Nivelul Prezentare

Rolul nivelului prezentare

reprezentarea datelor cu tip, sintaxa de transfer, conversia (Ex. ASN.1)

compresia datelor

criptarea

Scopul securitatii

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

Probleme derivate

- autentificarea
  - determinarea identităţii persoanei cu care schimbi mesaje înainte de a dezvălui informaţii importante
- controlul accesului
  - protectia impotriva 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

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

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

Metode de rezolvare

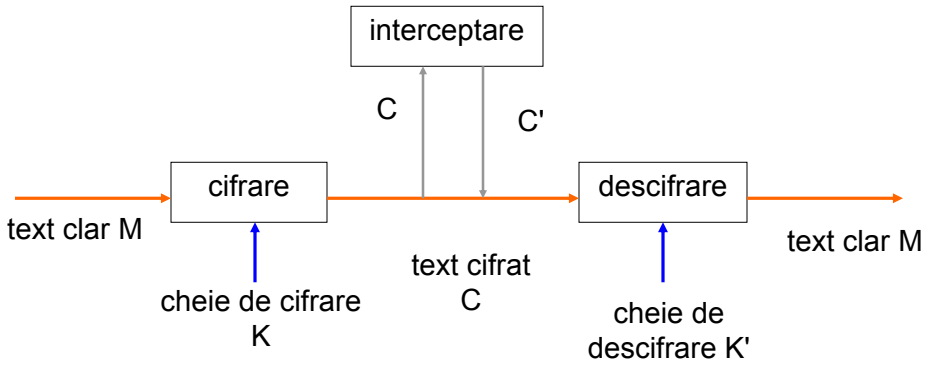
- Organizare
  - Servicii (protocoale) de securitate
  - Mecanisme de securitate
    - **criptare, rezumare (hash), semnatura digitala**
  - Algoritmi de criptare si hash
- Securitatea in ierarhia de protocoale
  - fizic – tuburi de securizare a liniilor de transmisie
  - legatura de date – legaturi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

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Alte aspecte

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

Modelul de bază al criptării



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

Definitii

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

Transformarea realizată la **cifrarea** unui mesaj

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

{M} = multimea mesajelor
{K} = multimea cheilor
{C} = multimea criptogramelor.

cifrarea $C = E_k(M)$
descifrarea $M = D_{k'}(C)$.

Conotatie 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 portiuni de text**
  - **ex. bază de date - modificare / efect**

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

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 probabilităti apropriate

Cerinte criptosisteme cu chei secrete

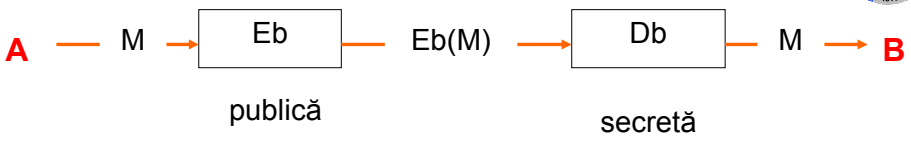
- Cerinte **generale**:
  - cifrare si descifrare eficiente pentru toate cheile
  - sistem usor de folosit (chei de transformare)
  - securitatea să depindă de chei nu de algoritm
- Cerinte specifice **confidentialitate**: să fie imposibil computational ca un criptanalist să det. sistematic
  - transformarea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

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

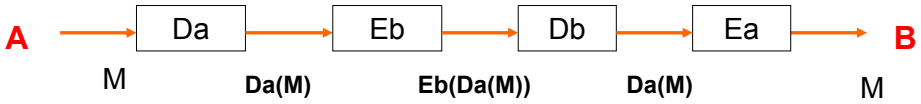
Modelul criptografic cu chei publice

- 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) **Eu** de cifrare (**cheia publica**)
  - păstrează secretă cheia (transformarea) **Du** de descifrare (**cheia privata**).
- **Schema de autentificare**
  - conditia necesară este ca transformările **Ea** si **Da** să comute, adică
  - Ea(Da(M)) = Da(Ea(M)) = M.

