Protocoale de Securitate
-
Criptologie

Cuprins

- Scopul securitatii si metode de rezolvare
- Modele criptografice cu chei simetrice si publice
- Cifrarea prin substitutie si transpozitie
- DES si AES
- RSA
- Analiza algoritmilor criptografici

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 al 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
- autorizarea (controlul accesului)
  - protectia impotriva accesului ne-autorizat
- non-repudierea
  - transmitatorul nu poate nega transmiterea unui mesaj pe care un receptor l-a primit

Metode de rezolvare

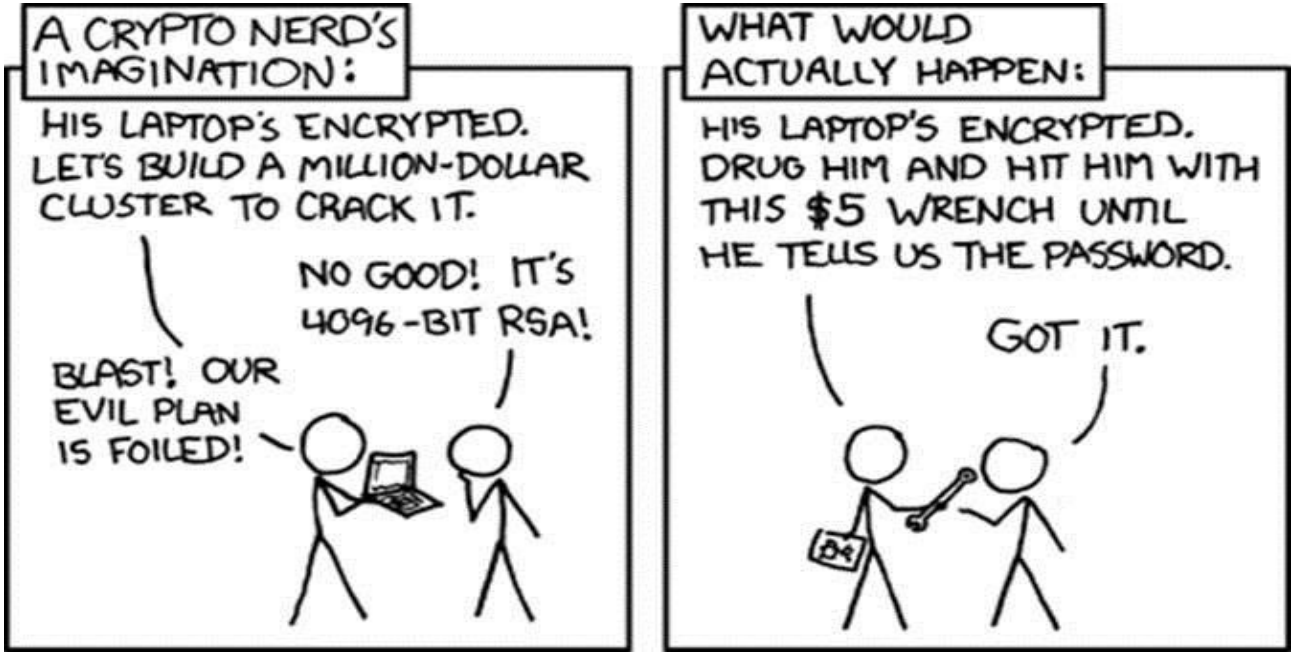
- Organizare
  - Algoritmi de criptare si hash
  - Mecanisme de securitate
    - **criptare, rezumare (hash), semnatura digitala**
  - Servicii si **protocoale** de securitate

- Securitatea in ierarhia de protocoale
  - considerata initial in nivelul prezentare al ISO OSI
  - este distribuita, in realitate, diverselor nivele
    - 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

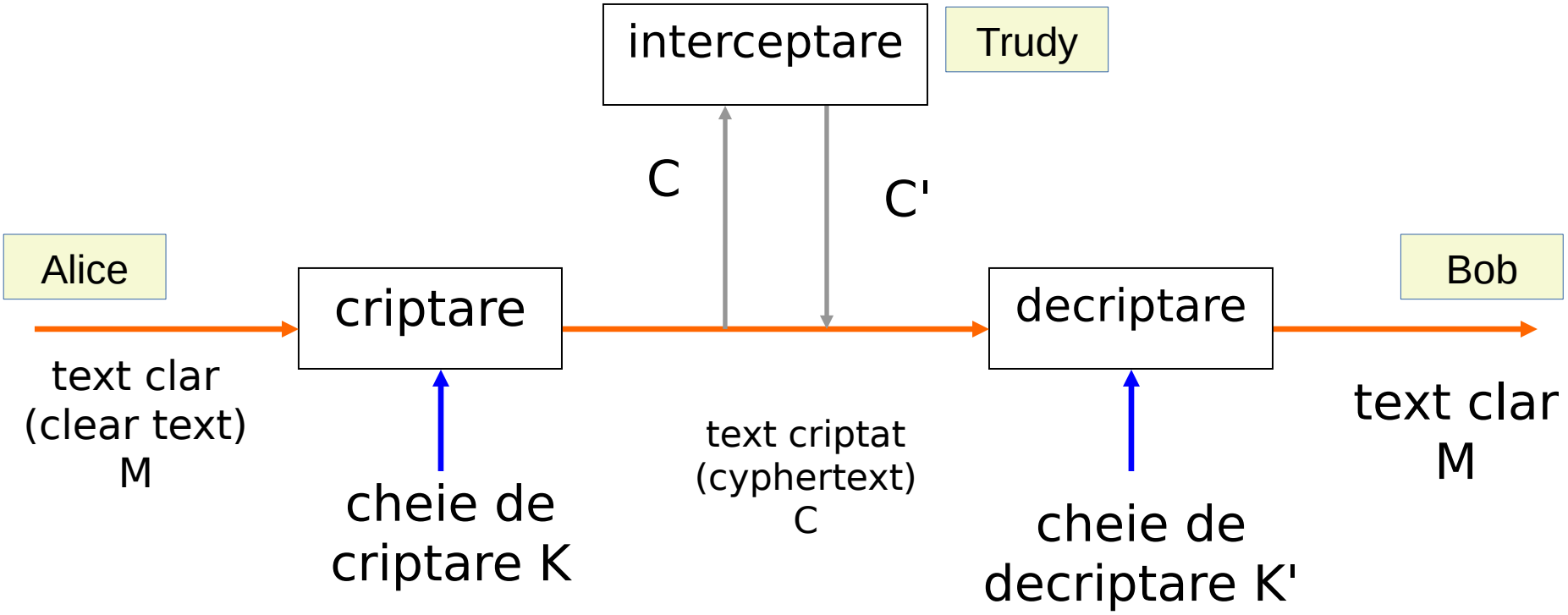
Alte aspecte

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

Elemente de criptografie



Modelul de bază al criptării



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

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

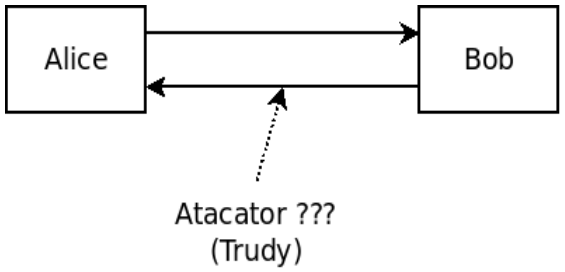
Criptografie – concepte de baza

Mecanism de baza pentru asigurarea securitatii pentru transferul datelor sau stocarea lor.

Criptologie = Criptografie + Criptanaliza

Utilizare:

- Criptarea datelor, protectie impotriva interceptarii datelor
- Comunicatie anonima
- Asigurarea integritatii datelor (functii Hash)
- Verificarea originii datelor (semnaturi digitale)
- Verificarea identitatii (certificate X.509)
- ….



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:
  - Criptarea (Encryption): C = E(k,M).
  - Decriptarea (Decryption): M = D(k’,C).
- D(k’,E(k,M))=M

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.

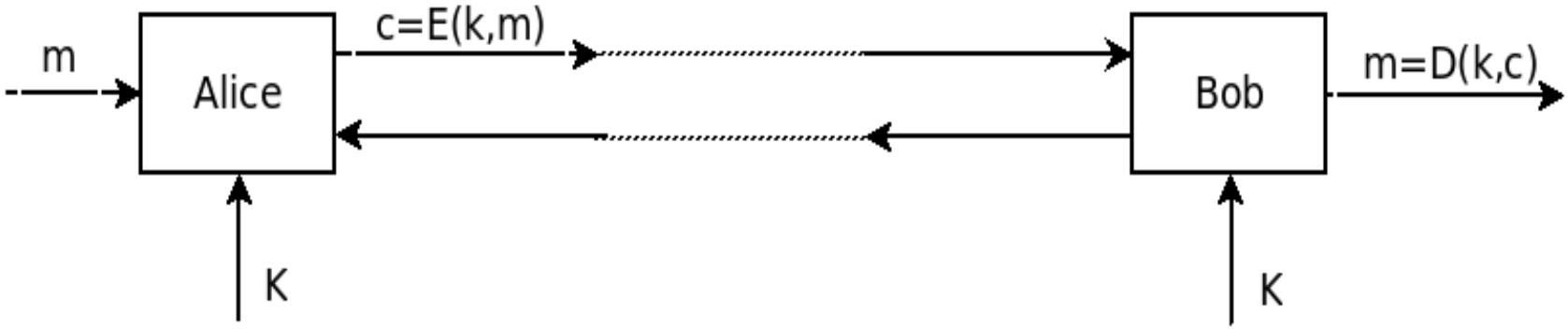
Caracteristicile sistemelor secrete

- sistem **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 de calcul disponibile.
- sistem **ideal**
  - indiferent de volumul textului cifrat care este interceptat, o criptogramă nu are o rezolvare unică, ci mai multe, cu probabilităti apropriate

Cerinţe criptosisteme cu chei secrete

- Cerinţe generale:
  - Cifrare şi descifrare eficiente pentru toate cheile.
  - Sistem uşor de folosit (gasire 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:
  - Cheia de criptare din C, chiar dacă ar cunoaşte M.
  - M din C (fără a cunoaste cheia k).
- Cerinţe specifice pentru integritate: să fie imposibil computaţional ca un criptanalist să determine sistematic:
  - Cheia de criptare k, din C, chiar dacă ar cunoaşte M.
  - Cifrul C' astfel ca D(k,C') să fie un mesaj valid (fără a cunoaşte cheia de criptare).

Criptare simetrică



Foloseste aceeasi cheie k pentru criptare si pentru decriptare

- m – mesajul în clar (clear text)
- c – mesajul criptat (cyphertext)
- Criptare: c=E(k,m)
- Decriptare: m=D(k,c)

m=D(k,E(k,m))

Vulnerabilitate: Cum ajung Alice si Bob sa cunoască valoarea K?

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).

Criptare simetrică - exemple

Algoritmi istorici	Algoritmi reali/eficienti
• Cifrul lui Cezar • Cifruri de substitutie • Substituție polialfabetică • ROT13	• Data Encryption Standard(DES) • 3DES • Advanced Encryption Standard(AES) • Foarte multi alti algoritmi: RC4, Twofish, Blowfish, IDEA, etc.

