, actiunea utilizatorului, eveniment aleator etc.)
- Erori in implementarea programelor
- Cod mobil malitios
- Modificari in serverele Web, DNS



Securitate - Persoane ce generează probleme

Adversar	Scop
Student	Pentru a se distra furând poșta electronică a celorlalți
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 funcționar	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 poșta electronică
Șarlatan	Pentru a fura numere de cărți de credit și a le vinde
Spion	Pentru a afla puterea militară a inamicului sau secrete industriale
Terorist	Pentru a fura secrete legate de conflicte armate



Metode de rezolvare

- Organizare
 - Servicii (protocoale) de securitate
 - Mecanisme de securitate
 - criptare, rezumare (hash), semnatura digitala
 - Algoritmi de criptare si hash
- Securitatea in ierarhia de protocoale
 - fizic – tuburi de securizare a liniilor de transmisie
 - legatura de date – legaturi criptate
 - retea – ziduri de protectie (firewalls), IPsec
 - transport – end-to-end security
 - aplicatie – autentificarea, non-repudierea

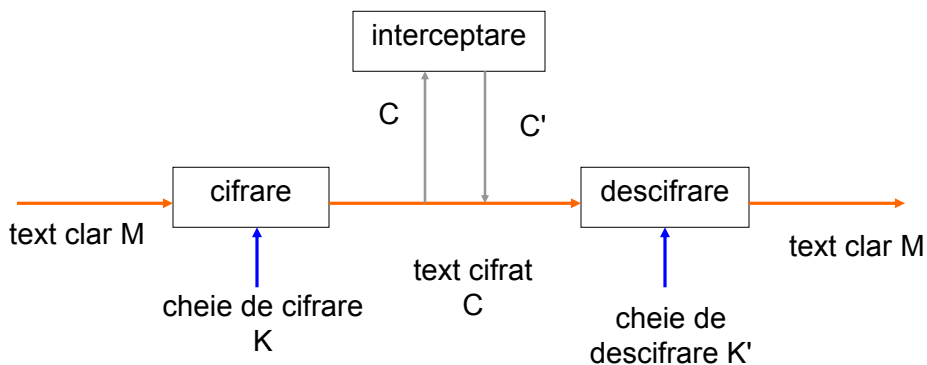


Alte aspecte

- Politici de securitate
- Administrarea securitatii
- Control software (ex. antivirus)
- Control hardware
 - cartele inteligente, biometrie
- Control fizic (protecție)
- Educație
- Măsuri legale



Modelul de bază al criptării



confidentialitatea - intrusul să nu poată reconstitui M din C (să nu poată descoperi cheia de descifrare K').

autenticarea - intrusul să nu poată introduce un text cifrat C', fără ca acest lucru să fie detectat (sa nu poată descoperi cheia de cifrare K).



Definiii

Spargerea cifrurilor = criptanaliză,
 Proiectarea cifrurilor = criptografie.
 Ambele sunt subdomenii ale criptologiei.

Transformarea realizată la **cifrarea** unui mesaj

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

$\{M\}$ = multimea mesajelor

$\{K\}$ = multimea cheilor

$\{C\}$ = multimea criptogramelor.

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
 - ex. cuvinte cheie (login)
- criptanaliză cu **text clar ales**; se cunosc
 - mod cifrare anumite porțiuni de text
 - ex. bază de date - modificare / efect



Caracteristicile sistemelor secrete

- sistem **neconditionat sigur**
 - rezistă la orice atac, indiferent de cantitatea de text cifrat interceptat (ex. one time pad).
- **computational 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



Cerinte criptosisteme cu chei secrete

- Cerinte **generale**:
 - cifrare si descifrare eficiente pentru toate cheile
 - sistem usor de folosit (chei de transformare)
 - securitatea să depindă de chei nu de algoritm
- Cerinte specifice **confidentialitate**: să fie imposibil computational ca un criptanalist să det. sistematic
 - transformarea D_k (mai precis, cheia k), din C , chiar dacă ar cunoaste M
 - M din C (fără a cunoaste D_k)
- Cerinte specifice **autentificare**: să fie imposibil computational ca un criptanalist să det. sistematic
 - transformarea E_k (mai precis, cheia k), din C , chiar dacă ar cunoaste M
 - cifra C' astfel ca $D_k(C')$ să fie un mesaj valid (fără a cunoaste E_k)

