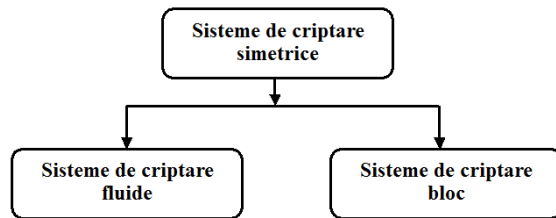
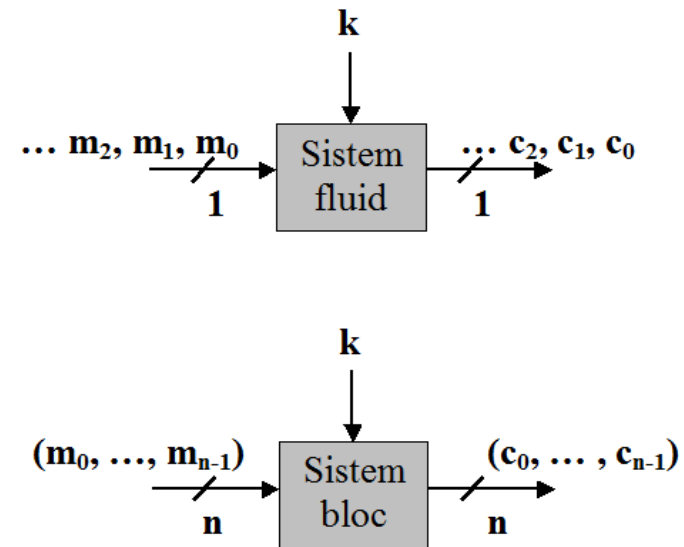


Criptografia simetrică

- ▶ Am studiat sisteme simetrice care criptează **bit cu bit** - sisteme de criptare fluide;
- ▶ Vom studia sisteme simetrice care criptează **câte n biți simultan** - sisteme de criptare bloc;



Sisteme bloc vs. sisteme fluide



Sisteme bloc vs. sisteme fluide

... d.p.d.v. al modului de criptare:

Sisteme fluide

- ▶ criptarea biților se realizează **individual**
- ▶ criptarea unui bit din textul clar este **independentă** de orice alt bit din textul clar

Sisteme bloc

- ▶ criptarea se realizează în **blocuri** de câte n biți
- ▶ criptarea unui bit din textul clar este **dependentă** de biții din textul clar care aparțin aceluiași bloc

Sisteme bloc vs. sisteme fluide

... d.p.d.v. *tradițional*, în practică:

Sisteme fluide

- ▶ necesități computaționale reduse
- ▶ utilizare: telefoane mobile, dispozitive încorporate, PDA
- ▶ par să fie mai puțin sigure, multe sunt sparte

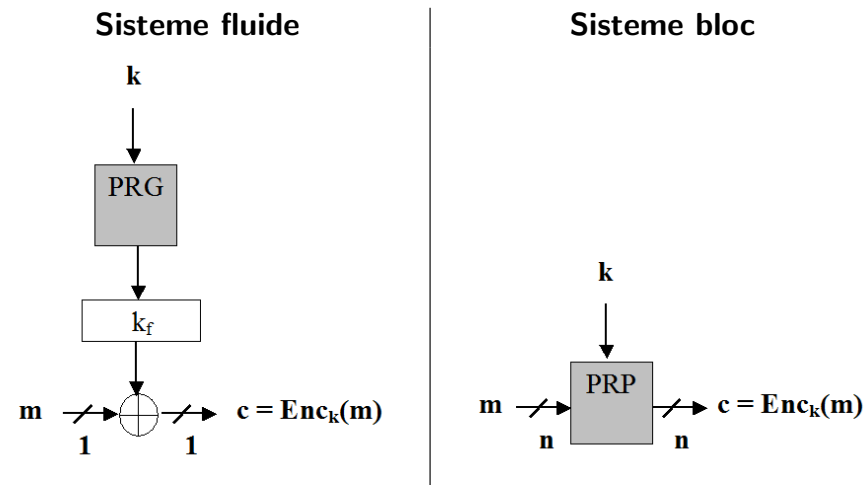
Sisteme bloc

- ▶ necesități computaționale mai avansate
- ▶ utilizare: internet
- ▶ par să fie mai sigure, prezintă încredere mai mare

Sisteme bloc

- ▶ Introducem noțiunea de **permutare pseudoaleatoare** sau **PRP** (*PseudoRandom Permutation*)
 - ▶ În analogie cu ce știm deja:
 - ▶ **PRP** sunt necesare pentru construcția **sistemelor bloc**
- așa cum*
- ▶ **PRG** sunt necesare pentru construcția **sistemelor fluide**

Sisteme bloc



PRP

- ▶ Ramâne să definim noțiunea de **permutare pseudoaleatoare** sau **PRP** (*PseudoRandom Permutation*);
- ▶ Acesta este o funcție **deterministă** și **bijectivă** care pentru o cheie fixată produce la ieșire o **permutare** a intrării ...
- ▶ ... **indistinctibilă** față de o permutare aleatoare;
- ▶ În plus, atât funcția cât și inversa sa sunt **eficient calculabile**.

