Securitatea Rețelor și Serviciilor de Comunicații

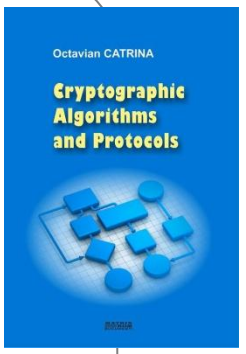
Algoritmi criptografici: Confidențialitatea datelor

Partea 1

Bibliografie

Octavian Catrina. Cryptographic Algorithms and Protocols.

Editura MATRIX ROM, București, 2016 (320 pg.).



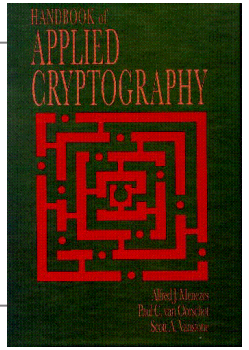
1. Encryption Schemes
2. Block Ciphers
3. Private-Key (Symmetric) Encryption
4. Hash Functions
5. Message Authentication Codes
6. Authenticated Encryption

7. Background for Public-key Cryptography
8. Public-Key (Asymmetric) Encryption
9. Digital Signatures
10. Public-Key Infrastructure
11. Entity Authentication
12. Key Exchange Protocols

Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone.

Handbook of Applied Cryptography.

CRC Press, 1996, 2001. E-book (pdf): http://cacr.uwaterloo.ca/hac/



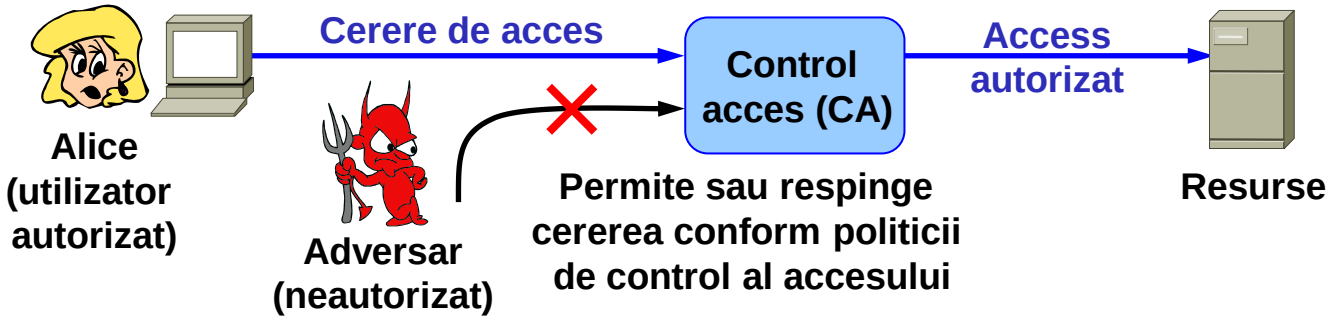
Christof Paar, Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2010.

Introducere

Servicii de securitate bazate pe criptografie

Controlul accesului

Controlul accesului (CA): Serviciu de securitate care *protejează resursele împotriva utilizării neautorizate*.



- Interceptează fiecare cerere de acces (utilizator, resursă, acțiune).
- Determină dacă acțiunea este autorizată sau nu de *politica de acces*.
- Permite efectuarea acțiunii dacă este autorizată, altfel o respinge.

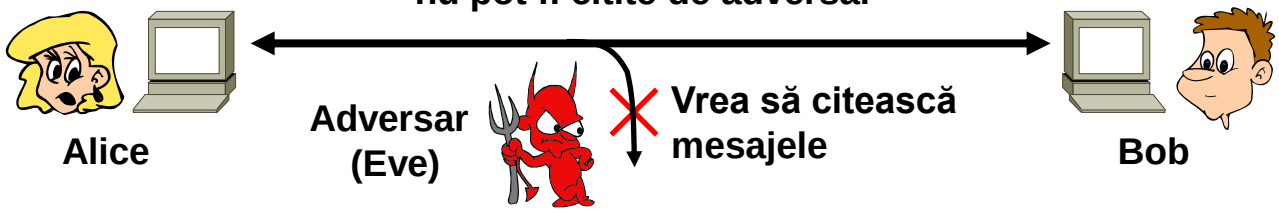
Exemplu: Cererea conține tripletul: uid (utilizator), oid (obiect), act (acțiune). Politica de acces specifică acțiunile permise fiecărui utilizator pentru fiecare obiect. Faze:

- *Autentificare*: Sistemul CA verifică identitatea utilizatorului.
- *Autorizare*: Sistemul CA verifică dacă operația solicitată (oid, act) apare în lista de permisiuni a utilizatorului uid.

Confidențialitatea datelor

Confidențialitatea datelor: Serviciu de securitate care *previne dezvăluirea neautorizată a informației (citire neautorizată)*.

Confidențialitatea datelor într-un sistem de comunicații:
Alice și Bob vor să se asigure că mesajele lor nu pot fi citite de adversar

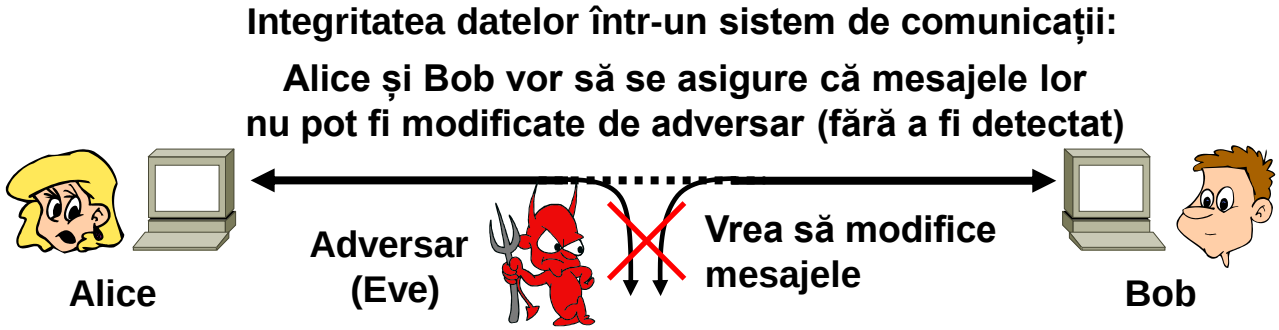


Metode de protecție a confidențialității datelor

- ***Controlul accesului:*** Împiedică accesul neautorizat (pentru citire) la sistemul de stocare sau rețeaua de comunicație. Aplicabilitate limitată.
- ***Criptografie - Criptare:*** Aplică datelor o transformare reversibilă, numită *criptare*, cu proprietatea că datele criptate nu pot fi distinse de o secvență binară aleatoare, iar transformarea inversă, numită *decriptare*, rămâne *secretă* și poate fi calculată doar de agenți autorizați.

Integritatea datelor

Integritatea datelor: Serviciu de securitate care *previne/detectează modificarea neautorizată a informației (scriere neautorizată)*.