Criptare simetrică - exemple

DES:

- Dezvoltat în 1977, cheie k de 56b, blocuri de 64b, 16 runde
- Considerat nesigur din anii '90

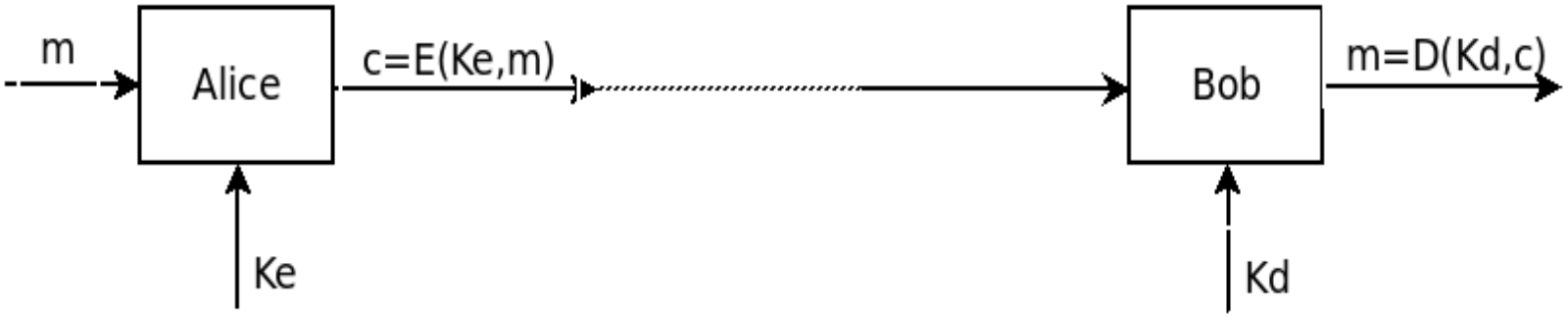
3DES:

- Rulează algoritmul DES de 3 ori, cu 3 chei K1, K2, K3 (max 168b)\
- Mare îmbunătățire a securității față de DES

AES:

- Dezvoltat în 2001
- Chei de 128b (10 runde), 196b (12 runde) sau 256b (14 runde)
- Considerat sigur, poate fi folosit pentru doc guvernamentale

Criptare asimetrică



Folosește o cheie publică penru criptare (Ke) și o cheie privată pentru decriptare (Kd), cunoscută doar de Bob.

- Ke – cheie de criptare (publică)
- Kd – cheie de decriptare (privată)
- Criptare: c=E(ke,m)
- Decriptare: m=D(kd,c)

m=D(kd,E(ke,m))

Criptare asimetrică - exemple

Exemple de algoritmi: ElGamal, RSA, etc.

RSA:

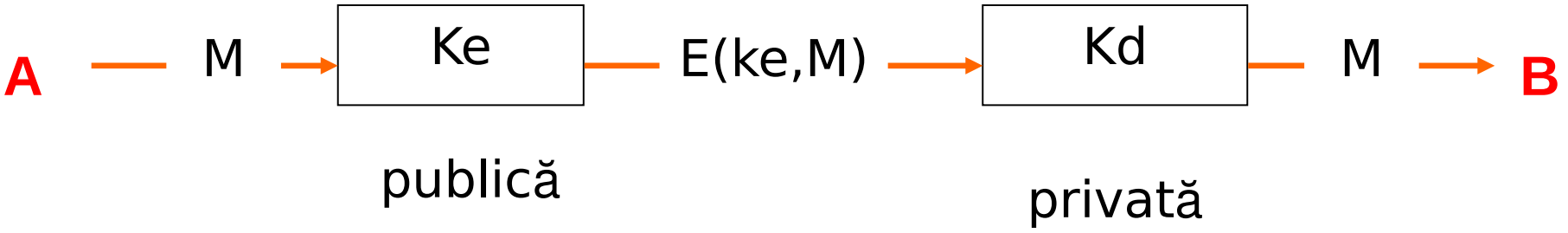
- Dezvoltat în 1977 de R. Rivest, A. Shamir, L. Adleman
- Dimensiunea cheilor: 1024b – 4096b
- Este considerat sigur pentru chei peste 2048b

Modelul criptografic cu chei publice

- Sistemele criptografice:
  - Simetrice.
  - Asimetrice:
    - Propuse de Diffie şi Hellman în 1976.
    - Chei diferite de cifrare Ke şi descifrare Kd cu proprietatea

      D(kd,E(ke,m)) =m
    - Nu se pot deduce (uşor) una din alta, mai precis:
      - Este extrem de greu să se deducă Kd din Ke;
      - D nu poate fi "spart" prin criptanaliză cu text clar ales.

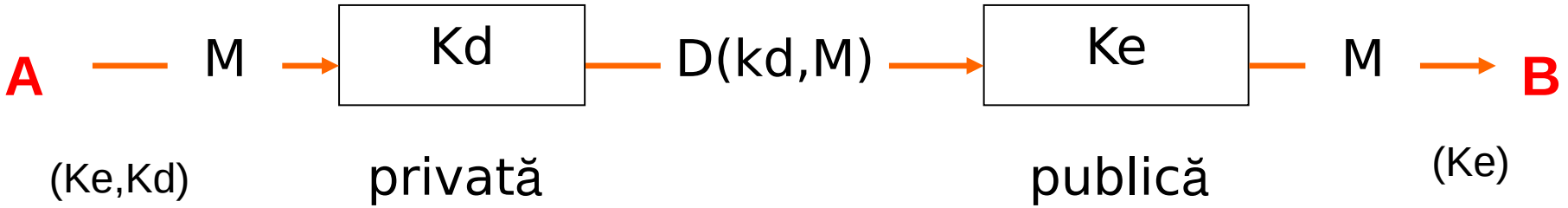
Schema de confidențialitate



- Într-un sistem asimetric, un utilizator B:
  - Face publică cheia (transformarea) E_b de cifrare.
  - Păstrează secretă cheia (transformarea) D_b de descifrare.

- Se asigură confidentialitatea
  - doar B, care are cheia privată Db poate intelege mesajul M

Schema de integritate (semnatura digitala)



- Pentru integritate / autentificare:

condiţia necesară este ca transformările E şi D să comute, adică

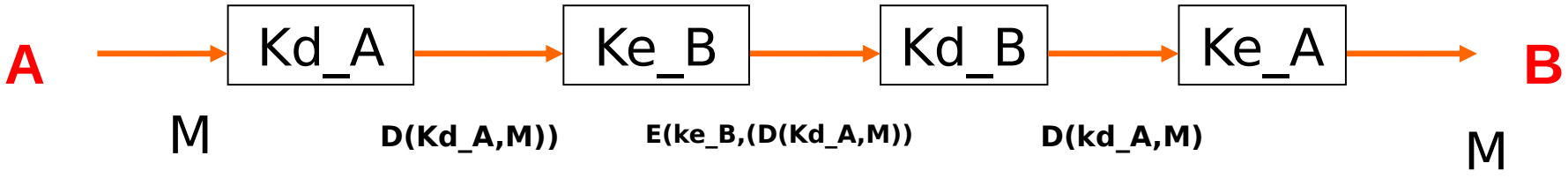
$$E(ke,D(kd,M)) = D(kd,E(ke,M)) = M.$$

Cheia Kd se foloseste pentru criptarea mesajului M

Se asigură integritatea

- Oricine poate folosi cheia publica Ke pentru a decripta mesajul
- Nimeni nu poate modifica mesajul M deoarece nu cunoaste cheia privata kd

Schema de autentificare



autentificare M este criptat mai intai cu cheia privata a lui A

- D(Kd_A,M) este un fel de “semnatura” a mesajului

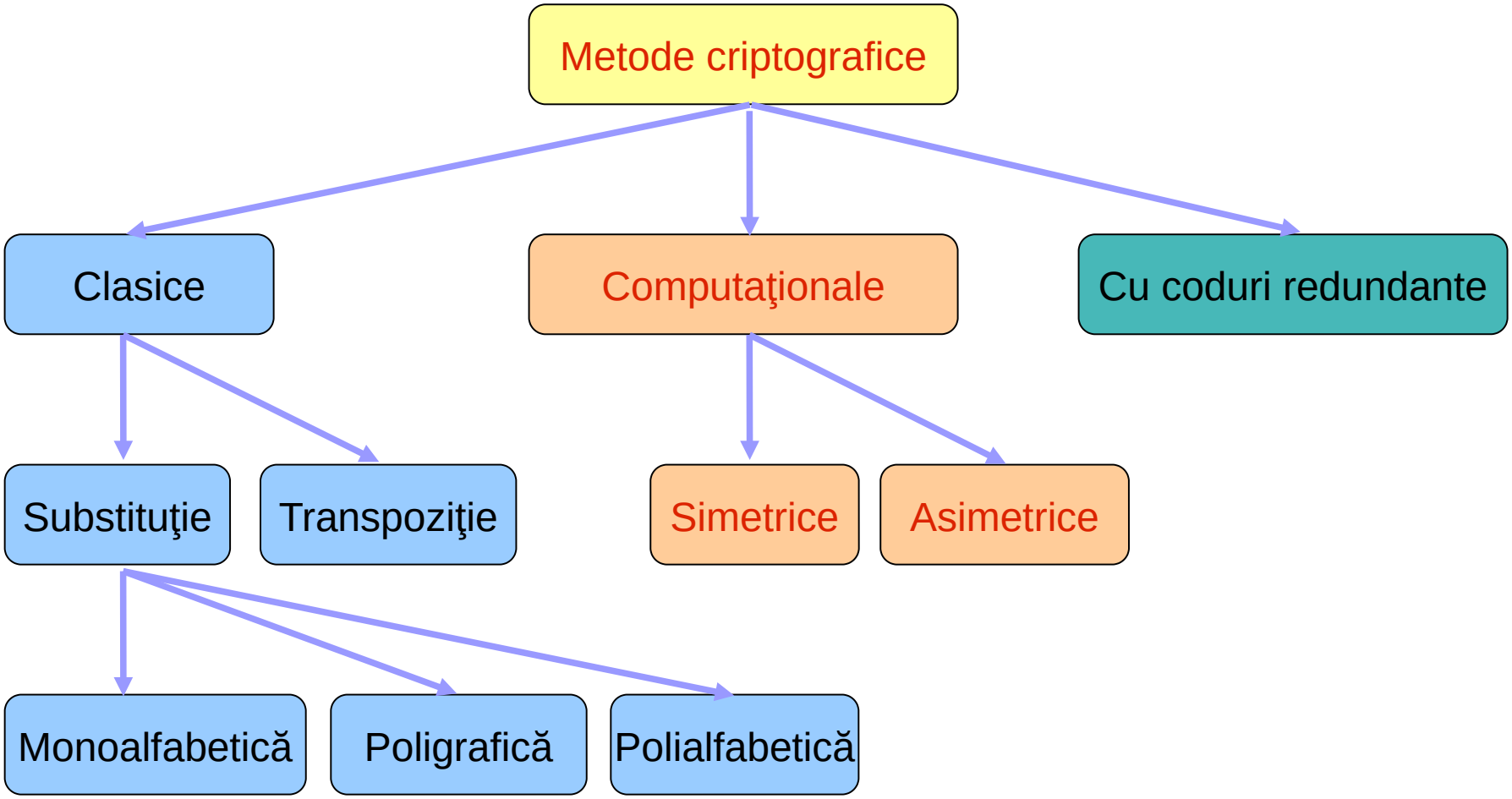
rezultatul este apoi criptat cu cheia publica a lui B

Se asigura că A este sursa mesajului

și că mesajul este confidențial

non-repudiere folosind perechea Da(M) si M, B poate demonstra ca a primit mesajul de la A

Clasificare generală



Algoritmi de criptare simetrica

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**

fiecare litera este inlocuita de litera aflata la distanta 3 de ea

Relatia de calcul

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

In general

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

- **ROT-13** este o alta varianta de substitutie monoalfabetica in care pentru criptare se realizeaza o rotatie cu 13 pozitii.