PRP

Definiție

O **permutare pseudoaleatoare** definită peste $(\mathcal{K}, \mathcal{X})$ este o funcție **bijectivă**

$$F : \mathcal{X} \times \mathcal{K} \rightarrow \mathcal{X}$$

care satisface următoarele proprietăți:

1. **Eficiență:** $\forall k \in \mathcal{K}, x \in \mathcal{X}, \exists$ algoritmi determinați polinomiali care calculează $F_k(x)$ și $F_k^{-1}(x)$
2. **Pseudoaleatorism:** \forall algoritm PPT \mathcal{D} , \exists o funcție neglijabilă negl a.î.:

$$|Pr[D(r) = 1] - Pr[D(F_k(\cdot)) = 1]| \leq \text{negl}(n)$$

$$\text{unde } r \leftarrow^R \text{Perm}(X), k \leftarrow^R \mathcal{K}$$

Notății

- ▶ $F_k(x) = F(k, x)$
o cheie este în general (aleator) aleasă și apoi fixată
- ▶ $\text{Perm}(X) =$ mulțimea tuturor funcțiilor bijective de la \mathcal{X} la \mathcal{X}
- ▶ $\mathcal{X} = \{0, 1\}^n$
- ▶ $\mathcal{D} = \text{Distinguisher}$ care are acces la *oracolul* de evaluare a funcției

PRF

- ▶ Introducem noțiunea de **funcție pseudoaleatoare** sau **PRF** (*PseudoRandom Function*)...
- ▶ ... ca o generalizare a noțiunii de **permutare pseudoaleatoare**;
- ▶ Acesta este o funcție **cu cheie** care este **indistinctibilă** față de o funcție aleatoare (cu același domeniu și mulțime de valori).

PRF

Definiție

O **funcție pseudoaleatoare** definită peste $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ este o funcție bijectivă

$$F : \mathcal{X} \times \mathcal{K} \rightarrow \mathcal{Y}$$

care satisface următoarele proprietăți:

1. **Eficiență:** $\forall k \in \mathcal{K}, x \in \mathcal{X}, \exists$ *algorithm determinist polinomial* care calculează $F_k(x)$
2. **Pseudoaleatorism:** \forall *algorithm PPT* \mathcal{D} , \exists o funcție neglijabilă *negl* a.î.:

$$|\Pr[D(r) = 1] - \Pr[D(F_k(\cdot)) = 1]| \leq \text{negl}(n)$$

$$\text{unde } r \leftarrow^R \text{Func}(\mathcal{X}, \mathcal{Y}), k \leftarrow^R \mathcal{K}$$

Notății

- ▶ $F_k(x) = F(k, x)$
o cheie este în general (aleator) aleasă și apoi fixată
- ▶ $\text{Func}(X, Y) =$ mulțimea funcțiilor de la \mathcal{X} la \mathcal{Y}
- ▶ $\mathcal{X} = \{0, 1\}^n, \mathcal{Y} = \{0, 1\}^n$
considerăm în general că *PRF păstrează lungimea*
- ▶ $\mathcal{D} = \text{Distinguisher}$ care are acces la *oracolul* de evaluare a funcției

$PRP \subseteq PRF$

- ▶ **Întrebare:** De ce PRF poate fi privită ca o generalizare a PRP ?
- ▶ **Răspuns:** PRP este PRF care satisface:
 1. $\mathcal{X} = \mathcal{Y}$
 2. este inversabilă
 3. calculul funcției inverse este eficient

Construcții

- ▶ **$PRF \Rightarrow PRG$**
Pornind de la PRF se poate construi PRG
- ▶ **$PRG \Rightarrow PRF$**
Pornind de la PRG se poate construi PRF
- ▶ **$PRP \Rightarrow PRF$**
Pornind de la PRP se poate construi PRF
- ▶ **$PRF \Rightarrow PRP$**
Pornind de la PRF se poate construi PRP

Întrebare: Care dintre aceste construcții este trivială?