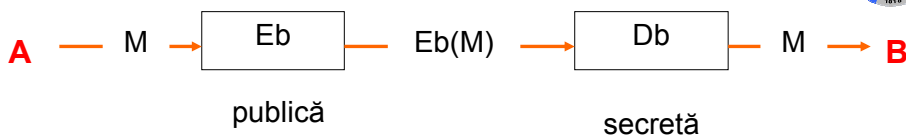


Modelul criptografic cu chei publice

- Sistemele criptografice:
 - simetrice
 - asimetrice
- Sistemele **asimetrice**
 - propuse de Diffie si Hellman în 1976
 - chei diferite de cifrare si descifrare
 - nu se pot deduce 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**.
- Fiecare utilizator U
 - face publică cheia (transformarea) **E_u** de cifrare (**cheia publica**)
 - păstrează secretă cheia (transformarea) **D_u** de descifrare (**cheia privata**).
- **Schema de autentificare**
 - condiția necesară este ca transformările **E_a** si **D_a** să comute, adică
 - $E_a(D_a(M)) = D_a(E_a(M)) = M$.



Schema de protecție



Schema de autentificare



Se asigură:

confidențialitate

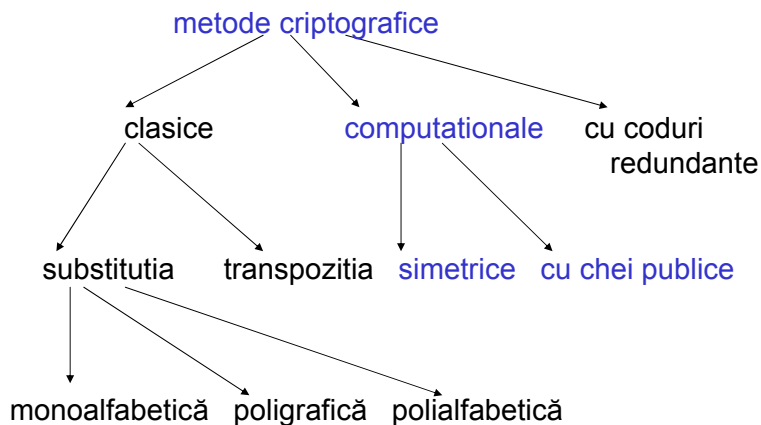
autentificare

ne-repudiare

B are garanția că A este sursa mesajului
(semnătură digitală - se poate semna doar rezumat)
folosind perechea $D_a(M)$ și M



Clasificare generală





Utilizarea TI în evaluarea algoritmilor criptografici

Analogia transmisie - confidentialitate
 perturbarea $\Leftrightarrow$ cifrarea mesajului
 mesajul recepționat cu erori $\Leftrightarrow$ 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: $\log(1/p(X))$ = nr biti codif. 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 multimea mesajelor $Y_1, \dots, Y_n$ cu

$$\sum_{i=1,n} p(Y_i) = 1$$

fie $p_Y(X)$ prob mesajului X condiționat de Y

$p(X, Y)$ prob mesajelor X și Y luate împreună: $p(X, Y) = p_Y(X) \cdot p(Y)$

Echivocitatea este entropia lui X condiționat de Y:

$$\begin{aligned} H_Y(X) &= -\sum_{X,Y} p(X, Y) \log p_Y(X) \\ &= \sum_{X,Y} p_Y(X) \cdot p(Y) \log(1/p_Y(X)) = \sum_Y p(Y) \sum_X p_Y(X) \log(1/p_Y(X)) \end{aligned}$$

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:

$Y_1 - X_1$ sau X_2 $Y_3 - X_2$ sau X_3

$Y_2 - X_3$ sau X_4 $Y_4 - X_4$ sau X_1

Echivocitatea este:

$$H_Y(X) = 4 \left(\left(\frac{1}{4} \right) 2 \left(\frac{1}{2} \right) \log 2 \right) = \log 2 = 1.$$

$\Rightarrow$ cunoașterea lui Y reduce incertitudinea lui X la un bit.

Confidentialitatea perfectă

Fie: M - texte clare cu prob $p(M)$, $\sum_M p(M) = 1$
 C criptograme, cu prob $p(C)$, $\sum_C p(C) = 1$
 K chei cu prob $p(K)$, $\sum_K p(K) = 1$

$p_C(M)$ prob să se fi transmis M când se recepționează C.

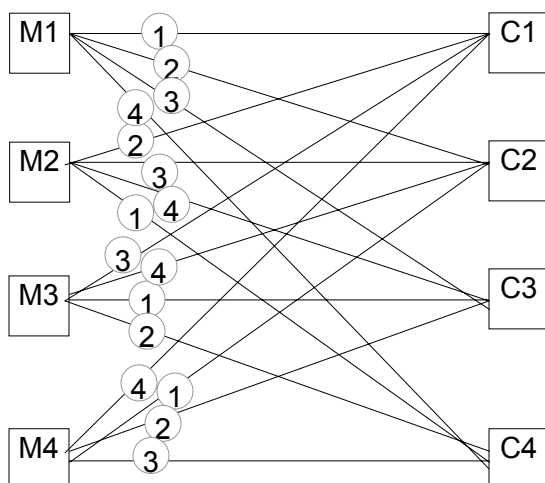
Confidentialitatea perfectă $\Leftrightarrow p_C(M) = p(M)$.