Substitutia polialfabetică (Vigenere)

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

fiecare litera din cheie = substitutul literei A din textul clar

Exemplu: cheia POLIGRAF

POLIGRAFPOLIGRAGPOLIGRAFPOLIGRAFPOLI	*cheie*
AFOSTODATACANPOVESTIAFOSTCANICIODATA	*clar*
PTZAZFDFIONITGOATGEQGWOXIQLVOTITSOEI	*cifrat*

Litera O din cheie substituie A din textul clar;
Litera F (situata la 5 pozitii de A) este inlocuita de T (aflata la 5 pozitii de O)

Cifrarea prin transpozitie

Modifică ordinea caracterelor. Uzual:

- textul clar este 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 sunt citite pe coloane,
- ordinea coloanelor este dată de ordinea alfabetică a literelor unei chei.

`cheie:` `POLIGRAF`

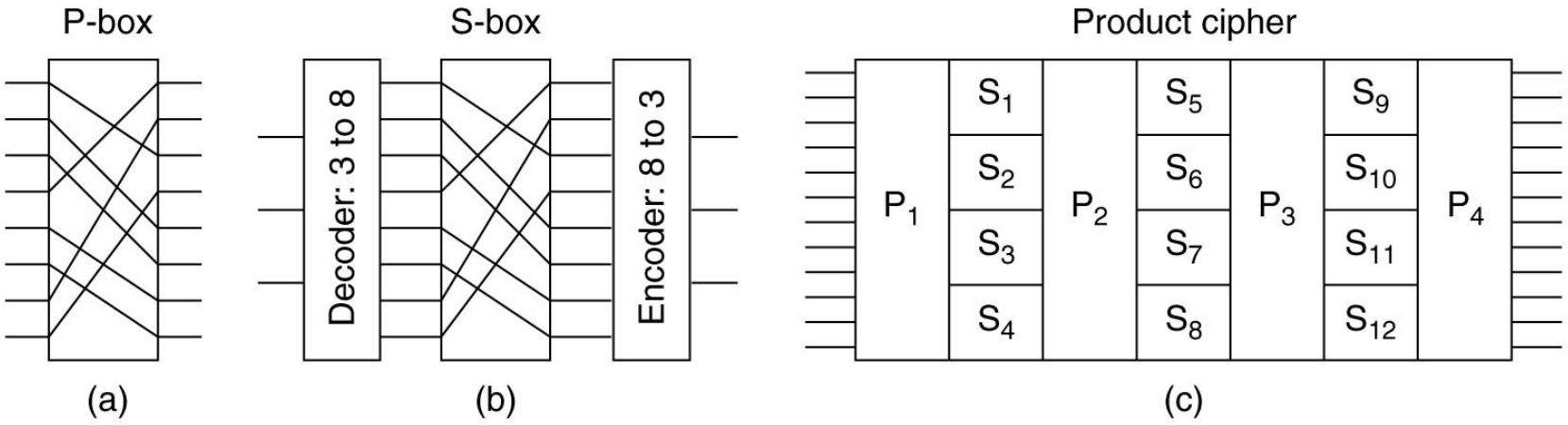
`ordine:` `76543812`

text clar: AFOSTODATACANPOVESTIAFOSTCANICIO

POLIGRAF
AFOSTODA
TACANPOV
ESTIAFOS
TCANICIO

text cifrat: DOOIAVSOTNAISAINOCTAFASCATETOPFC

Cifruri produs

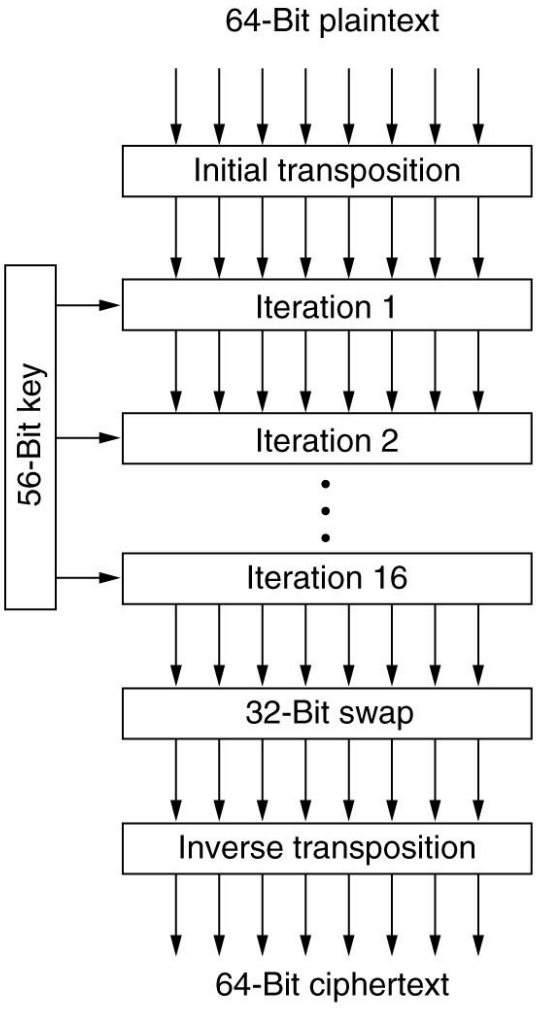


Principii pentru a obţine o securitate mai mare:

- compune două cifruri "slabe", complementare
  - P-box – permutare (transpozitie) - asigură difuzia
  - S-box – substitutie - asigură confuzia
- repetă aplicarea permutării şi substituţiei

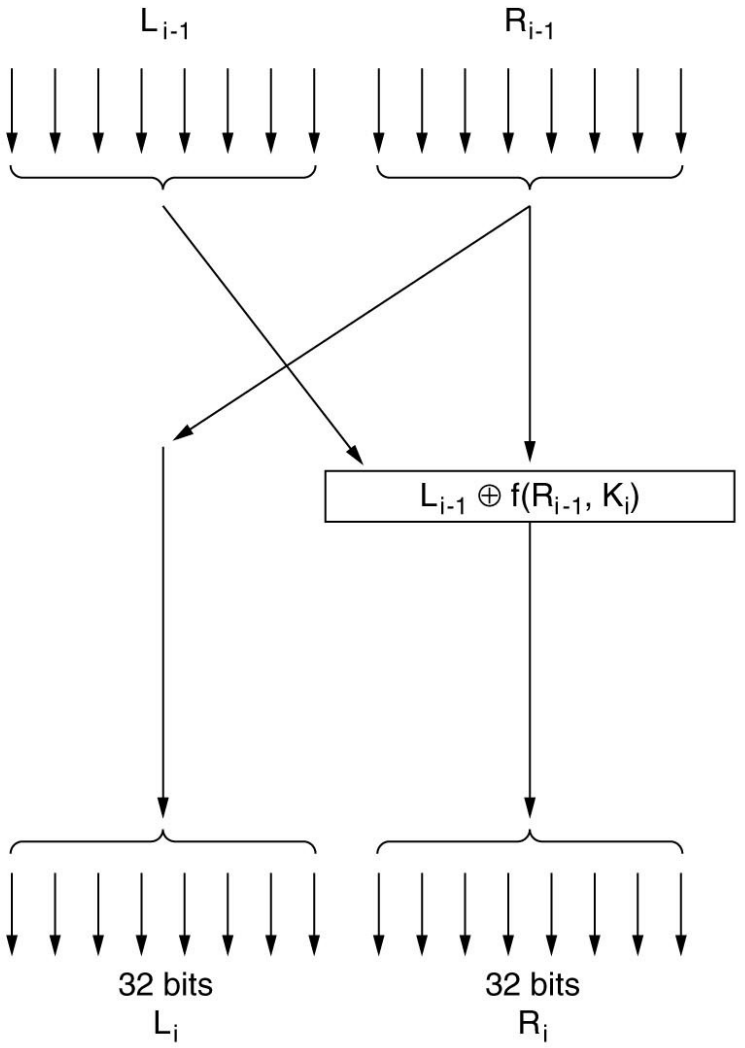
DES (Data Encryption Standard)

Schema generală



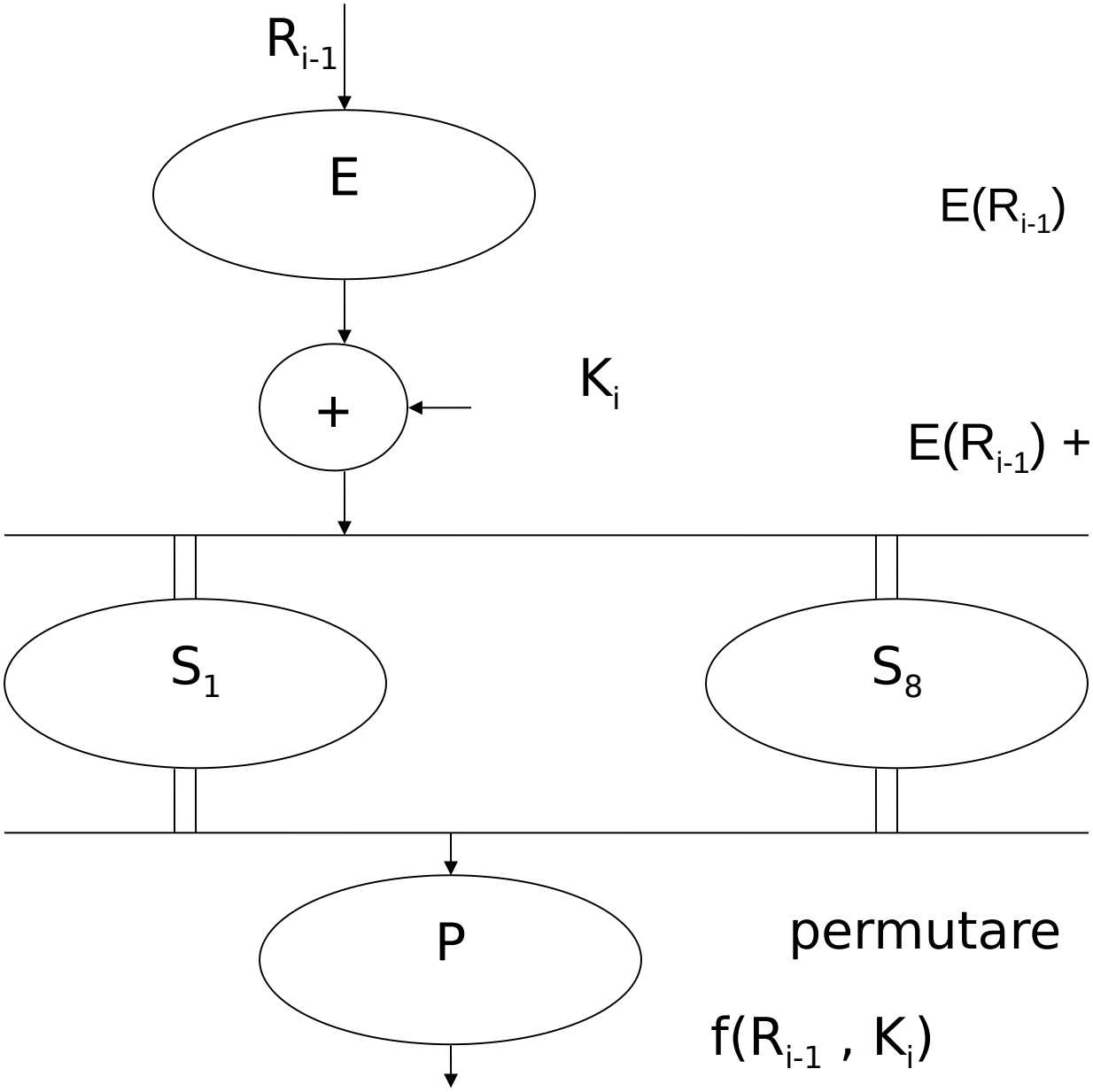
(a)