Construcții

Răspuns: $PRP \Rightarrow PRF$

PRP este o particularizare a $PRF : \mathcal{X} \times \mathcal{K} \rightarrow \mathcal{Y}$ care satisface:

1. $\mathcal{X} = \mathcal{Y}$
2. este inversabilă
3. calculul funcției inverse este eficient

Construcții

- ▶ **$PRF \Rightarrow PRG$**
Pornind de la PRF se poate construi PRG
- ▶ **$PRG \Rightarrow PRF$**
Pornind de la PRG se poate construi PRF
- ▶ **$PRP \Rightarrow PRF$ ✓**
Pornind de la PRP se poate construi PRF
- ▶ **$PRF \Rightarrow PRP$**
Pornind de la PRF se poate construi PRP

$PRF \Rightarrow PRG$

- ▶ Considerăm $F : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ PRF;
- ▶ Construim $G : \mathcal{K} \rightarrow \{0, 1\}^{nl}$ PRG sigur:

$$G(k) = F_k(0) || F_k(1) || \dots || F_k(l-1)$$

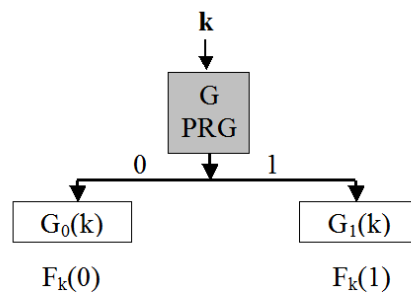
- ▶ **Întrebare:** De ce este G sigur?
- ▶ **Răspuns:** $F_k(\cdot)$ este *indistingibilă* față de o funcție aleatoare
 $\Rightarrow G(k)$ este *indistingibilă* față de o secvență aleatoare de lungime ln .
- ▶ **Avantaj:** Construcția este *paralelizabilă*

Construcții

- ▶ **PRF \Rightarrow PRG** ✓
Pornind de la PRF se poate construi PRG
- ▶ **PRG \Rightarrow PRF**
Pornind de la PRG se poate construi PRF
- ▶ **PRP \Rightarrow PRF** ✓
Pornind de la PRP se poate construi PRF
- ▶ **PRF \Rightarrow PRP**
Pornind de la PRF se poate construi PRP

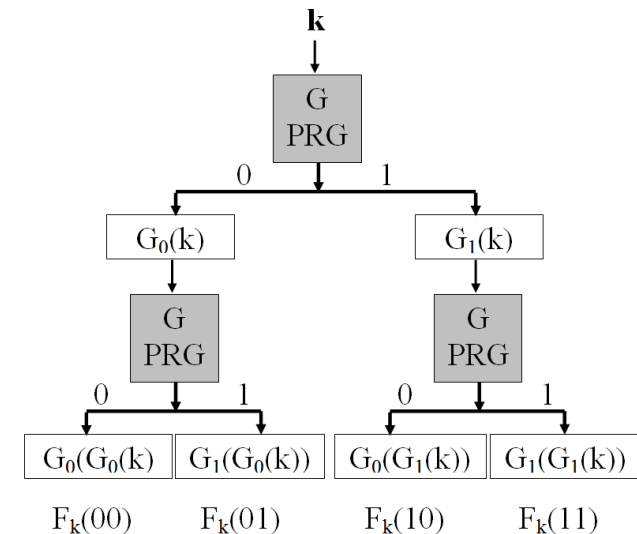
$PRG \Rightarrow PRF$

- ▶ Construcția pentru un singur bit de intrare...



$PRG \Rightarrow PRF$

- ▶ ...se poate generaliza pentru un număr oarecare de biți



PRG \Rightarrow PRF

- ▶ Construcția poate fi reprezentată ca un arbore binar cu cheia k rădăcină;
- ▶ Pentru un nod de valoare k' , copilul stâng ia valoarea $G_0(k')$ și copilul drept ia valoarea $G_1(k')$;
- ▶ Valoarea funcției $F_k(x) = F_k(x_0, \dots, x_{n-1})$ este obținută prin parcurgerea arborelui în funcție de x ;
- ▶ Adâncimea arborelui este *liniară* în n (n);
- ▶ Dimensiunea arborelui este *exponențială* în n (2^n);
- ▶ NU se utilizează în practică din cauza performanței scăzute.