Schema de protectie



Schema de autentificare

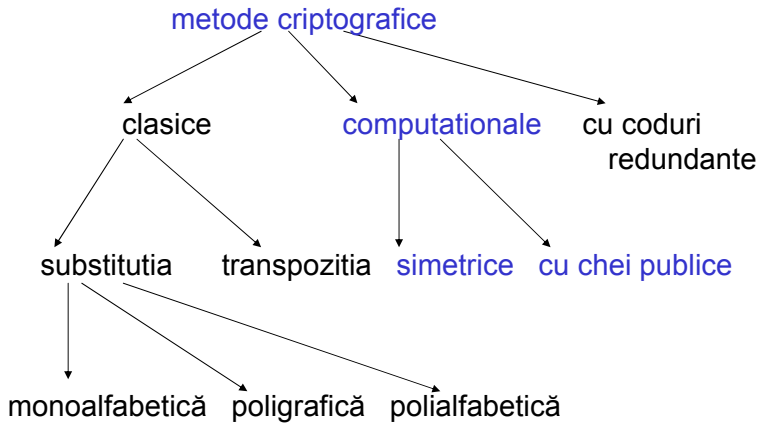


Se asigură:

confidentialitate
autentificare — B are garantia că A este sursa mesajului (semnătură digitală - se poate semna doar rezumat)
ne-repudiere — folosind perechea Da(M) si M

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Clasificare generală



Utilizarea TI în evaluarea algoritmilor criptografici

Analogia transmisie - confidentialitate

perturbarea <=> cifrarea mesajului
mesajul receptionat cu erori <=> text cifrat

Cantitatea de informatie = **entropie**.

X1,..., Xn mesajele unei surse

p(X1), ... , p(Xn) probabilitătile ($\Sigma_{i=1,n}$ p(Xi) = 1)

Entropia unui mesaj $H(X) = -\Sigma_{i=1,n}\ p(Xi) \log p(Xi)$

Intuitiv: log(1/p(X)) = nr biti codif. optimă a lui X.

Entropia măsoară si incertitudinea.

H(X) maxim când p(X1) = p(X2) = ... = p(Xn) = 1/n

H(X) descreste când distributia mesajelor se restrânge.

H(X) = 0 când p(Xi) = 1 pentru un mesaj i.

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Echivocitatea

Dat fiind Y din multimea mesajelor Y1, ..., Yn cu

$\Sigma_{i=1,n}\ p(Yi) = 1$

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

p(X,Y) prob mesajelor X si Y luate împreună: $p(X,Y) = p_Y(X)*p(Y)$

Echivocitatea este entropia lui X conditionat de Y:

$$H_Y(X) = -\Sigma_{X,Y}\ p(X,Y) \log p_Y(X)$$

$$= \Sigma_{X,Y}\ p_Y(X)*p(Y) \log (1/p_Y(X)) = \Sigma_Y\ p(Y) \Sigma_X\ p_Y(X)* \log (1/p_Y(X))$$

Exemplu:

n = 4 si p(X) = 1/4 pentru fiecare X => H(X) = log 4 = 2

Fie m=4 si p(Y) = 1/4 pentru fiecare Y.

Presupunem că fiecare Y restrânge X:

Y1 - X1 sau X2 Y3 - X2 sau X3

Y2 - X3 sau X4 Y1 - X4 sau X1

Echivocitatea este:

$H_Y(X) = 4 ((1/4) 2 (1/2) \log 2) = \log 2 = 1$.

=> cunoasterea lui Y reduce incertitudinea lui X la un bit.

Confidentialitatea perfectă

Fie: M - texte clare cu prob p(M), Σ_M p(M) = 1
C criptograme, cu prob p(C), Σ_C p(C) = 1
K chei cu prob p(K), Σ_K p(K) = 1

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

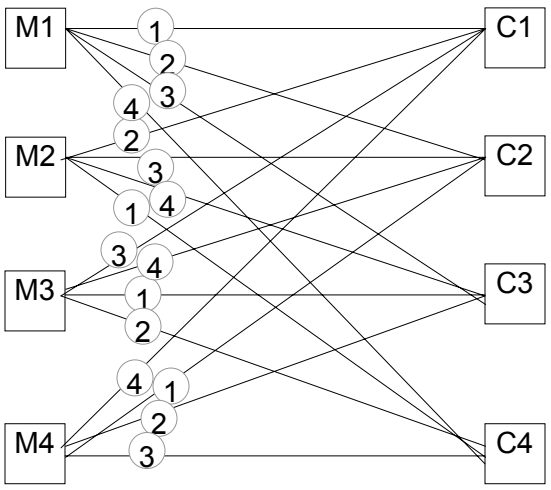
Confidentialitatea perfectă <=> $p_C(M) = p(M)$.