O iteraţie



(b)

Calculul lui $f(R_{i-1},k_i)$



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

$S_j(B_j)$

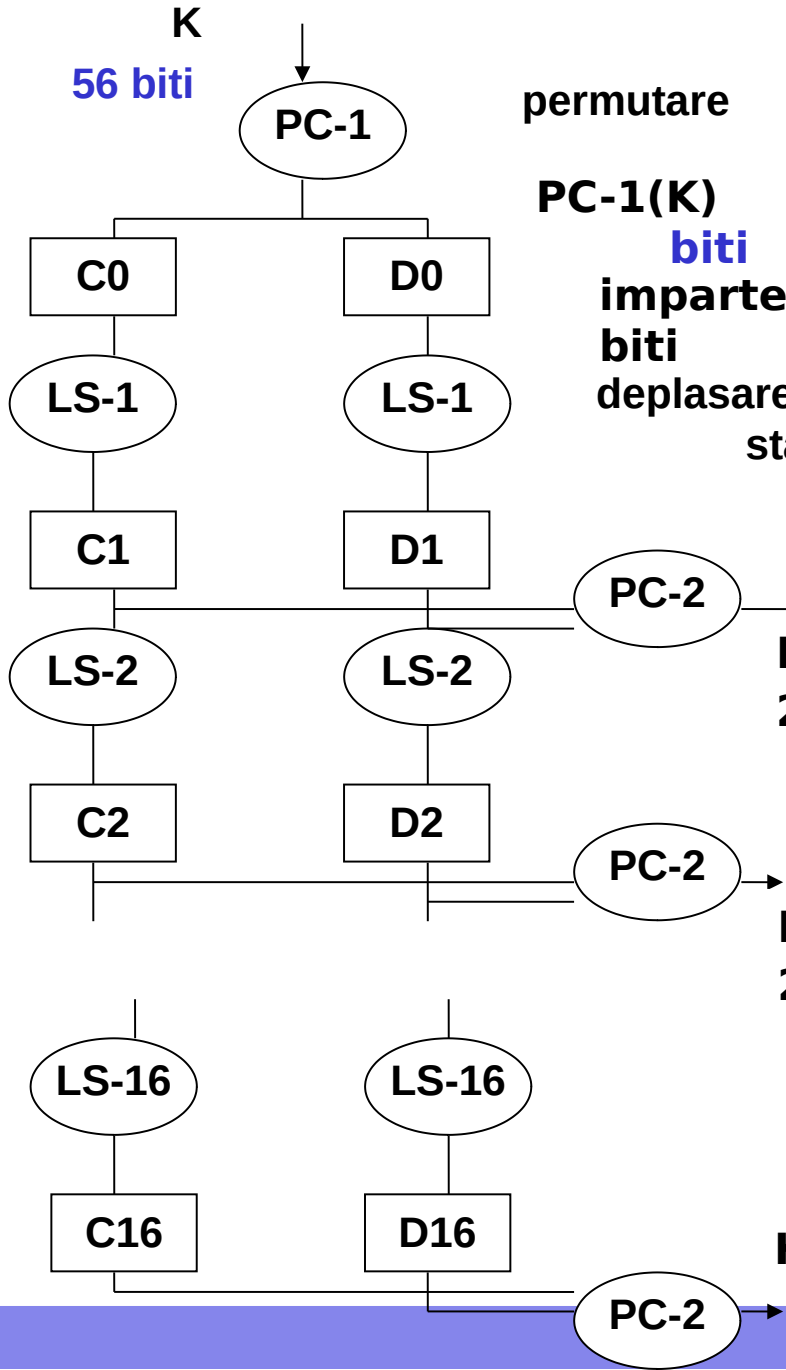
Fiecare Sj produce un cod de 4 biti

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

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

32 biti

Calculul cheilor



permutare

PC-1(K) 56 biti

imparte 56 biti in 2 blocuri x 28 biti

deplasare circulara la stânga

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

K_1=PC-2(C1D1)

extrage si permuta 48 biti din cei 56 ai C1D1

K_2=PC-2(C2D2)

K_{16}=PC-2(C16D16)

Comentarii

- Elementele cheie ale algoritmului nu au fost făcute publice
  - Controverse
    - Există trape (capcane) care să uşureze decriptarea de către NSA?

      NSA declară că NU.
    - Descoperirea şi folosirea unei astfel de trape de un criptanalist răuvoitor

 unele detalii despre S-box au fost dezvăluite de NSA

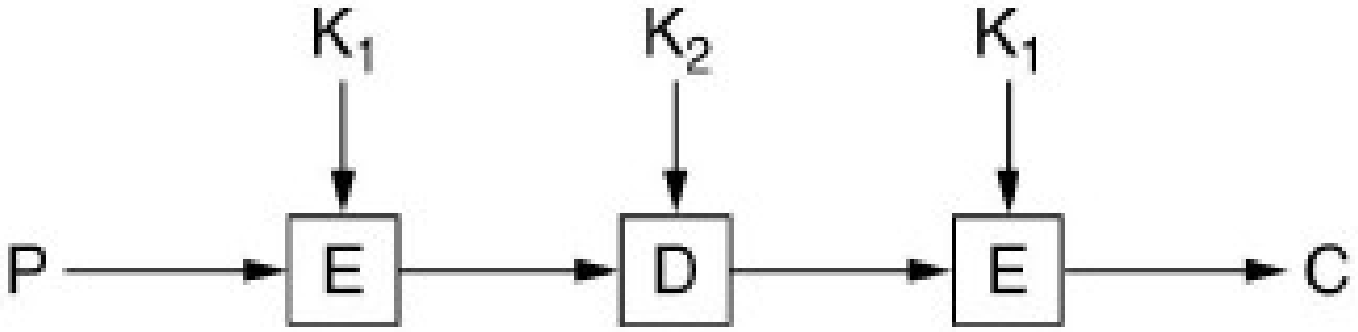
Comentarii

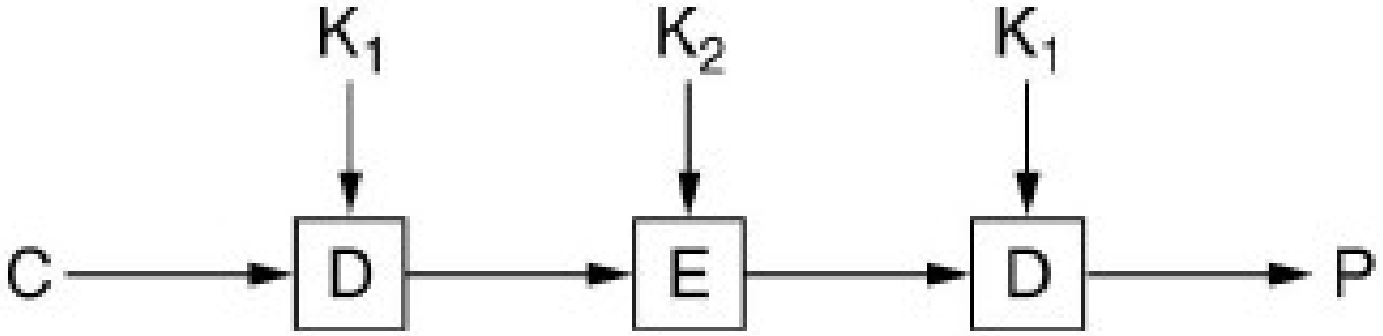
- Număr de iteraţii (16) sunt suficiente pentru difuzie?
  - Experimental, după 8 iteraţii nu se mai văd dependenţe ale biţilor de ieşire de grupuri de biţi din intrare
- Lungimea cheii
  - Cheie DES de 56 biţi spartă prin forţă brută (4 luni * 3500 maşini) în 1997
  - Dar, nu au fost raportate deficienţe în algoritm
  - Triple DES “măreşte” lungimea cheii