Construcții

- ▶ **PRF \Rightarrow PRG** ✓
Pornind de la PRF se poate construi PRG
- ▶ **PRG \Rightarrow PRF** ✓
Pornind de la PRG se poate construi PRF
- ▶ **PRP \Rightarrow PRF** ✓
Pornind de la PRP se poate construi PRF
- ▶ **PRF \Rightarrow PRP**
Pornind de la PRF se poate construi PRP

PRF \Rightarrow PRP

Teoremă (Luby-Rackoff 5)

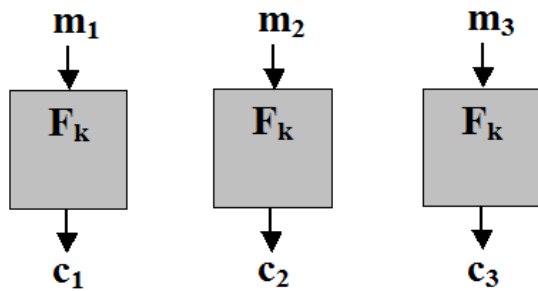
Dacă $F : \mathcal{K} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ este PRF, se poate construi $F' : \mathcal{K} \times \{0, 1\}^2 \rightarrow \{0, 1\}^2$ PRP.

- ▶ Construcția folosește runde **Feistel**, pe care le vom prezenta într-un curs ulterior.

Moduri de utilizare

- ▶ Să continuăm cu ceva mai practic...
- ▶ **Întrebare:** Ce se întâmplă dacă lungimea mesajului clar este **mai mică** decât dimensiunea unui bloc?
- ▶ **Răspuns:** Se completează cu biți: **1 0 ... 0**;
- ▶ **Întrebare:** Ce se întâmplă dacă lungimea mesajului clar este **mai mare** decât lungimea unui bloc?
- ▶ **Răspuns:** Se utilizează un **mod de operare** (ECB, CBC, OFB, CTR);
- ▶ Notăm cu F_k un sistem de criptare bloc (i.e. PRP) cu cheia k fixată.

Modul ECB (Electronic Code Book)



Modul ECB (Electronic Code Book)

- ▶ Pare modul cel mai **natural** de a cripta mai multe blocuri;
- ▶ Pentru decriptare, F_k trebuie să fie **inversabilă**;
- ▶ Este **paralelizabil**;
- ▶ Este **determinist**, deci este **nesigur**;
- ▶ **Întrebare**: Ce informații poate să ofere modul de criptare ECB unui adversar pasiv?
- ▶ **Răspuns**: Un adversar pasiv detectează repetarea unui bloc de text clar pentru că se repetă blocul criptat corespunzător;
- ▶ Modul ECB **NU** trebuie utilizat în practică!

Modul ECB (Electronic Code Book)

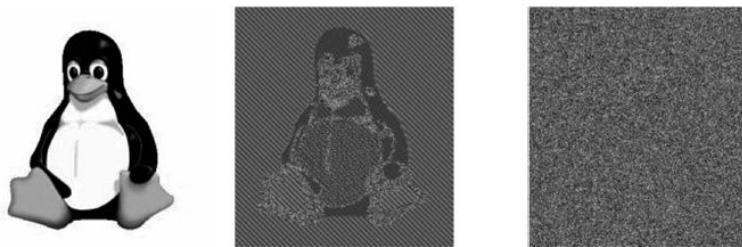
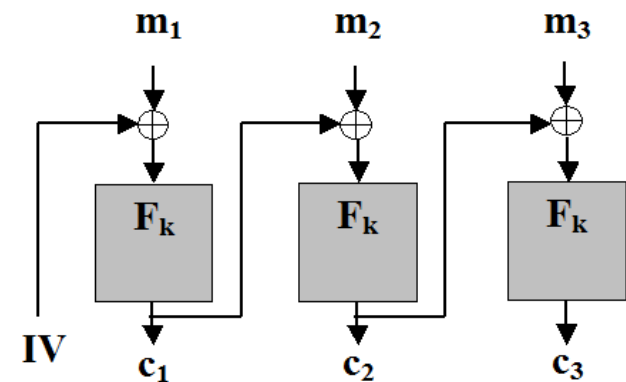


Figure: Imagine preluată de pe <https://en.wikipedia.org/>

Figura din mijloc este criptarea imaginii din stânga în modul ECB. În dreapta este aceeași imagine criptată folosind un mod sigur.