Metode de protecție a integrității datelor

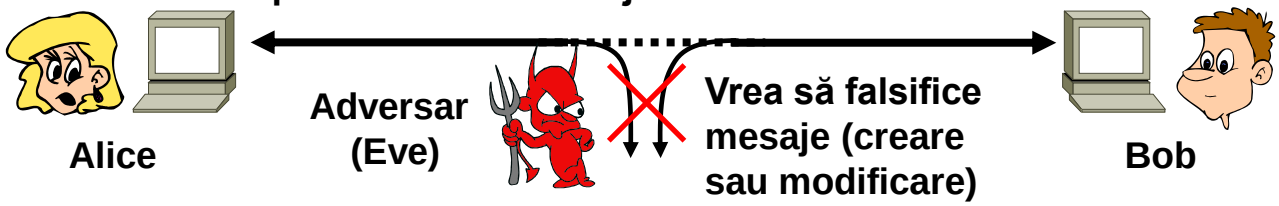
- ***Controlul accesului:*** Împiedică accesul neautorizat (pentru scriere) la sistemul de stocare sau rețeaua de comunicație. Aplicabilitate limitată.
- ***Criptografie - Cod de integritate:*** Asociază mesajului un cod binar cu proprietatea că nu pot fi găsite (calculate) mesaje diferite cu același cod, deci orice modificare va fi detectată. Limitare: codul trebuie transmis separat, pe un canal care nu este controlat de adversar.

Autentificarea (originii) datelor

Autentificarea datelor: Serviciu de securitate care *permite verificarea originii și (implicit) a integrității datelor*.

Autentificarea datelor într-un sistem de comunicații:

Alice și Bob vor să se asigure că adversarul nu poate falsifica mesajele lor fără a fi detectat



Metode de autentificare a (originii) datelor

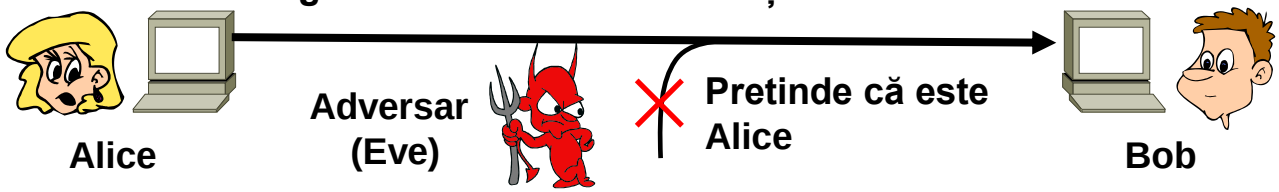
- ***Criptografie - Cod de autentificare (sau semnătură digitală)***:
  Asociază mesajului un cod binar care este o *funcție secretă*, cunoscută doar de agenții autorizați, și are proprietatea că nu pot fi găsite (calculate) mesaje diferite cu același cod.
  Adversarul nu poate falsifica (crea, modifica) mesaje fără a fi detectat pentru că nu poate calcula codul de autentificare corect.

Autentificare entităților

Autentificarea entităților: Serviciu de securitate care *permite verificarea identității participanților la o comunicație.*

Autentificare entităților într-un sistem de comunicații:

Exemplu: Autentificare unilaterală. Bob vrea să se asigure că într-adevăr interacționează cu Alice



- ***Metode de verificare a identității unei enități:***
  - Ceea ce știe: un secret (exemple: parolă, PIN).
  - Ceea ce are: o dovadă fizică (exemple: cartelă).
  - Ceea ce este: date biometrice (exemplu: amprentă digitală).
- ***Criptografie - Protocol (criptografic) de autentificare:*** Permite unei entități să verifice identitatea alteia prin achiziționarea interactivă a unor dovezi verificabile (folosind tehnici criptografice).

Considerații generale

Securitatea nu se poate baza pe algoritmi secreți

- Un algoritm nu poate rămâne secret: poate fi aflat prin “reverse engineering” sau publicat fără permisiune.
- “Security by obscurity” nu este o soluție.

Securitatea nu trebuie să depindă de ceva care nu poate fi ușor schimbat (principul lui Kerckhoffs,1883)

- Folosim *algoritmi publici* cu *parametri secreți* (chei secrete).
- Atenție la utilizarea datelor biometrice.

Criptografia rămâne un domeniu deosebit de dificil

- Folosim doar algoritmi validați de comunitatea internațională de experți criptologi, prin analize de securitate riguroase, publicate.

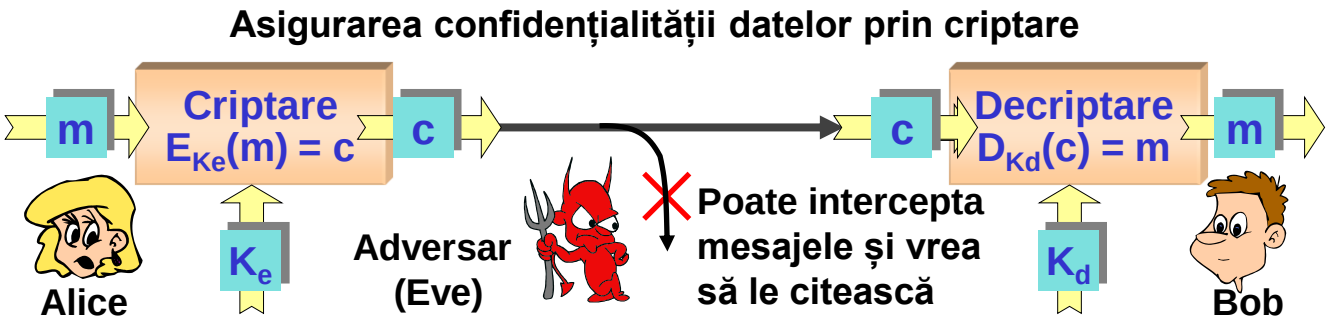
Avem de nevoie de interoperabilitate

- Trebuie să folosim soluții standardizate (algoritmi/protocoale).

Confidențialitatea datelor

Criptare - definiții

Confidențialitate prin criptare



- Mesajul *m* care trebuie protejat se numește *text clar (plaintext)*.
- Alice aplică textului clar *m* o transformare reversibilă *E*, numită *criptare (encryption)*, și obține un *text cifrat (ciphertext) c*.
- Bob recuperează textul clar *m* aplicând textului cifrat *c* transformarea inversă *D*, numită *decriptare (decryption)*.

- Securitate (intuitiv): Adversarul nu cunoaște transformarea inversă și nu poate să obțină din textul cifrat nici o informație despre textul clar.
- Algoritmii folosiți pentru criptare și decriptare sunt publici.
- Secretul transformării inverse este asigurat de un parametru numit *cheie*.

Scheme de criptare

Definiție: Familie de funcții

- Vom numi *familie de funcții* cu *domeniu* $\mathcal{X}$, *codomeniu* $\mathcal{Y}$ și mulțimea *cheilor* $\mathcal{K}$, o aplicație $\mathcal{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. Pentru fiecare cheie $K \in \mathcal{K}$, funcția $F_K: \mathcal{X} \to \mathcal{Y}$, $F_K(x) = F(K, x)$ este o *instanță* a familiei de funcții.
- Echivalent, $\mathcal{F}$ este o mulțime de funcții definite pe $\mathcal{X}$ cu valori în $\mathcal{Y}$, indexată de elementele mulțimii cheilor: $\mathcal{F} = \{ F_K \mid K \in \mathcal{K}, F_K: \mathcal{X} \to \mathcal{Y} \}$.

