

Securitate perfectă

- ▶ Primul curs: Sisteme de criptare istorice (substituție, Vigenere etc.) care pot fi sparte cu **efort computațional foarte mic**
- ▶ Cursul de azi: Scheme perfect sigure care rezistă în fața unui adversar cu **putere computațională nelimitată**
- ▶ Însă...limitările sunt inevitabile

Securitate perfectă (Shannon 1949)

Definiție

O schemă de criptare peste un spațiu al mesajelor \mathcal{M} este perfect sigură dacă pentru orice probabilitate de distribuție peste \mathcal{M} , pentru orice mesaj $m \in \mathcal{M}$ și orice text criptat c pentru care $Pr[C = c] > 0$, următoarea egalitate este îndeplinită:

$$Pr[M = m|C = c] = Pr[M = m]$$

- ▶ $Pr[M = m]$ - probabilitatea *a priori* ca Alice să aleagă mesajul m ;
- ▶ $Pr[M = m|C = c]$ - probabilitatea *a posteriori* ca Alice să aleagă mesajul m , chiar dacă textul criptat c a fost văzut ;
- ▶ **securitate perfectă** - dacă Oscar afla textul criptat nu are nici un fel de informație în plus decât dacă nu l-ar fi aflat.

Securitate perfectă (Shannon 1949)

Definiție echivalentă

O schemă de criptare (Enc, Dec) este perfect sigură dacă pentru orice mesaje $m_0, m_1 \in \mathcal{M}$ cu $|m_0| = |m_1|$ și $\forall c \in \mathcal{C}$ următoarea egalitate este îndeplinită:

$$Pr[M = m_0|C = c] = Pr[M = m_1|C = c]$$

unde $k \in \mathcal{K}$ este o cheie aleasă uniform.

- ▶ fiind dat un text criptat, este imposibil de ghicit dacă textul clar este m_0 sau m_1
- ▶ cel mai puternic adversar nu poate deduce nimic despre textul clar dat fiind textul criptat

Un exemplu de cifru sigur - One Time Pad (OTP)

- ▶ Patentat în 1917 de Vernam (mai poartă denumirea de Cifru Vernam)
- ▶ Algoritmul:
 1. Fie $l > 0$ iar $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1\}^l$
 2. Cheia k se alege cu distribuție uniformă din spațiul cheilor \mathcal{K}
 3. **Enc:** dată o cheie $k \in \{0, 1\}^l$ și un mesaj $m \in \{0, 1\}^l$, întoarce $c = k \oplus m$.
 4. **Dec:** dată o cheie $k \in \{0, 1\}^l$ și un mesaj criptat $c \in \{0, 1\}^l$, întoarce $m = k \oplus c$.

Un exemplu de cifru sigur - One Time Pad (OTP)

mesaj:	0	1	1	0	0	1	1	1	1	\oplus
cheie:	1	0	1	1	0	0	1	1	0	
text criptat:	1	1	0	1	0	1	0	0	1	

- ▶ **avantaj** - criptare și decriptare rapide
- ▶ **dezavantaj** - cheia foarte lungă (la fel de lungă precum textul clar)
- ▶ Este OTP sigur?

Teoremă

Schema de criptare OTP este perfect sigură.

- ▶ securitatea perfectă nu este imposibilă dar...
- ▶ cheia trebuie să fie la fel de lungă precum mesajul
- ▶ inconveniente practice (stocare, transmitere)
- ▶ cheia trebuie să fie folosită o singură dată - **one time pad** - de ce?

Exercițiu Ce se întâmplă dacă folosim o aceeași cheie de două ori cu sistemul OTP ?

Limitările securității perfecte - optimalitate OTP

Teoremă

Fie o schemă (Enc, Dec) de criptare perfect sigură peste un spațiu al mesajelor \mathcal{M} și un spațiu al cheilor \mathcal{K} . Atunci $|\mathcal{K}| \geq |\mathcal{M}|$.