Triplu DES

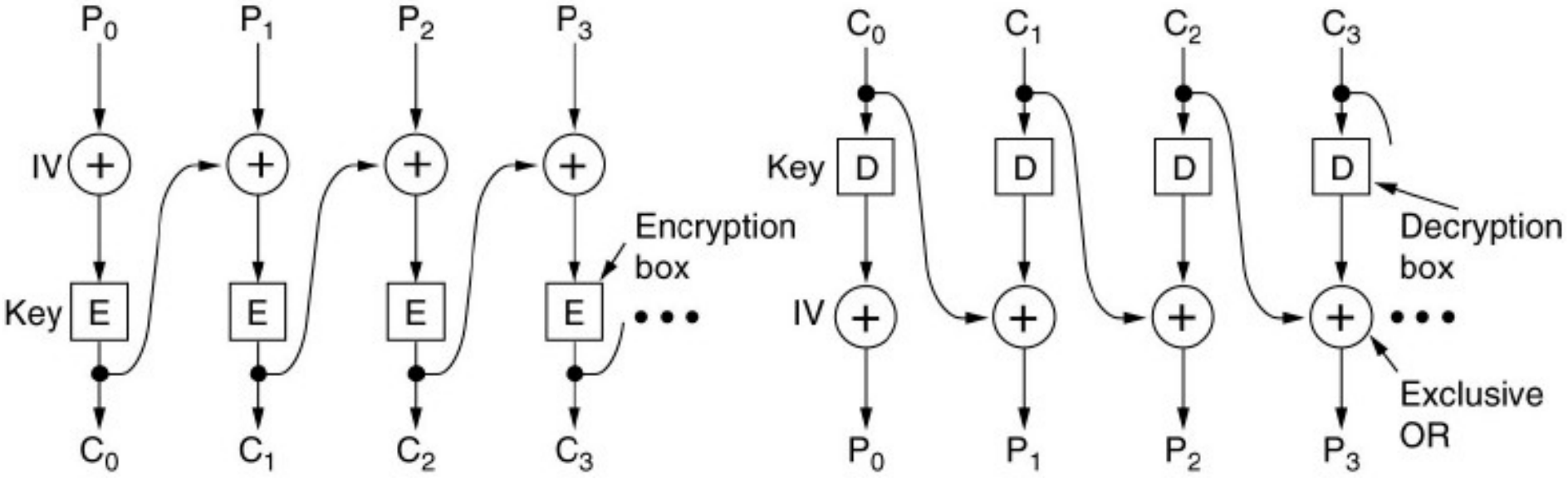
"Creste" lungimea cheii folosind

- 2 chei
- 3 runde de criptare / decriptare





Inlantuirea blocurilor cifrate: CBC - Cipher Block Chaining



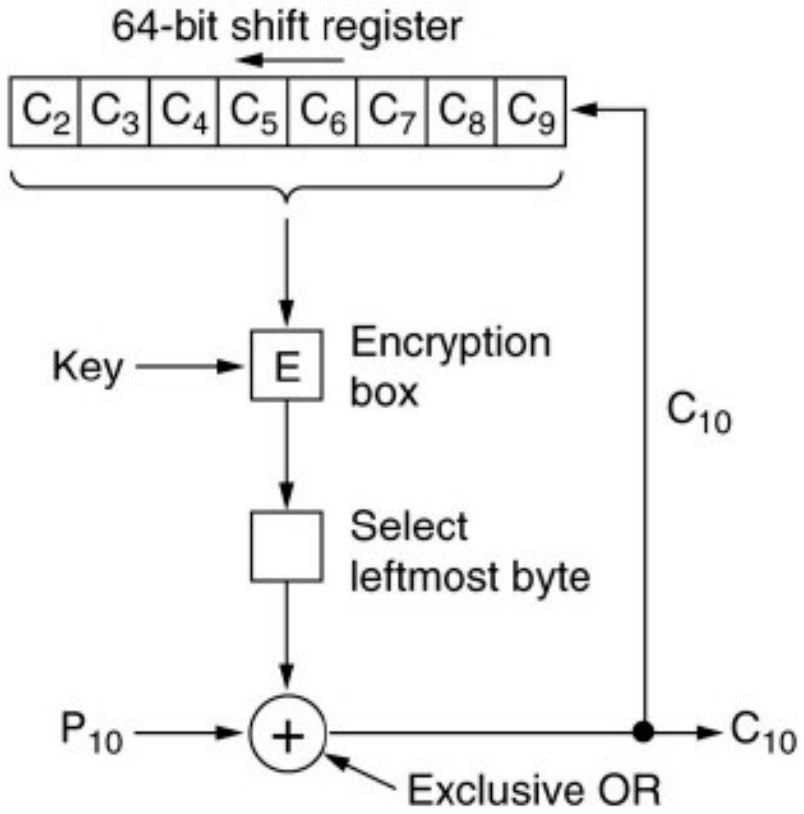
Key – cheie secreta

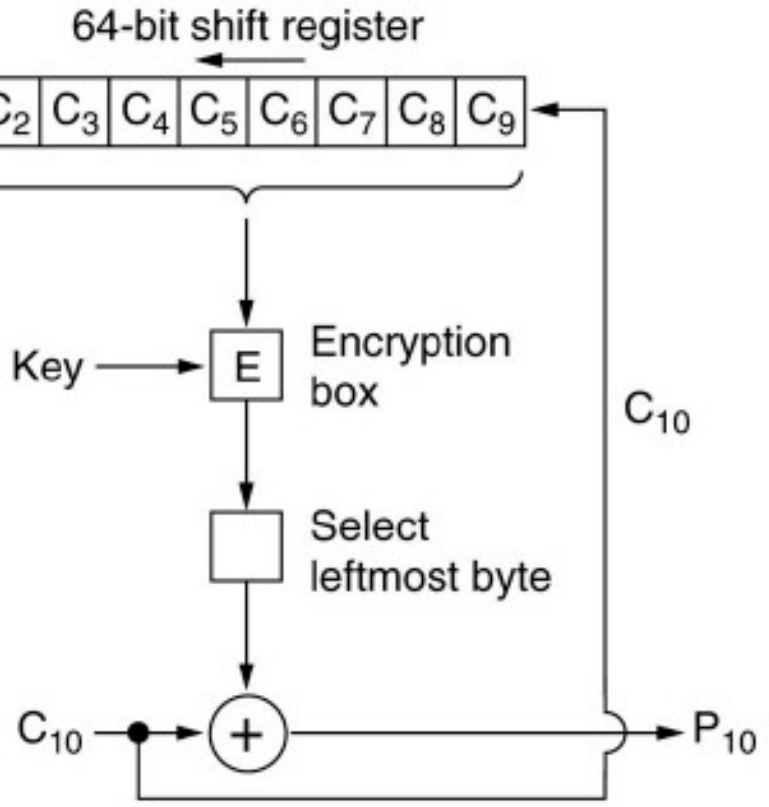
IV – Initialization Vector

- ales aleator, acelasi pentru criptare si decriptare
- folosit pentru combinarea cu primul bloc de text clar

Urmarea: un acelasi text clar repetat in mesaj va fi criptat diferit

Reactie cifrata: CFB - Cipher-Feed Back



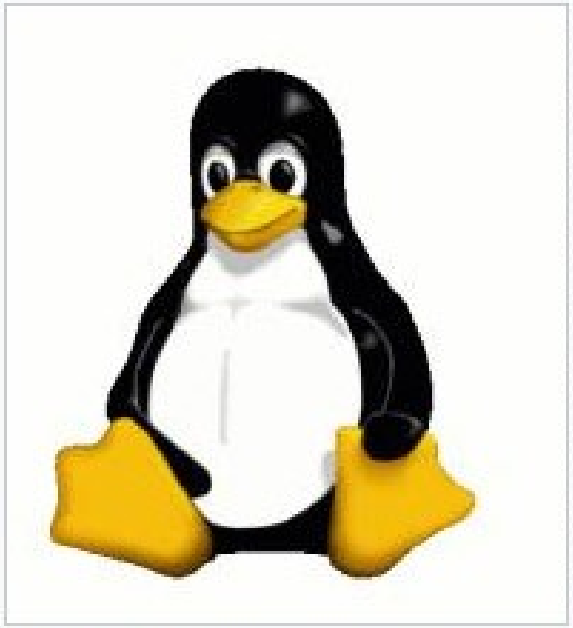


Foloseste un Initialization Vector ca prima valoare in Registrul de deplasare

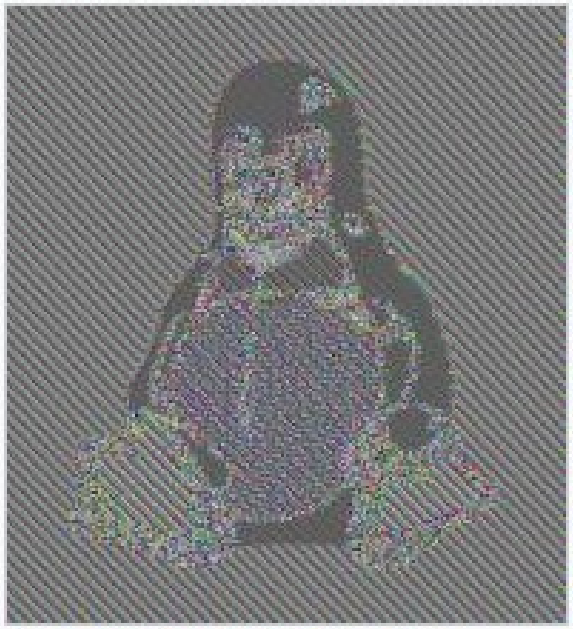
O eroare de un bit in criptograma conduce la decriptarea eronata a 8 octeti

- generati in pasii in care bitul se afla in registrul de deplasare

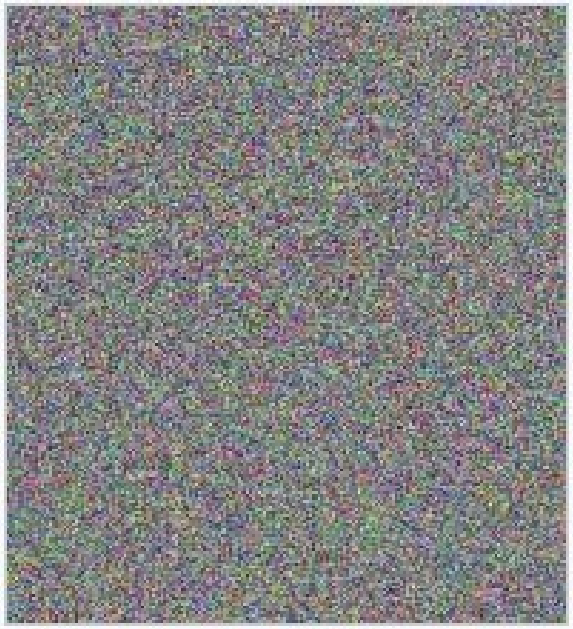
Exemplu cifrare CBC



Original image



Encrypted using ECB mode

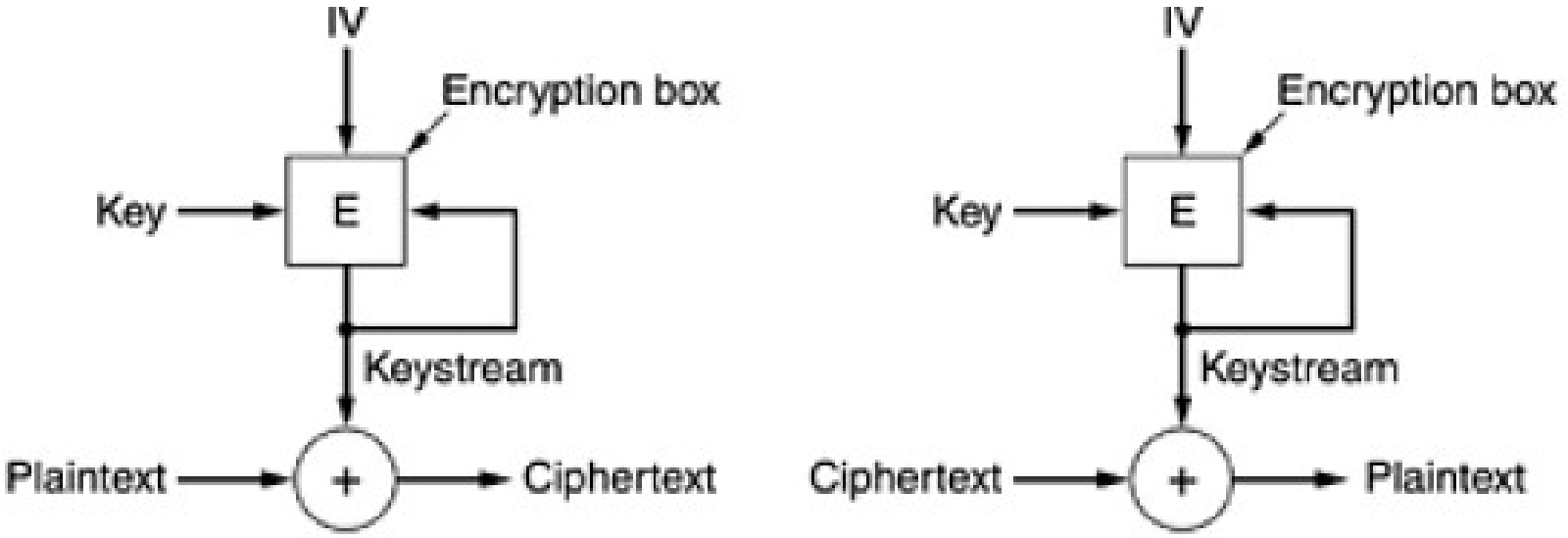


Modes other than ECB result in pseudo-randomness

The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".