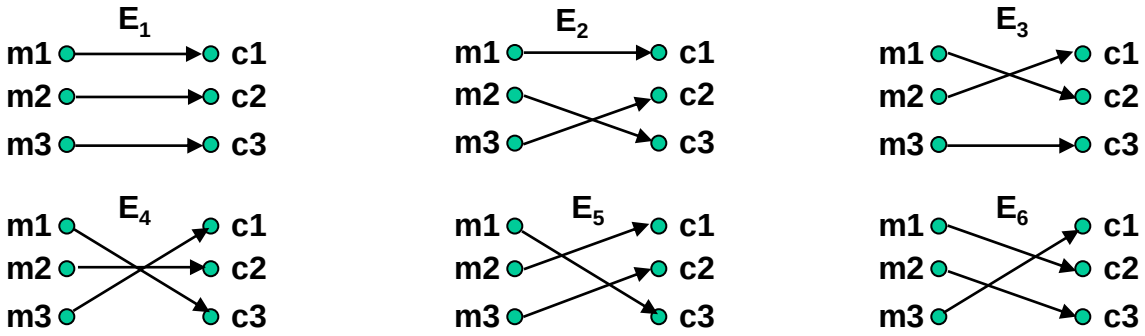
Definiție: Schemă de criptare

- O schemă de criptare cu *mulțimea textelor clare* $\mathcal{M}$, *mulțimea textelor cifrate* $\mathcal{C}$ și *mulțimea cheilor* $\mathcal{K}$, este alcătuită din trei algoritmi:
- *Algoritmul de generare a cheilor*, care alege aleator o cheie din $\mathcal{K}$.
- *Algoritmul de criptare*, care este o familie de funcții inversabile, numite *funcții de criptare*, $\mathcal{E} = \{ E_K \mid K \in \mathcal{K}, E_K: \mathcal{M} \to \mathcal{C} \}$.
- *Algoritmul de decriptare*, care este familia de *funcții de decriptare* $\mathcal{D} = \{ D_K \mid K \in \mathcal{K}, D_K: \mathcal{C} \to \mathcal{M} \}$, astfel încât $D_K = E_K^{-1}$, pentru oricare $K \in \mathcal{K}$.

Exemplu generic

- Schemă de criptare pentru $\mathcal{M}$ = { m_1, m_2, m_3 } și $\mathcal{C}$ = { c_1, c_2, c_3 }.
- Există n = 3! = 6 funcții bijective $E_K : \mathcal{M} \to \mathcal{C}$. Vom folosi pentru criptare toate aceste funcții, deci mulțimea cheilor este $\mathcal{K}$ = { 1, 2, 3, 4, 5, 6 }.

Famila de funcții de criptare:



- Alg. de generare a cheilor: alege K$\in$ { 1, 2, 3, 4, 5, 6 }, uniform aleator.
- Alg. de criptare: $\mathcal{E}$ = { E_1, E_2, E_3, E_4, E_5, E_6 }, cu E_K definite în figură.
- Alg. de decriptare: $\mathcal{D}$ = { D_1, D_2, D_3, D_4, D_5, D_6 }, unde $D_K = E_K^{-1}$.

Exemplu: $\mathcal{M}$ = { fwd, back, stop } și $\mathcal{C}$ = { 1, 2, 3 }.

K = 3 $\Rightarrow$ E_3(fwd) = 2, E_3(back) = 1, E_3(stop) = 3.

Cât de sigură este această schemă? Ce facem dacă $|\mathcal{M}| = 2^{100}$?

Exemplu: Cifru cu substituție

- Alfabet: $\mathcal{A}$ = { a, b, c, ... y, z } $\Rightarrow |\mathcal{A}| = 26$ caractere.
- $\mathcal{M} = \mathcal{C}$ = { toate șirurile finite de caractere din $\mathcal{A}$ }.
- Principiu: fiecare caracter din textul clar este înlocuit cu un alt caracter.
- $\mathcal{K}$ = Perm($\mathcal{A}$) (toate permutările p : $\mathcal{A} \to \mathcal{A}$) $\Rightarrow |\mathcal{K}| = 26! \approx 4 \times 10^{26}$ chei.

- Alg. de generare a cheilor: alege p $\in$ Perm($\mathcal{A}$), uniform aleator.
- Alg. de criptare: c[i] = p(m[i]), i = 1, 2, ... |m|.
- Alg. de decriptare: m[i] = p^{-1}(c[i]), i = 1, 2, ... |m|.

Exemplu: **Cheie** **O permutare p : $\mathcal{A} \to \mathcal{A}$:**

	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
p =	f	q	i	s	h	n	c	v	j	t	y	a	u	w	d	r	e	x	l	b	m	z	o	g	k	p

Text clar TO BE OR NOT TO BE

Criptare

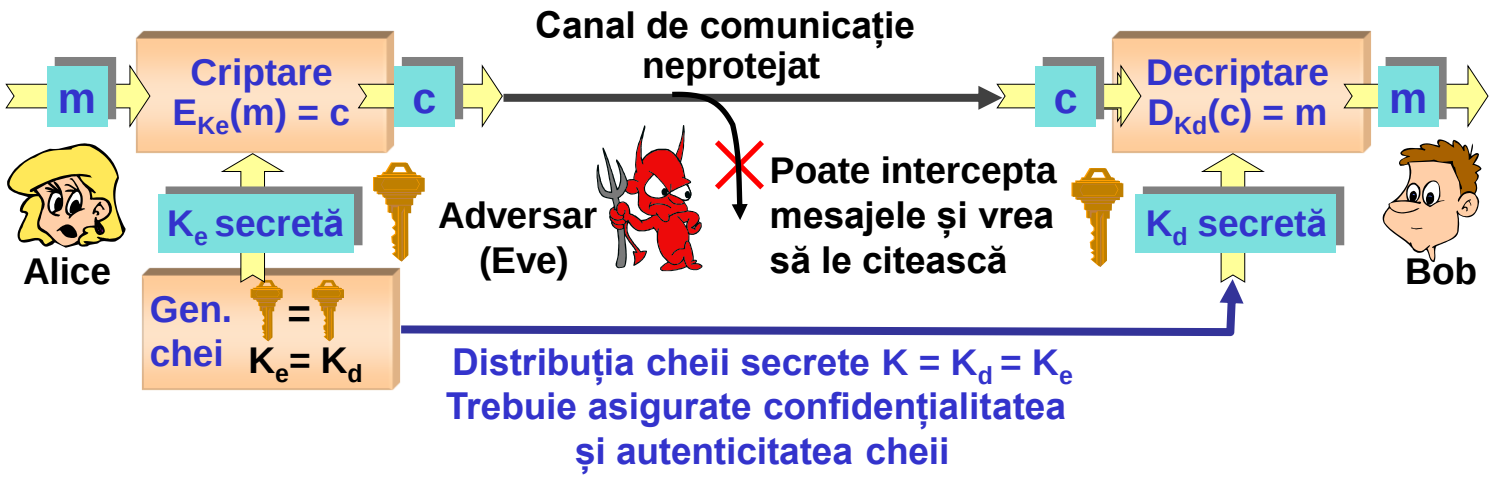
Text cifrat BD QH DX WDB BD QH

Decriptare

TO BE OR NOT TO BE

Cât de sigură este această schemă?