Demonstrație

Intuitie:

- ▶ Pentru orice text criptat, se încearcă decriptarea lui cu toate cheile posibile din \mathcal{K} și se obține o listă de cel mult $|\mathcal{K}|$ elemente
- ▶ Dacă $|\mathcal{K}| < |\mathcal{M}|$ unele mesaje nu sunt pe listă - contradicție cu securitatea perfectă (vezi definiția)

Securitate perfectă vs. Criptografie computațională

- ▶ Am văzut scheme de criptare care pot fi demonstrate ca fiind sigure în prezența unui adversar cu putere computațională nelimitată;
- ▶ Se mai numesc și **informational-teoretic sigure**;
- ▶ Adversarul nu are suficientă informație pentru a efectua un atac;
- ▶ Majoritatea construcțiilor criptografice moderne → **securitate computațională**;
- ▶ Schemele moderne *pot fi sparte* dacă un atacator are la dispoziție suficient spațiu și putere de calcul.

Securitate perfectă vs. Criptografie computațională

- ▶ **Securitatea computațională** mai slabă decât **securitatea informațional-teoretică**;
- ▶ Prima se bazează pe prezumpții de securitate; a doua este necondiționată;
- ▶ **Întrebare**: de ce renunțăm la securitatea perfectă?
- ▶ **Raspuns**: datorită limitărilor practice!
- ▶ Preferăm un compromis de securitate pentru a obține construcții practice.

Securitate computațională

- ▶ **Ideea de bază**: principiul 1 al lui Kerckhoffs
Un cifru trebuie să fie practic, dacă nu matematic, indescifrabil.
- ▶ Sunt de interes mai mare schemele care **practic nu pot fi sparte** deși nu beneficiază de securitate perfectă;
 1. Adversari **limitați computațional/eficienti/timp polinomial**
Exemplu: Un atacator care realizează un atac prin forta brută peste spațiul cheilor și testează o cheie/ciclu de ceas
 - ▶ calculator desktop - se pot testa aprox. 2^{57} chei/an
 - ▶ supercalculator - se pot testa aprox. 2^{80} chei/an
 - ▶ supercalculator, vârsta universului - 2^{112} chei

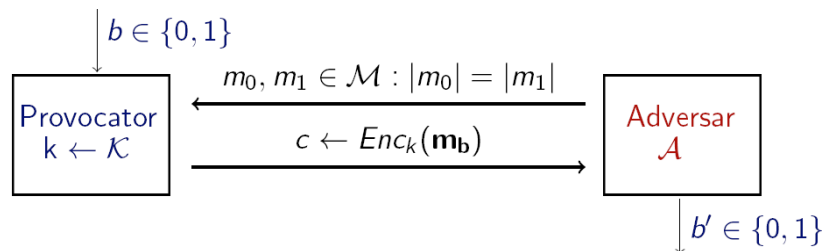
Securitate computațională

2. Adversarii pot efectua un atac cu succes cu o **probabilitate foarte mică**;
Exemplu: un adversar află textul clar cu probabilitate 2^{-60} într-un an
 - ▶ sunt șanse mai mari ca Alice și Bob să fie loviți de fulger în aceeași perioadă de timp
 - ▶ un eveniment cu prob. $2^{60}/\text{sec.}$ se produce o dată la un miliard de ani

Indistinctibilitate perfectă

- ▶ Pentru securitatea perfectă am dat două definiții echivalente, a doua subliniază ideea de **indistinctibilitate**: adversarul nu poate distinge între criptările a două mesaje diferite
- ▶ Vom defini indistinctibilitatea pe baza unui experiment $\text{Priv}_{\mathcal{A},\pi}^{\text{eav}}$ unde $\pi = (\text{Enc}, \text{Dec})$ este schema de criptare
- ▶ Personaje participante: **adversarul** \mathcal{A} care încearcă să spargă schema și un **provocator (challenger)**.
- ▶ Trebuie să definim capacitățile adversarului: el poate vedea **un singur text criptat cu o anume cheie**, fiind un adversar *pasiv* care poate rula atacuri în timp polinomial, și nu are nici o altă interacțiune cu Alice sau Bob

Experimentul $Priv_{\mathcal{A},\pi}^{eav}$



- Output-ul experimentului este 1 dacă $b' = b$ și 0 altfel. Dacă $Priv_{\mathcal{A},\pi}^{eav} = 1$, spunem că \mathcal{A} a efectuat experimentul cu succes.
- Schema π este *perfect indistinctibilă* dacă

$$Pr[Priv_{\mathcal{A},\pi}^{eav}(n) = 1] = \frac{1}{2}$$

- Reamintim ca **indistinctibilitatea perfectă** este doar o definiție alternativă pentru **securitatea perfectă**