$p_M(C)$ prob să se recepționeze C când s-a transmis M:

$$p_M(C) = \sum_{k, E_k(M)=C} p(k)$$

Confidentialitate perfectă

$p_M(C) = p(C)$, pentru toate M și orice C.



Confidentialitatea 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 (cifru se poate sparge).

Când crește lungimea N a textelor cifrate echivocitatea descrește.

Distanța de unicitate = cel mai mic N pentru care $H_C(K)$ este foarte apropiat de 0.

Cifru necondiționat sigur =>

$H_C(K)$ nu se apropie niciodată de 0.



Calcul aproximativ distanța unic.

Notatii

Pt. un limbaj, luăm mulțimea mesajelor de lungime N.

Rata limbajului

$$r = H(X) / N$$

$r = 1 \dots 1.5$ pentru l. engleză.

Rata absolută a limbajului (pentru L simboluri)

$$R = -\sum_{i=1,L} (1/L) \log (1/L) = \log L$$

$R = \log 26 = 4.7$ biti pe literă pentru l. engleză.

Redundantele apar din structura limbaj: distribuția frecvențelor literelor, digramelor, trigramelor etc)

$$D = R - r$$

$D = 3.2 \dots 3.7$ în l. engleză.



Ipoteze:

sunt 2^{rN} mesaje posibile de lungime N

2^{rN} mesaje 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 prob q de a produce o soluție falsă

(**același C se obține criptând un alt mesaj M' cu înțeles, cu altă cheie K'**).

$$q = 2^{rN} / 2^{rN} = 2^{-DN} \quad (D = R-r \text{ este redundanța limbajului})$$

Numărul de soluții false

$$F = (2^{H(K)} - 1)q = (2^{H(K)} - 1) 2^{-DN} \approx 2^{H(K)-DN}$$

$$\log F = H(K) - DN = 0$$

$$N = H(K) / D$$



Confuzie și difuzie

- Definite de Shannon
- Confuzia**
 - relația între **cheie** și **textul cifrat** trebuie să fie cât mai complexă
 - efect – un atacator are nevoie de mult timp să afle relația
- Difuzia**
 - redundanța în statisticile textului clar este "disipată" în statisticile textului cifrat.
 - este asociată cu dependența între biții ieșirii și biții intrării (modif unui bit la intrare modifică fiecare bit la ieșire cu probabilitatea 1/2)
 - ex. difuzie bună – modificarea unui caracter în intrare este distribuită întregii ieșiri
 - efect – un atacator trebuie să intercepteze mult text cifrat
- Mecanisme pentru confuzie și difuzie
 - Confuzie** - substituția
 - Difuzie**
 - Transpoziția
 - Transformări lineare



Cifrarea prin substitutie

Cifrul lui Cezar (substitutie 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

textul clar: **CRIPTOGRAFIE**

text cifrat: **FULSWRJUDILH**

Relatia de calcul

$$c[i] = (m[i] + 3) \bmod 26$$

In general

$$c[i] = (a \cdot m[i] + b) \bmod n.$$

Substitutia polialfabetică (Vigenere)

foloseste 36 de cifruri Cezar si o cheie de cifrare de lungime l

Exemplu: cheia POLIGRAF.

POLIGRAFPOLIGRAG**POLIGRAF**POLIGRAF**POLI**
AFOSTODATACANPOVESTIAFOSTCANICIODATA
PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI



Cifrul Beaufort

$$c[i] = (k[i] - m[i]) \bmod n$$

Pentru descifrare

$$m[i] = (k[i] - c[i]) \bmod n$$

Substitutia poligrafică

un grup de n litere este înlocuit cu un alt grup de n litere.

Analiza

Substitutie **monoalfabetică**:

$$N = H(K) / D = \log n! / D$$

Pentru l. engleză: $N = \log 26! / 3.2 = 27.6$

Substitutie **periodică** cu perioada d

sunt s^d chei posibile pentru fiecare substitutie simplă =>

$$N = H(K) / D = \log s^d / D = (d \cdot \log s) / D$$

Pentru cifrul Vigenere $s = 26$ deci

$$N = d \cdot 4.7 / 3.2 = 1.5 d$$



Cifrarea prin transpozitie

Modifică ordinea caracterelor. Uzual:

- textul clar dispus în liniile succesive ale unei matrice și
- parcurgerea acesteia după o anumită regulă pentru stabilirea noii succesiuni de caractere.