Criptare simetrică (cheie secretă)

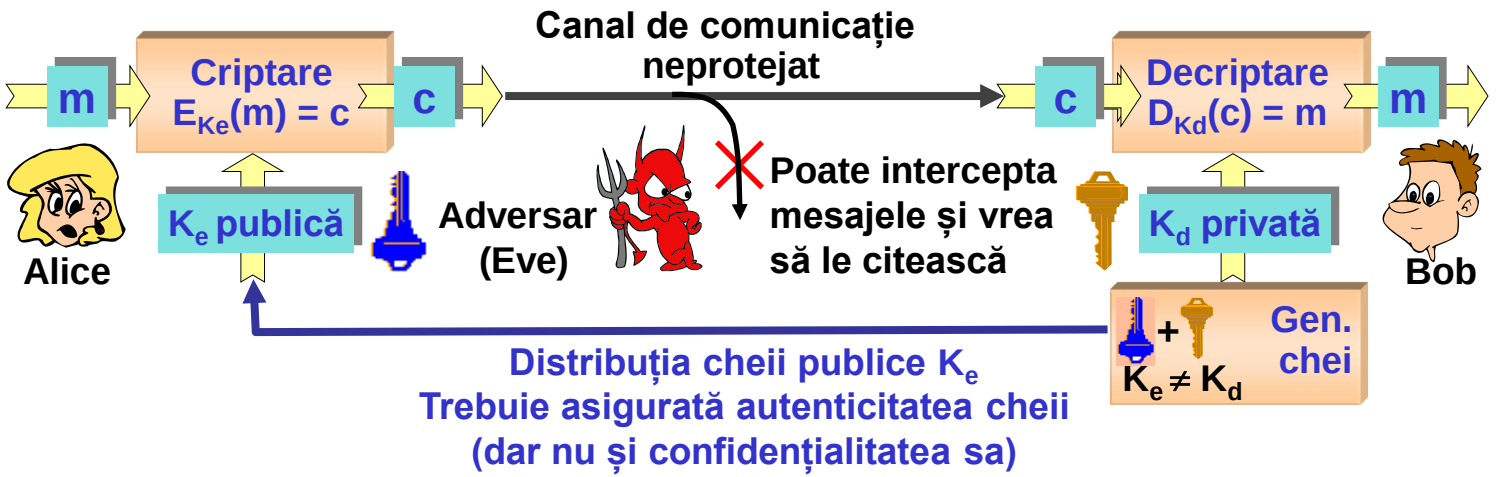


Schemă de criptare *simetrică* (sau criptare cu *cheie secretă*)

- Algoritmul de criptare și cel de decriptare folosesc aceeași cheie K.
- Cheia este secretă: o cunosc doar participanții legitimi.
- Adversarul nu cunoaște cheia, deci nu cunoaște funcția de decriptare.

- Avantaje: Oferă cei mai rapizi și eficienți algoritmi.
- Dezavantaje: Distribuția cheilor este dificilă. Un utilizator trebuie să distribuie o cheie secretă diferită pentru fiecare partener de comunicație, garantând confidențialitatea și autenticitatea cheii.

Criptare asimetrică (cheie publică)

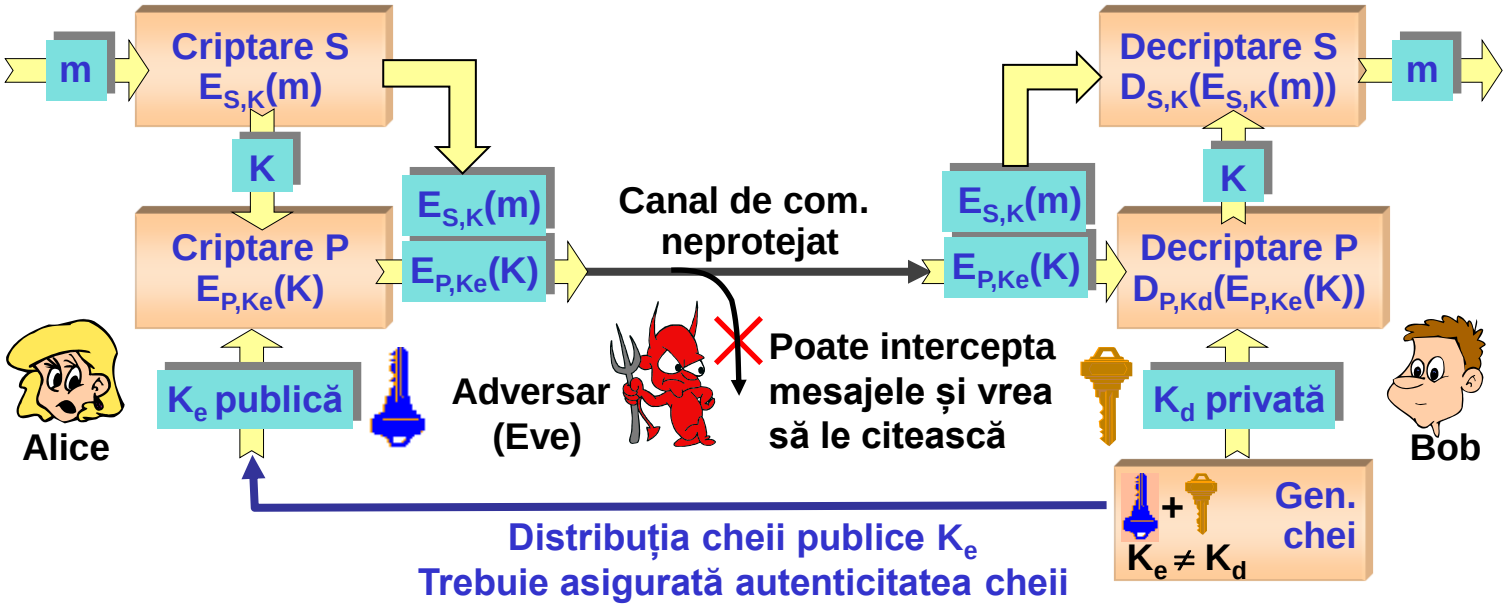


Schemă de criptare *asimetrică* (sau criptare cu *cheie publică*)

- Un utilizator are o pereche de chei: cheia privată K_d și cheia publică K_e.
- Cheia publică este distribuită partenerilor și este folosită pentru criptare.
- Cheia privată rămâne secretă și este folosită pentru decriptare.

- Avantaje: Simplifică distribuția cheilor. Un utilizator distribuie o singură cheie, care nu este secretă, deci trebuie garantată doar autenticitatea.
- Dezavantaje: Algoritmii sunt mult mai lenți și ineficienți. În practică, se folosește doar pentru distribuirea unor chei secrete.

Criptare hibridă



Schemă de criptare *hibridă*

- Combină avantajele criptării cu cheie secretă (S) și cu cheie publică (P):
- Alice deține o copie a cheii publice de criptare a lui Bob, K_e.
  - Criptează cu cheie secretă mesajul m: $E_{S,K}(m)$, K aleasă aleator.
  - Criptează cu cheie publică cheia K: $E_{P,Ke}(K)$.
  - Transmite $E_{P,Ke}(K)$ și $E_{S,K}(m)$.
- Bob efectuează operațiile inverse pentru a recupera K și apoi m.

Funcții aleatoare

Definiție: Funcție aleatoare (Random Function)

- Fie $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^*$ mulțimi finite de șiruri binare.
- Notăm $\mathrm{Func}(\mathcal{X}, \mathcal{Y})$ familia *tuturor* funcțiilor definite pe $\mathcal{X}$ cu valori în $\mathcal{Y}$.
- *O funcție aleatoare* $\mathrm{F} : \mathcal{X} \to \mathcal{Y}$ este o instanță a familiei $\mathrm{Func}(\mathcal{X}, \mathcal{Y})$ aleasă uniform aleator.