Source:
https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Cifrarea secvențială (Stream Cipher)



Foloseste un Initialization Vector ca intrare DES pentru a genera prima cheie de criptare/decritare, care este folosita ca intrare DES pentru a genera a doua cheie etc.

Sirul de chei generat este combinat XOR cu textul clar pentru a produce criptograma

Sirul de chei depinde doar de cheia principala Key si de vectorul de initializare IV

 Nu trebuie refolosita repetat aceeasi pereche (Key, IV)

AES – Advanced Encryption Standard

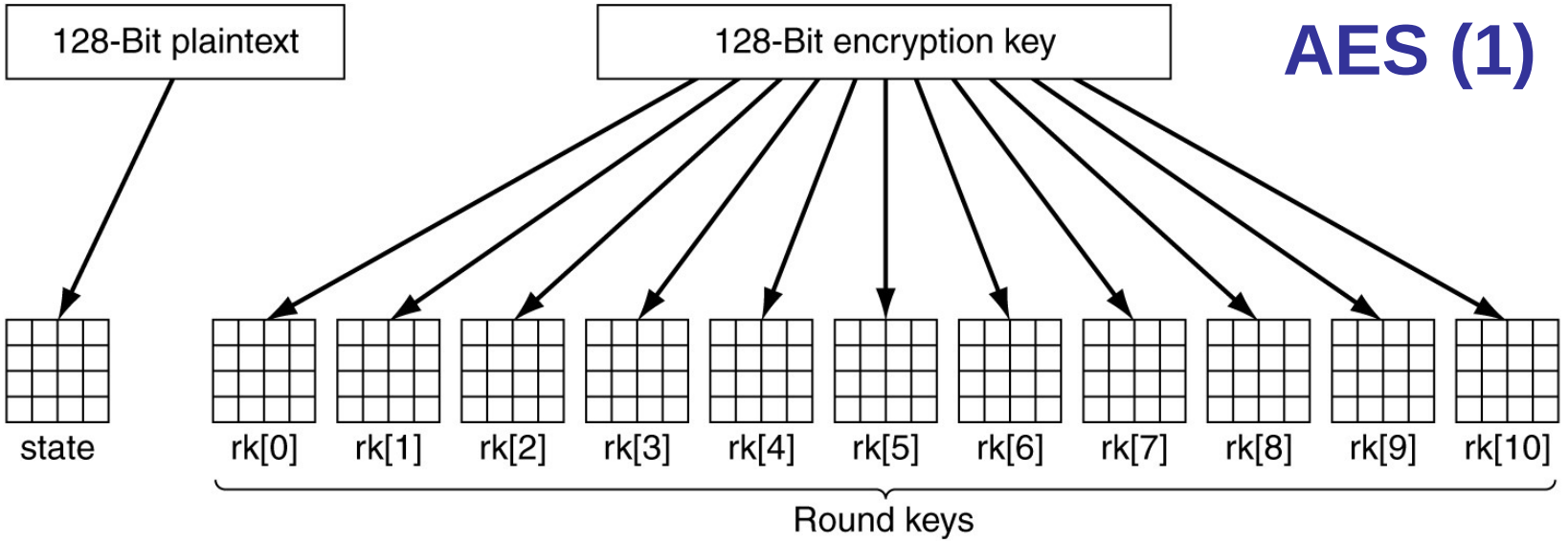
Regulile concursului organizat de NIST (ianuarie 1997) erau:

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 (1)



Foloseste o matrice de stare de 4*4 octeţi (bloc de 128 biţi)

si mai multe chei de runda fabricate din cheia de baza

Numarul de runde **n** depinde de lungimea cheii

n=10 pentru cheie de lungime 128; 12/ 192, 14/ 256

Initial

- se copiaza un bloc de 128 biti text clar in state
- se face XOR intre state si cheia rk[0]

AES (2)

In fiecare runda se fac patru operatii

substitutie – la nivel octet, foloseşte tabel substituţie
rotate_rows – prin deplasare circulară la stânga la nivel octet

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

1	5	9	13
6	10	14	2
11	15	3	7
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 – combinare cu o cheie de rundă rk[i]

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

număr = coeficienţii unui polinom

Ex. $23_{(10)} = 10111_{(2)}$ 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)

Algoritmi de criptare asimetrica

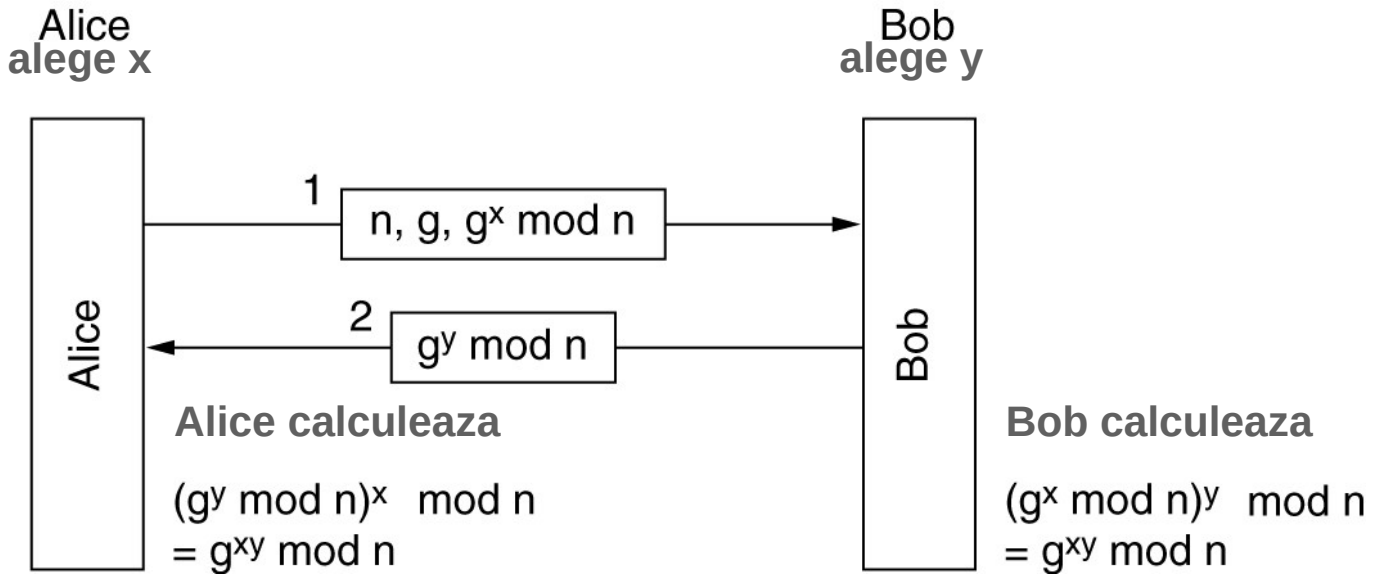
Cifrarea prin functii greu inversabile

- 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
    - problemă intratabilă - nu există un algoritm de rezolvare în timp polinomial.
  - destinatarul autorizat
    - **are cheia sau**
    - **dispune de o trapă ce face problema usor de rezolvat.**

- Metode
  - algoritmi exponentiali
  - problema rucsacului.

Stabilire cheie partajata: Diffie-Hellman key exchange



n, g – numere mari
n prim
(n-1)/2 prim

x nu poate fi calculat din $g^x \bmod n$
$g^{xy} \bmod n$ nu poate fi calculat din $g^x \bmod n$ si $g^y \bmod n$ cand n este mare

g < n (generator) are proprietatea: orice p poate fi scris ca $g^k \bmod n$

adica: pentru fiecare *p* intre 1 si *n*-1 inclusiv, exista o putere *k* a lui *g* astfel ca $p = g^k \bmod n$.

Algoritmi exponentiali – RSA

In RSA (Rivest, Shamir si Adleman):

Criptarea si decriptarea se fac prin functii exponentiale
Criptarea se face prin calculul

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

unde **(e, n)** reprezintă cheia de criptare.
M este un bloc de mesaj (valoare întreagă între 0 si n-1)
C este criptograma.

Decriptarea se face prin calculul

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

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

Algoritmi exponentiali – RSA

Conditia: functiile de criptare si decriptare trebuie sa fie inverse una alteia:

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

Conditia poate fi indeplinita daca

- e este un intreg relativ prim cu $\Phi(n)$

$\Phi(n)$ este Functia lui Euler
adica nr de întregi pozitivi <n relativ primi cu n
d este inversul multiplicativ al lui e modulo $\Phi(n)$

$$e*d \bmod \Phi(n) = 1$$

- n este produsul a doua numere prime, $n = p * q$

caz in care $\Phi(n) = (p-1)(q-1)$

Motivatie

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

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

daca $n = p*q$ cu p, q prime atunci

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

Teorema (Euler). Pentru orice a si n cu $(a,n) = 1$ avem

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

Aceasta proprietate este folosita in demonstrarea urmatoarei

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

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

si un mesaj $M \in [0, n-1]$, avem

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

Metoda RSA

1. Se aleg două numere prime *p* şi *q*,

   (de ex. de 1024 biţi).

2. Se calculează $n = p \xi q$

şi $z = (p - 1)\xi(q - 1)$.

3. Se alege *d* un număr relativ prim cu *z*

d poate fi un numar prim care satisface

d > (p-1) si d > (q-1)

4. Se găseşte *e* astfel încât $e \xi d = 1 \bmod z$.

5. (e, n) este cheia de criptare.

(d, n) este cheia de decriptare.

Comentarii

Cheia de criptare (e,n) se face publica

Problema:

- cunoasterea lui (e,n) sa nu permita deducerea lui d

Securitate pastrata deoarece:

- p si q sunt numere prime foarte mari
- p si q sunt pastrate secrete

Cifrarea si descifrarea sunt comutative si mutual inverse

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

- RSA utilizată ptr confidentialitate si autentificare.

Nu au fost identificate atacuri reusite cu RSA

Rezumatele mesajelor

Rezumatele mesajelor

Este mai usor sa semnezi digital rezumatul decat mesajul intreg !