Exemplu

- caracterele dispuse pe linii sînt citite pe coloane,
- ordinea coloanelor este dată de ordinea alfabetică a literelor unei chei.

cheie: POLIGRAF

ordine: 76543812

text clar:

AFOSTODATACANPOVESTIAFOSTCANICIO

POLIGRAF
AFOSTODA
TACANPOV
ESTIAFOS
TCANICIO

text cifrat:

DOOIAVSOTNAISAINOCTAFASCATETOPFC



Analiza

Pentru spargerea cifrului:

cifrul permută caracterele cu o perioadă fixă d .

sunt $d!$ permutări posibile

toate sunt echiprobabile

$$H(K) = \log d!$$

=>

$$N = H(K) / D = \log d! / D$$

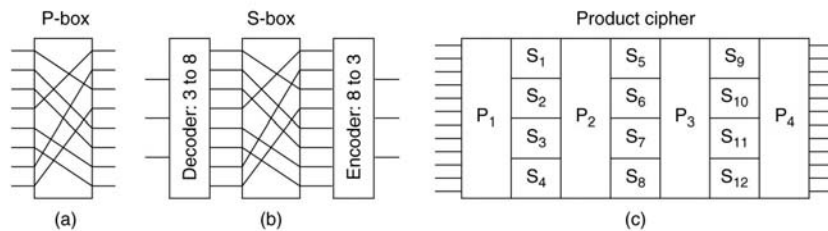
$$N = d \log (d/e) / D$$

Pentru $d = 2.7$

$D = 3.2$

=> $N = 27$

Cifruri produs



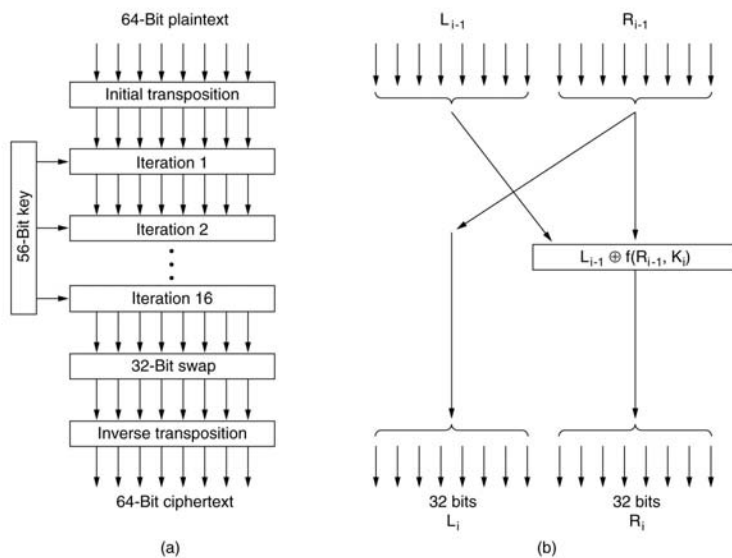
Principii pentru a obține o securitate mai mare:

- **compune** două cifruri "slabe", complementare
 - P-box asigură difuzia
 - S-box asigură confuzia
- **repetă** aplicarea permutării și substituției

DES (Data Encryption Standard)