Algoritm echivalent

```
Inițializare:
  For all x∈𝒳
    F[x] ← null.
Funcție aleatoare:
  Input x
  If (F[x] = null)
    Alege aleator y∈𝒴
    F[x] ← y
  Output F[x].
```

Proprietăți

Fiecărui $\mathrm{x} \in \mathcal{X}$ îi este asociat $\mathrm{y} \in \mathcal{Y}$, ales uniform aleator și independent:

- Distribuție uniformă: $\forall \mathrm{x} \in \mathcal{X}, \forall \mathrm{y} \in \mathcal{Y}, \Pr[\mathrm{F}(\mathrm{x}) = \mathrm{y}] = 1/|\mathcal{Y}|$
- Indepedență: $\forall \mathrm{x}_1 \neq \mathrm{x}_2 \in \mathcal{X}$ și $\forall \mathrm{y}_1, \mathrm{y}_2 \in \mathcal{Y}, \Pr[\mathrm{F}(\mathrm{x}_1) = \mathrm{y}_1 \mid \mathrm{F}(\mathrm{x}_2) = \mathrm{y}_2] = 1/|\mathcal{Y}|$

- *Acest model oferă proprietăți statistice esențiale pentru construcția schemelor de criptare, autentificare, generare a șirurilor aleatoare, etc.*

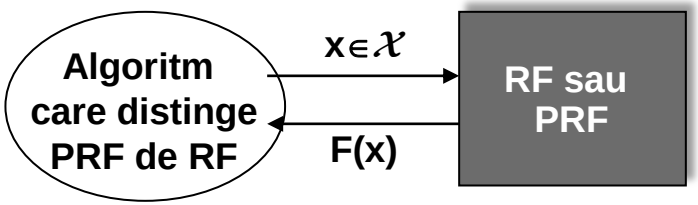
Funcții pseudoaleatoare

Definiție: Funcție pseudoaleatoare (PRF - Pseudorandom Function)

- Aproximare a unei funcții aleatoare (construcții practice, eficiente).
- O funcție pseudoaleatoare este o familie de funcții $\mathcal{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ (cu domeniu $\mathcal{X}$, codomeniu $\mathcal{Y}$ și chei $\mathcal{K}$), cu proprietatea că instanțele $F_K \in \mathcal{F}$ alese uniform aleator nu pot fi distinse de funcția aleatoare $F: \mathcal{X} \to \mathcal{Y}$.

Experiment: RF sau PRF?

- "Cutie neagră" care conține fie funcția aleatoare $F: \mathcal{X} \to \mathcal{Y}$, fie funcția pseudoaleatoare $\mathcal{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$.
- Algoritmul interacționează cu cutia neagră și decide dacă este PRF sau RF.



RF: Random Function
PRF: Pseudo-Random Function

- Familia de funcții $\mathcal{F}: \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ este o funcție pseudoaleatoare dacă nu există nici un algoritm eficient care să o distingă de funcția aleatoare $F: \mathcal{X} \to \mathcal{Y}$. cu probabilitate $\Pr > 0.5 + \varepsilon$, cu ε neglijabil.

"Paradoxul zilei de naștere"

Paradoxul zilei de naștere (birthday paradox)

Câți studenți trebuie să fie în clasă astfel încât cu probabilitate 50%:

cel puțin un student are aceeași zi de naștere ca profesorul?

- q studenți, N zile/an:
  $P(N, q) \approx q/N$
- $N = 365, P = 0.5$:
  $q \approx N/2 \approx$ 183 studenți

cel puțin doi studenți au aceeași zi de naștere?

- q studenți, N zile/an:
  $P(N, q) \approx 1 - e^{-q(q-1)/2N} = O(q^2/N)$
- $N = 365, P = 0.5$:
  $q \approx (2\ln 2)^{1/2} \cdot N^{1/2} \approx$ 23 studenți

Coliziuni și preimagini în funcții (pseudo)aleatoare

- O *coliziune* a unei funcții $F : \mathcal{X} \to \mathcal{Y}$ este o pereche de intrări distincte, $x_1 \neq x_2$, cu proprietatea că $F(x_1) = F(x_2)$.
- Pentru o funcție (pseudo)aleatoare $F : \mathcal{X} \to \mathcal{Y}$ cu $\mathcal{Y} = \{0,1\}^n$:
  - Găsim o **preimagine** x a.î. $F(x) = y$ testând $O(2^n)$ intrări.
  - Găsim o **coliziune** $x_1 \neq x_2$ a.î. $F(x_1) = F(x_2)$ testând $O(2^{n/2})$ intrări.

$\Rightarrow$ Atacuri eficiente bazate pe coliziuni asupra unor algoritmi criptografici.

Permutări aleatoare

Definiție: Permutare aleatoare

- Fie $\mathcal{X} \subseteq \{0,1\}^*$ o mulțime finită de șiruri binare.
- Notăm $\mathrm{Perm}(\mathcal{X})$ familia *tuturor* funcțiilor bijective (permutări) definite pe $\mathcal{X}$.
- *O permutare aleatoare* $\mathrm{P} : \mathcal{X} \to \mathcal{X}$ este o instanță a familiei $\mathrm{Perm}(\mathcal{X})$ aleasă uniform aleator.

Algoritm echivalent

```
Inițializare:
  For all x∈𝒳
    P[x] ← null
  S ← ∅.
Permutare aleatoare:
  Input x
  If (P[x] = null)
    Alege aleator y∈𝒳-S
    P[x] ← y
    S ← S ∪ {y}
  Output P[x].
```

Proprietăți

Fiecărui $x \in \mathcal{X}$ îi este asociat $y \in \mathcal{X}$ ales uniform aleator; y nu e ales complet independent, deoarece P este o funcție bijectivă, deci $x_1 \neq x_2 \Rightarrow y_1 \neq y_2$:

- Distribuție uniformă: $\forall x \in \mathcal{X}$, $\forall y \in \mathcal{X}$, $\Pr[P(x) = y] = 1/|\mathcal{X}|$
- Indepedență (imperfectă): $\forall x_1 \neq x_2 \in \mathcal{X}$ și $\forall y_1, y_2 \in \mathcal{X}$,

$$\Pr[P(x_1) = y_1 \mid P(x_2) = y_2] = \begin{cases} 1/(|\mathcal{X}|-1) & \text{dacă } y_1 \neq y_2, \\ 0 & \text{dacă } y_1 = y_2. \end{cases}$$

- *Acest model oferă proprietăți esențiale ale schemelor de criptare*.

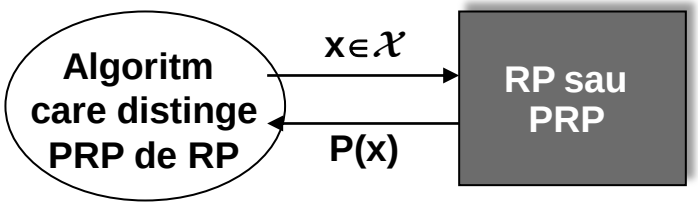
Permutări pseudoaleatoare

Definiție: Permutare pseudoaleatoare (PRP)

- Aproximare a unei permutări aleatoare (construcții practice, eficiente).
- O permutare pseudoaleatoare este o familie de funcții bijective $\mathcal{P} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ (domeniu $\mathcal{X}$ și chei $\mathcal{K}$), cu proprietatea că instanțele $P_K \in \mathcal{P}$ alese uniform aleator nu pot fi distinse de permutarea aleatoare $P : \mathcal{X} \to \mathcal{X}$.

Experiment: RP sau PRP?

- "Cutie neagră" care conține permutarea aleatoare $P : \mathcal{X} \to \mathcal{X}$ sau permutarea pseudoaleatoare $\mathcal{P} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$.
- Algoritmul interacționează cu cutia neagră și decide dacă este PRP sau RP.