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

$$p_M(C) = \Sigma_{k,\ Ek(M)=C}\ p(k)$$

Confidentialitate perfectă

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



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

Distanta de unicitate

Confidentialitatea = cantitatea de incertitudine in 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 (cifrul se poate sparge).
Când creste lungimea N a textelor cifrate echivocitatea descreste.

Distanta 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.

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Calcul aproximativ distanta unic.

Notatii

Pt. un limbaj, luam multimea mesajelor de lungime N.

Rata limbajului

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

r = 1 ... 1.5 pentru l. engleză.

Rata absolută a limbajului (pentru L simboluri)

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

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

Redundantele apar din structura limbaj: distributia frecventelor literelor, digramelor, trigramelor etc)

$$D = R - r$$

D = 3.2 ... 3.7 în l. engleză.

Ipoteze:

sunt 2^{RN} mesaje posibile de lungime N

2^{rN} mesaje au sens

toate mesajele cu sens au aceeasi probabilitate, $1/2^{rN}$

toate mesajele fără sens au probabilitate 0

sunt $2^{H(K)}$ chei cu probabilitati egale

cifrul este aleator

pentru fiecare k si 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 aceeasi prob **q** de a produce o solutie **falsă**
(acelasi C se obtine criptând un alt mesaj M' cu înteles, cu altă cheie K').

$$q = 2^{rN} / 2^{RN} = 2^{-DN}$$ **(D = R-r este redundanta limbajului)**

Numărul de solutii false

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

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

$N = H(K) / D$

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Confuzie şi difuzie

- Definite de Shannon
- *Confuzia*
  - relaţia între cheie şi textul cifrat trebuie să fie cât mai complexă
  - efect – un atacator are nevoie de mult timp să afle relaţia
- *Difuzia*
  - redundanţa în statisticile textului clar este "disipată" în statisticile textului cifrat.
  - este asociată cu dependenţa între biţii ieşirii şi biţii intrării (modif unui bit la intrare modifică fiecare bit la ieşire cu probabilitatea 1/2)
  - ex. difuzie bună – modificarea unui caracter în intrare este distribuită întregii ieşiri
  - efect – un atacator trebuie să intercepteze mult text cifrat
- Mecanisme pentru confuzie şi difuzie
  - Confuzie - substituţia
  - Difuzie
    - Transpoziţia
    - 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) mod 26

In general

c[i] = (a.m[i] +b) mod n.

Substitutia polialfabetică (Vigenere)

foloseste 36 de cifruri Cezar si o cheie de cifrare de lungime l
Exemplu: cheia POLIGRAF.

POLIGRAFPOLIGRAGPOLIGRAFPOLIGRAFPOLI
AFOSTODATACANPOVESTIAFOSTCANICIODATA
PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI

Cifrul Beaufort

c[i] = (k[i] - m[i]) mod n

Pentru descifrare

m[i] = (k[i] - c[i]) mod 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 =>

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

Pentru cifrul Vigenere s = 26 deci

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

Cifrarea prin transpozitie

Modifică ordinea caracterelor. Uzual:
- **textul clar dispus în liniile succesive ale unei matrice si**
- **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

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Analiza

Pentru spargerea cifrului:

cifrul permută caracterele cu o perioadă fixă d.
sunt d! permutări posibile
toate sunt echiprobabile