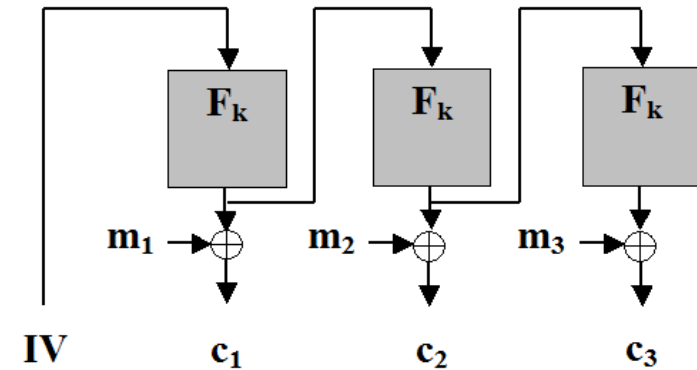
Modul CBC (Cipher Block Chaining)



Modul CBC (Cipher Block Chaining)

- ▶ IV este ales în mod aleator la criptare;
- ▶ IV se transmite în clar pentru ca este necesar la decriptare;
- ▶ Pentru decriptare, F_k trebuie să fie **inversabilă**;
- ▶ Este **secvențial**, un dezavantaj major dacă se poate utiliza procesarea paralelă.

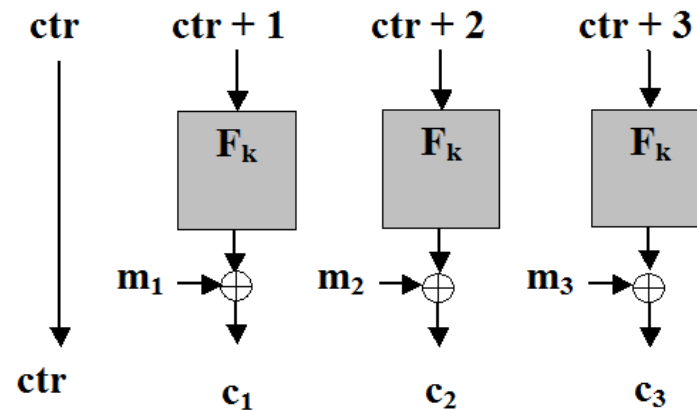
Modul OFB (Output FeedBack)



Modul OFB (Output FeedBack)

- ▶ Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- ▶ IV este ales în mod aleator la criptare;
- ▶ IV se transmite în clar pentru ca este necesar la decriptare;
- ▶ F_k nu trebuie neapărat să fie inversabilă;
- ▶ Este **secvențial**, însă secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.

Modul CTR (Counter)



Modul CTR (Counter)

- ▶ Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- ▶ ctr este ales în mod aleator la criptare;
- ▶ ctr se transmite în clar pentru ca este necesar la decriptare;
- ▶ F_k nu trebuie neapărat să fie inversabilă;
- ▶ Este **paralelizabil**;
- ▶ În plus, secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.

Modul CTR (Counter)

- ▶ Generează o secvență pseudoaleatoare care se XOR-ează mesajului clar;
- ▶ ctr este ales în mod aleator la criptare și se transmite în clar pentru ca este necesar la decriptare;
- ▶ Este **paralelizabil**; F_k nu trebuie neapărat să fie inversabilă;
- ▶ În plus, secvența pseudoaleatoare poate fi pre-procesată anterior decriptării.
- ▶ CTR poate fi văzut și ca un sistem fluid nesincronizat:
 - ▶ pentru criptarea unui mesaj de lungime $l < 2^{n/4}$ blocuri, se alege un IV uniform din $\{0, 1\}^{2n/4}$
 - ▶ fiecare bloc de text criptat este calculat $y_i = F_k(IV || i)$ unde i este codificat ca un string pe $n/4$ biți

Câteva considerații practice

- ▶ modurile CTR, OFB și CBC sunt CPA-sigure
- ▶ modurile CBC, OFB și CTR folosesc un IV uniform aleator - asigură faptul că F_k este mereu evaluat pe intrări diferite (previne situația în care adversarul afla informații la vederea de intrări identice)
- ▶ CTR - IV ales uniform de lungime $3n/4$ înseamnă că IV se repetă după criptarea aprox. $q(n) = 2^{2n/8}$ mesaje
- ▶ Dacă $n = 64$ atunci $q \approx 17.000.000$ ceea ce e puțin pentru zilele noastre
- ▶ Dacă $n = 128$ și vrem să folosim CTR având garanția că IV se repetă cu probabilitate cel mult 2^{-40} , rezultă $q \approx 2^{28}$ mesaje (calculând q din $\frac{q^2}{2^{3n/4}+1} \leq 2^{-40}$)

Câteva considerații practice - IV folosit greșit

- ▶ Ce se întâmplă dacă IV se repetă?
- ▶ Pentru modurile OFB și CTR, întregul stream pseudoaleator (cu care se face xor pe mesaj) se repetă
- ▶ Dacă IV nu este uniform aleator (deci este predictibil), CTR este sigur dar CBC nu este sigur.