Folosite in semnaturi digitale datorita proprietatilor utile

1. Rezumatul lui P - MD(P) - este usor de calculat
2. Este imposibil sa se afle P din MD(P)
3. Rezumatul nu poate trucat: nimeni nu poate gasi P' avand un rezumat identic cu P, adica MD(P') = MD(P)
4. O schimbare de 1 bit a intrarii schimba multi biti din iesire

Functii hash

- MD5 (Message Digest)
- SHA-1 (Secure Hash Algorithm)

Functii Hash: MD5 - Message Digest 5

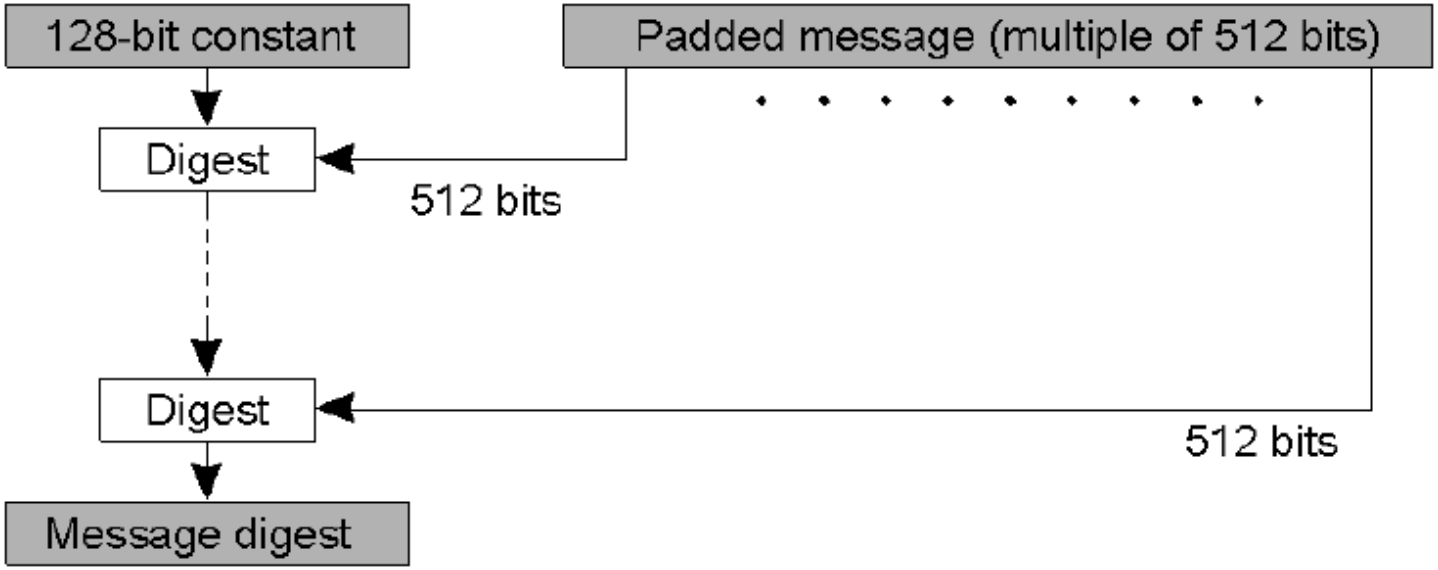
MD5 calculeaza un rezumat de 128 biti dintr-un mesaj de lungime multipla de 512 biti

Mesajul este completat cu biti pentru a respecta regula

Ultimii 64 biti precizeaza lungimea originala a mesajului

In fiecare faza algoritmul calculeaza un nou rezumat din rezumatul anterior si rezumatul unui bloc de 512 biti.

Primul rezumat este o constanta de 128 biti



Functii Hash: MD5 (2)

O **faza** transforma un **bloc** de mesaj de 512 biti. Are 4 runde.

Fiecare **runda** foloseste o functie diferita:

F(x,y,z) = (x AND y) OR ((NOT x) AND z)
G(x,y,z) = (x AND z) OR (y AND (NOT z))
H(x,y,z) = x XOR y XOR z
I(x,y,z) = y XOR (x OR (NOT z))

O runda are 16 **iteratii**.

$b_0,\ldots,b_{15}$ – **sub-blocuri** 32-biti (total 512 biti)

p, q, r, s – variabile *digest*

$C_1, \ldots, C_{16}$ – constante (in total 64)

<<< denota rotatie stanga

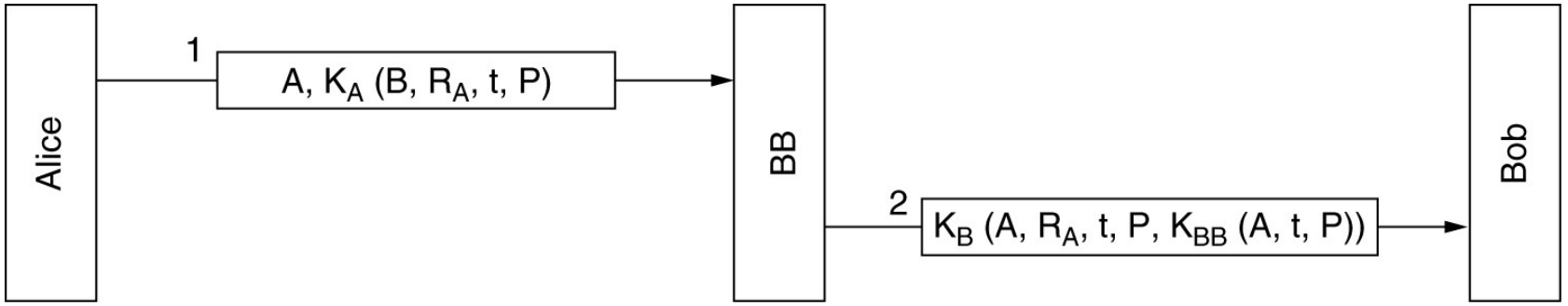
Iterations 1-8	Iterations 9-16
$p \leftarrow (p + F(q,r,s) + b_0 + C_1) \lll 7$	$p \leftarrow (p + F(q,r,s) + b_8 + C_9) \lll 7$
$s \leftarrow (s + F(p,q,r) + b_1 + C_2) \lll 12$	$s \leftarrow (s + F(p,q,r) + b_9 + C_{10}) \lll 12$
$r \leftarrow (r + F(s,p,q) + b_2 + C_3) \lll 17$	$r \leftarrow (r + F(s,p,q) + b_{10} + C_{11}) \lll 17$
$q \leftarrow (q + F(r,s,p) + b_3 + C_4) \lll 22$	$q \leftarrow (q + F(r,s,p) + b_{11} + C_{12}) \lll 22$
$p \leftarrow (p + F(q,r,s) + b_4 + C_5) \lll 7$	$p \leftarrow (p + F(q,r,s) + b_{12} + C_{13}) \lll 7$
$s \leftarrow (s + F(p,q,r) + b_5 + C_6) \lll 12$	$s \leftarrow (s + F(p,q,r) + b_{13} + C_{14}) \lll 12$
$r \leftarrow (r + F(s,p,q) + b_6 + C_7) \lll 17$	$r \leftarrow (r + F(s,p,q) + b_{14} + C_{15}) \lll 17$
$q \leftarrow (q + F(r,s,p) + b_7 + C_8) \lll 22$	$q \leftarrow (q + F(r,s,p) + b_{15} + C_{16}) \lll 22$

Comentarii

- Rezistenta la coliziuni
  - o functie H este rezistenta la coliziuni daca este foarte greu sa se gaseasca a si b, a!=b astfel ca H(a) = H(b)
- In 2004 s-a aratat ca MD5 nu este rezistent la coliziuni
- S-au dezvoltat si recomandat alte functii de hash
  - SHA1, SHA2
- Obs.
  - criptare != rezumare!

Semnaturi digitale

Semnaturi cu chei simetrice



Semnaturi digitale cu Big Brother.

- R_A – numar aleator (control replici)
- t – timestamp (mesaj recent)
- K_A – cheie secreta a lui A (cunoscuta de BB)
- K_B – cheie secreta a lui B (cunoscuta de BB)
- K_{BB} – cheie secreta Big Brother (doar el o cunoaste)

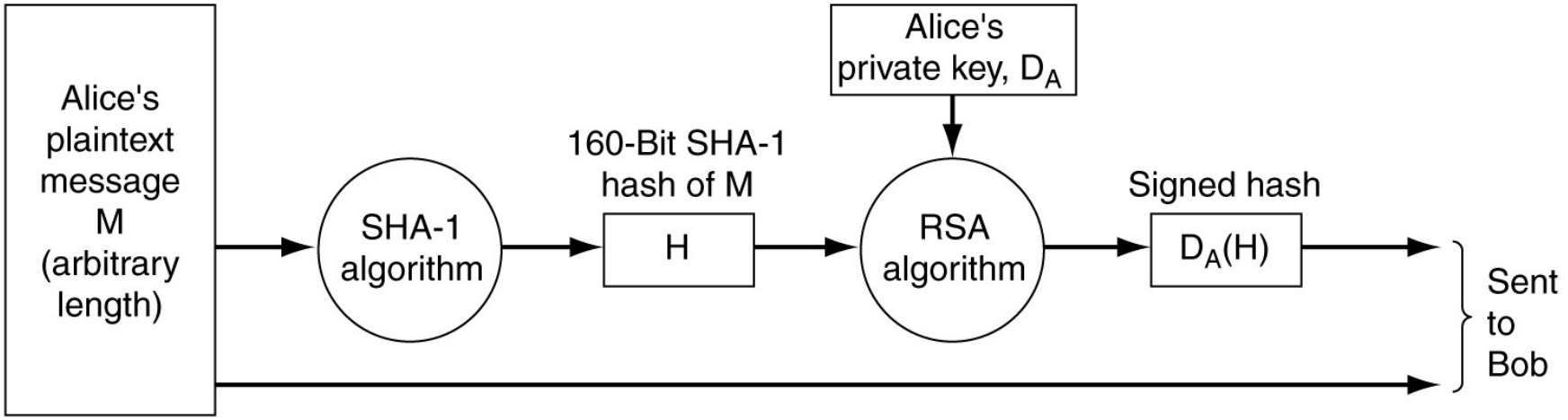
Comentarii

t si R_A folosite ptr. detectie atacuri prin replicare mesaje mai vechi

K_{BB} (A, t, P) folosit pentru non-repudiere

Semnaturi cu chei publice

Utilizarea SHA-1 si RSA pentru semnarea mesajelor nesecrete.