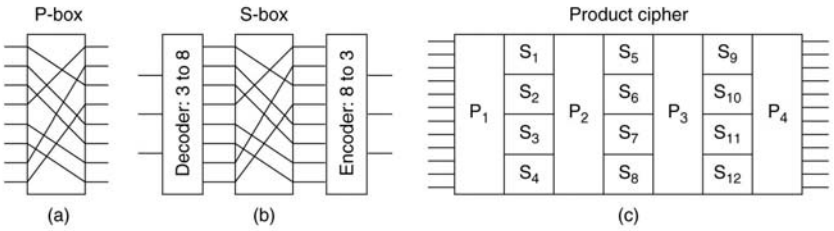
$$H(K) = \log d!$$

=>

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

Pentru d = 2.7
D = 3.2
=> N = 27

Cifruri produs

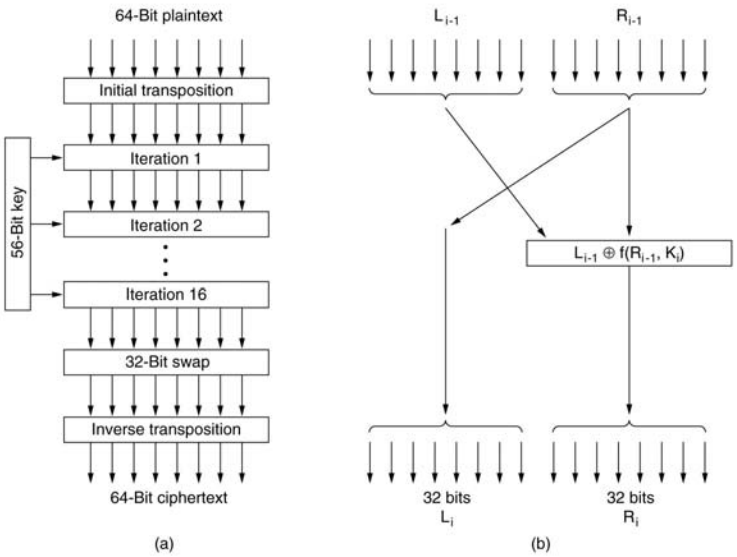


Principii pentru a obţine o securitate mai mare:

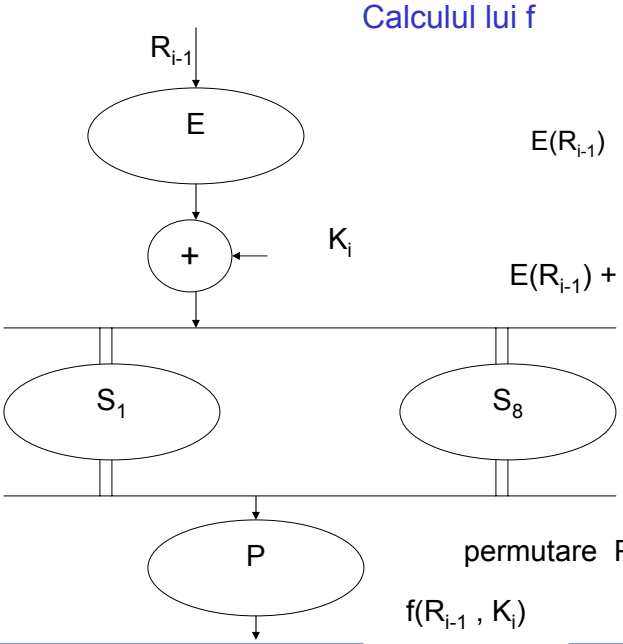
- compune două cifruri "slabe", complementare
  - P-box asigură difuzia
  - S-box asigură confuzia
- repetă aplicarea permutării şi substituţiei

DES (Data Encryption Standard)