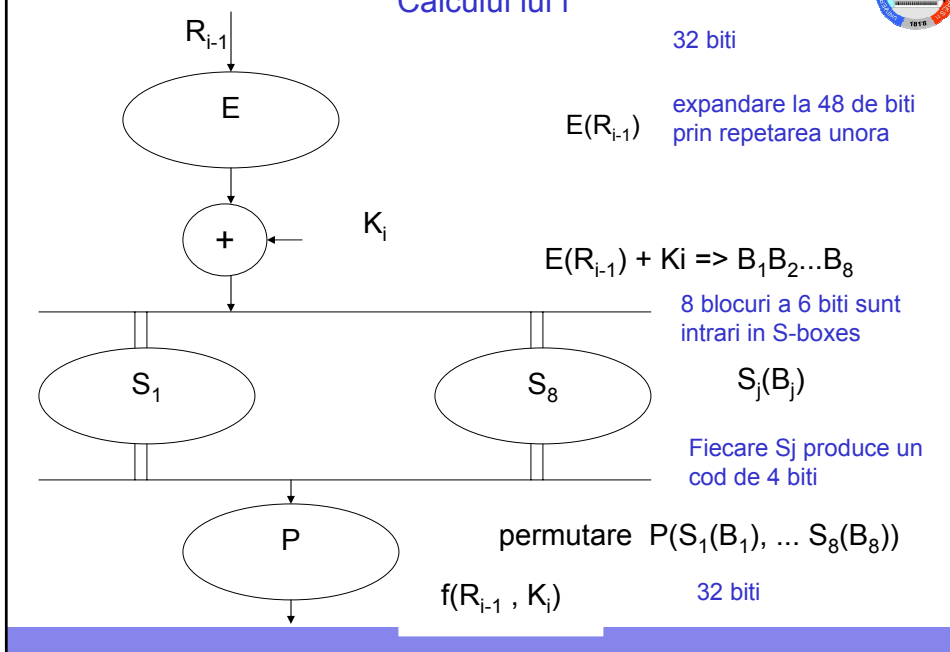
Schema generală

0 iterație

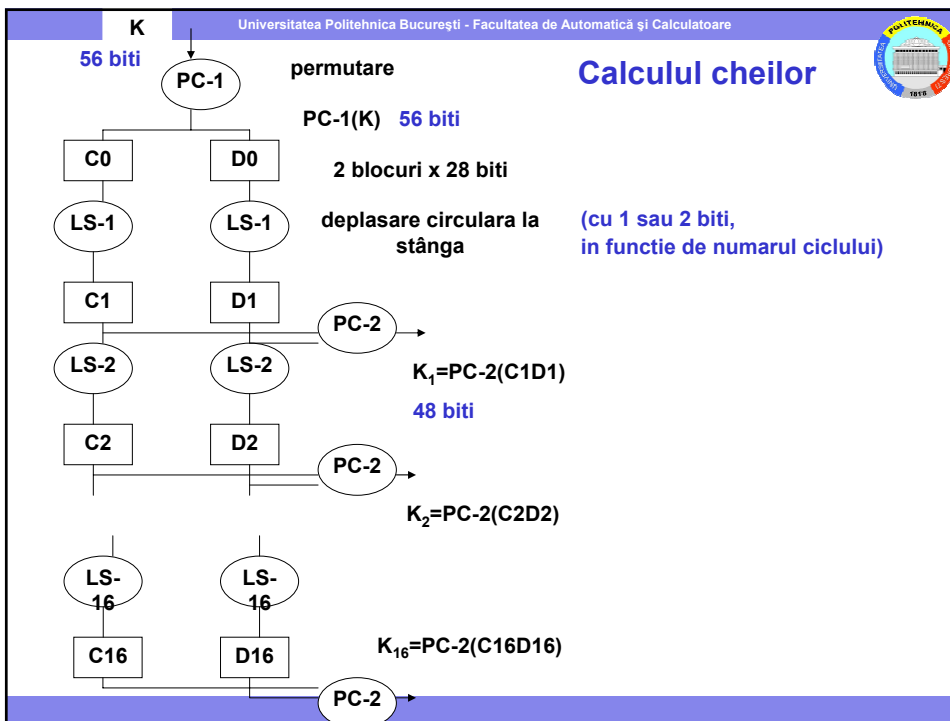




Calculul lui f



Calculul cheilor





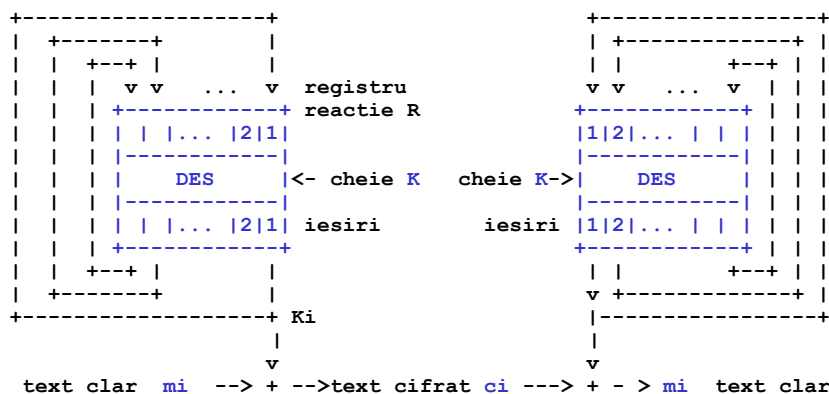
Comentarii

- Transpozițiile, expandările, substituțiile, permutările sunt definite în DES
- Același algoritm folosit la criptare și decriptare
 - La criptare: $L_j = R_{j-1}$
 $R_j = L_{j-1} (+) f(R_{j-1}, k_j)$
 - De unde: $R_{j-1} = L_j$
 $L_{j-1} = R_j (+) f(R_{j-1}, k_j)$
 - și $L_{j-1} = R_j (+) f(L_j, k_j)$
- Decriptare = criptare în ordine inversă (cu cheile în ordinea $k_{16} - k_1$)
- Elementele cheie ale algoritmului nu au fost făcute publice
 - Controverse
 - Există trape care să ușureze decriptarea de către NSA? NSA declară că NU.
 - Descoperirea și folosirea unei astfel de trape de un criptanalist răuvoitor
 - Urmarea – unele detalii despre S-box au fost dezvăluite de NSA
 - Număr de iterații – suficiente pentru difuzie?
 - Experimental, după 8 iterații nu se mai văd dependențe ale ieșirii de biți / grupuri de biți din intrare
 - Lungimea cheii