RP: Random Permutation
PRF: Pseudo-Random Permutation

- Familia de funcții $\mathcal{P} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ este o permutare pseudoaleatoare dacă nu există nici un algoritm eficient care să o distingă de permutarea aleatoare $P : \mathcal{X} \to \mathcal{X}$ cu probabilitate $Pr > 0.5 + \varepsilon$, cu ε neglijabil.

Confidențialitatea datelor

Modele de securitate pentru criptare

Modele de securitate

Care sunt cerințele de securitate ale unei scheme criptografice?

Criptarea trebuie să protejeze confidențialitatea datelor împotriva unui adversar ...

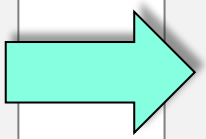
Ce înseamnă confidențialitate?
Ce este un adversar?
Ce fel de protecție, în ce condiții?

Modelul de securitate

Obiectivele adversarului
Ce urmărește adversarul?
Ce încercăm să prevenim?

Modelul de atac
Ce poate face adversarul?
Ce informații poate obține?

Resursele adversarului
Ce resurse (calcul, stocare, comunicație) are adversarul?



Cerința de securitate (sau evaluare a securității)

Dorim ca adversarul

- să nu poate atinge un anumit obiectiv,
- în condițiile unui anumit model de atac,
- dispunând doar de resursele specificate.

Modele de atac

Ipoteze generale despre adversar

- Cunoaște toate tehnicile criptografice folosite (algoritmi, protocoale).
- Nu cunoaște anumiți parametri criptografici secreți ai algoritmilor.
  Exemplu: Cheile secrete.

Modelul de atac

- Descrie alte capabilități ale adversarului, utile atingerii obiectivului său.
  Exemplu: Ce informații suplimentare poate să obțină și în ce mod, despre textele clare sau despre parametrii criptografici secreți.

Pentru criptare, vom presupune că adversarul dispune de anumite *informații preliminare (a priori) limitate* privind textele clare.

Exemplu: Textul criptat este un e-mail în format standard, conținând un text cu caractere ASCII în limba engleză.

```
From: alice@some.org
To: bob@other.com
Subject: Out of office next week
Date: Wed, 22 Mar 2017 20:15:48 +0100
MIME-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
Content-Length: 123

Dear Bob,
Next week I'll be on vacation.
Alice
```

Tipuri generice de atac / adversar

Atac pasiv (adversar pasiv)

- Adversarul pasiv nu poate influența funcționarea participanților legitimi.
- Poate captura mesajele transmise pe canalul de comunicație.
- Nu poate modifica, crea sau șterge mesaje, nu poate iniția interacțiuni.

Atac activ (adversar activ (ne-adaptiv))

- Adversarul activ poate iniția interacțiuni cu participanții legitimi.
- Poate captura, modifica, sau șterge mesajele transmise pe canalul de comunicație.

Atac adaptiv (adversar activ adaptiv)

- Într-un atac adaptiv, un adversar activ interacționează cu participanții legitimi și își adaptează atacul în funcție de informația obținută.

Periculozitatea tipurilor generice de atacuri

Atac pasiv < Atac activ < Atac (activ) adaptiv

Criptare: Modele de atac (1)

Atac cu text cifrat (Known Ciphertext Attack (KCA))

- Adversarul obține doar texte cifrate (criptate cu aceeași cheie).
- Deține și informații a priori limitate despre textul clar.
- Atac pasiv. Este suficient ca adversarul să poată captura mesaje cifrate.

Atac cu text clar (Known Plaintext Attack (KCA))

- Adversarul obține texte cifrate împreună cu textele clare asociate.
- Informațiile a priori pot oferi fragmente de text clar și text cifrat (de exemplu, mesaje în format standard).
- Atac pasiv/activ. Suficient pentru căutarea exhaustivă a cheii.

Atac cu text clar ales (Chosen Plaintext Attack (CPA))

- Adversarul obține textele cifrate pentru *anumite texte clare*, pe care le alege el însuși. Textele clare sunt alese astfel încât să permită exploatarea unor vulnerabilități ale schemei de criptare.
- Atac activ. Echivalent cu accesul la funcția de criptare.

Criptare: Modele de atac (2)

Atac cu text cifrat ales (Chosen Ciphertext Attack (CCA))

- Adversarul obține textele clare pentru *anumite texte cifrate*, pe care le alege el însuși. Textele cifrate sunt alese astfel încât să permită exploatarea unor vulnerabilități ale schemei de criptare.
- Atac activ. Echivalent cu *acces parțial/temporar* la funcția de decriptare.

Atacuri adaptive cu text clar/cifrat ales

- Adversarul interacționează în mod repetat cu victima și algoritmul său adaptează textele alese în funcție de informațiile obținute anterior.

Atacuri side-channel

- Exploatează informații obținute prin măsurători ale unor parametri ai unei implementări: durata operațiilor, consumul de curent, etc.

Atac cu text cifrat < Atac cu text clar <
Atac cu text clar ales (< adaptiv) < Atac cu text cifrat ales (< adaptiv)

Criptare: Obiectivele adversarului (1)

Criptarea trebuie să protejeze confidențialitatea datelor ...
Ce înseamnă acest lucru?

Adversarul dorește informații despre datele confidențiale.
Ce fel de informații, cum?

Să încercăm câteva răspunsuri ...

Recuperarea cheii de decriptare (Key Recovery)

- Adversarul vrea să determine cheia de decriptare.
- Obiectivul cel mai ambițios (recompensă maximă): aflând cheia, poate decripta orice text cifrat (pe durata de viață a cheii, în trecut și în viitor).

Recuperarea textului clar (Plaintext Recovery)

- Adversarul vrea să determine textul clar corespunzător unui text cifrat, fără cheia de decriptare.
- Obiectiv mai puțin ambițios: efort repetat pentru fiecare text cifrat.

Alte obiective posibile?

- Să determine un fragment din textul clar, un caracter, probabilitatea sa?

Criptare: Obiectivele adversarului (2)

Securitatea aplicațiile informatice

- Problema confidențialității apare întotdeauna în contextul unei aplicații.
- Motivația reală a adversarului este să atace o aplicație.
- Ce încălcări ale confidențialității datelor pot compromite aplicația?

  Estimarea valorii unui singur bit din textul clar, cu probabilitate mare, ar putea compromite unele aplicații ...

Considerente privind definirea noțiunilor de securitate

- Cerințele de securitate ale aplicațiilor.

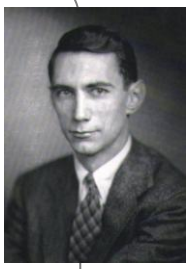
  Suport pentru o gamă largă de aplicații, în prezent și în perspectivă.
- Cerințele de performanță ale aplicațiilor.

  Noțiuni de securitate care să poată fi îndeplinite de algoritmi eficienți.
- Capabilitățile și resursele adversarilor.

  Ipoteze realiste privind adversarul. Resurse limitate? Nelimitate? Evoluția tehnologică în următoarele decenii?

Securitate necondiționată (perfectă)

Criptare cu securitate perfectă / necondiționată (perfect secrecy)



- Primul studiu riguros al securității sistemelor de criptare a fost publicat 1949 de Claude Shannon: "*Communication Theory of Secrecy Systems*", *Bell Systems Technical Journal*.
- Definește securitatea perfectă și prezintă și o soluție ideală/optimă.