Schema generală **O iterație**



Calculul lui f



32 biti

$E(R_{i-1})$ expandare la 48 de biti prin repetarea unora

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

8 blocuri a 6 biti sunt intrari in S-boxes

$S_j(B_j)$

Fiecare Sj produce un cod de 4 biti

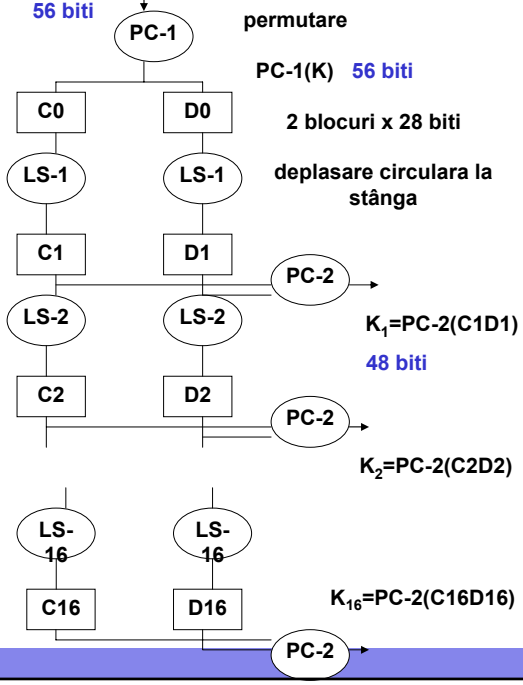
permutare $P(S_1(B_1), ... S_8(B_8))$

$f(R_{i-1}, K_i)$ 32 biti

Calculul cheilor

K

56 biti



permutare

PC-1(K) 56 biti

2 blocuri x 28 biti

deplasare circulara la stânga

(cu 1 sau 2 biti, in functie de numarul ciclului)

K_1=PC-2(C1D1)

48 biti

K_2=PC-2(C2D2)

K_{16}=PC-2(C16D16)

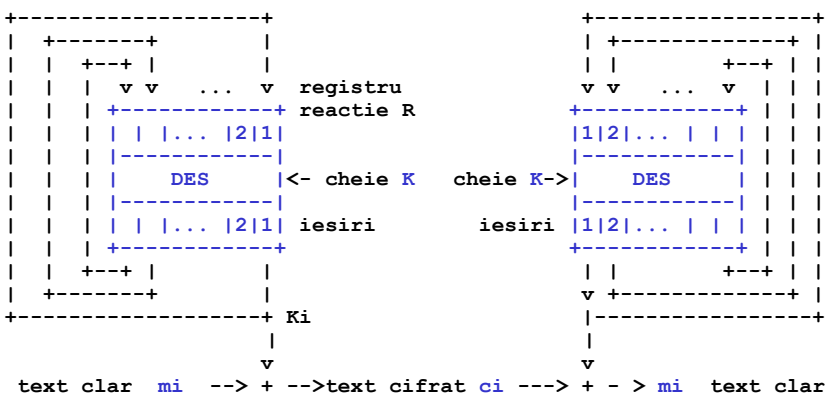
Comentarii

- Transpoziţiile, expandările, substituţiile, permutările sunt definite în DES
- Acelaşi algoritm folosit la criptare si decriptare

  La criptare: $L_j = R_{j-1}$
  $R_j = L_{j-1} (+) f(R_{j-1},k_j)$
  De unde: $R_{j-1} = L_j$
  $L_{j-1} = R_j (+) f(R_{j-1},k_j)$
  şi $L_{j-1} = R_j (+) f(L_j,k_j)$
  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 uşureze decriptarea de către NSA? NSA declară că NU.
    - Descoperirea şi folosirea unei astfel de trape de un criptanalist răuvoitor
    - Urmarea – unele detalii despre S-box au fost dezvăluite de NSA
  - Număr de iteraţii – suficiente pentru difuzie?
    - Experimental, după 8 iteraţii nu se mai văd dependenţe ale ieşirii de biţi / grupuri de biţi din intrare
  - Lungimea cheii
    - Cheie DES de 56 biţi spartă prin forţă brută (4 luni * 3500 maşini) în 1997
    - Dar, nu au fost reportate deficienţe în algoritm
    - Triple DES “măreşte” lungimea cheii

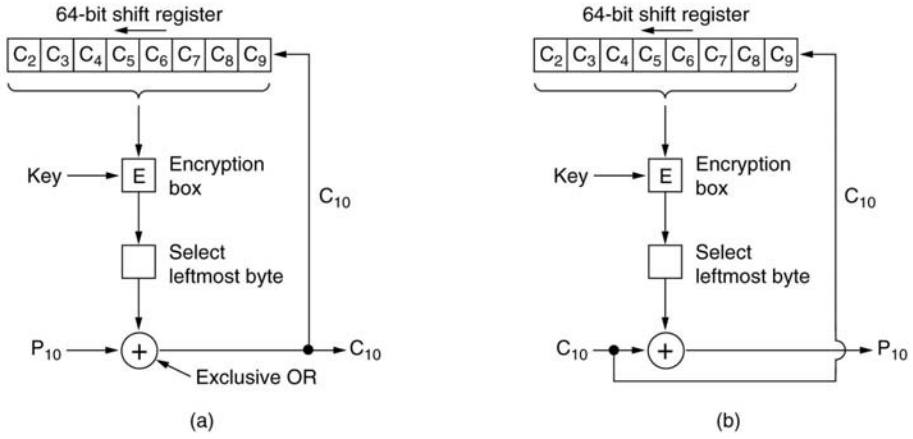
Cifrarea secvențială

Sistem secvential sincron cu reactie bloc (OFB - Output Fead Back)