 - Cheie DES de 56 biți spartă prin forță brută (4 luni * 3500 mașini) în 1997
 - Dar, nu au fost reportate deficiențe în algoritm
 - Triple DES "mărește" lungimea cheii



Cifrarea secvențială

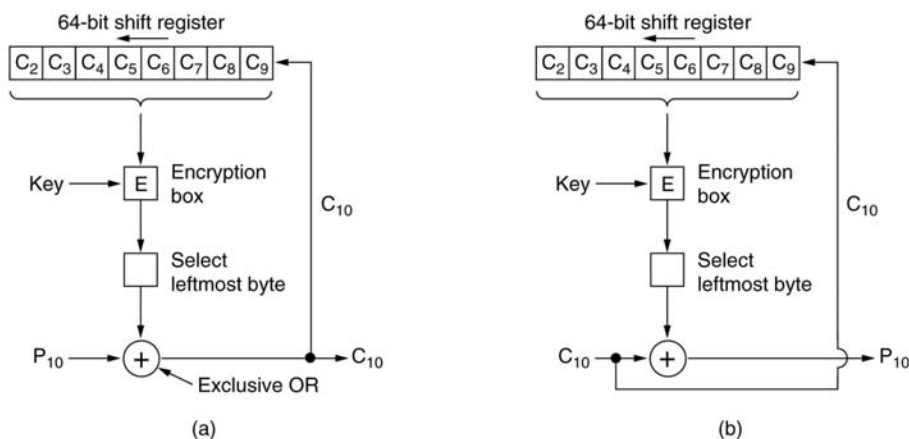
Sistem secvențial sincron cu reacție bloc (OFB - Output Feed Back)



Folosește un **Initialization Vector** ca prima intrare în R
 Nu trebuie refolosită aceeași pereche (K, IV)



CFB - K-bit Cipher-Feed Back

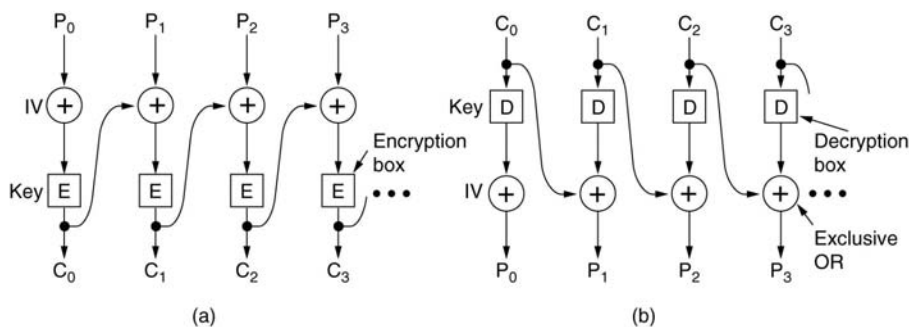


Folosește un **Initialization Vector** ca prima valoare în **Registrul de deplasare**

O eroare de un bit în criptograma conduce la decriptarea eronată a 8 octeți



CBC - Cipher Block Chaining



Key – cheie secretă

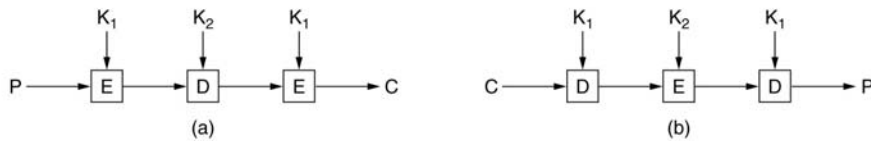
IV – Initialization Vector

ales aleator, același pentru criptare și decriptare

folosit pentru combinarea cu primul bloc

Avantaj: același text clar repetat în mesaj va fi criptat diferit

Triplu DES



AES – Advanced Encryption Standard

Regulile concursului organizat de NIST (ianuarie 1997) erau:

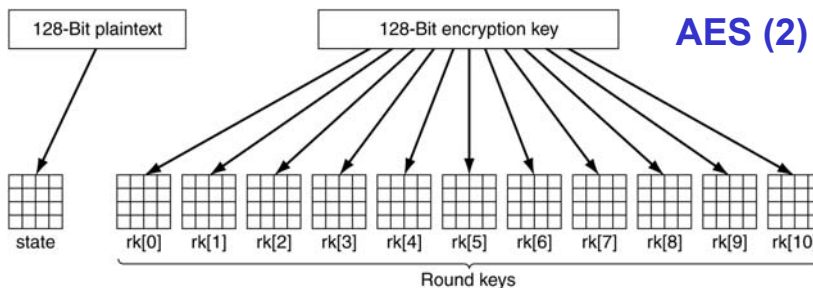
1. Algoritmul trebuie să fie un cifru bloc simetric.
2. Tot proiectul trebuie să fie public
3. Trebuie să fie suportate chei de 128, 192, și de 256 biți
4. Trebuie să fie posibile atât implementări hardware cât și software
5. Algoritmul trebuie să fie public sau oferit cu licență nediscriminatorie.

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)



AES (2)



n runde ($n=10$ pentru cheie de lungime 128; 12/ 192, 14/256)

Bloc 128 biți = matrice 4*4 octeți – **state**

Operații pe coloane sau pe linii; 4 operații pe rundă

substitute – la nivel octet, folosește tabel substituție

rotate_rows – prin deplasare circulară la stânga la nivel octet

1	5	9	13	→	1	5	9	13
2	6	10	14		6	10	14	2
3	7	11	15		11	15	3	7
4	8	12	16		16	4	8	12



mix_columns – elementele unei coloane sunt înmulțite cu o matrice

$$\begin{vmatrix} s'0i \\ s'1i \\ s'2i \\ s'3i \end{vmatrix} = \begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix} \begin{vmatrix} s0i \\ s1i \\ s2i \\ s3i \end{vmatrix}$$

xor_roundkey_into_state – adaugă o cheie $rk[i]$

Rijndael definit în **câmp Galois $G(2^8)$** prin polinomul $P = x^8 + x^4 + x^3 + x + 1$
 număr = coeficienții unui polinom

Ex. 23 = 10111₍₂₎ este polinomul $1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x + 1$
 $x^4 + x^2 + x + 1$

adunarea coeficienților făcută modulo 2

înmulțirea făcută ca la polinoame, dar modulo P

Ex. $(x^3 + 1) \cdot (x^4 + x) = x^7 + x^4 + x^4 + x = x^7 + x$



Algoritmul AES (3)

```

#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 */
}

```



Comentarii

- Nu au fost probleme la utilizare
- Experimental – difuzie bună
- Metodă bazată pe algebră (câmpuri Galois)
 - substituții și mixare coloane folosesc operații cu sens în teoria algebrică (nu simple tabele greu de explicat)
 - autorii nu au oferit argumente matematice
 - nu sunt suspectate trape sau scurtături ascunse



Cifrarea prin functii greu inversabile

- functii 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 intratabilă doar pentru criptanalist
 - nu 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 - RSA

În 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 $n-1$)

C este criptograma.

Descifrarea se face prin calculul

$$M = (C^d) \bmod n$$

unde (d, n) este cheia de descifrare



RSA

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 d un număr relativ prim cu z
de regula, d este un număr prim mai mare ca $(p-1)$ și $(q-1)$
4. Se găsește e astfel încât $e \times d = 1 \bmod z$.

Ex:

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 $7^e = 1 \bmod 20$, care dă $e = 3$.

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

$\Phi(n)$ = nr de întregi pozitivi $< n$ relativ primi cu n

p prim $\Rightarrow \Phi(p) = p-1$.

T. Pentru $n = p \times q$ cu p, q prime

$$\Phi(n) = \Phi(p) \times \Phi(q) = (p-1)(q-1)$$

T. (**Fermat**). Fie p un număr prim. Pentru orice a cu $(a, p) = 1$:

$$a^{p-1} \bmod p = 1$$

T. (**Euler**). Pentru orice a și n cu $(a, n) = 1$ avem

$$a^{\Phi(n)} \bmod n = 1$$



T. (cifrare). Date fiind e și d care satisfac

$$ed \bmod \Phi(n) = 1$$

și un mesaj $M \in [0, n-1]$ astfel că $\gcd(M, n) = 1$, avem

$$(M^e \bmod n)^d \bmod n = M$$

Dem. $ed \bmod \Phi(n) = 1 \Rightarrow ed = t \Phi(n) + 1$ ptr. un anumit t .