Modelul adversarului

- Dispune de *resurse nelimitate* $\Leftrightarrow$ *securitate necondiționată*.
- Deține *informații "a priori"* despre textele clare, $\mathcal{M} = \{m_i\}$, exprimate prin distribuția de probabilitate $Pr[m_i]$ ca $m_i \in \mathcal{M}$ să apară în comunicație.

Definiție: Securitate perfectă (necondiționată)

- Pentru toate textele cifrate $c \in \mathcal{C}$ și orice distribuție de probabilitate $Pr(m_i)$, $\mathbf{Pr[m_i \mid c] = Pr[m_i]}$, (prob. condiționate ale textelor clare după observarea textului cifrat c, $Pr[m_i \mid c]$, sunt egale cu probabilitățile lor a priori, $Pr[m_i]$).

$\Rightarrow$ Textele cifrate nu oferă adversarului nici o informație despre textele clare.

Criptare cu securitate perfectă

Schema de criptare "one-time pad" (Shannon)

- Familie de funcții $\mathcal{E} : \mathcal{K}\times\mathcal{M} \to \mathcal{C}$, unde $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$.
- Generarea cheii: Pentru *fiecare text clar* de n biți, alege indepedent, uniform aleator o cheie $K = (K_1, K_2, ..., K_n) \in \mathcal{K}$ (deci $\Pr[K] = 1/2^n$).
- Criptare: Pentru oricare $m = (m_1, m_2, ..., m_n) \in \mathcal{M}$ și $K = (K_1, K_2, ..., K_n) \in \mathcal{K}$, $c = E_K(m) = m \oplus K = (m_1 \oplus K_1, m_2 \oplus K_2, ... m_n \oplus K_n)$.
- Decriptare: Pentru oricare $c = (c_1, c_2, ..., c_n) \in \mathcal{C}$ și $K = (K_1, K_2, ..., K_n) \in \mathcal{K}$, $m = D_K(c) = c \oplus K = (c_1 \oplus K_1, c_2 \oplus K_2, ... c_n \oplus K_n)$.

Proprietățile schemei "one-time pad"

- *Îndeplinește cerința de securitate perfectă* definită de Shannon.
- Este *soluția optimă* pentru această cerință (nu există soluții mai eficiente).
- Fiecare text cifrat necesită o cheie cu aceeași lungime, generată indepedent și uniform aleator. Generare, distribuție, stocare a cheilor?

Schema "one-time pad" este extrem de ineficientă.

⇒ Noțiunea de securitate perfectă nu este practică.

"Computational security"

Securitate bazată pe efortul de calcul - "Computational security"

- Concept fundamental al criptografiei moderne.
- Model mai realist al adversarului:
  Adversarul dispune de resurse limitate (calcul, memorie, comunicație).
- Cerințe de securitate mai pragmatice:
  Adversarul poate realiza un anumit obiectiv cu probabilitate neglijabilă.

Consecințe asupra eficienței și securității algoritmilor criptografici

- Securitatea unei scheme criptografice se bazează pe un *decalaj enorm între complexitatea algoritmilor* executați de utilizatorii legitimi și cei prin care adversarul ar putea ataca proprietățile de securitate.
- Complexitatea algoritmilor depinde de un *parametru de securitate*.

  Exemplu: Parametru de securitate k. Complexitate polinomială, $O(k^e)$, dacă este cunoscută cheia secretă (exp. mic), altfel complexitate exponențială, $O(2^k)$.
- Securitatea pot fi asigurată de *algoritmi eficienți, cu chei relativ scurte*.

  Exemplu: Criptare cu cheie secretă cu lungime k. Cheia poate fi ghicită cu probabilitate 2^{-k} sau găsită prin căutare exhaustivă cu $\approx 2^k$ operații.

Securitatea semantică

Securitatea criptării în criptografia modernă

- Criptografia modernă utilizează *două formulări echivalente* pentru securitatea criptării, introduse în 1982 de Goldwasser și Micali:
  - Securitatea semantică (semantic security).
  - Nedistingerea textelor criptate (indistinguishability of encryptions).

Securitatea semantică (semantic security)

- Intuitiv: Informația pe care un adversar cu resurse limitate o poate *calcula eficient* din texte cifrate și din informația disponibilă a priori poate fi calculată aproape *la fel de eficient* doar din informația a priori.

Nedistingerea textelor cifrate (IND: Indistinguishability of encryptions)

- Intuitiv: Pentru orice pereche de texte clare $m_1 \in \mathcal{M}$ și $m_2 \in \mathcal{M}$, cu aceeași lungime, și $c \in \{ E_K(m_1), E_K(m_2) \}$, un adversar cu resurse limitate nu poate determina dacă $c = E_K(m_1)$ sau $c = E_K(m_2)$.
- Preferată în practică, mai convenabilă pentru demonstrații de securitate.

Cerințe de securitate

Formularea cerințelor de securitate

- O cerință de securitate afirmă că adversarul nu poate atinge un anumit obiectiv folosind un anumit model de atac și anumite resurse.
- O cerință este mai puternică (strictă) dacă afirmă că adversarul nu poate atinge un obiectiv mai slab folosind un model de securitate mai puternic.

- Obiective:
  Distingere texte cifrate < Recuperare text clar < Recuperare cheie.
- Modele de atac:
  Atac cu text cifrat (KCA) < Atac cu text clar (KPA) <
  Atac cu text clar ales (CPA) < Atac cu text cifrat ales (CCA).
  Atac neadaptiv (CPA, CCA) < Atac adaptiv (CPA, CCA).

Cerințe de securitate pentru criptare

- IND-CCA: Nedistingerea textelor cifrate în atacuri cu text cifrat ales.
  Cerința standard (cea mai strictă: obiectiv minim, cel mai puternic atac).
- IND-CPA: Nedistingerea textelor cifrate în atacuri cu text clar ales.
  Cerința minimă, acceptată în anumite contexte.

Exemplu: Criptare simetrică IND-CPA

Definiție: O schemă de criptare simetrică cu parametrul de securitate k îndeplinește cerința IND-CPA dacă pentru orice adversar există o funcție neglijabilă $\nu(k)$ astfel încât probabilitatea să câștige jocul de mai jos este:

$$\Pr[k] \leq 0.5 + \nu(k).$$

Adversar

Oracol de criptare

$E : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

Alege uniform aleator cheia secretă $K \in \mathcal{K}$.

Alege uniform aleator bitul secret $b \in \{0,1\}$

Alege $m_0, m_1 \in \mathcal{M}$, de lungime egală.

Găsește b' astfel încât $c = E_K(m_{b'})$

Interogare (m_0, m_1) →

← Provocare (c)

Rezultat (b') →

Calculează $c = E_K(m_b)$

Adversarul câștigă dacă $b' = b$

Înainte și după provocare, adversarul poate solicita oracolului textul cifrat pentru orice text clar.

Teoremă: Un algoritm de criptare care este determinist și nu menține informație de stare nu poate îndeplini cerința de securitate IND-CPA.