Foloseste un Initialization Vector ca prima intrare in R
Nu trebuie refolosita aceeasi pereche (K, IV)

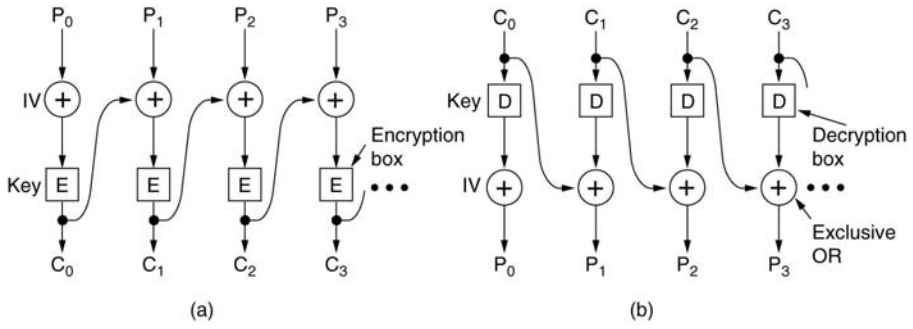
CFB - K-bit Cifer-Feed Back



Foloseste un Initialization Vector ca prima valoare in Registrul de deplasare
O eroare de un bit in criptograma conduce la decriptarea eronata a 8 octeti

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

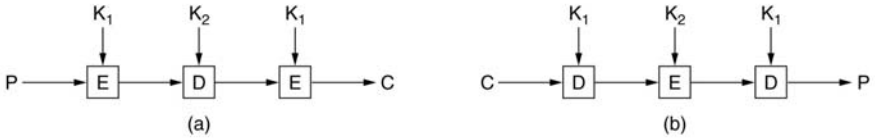
CBC - Cipher Block Chaining



Key – cheie secreta
IV – Initialization Vector
- ales aleator, acelasi pentru criptare si decriptare
- folosit pentru combinarea cu primul bloc

Avantaj: acelasi text clar repetat in mesaj va fi criptat diferit

Triplu DES



AES – Advanced Encryption Standard

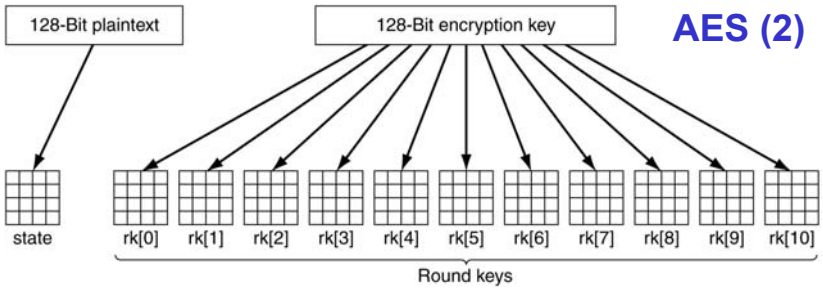
Regulile concursului organizat de NIST (ianuarie 1997) erau:

1. Algoritmul trebuie să fie un cifru bloc simetric.
2. Tot proiectul trebuie sa fie public
3. Trebuie să fie suportate chei de 128, 192, şi de 256 biţi
4. Trebuie să fie posibile atât implementări hardware cât şi software
5. Algoritmul trebuie să fie public sau oferit 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)

AES (2)



n runde (n=10 pentru cheie de lungime 128; 12/ 192, 14/256)
Bloc 128 biţi = matrice 4*4 octeţi – state
Operaţii pe coloane sau pe linii; 4 operaţii pe rundă
- substitute – la nivel octet, foloseşte tabel substituţie
- 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 înmulţite 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 definint în câmp Galois $G(2^8)$ prin polinomul $P = x^8+x^4+x^3+x+1$
- număr = coeficienţii unui polinom
- Ex. $23 = 10111_{(2)}$ este polinomul $1*x^4+0*x^3+1*x^2+1*x+1$
  $x^4+ x^2+ x+1$
- adunarea coeficienţilor făcută modulo 2
- înmulţirea făcută ca la polinoame, dar modulo P
- Ex. $(x^3+1)*(x^4+x) = x^7+x^4+x^4+x = x^7+x$

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)
  - substituţii şi mixare coloane folosesc operaţii 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
  - cunoscînd x este usor 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 usor de rezolvat.**
- problemă intratabilă - nu există un algoritm de rezolvare în timp polinomial.
- Metode
  - algoritmi exponentiali
  - problema rucsacului.

