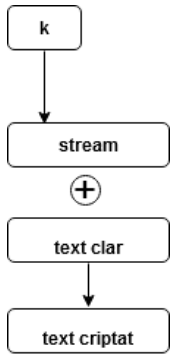
Recapitulare - PRG

- am definit generatoarele de numere pseudo-aleatoare, am văzut că ele sunt vulnerabile în fața unui adversar nelimitat computațional și că putem construi sisteme de criptare sigure bazate pe ele
- Intrebare PRG există?
- Răspuns: nu putem demonstra necondiționat, dar credem cu tărie că există
- Explicație: d.p.d.v. teoretic, putem construi PRG condiționat, bazat pe existența funcțiilor one-way
- In practică, construcțiile existente pentru PRG nu pot fi demonstrate ca fiind sigure, dar credem că sunt întrucât nu se cunosc algoritmi "distinguisher" ($\mathcal{D}$) eficienți $\rightarrow$ prezumpție: PRG există.

PRG-uri în practică

- Dezavantaj: PRG, asa cum le-am definit, produc tot output-ul odată și acesta este de lungime fixă
- In practică, PRG-urile sunt instanțiate cu sisteme de criptare fluide

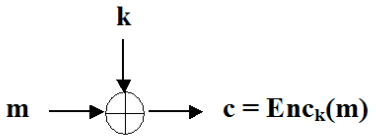


Sisteme fluide

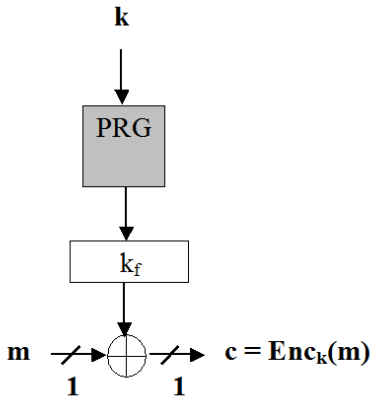
- sistemele fluide produc biții de output (pseudo-aleatori) gradual și la cerere, fiind mai eficiente și flexibile
- criptarea cu un sistem fluid presupune 2 faze:
  - **Faza 1**: se generează o secvență pseudoaleatoare de biți, folosind un generator de numere pseudoaleatoare (PRG)
  - **Faza 2**: secvența obținută se XOR-ează cu mesajul clar
- Atenție! De multe ori când ne referim la un sistem de criptare fluid considerăm doar Faza 1

Sisteme fluide

OTP (One Time Pad)



Sisteme fluide



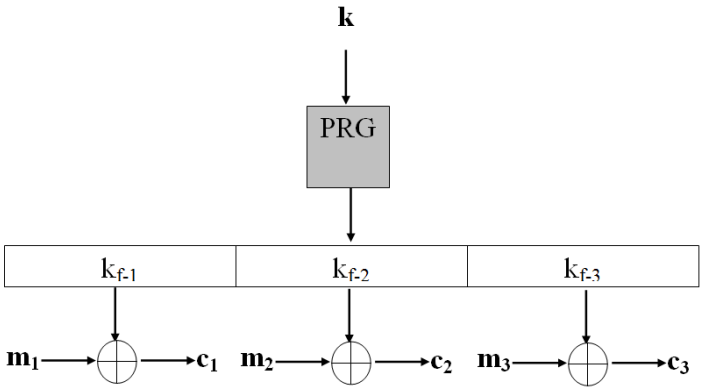
Securitate - interceptare multiplă

- Un sistem de criptare fluid în varianta prezentată este determinist: *unui text clar îi corespunde întotdeauna același mesaj criptat*;
- În consecință, utilizarea unui sistem fluid în forma prezentată pentru criptarea mai multor mesaje (cu aceeași cheie) este nesigură;
- Un sistem de criptare fluid se folosește în practică în 2 moduri: sincronizat și nesincronizat.

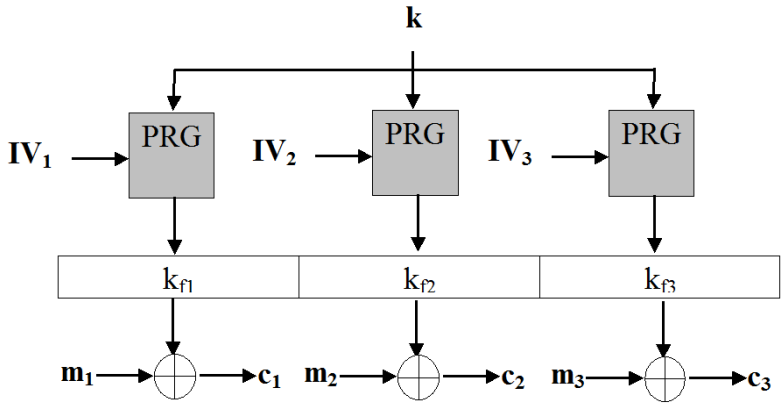
Moduri de utilizare

- modul sincronizat: partenerii de comunicație folosesc pentru criptarea mesajelor *părți succesive* ale secvenței pseudoaleatoare generate;
- modul nesincronizat: partenerii de comunicație folosesc pentru criptarea mesajelor secvențe pseudoaleatoare *diferite*.

Modul sincronizat



Modul nesincronizat



Moduri de utilizare

Modul sincronizat

- mesajele sunt criptate în mod succesiv (participanții trebuie să știe care părți au fost deja folosite)
- necesită păstrarea stării
- mesajele succesive pot fi percepute ca un *singur mesaj clar lung*, obținut prin concatenarea measajelor succesive
- se pretează unei singure sesiuni de comunicații

Modul nesincronizat

- mesajele sunt criptate în mod independent
- NU necesită păstrarea stării
- valorile $IV_1, IV_2, \ldots$ sunt alese uniform aleator pentru fiecare mesaj transmis
- valorile $IV_1, IV_2, \ldots$ (dar si IV în modul sincronizat) fac parte din mesajul criptat (sunt necesare pentru decriptare)

Proprietăți necesare ale PRG în modul nesincronizat

Fie $G(s, IV)$ un PRG cu 2 intrări:

- s = *seed*
- IV = *Initialization Vector*

PRG trebuie să se satisfacă (cel puțin):

1. $G(s, IV)$ este o secvență pseudoaleatoare chiar dacă IV este public (i.e. securitatea lui G constă în securitatea lui s);
2. dacă IV_1 și IV_2 sunt valori uniform aleatoare, atunci $G(s, IV_1)$ și $G(s, IV_2)$ sunt indistinctibile.

Exemple

- RC4 (Ron"s Cipher 4):
  - definit de R.Rivest, în 1987
  - utilizat în WEP
  - inițial secret !
- WEP (Wired Equivalent Privacy):
  - standard IEEE 802.11, 1999 (rețele fără fir)
  - înlocuit în 2003 de WPA (Wi-Fi Protected Access), 2004 WPA2 - IEEE 802.11i

Exemple

- A5/1:
  - definit în 1987 pentru Europa și SUA
  - A5/2 definit în 1989 ca o variantă mai slabă pentru alte zone geografice
  - utilizat în rețelele de telefonie mobilă GSM
  - inițial secret !
- SEAL (Software-Optimized Encryption Algorithm)
  - definit de D.Coppersmith și P.Rogaway, în 1993
  - prezintă o implementare foarte eficientă pe procesoarele pe 32 de biți
  - versiunea curentă (SEAL 3.0) este patentată IBM