Securitate computațională concretă

O schemă de criptare este (t, ϵ) -indistinctibilă dacă orice adversar care rulează în timp cel mult t

$$Pr[Priv_{\mathcal{A},\pi}^{eav} = 1] \leq \frac{1}{2} + \epsilon$$

- probabilitatea de succes a adversarului $\leq \epsilon$
- adversarul rulează în timp $\leq t$
- dezavantaj: am dori să avem scheme în care utilizatorul își poate ajusta nivelul de securitate dorit

Securitate computațională asimptotică

- parametru de securitate n atât pentru schema de criptare cât și pentru părțile oneste și adversar
 - poate fi văzut ca lungimea cheii
 - timpul în care rulează adversarul și probabilitatea lui de succes sunt funcții de n
 - este cunoscut adversarului
 - permite utilizatorului să își aleagă nivelul de securitate dorit - este fixat la momentul inițializării schemei de criptare

Securitate computațională asimptotică

- Se impune o nouă modalitate de a defini securitatea:

Definiție

O schemă de criptare este *indistinctibilă* dacă pentru orice adversar PPT, există o funcție neglijabilă ϵ așa încât

$$Pr[Priv_{\mathcal{A},\pi}^{eav}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

Neglijabil și ne-neglijabil

- ▶ **în practică:** ϵ este scalar și
 - ▶ ϵ ne-neglijabil dacă $\epsilon \geq 1/2^{30}$
 - ▶ ϵ neglijabil dacă $\epsilon \leq 1/2^{80}$
- ▶ **în teorie:** ϵ este funcție $\epsilon : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ și $p(n)$ este o funcție polinomială în n (ex.: $p(n) = n^d$, d constantă)
 - ▶ ϵ ne-neglijabilă în n dacă $\exists p(n) : \epsilon(n) > 1/p(n)$
 - ▶ ϵ neglijabilă în n dacă $\forall p(n), \exists n_d$ a.î. $\forall n \geq n_d : \epsilon(n) \leq 1/p(n)$

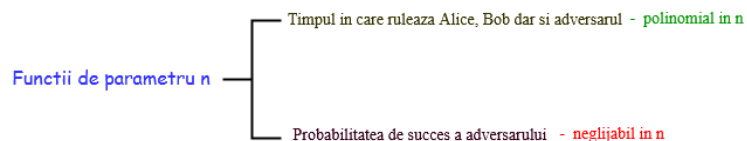
Valori concrete pentru n

Presupunem ca pentru o schema de criptare concreta, un adversar care ruleaza in timp n^3 minute reuseste sa sparga schema cu probabilitate $2^{40} * 2^{-n}$.

- ▶ Ce valori alegem pentru n la implementare?
 - ▶ pentru $n \leq 40$, atunci un adversar care rulează în 40^3 minute (adica 6 saptamani) sparge schema cu probabilitate 1
 - ▶ pentru $n = 50$, atunci un adversar care rulează în 50^3 minute (adica aproximativ 3 luni) sparge schema cu probabilitate aprox. $1/1000$ (ar putea sa nu fire acceptabil)
 - ▶ pentru $n = 500$, atunci un adversar care rulează în 200 de ani sparge schema cu probabilitate aprox. 2^{-500} (acceptabil)

Important de reținut!

- ▶ Parametrul de securitate n este public cunoscut si parte din schema
- ▶ Input-urile pentru toti algoritmi, inclusiv adversarul, sunt polinomiale in n
- ▶ Tipic, n este lungimea cheii secrete (de ex. $n = 128, 256$ etc.)



Criptarea simetrică - redefinită

Definiție

Un **sistem de criptare simetric** definit peste $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, cu:

- ▶ \mathcal{K} = spațiul cheilor
- ▶ \mathcal{M} = spațiul textelor clare (mesaje)
- ▶ \mathcal{C} = spațiul textelor criptate

este un triplet $(\text{Gen}, \text{Enc}, \text{Dec})$, unde:

1. **Gen(1^n):** este algoritmul probabilistic de generare a cheilor care întoarce o cheie k conform unei distribuții
2. **Enc:** primește o cheie k și un mesaj $m \in \{0, 1\}^*$ și întoarce $c \leftarrow \text{Enc}_k(m)$
3. **Dec:** primește cheia k și textul criptat și întoarce m sau "eroare".