Confidențialitatea datelor

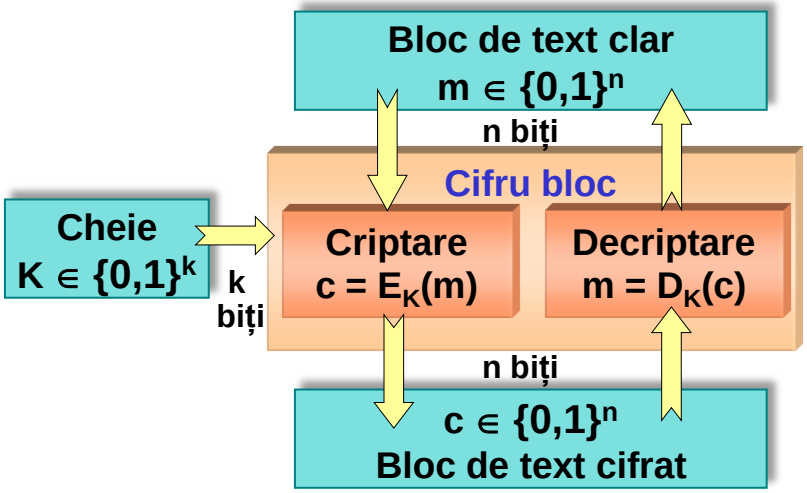
Cifruri bloc

Cifru bloc

Definiție: Cifru bloc (Block cipher)

Un cifru bloc este o *familie de funcții bijective* $\mathcal{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, unde:

- $\mathcal{K} = \{0,1\}^k$ este mulțimea cheilor.
- $\mathcal{M} = \{0,1\}^n$ este mulțimea blocurilor de intrare ("texte clare").
- $\mathcal{C} = \{0,1\}^n$ este mulțimea blocurilor de ieșire ("texte cifrate").
- Pentru fiecare cheie $K \in \mathcal{K}$, $E_K : \mathcal{M} \to \mathcal{C}$, $E_K(m) = \mathcal{E}(K, m)$, este *funcția directă a cifrului* ("funcția de criptare").
- Pentru fiecare cheie $K \in \mathcal{K}$, $D_K = E_K^{-1}$, este *funcția inversă a cifrului* ("funcția de decriptare"), deci $D_K(E_K(m)) = m$ pentru fiecare $m \in \mathcal{M}$.



- *Un cifru bloc este o primitivă criptografică*. Este utilizat în scheme de criptare, de autentificare, etc.
- *Un cifru bloc nu este o schemă de criptare sigură*. Este un algoritm determinist fără stare, deci nu poate îndeplini IND-CPA.

Securitatea cifrurilor bloc (1)

Cerința de securitate fundamentală pentru cifruri bloc

- Un cifru bloc $\mathcal{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ trebuie să se comporte ca o *permutare pseudoaleatoare*: pentru o alegere uniform aleatoare a cheii $K \in \{0,1\}^k$, valorile sale de ieșire, $E_K(m)$, $m \in \{0,1\}^n$, nu pot fi distinse de cele ale permutării aleatoare $P : \{0,1\}^n \to \{0,1\}^n$.

Securitate aplicațiilor cifrurilor bloc

- Adversarul dorește să compromită aplicația care folosește cifrul bloc: schemă de criptare, schemă de autentificare, etc.
- Securitatea algoritmilor criptografici construiți cu cifruri bloc se bazează pe ipoteza că cifrul bloc utilizat este o permutare pseudoaleatoare.
- Această proprietate de securitate implică rezistență la diverse atacuri uzuale. De exemplu, un adversar cu resurse limitate:
  - Nu poate determina o intrare m cunoscând (doar) ieșirea $E_K(m)$.
  - Nu poate afla o cheie secretă K (aleasă uniform aleator) cunoscând perechi $(m, E_K(m))$, etc.

Securitatea cifrurilor bloc (2)

Căutare exhaustivă (exhaustive search)

- Adversarul caută informația necesară atingerii obiectivului său (cheie secretă, text clar, etc.) prin enumerarea situațiilor posibile.

- Exemplu: Căutarea cheii unui cifru bloc, știind câteva perechi (m_i, c_i).
- Pentru $\mathcal{K} = \{0,1\}^k$, există 2^k chei. Pentru $\forall K \in \mathcal{K}$ verificăm $c_1 = E_K(m_1)$. Obținem $1 + 2^{k-n}$ chei posibile. Repetăm cu alte texte până când rămâne o singură cheie. În total: $O(2^k)$ operații de criptare și $1 + k/n$ texte.

Atacuri criptanalitice

- Criptanaliza caută vulnerabilități în algoritmii criptografici care permit atacuri mai eficiente decât căutarea exhaustivă.
- Exemple de metode criptanalitice pentru cifruri bloc:
  - Criptanaliză lineară, criptanaliză diferențială: Caută anumite corelații între intrări și ieșiri care permit aflarea cheii secrete.
  - Criptanaliză algebrică: Descrie algoritmul ca sistem de ecuații a cărui rezolvare permite aflarea cheii secrete.

Securitatea cifrurilor bloc (3)

Parametri de securitate ai unui cifru bloc

- Determină complexitatea algoritmilor de atac, deci rezistența la atacuri.
- Principalii parametri: *lungimea cheii, k (biți), lungimea blocului, n (biți)*.
- Trebuie alese valori suficient de mari pentru ca atacurile să devină practic imposibile pentru un adversar cu resurse limitate.

Exemplu: Căutarea exhaustivă a cheii cifrului bloc

- Cel mai important atac prin forță brută. Recompensă maximă.

  Poate fi efectuat offline (fără interacțiune cu victima), alocând resurse considerabile (calcul paralel masiv folosind hardware dedicat).
- Complexitate $O(2^k) \Rightarrow$ Este contracarat alegând k suficient de mare.

Valori ale parametrilor recomandate în prezent

- Cheie: $k \geq 128$ biți. Bloc: $n \geq 128$ biți.
- Compromis de proiectare (trade-off): prag de securitate (pentru un anumit orizont de timp) și performanțe cât mai bune.

Securitatea cifrurilor bloc (4)

Cerință cantitativă de securitate

- Un cifru bloc trebuie proiectat astfel încât nici un atac criptanalitic să nu fie mai eficient decât atacul prin căutarea exhaustivă a cheii.
- Deci, pentru un cifru bloc cu cheie de k biți, complexitatea celui mai eficient atac ar trebui să fie $O(2^k)$.

Garanții de securitate oferite de cifrurile bloc

- Nu se poate demonstra matematic faptul că nu există nici un atac criptanalitic mai eficient decât căutarea exhaustivă.
- Se evaluează *rezistența la toate atacurile criptanalitice cunoscute* (criptanaliză liniară, diferențială, algebrică, etc.).
- Garanțiile de securitate se degradează (lent) în timp:
  - Progres tehnologic (microelectronică): reduce timpul/costul necesar atacurilor prin forță brută.
  - Ameliorări ale atacurilor criptanalitice.

  $\Rightarrow$ Chei mai lungi, noi algoritmi.

Exemplu: AES

Advanced Encryption Standard (AES)

- Parametri: n = 128 biți; k = 128 / 192 / 256 biți (trei variante).
- Algoritm standard dominant, publicat în 2001 de NIST (SUA), în urma unui proces public de analiză și selecție care a durat aproape 5 ani.

Procesul de elaborare și standardizare a cifrului bloc AES

- 1996: Cifrul bloc DES (n = 64, k = 56), standardizat de NIST în 1977, nu mai este considerat sigur, în special din cauza cheii prea scurte.
- 1997: NIST solicită propuneri de algoritmi pentru standardizarea unui nou cifru bloc, cu parametrii: n = 128, k = 128 / 192 / 256.
- 1998: NIST acceptă 15 propuneri de algoritmi și lansează procesul public de analiză și selecție, desfășurat în 2 runde, 1998-2000.
- 1999: După prima rundă, rămân 5 algoritmi pentru selecția finală.
- 2000: NIST selectează algoritmul Rijndael (Daemen și Rijmen, Belgia).
- 2001: NIST publică noul algoritm AES (FIPS PUB 197).