$$\begin{aligned} (M^e \bmod n)^d \bmod n &= M^{ed} \bmod n \\ &= M^{t \Phi(n) + 1} \bmod n \\ &= M * M^{t \Phi(n)} \bmod n = M(M^{t \Phi(n)} \bmod n) \bmod n \\ &= M((M^{\Phi(n)} \bmod n)^t \bmod n) \bmod n \\ &= M((1)^t \bmod n) \bmod n \\ &= (M \cdot 1) \bmod n = M \end{aligned}$$

Comentarii

Prin simetrie, cifrarea și descifrarea sunt comutative și mutual inverse.

$$(M^e \bmod n)^d \bmod n = M$$

$\Rightarrow$ **RSA utilizată ptr confidentialitate și autentificare.**

Nu au fost identificate atacuri reușite cu RSA



Metoda MH (Merkle și Hellman)

Problema rucsacului

Se cere determinarea lui $X = (x[1], x[2], \dots, x[n])$ cu elemente binare, a.i.

$$C = \sum_{i=1, n} x[i] * a[i]$$

Găsirea unei soluții = backtracking

$\Rightarrow$ număr operații care crește exponențial cu n .

O soluție x poate fi **verificată** prin cel mult n operații de adunare

Varianta **rucsac simplu** a problemei (trapa):

dacă A satisface **proprietatea de dominantă** (este o secvență supercrescătoare), adică

$$a[i] > \sum_{j=1, i-1} a[j]$$

atunci problema poate fi rezolvată în timp liniar.

Ex.

text clar	1	0	1	0	0	1
rucsac	1	2	5	9	20	43
text cifrat	1		5			43
suma	49					



Cheie publica: secventa (oarecare) de intregi

Cheie secreta: secventa supercrescatoare

Contributia Merkle si Hellman

conversie secventa (oarecare) $\Leftrightarrow$ secventa supercrescatoare

Solutia: aritmetica modulara

rucsac simplu $A = [a_1, a_2, \dots, a_n]$

rucsac greu $G = [g_1, g_2, \dots, g_n]$ cu $g_i = w * a_i \bmod n$

Ex.

rucsac simplu $A = [1, 2, 4, 9], w = 15, n = 17$

$$1 * 15 \bmod 17 = 15 \bmod 17 = 15$$

$$2 * 15 \bmod 17 = 30 \bmod 17 = 13$$

$$4 * 15 \bmod 17 = 60 \bmod 17 = 9$$

$$9 * 15 \bmod 17 = 135 \bmod 17 = 16$$

rucsac greu $G = [15, 13, 9, 16]$

Obs. inmultirea $\bmod n$ strica proprietatea de dominanta



Conditii:

Toate numerele din G trebuie sa fie distincte intre ele

Conversia inversa de la G la A trebuie sa produca o solutie unica

Ex.

x	$3*x$	$3*x \bmod 6$	x	$3*x$	$3*x \bmod 5$
1	3	3	1	3	3
2	6	0	2	6	1
3	9	3	3	9	4
4	12	0	4	12	2
5	15	3	5	15	0
6	18	0	6	18	3

Cerinte:

1. n trebuie sa fie mai mare decat suma tuturor a_i
2. w si n trebuie sa fie prime intre ele (se alege n prim)
 $\Rightarrow w$ are un invers multiplicativ w^{-1} ($w * w^{-1} = 1 \bmod n$)



Criptare

Obține criptograma C din textul clar P prin $C = G * P$

unde G este rucsacul greu, $G = w * A \bmod n$ (adică $g_i = w * a_i \bmod n$)

Ex: $P = [1, 0, 1, 0]$, $G = [15, 13, 9, 16]$, $C = 15 + 9 = 24$

Decriptare

Receptorul cunoaște A, w, n și, bineînțeles, G

Deoarece $C = G * P = w * A * P \bmod n$, rezulta

$$w^{-1} * C = w^{-1} * G * P = w^{-1} * w * A * P \bmod n = A * P \bmod n$$

din care P se afla prin rucsac simplu

Ex. $A = [1, 2, 4, 9]$, $w = 15$, $n = 17$

$$15^{-1} = 8 \bmod 17 \Rightarrow 8 * 24 = 192 \bmod 17 = 5 \Rightarrow P = [1, 0, 1, 0]$$

Comentarii

Metoda pare sigură

S-au găsit metode de atac prin ocolire rucsac greu în anumite cazuri

Oricum, algoritmul MH este greu de utilizat