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

Algoritmi exponentiali - RSA

In RSA (Rivest, Shamir si Adleman)

Cifrarea se face prin calculul

$$\mathbf{C = (M^e) \bmod n}$$

unde **(e, n)** reprezintă cheia de cifrare.

M este un bloc de mesaj (valoare întreagă între 0 si n-1)

C este criptograma.

Descifrarea e face prin calculul

$$\mathbf{M = (C^d) \bmod n}$$

unde **(d, n)** este cheia de descifrare

RSA

1. Se aleg două numere prime *p* şi *q*, (de obicei de 1024 biţi).
2. Se calculează $n = p \times q$ şi $z = (p - 1) \times (q-1)$.
3. Se alege *d* un număr relativ prim cu *z*
   de regula, d este un numar prim mai mare ca (p-1) si (q-1)
4. Se găseşte *e* astfel încât $e \times d = 1 \bmod z$.

Ex:

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 $7^e = 1 \bmod 20$, 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

Functia lui Euler

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

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

T. Pentru n = p*q cu p, q prime

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

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 si n cu (a,n) = 1 avem

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

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

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

si un mesaj $M \in [0, n-1]$ astfel că $(M,n) = 1$, 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.

$(M^e \bmod n)^d \bmod n = M^{ed} \bmod n$
$= M^{t\,\Phi(n)+1} \bmod n$
$= M*M^{t\,\Phi(n)} \bmod n = M(M^{t\,\Phi(n)} \bmod n) \bmod n$
$= M((M^{\Phi(n)} \bmod n)^t \bmod n) \bmod n$
$= M((1)^t \bmod n) \bmod n$
$= (M.1) \bmod n = M$

Comentarii

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

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

=> RSA utilizată ptr confidentialitate si autentificare.
Nu au fost identificate atacuri reusite cu RSA

Metoda MH (Merkle si Hellman)

Problema rucsacului

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

$$C = \Sigma_{i=1,m}\ x[i] * a[i]$$

Găsirea unei solutii = backtracking
=> număr operatii care creste exponential cu m.
O solutie **x** poate fi verificată prin cel mult m operatii de adunare

Varianta rucsac simplu a problemei (trapa):

dacă A satisface proprietatea de dominantă (este o secventa supercrescatoare), adică

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

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) ⇔ secventa supercrescatoare
Solutia: aritmetica modulara
rucsac simplu A = [a1, a2, …, am]
rucsac greu G = [g1, g2, …,gm] cu gi = w * ai mod n
Ex.
rucsac simplu A = [1, 2, 4, 9], w = 15, n = 17
1*15 mod 17 = 15 mod 17 = 15
2*15 mod 17 = 30 mod 17 = 13
4*15 mod 17 = 60 mod 17 = 9
9*15 mod 17 = 135 mod 17 = 16
rucsac greu G = [15, 13, 9, 16]

Obs. inmultirea mod n strica proprietatea de dominanta

Universitatea Politehnica Bucureşti - Facultatea de Automatică şi Calculatoare

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 mod 6	x	3*x	3*x mod 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 ai
2. w si n trebuie sa fie prime intre ele (se alege n prim)
=> w are un invers multiplicativ w^{-1} ($w * w^{-1} = 1 \bmod n$)

Criptare

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

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

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

Decriptare

Receptorul cunoaste A,w, n si, bineinteles, 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

$15^{-1} = 8 \bmod 17 \Rightarrow 8 * 24 = 192 \bmod 17 = 5$

A = [1, 2, 4, 9] => P = [1, 0, 1, 0]

Comentarii

Metoda pare sigura

S-au gasit metode de atac prin ocolire rucsac greu in anumite cazuri

Oricum, algoritmul MH este greu de utilizat