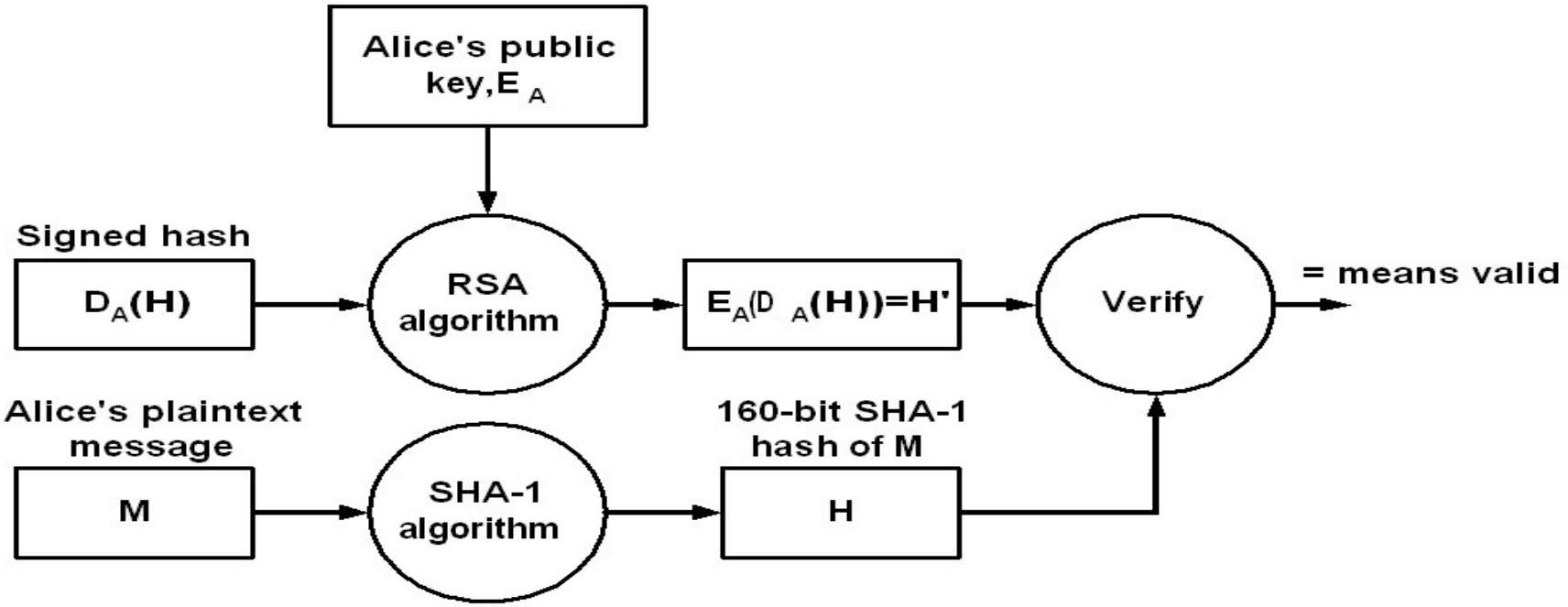


Caracteristici

Rezumatul SHA-1 este semnat cu cheia secreta a transmitatorului D_A

Mesajul M este transmis in clar

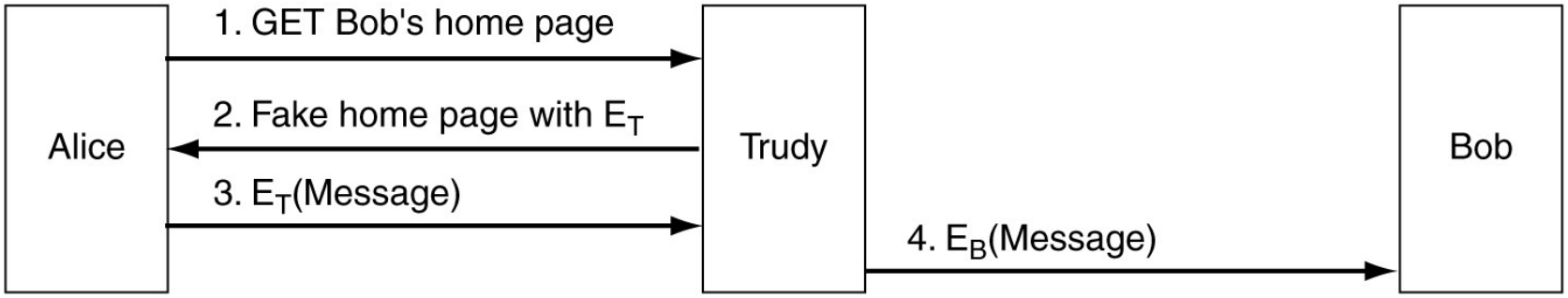
Verificare semnatura digitala



Orice modificare a textului clar M este detectata prin H<>H'

Un intrus nu poate modifica si M si rezumatul criptat $D_A(H)$

Probleme cu difuzarea cheilor publice



Problema: difuzarea cheii publice prin pagina de referinta a proprietarului

- Trudy raspunde in locul lui Bob cu cheia sa publica
- Trudy poate modifica mesajele trimise de Alice lui Bob
  - Man-in-the-middle attack

Certificate digitale

Certificate de securitate

- Certificate
  - Asociaza identitatea cu cheia publica
- X.509
  - Standard de certificate

Certificate

Rol: leaga cheia publica de un proprietar (principal) sau de un atribut.
Un certificat permite autentificarea unei entitati in cazul in care nu a mai fost vreun contact anterior.

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

SHA-1 hash of the above certificate signed with the CA's private key

Un certificat nu este secret - este semnat de o autoritate de certificare - CA (Certificate Authority)
CA cripteaza cu cheia sa privata rezumatul certificatului

Verificarea certificatului de catre Alice

- A aplica cheia publica a CA asupra semnaturii
- A calculeaza rezumatul SHA-1 al certificatului (fara semnatura)
- A compara cele doua rezultate

Campurile de baza dintr-un certificat X.509

Câmp	Semnificatie
Versiune	Ce versiune de X.509 este utilizată
Serial Number	Acest număr împreună numele CA-ului identifică în mod unic certificatul
Algoritm de semnare	Algoritmul folosit la semnarea certificatului (ex. MD5 cu RSA)
Emitent	Numele X.500 al CA-ului
Perioada de validitate	Momentele de început si sfârşit ale perioadei de validitate
Numele subiectului	**Entitatea care este certificată**
Cheia publică	**Cheia publică a subiectului şi ID-ul algoritmului folosit (ex. RSA)**
ID emitent	Un ID opţional identificând în mod unic emitentul certificatului (nume X.500 sau DNS)
ID subiect	Un ID opţional identificând în mod unic subiectul certificatului
Extensii	ptr identificarea cheii publice a emitentului, a certificatului care contine o anumita cheie publica, scopul utilizarii cheii (criptare, semnare,…) si altele
Semnătura	**Semnătura certificatului (semnat cu cheia privată a CA-ului)**

PKI - Public Key Infrastructure

- **PKI- Set de componente (hard & soft)** care asigura utilizarea corecta a tehnologiei de chei publice
  - programele,
  - echipamentele,
  - tehnologiile de criptare si
  - serviciile de gestiune a infrastructurii criptografice si a cheilor publice ale utilizatorilor.

CA - Certificate Authority

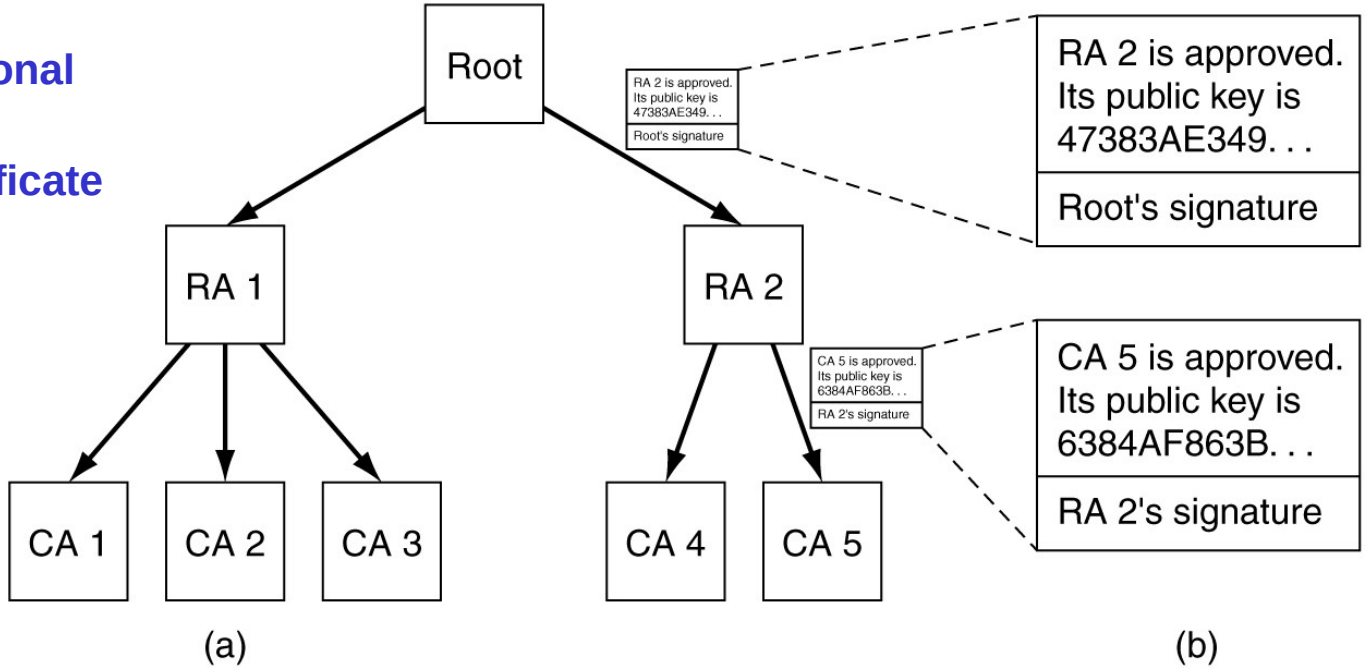
- autoritate de incredere care elibereaza certificate
  - atestă că cheia publica inclusa apartine persoanei cu numele atasat

- CA poate fi:
  - organizatie sau companie - pentru angajati
  - universitate - pentru studenti
  - CA publice (VeriSign) - pentru clienti

PKI – verificarea cheilor

RA – Regional Authority

CA – Certificate Authority



a) PKI ierarhic.

(b) Un lant de incredere (certification path).

A cunoaste si are incredere in Root

- gaseste certificatul lui B semnat de CA 5
- certificatul lui CA 5 semnat de RA 2
- certificatul lui RA 2 semnat de Root

Simplificare

A primeste de la B tot lantul de certificate

Revocarea Certificatelor

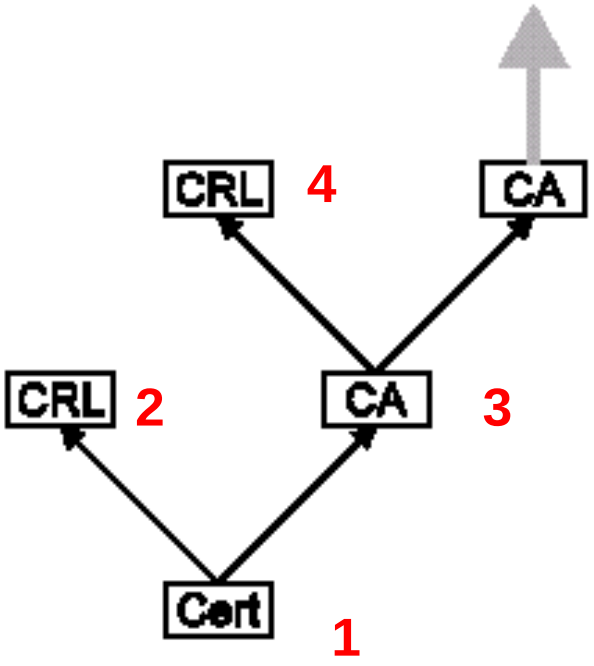
- Un certificat trebuie **revocat** cand:
  - cheia primara este **compromisa**;
  - cheia primara este **pierduta**;
  - o persoana pleaca din companie
  - altele.

- Revocarea trebuie anuntata tuturor utilizatorilor – dificil !
- Alternativa - se folosesc liste de revocare
  - **CRL – Certificate Revocation List**;

Metoda

- se verifica listele de revocare inainte de utilizarea certificatelor
- CRL sunt publicate de CA care a emis certificatele
- Listele pot fi consultate sau duplicate (cache)
  - difuzarea listelor de revocare – prin HTTP, LDAP sau alte protocoale

Verificarea revocarii Certificatelor



Verificare certificate

1 - verifica certificat

2 - verifica CRL

repeat

3 - verifica certificatul pentru CA

4 - verifica CRL al CA

until radacina


POLITEHNICA
UNIVERSITATEA
BUCUREŞTI
1818