True Randomness in Computers

De la Pseudo-Random la Hardware Random Number Generators

Manea Marius

Universitatea din Bucuresti

June 3, 2025

Agenda

1. Introducere - Experimentul cu Audiența
2. Ce Înseamnă "Random"?
3. Pseudo-Random Number Generators (PRNG)
4. Hardware Random Number Generators
5. Aplicații Practice
6. Provocări și Limitări
7. Concluzie

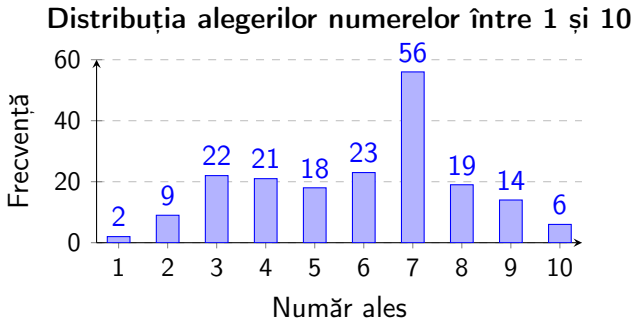
Să începem cu un experiment!

Instrucțiuni

Vă rog să scrieti pe o foaie un număr "random" între 1 și 10.

Să vedem cât de random suntem cu adevărat...

Rezultatele Experimentului



Observație

Oamenii nu sunt buni la generarea de numere random! Avem biasuri predictibile.

Date preluate din sondajul YouGov UK: *"Seven is the most random number"*, 2014

(https://yougov.co.uk/society/articles/51866-seven-is-the-most-random-number)

Definiția Randomness

Definition

Un proces este **random** dacă rezultatele sale nu pot fi prezise cu certitudine, chiar dacă cunoaștem toate condițiile inițiale.

Tipuri de Randomness

- **True Randomness** - bazat pe procese fizice impredictibile
- **Pseudo-Randomness** - algoritmic, deterministic dar aparent random
- **Cryptographically Secure** - pseudo-random dar rezistent la atacuri

Criterii matematice pentru randomness

1. **Uniformitate**: $P(X = x_i) = \frac{1}{n}$ pentru toate valorile
2. **Independență**: $P(X_{i+1}|X_i) = P(X_{i+1})$
3. **Nepredictibilitate**: Imposibil de prezis următoarea valoare

Testul Chi-Square pentru uniformitate

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

unde O_i = frecvența observată, E_i = frecvența așteptată

Nivel de semnificație (α): probabilitatea de a respinge greșit ipoteza de uniformitate.
Valori uzuale: 0.05, 0.01 sau 0.10.

Grade de libertate (df): $df = k - 1$, unde k este numărul de categorii.

Valoare critică: valoarea din tabelul Chi-Square corespunzătoare lui α și df — se compară cu valoarea calculată χ^2:

- Dacă $\chi^2 >$ valoarea critică respingem ipoteza de uniformitate
- Altfel nu avem dovezi să o respingem

Exemplu: Testul Chi-Square pentru Uniformitate

Context

Am generat 60 de valori presupus aleatoare între 1 și 6 (ca un zar). Ne așteptăm ca fiecare valoare să apară de **10 ori** dacă distribuția este uniformă.

Date

Valoare (i)	Frecvență Observată (O_i)	Frecvență Așteptată (E_i)
1	8	10
2	12	10
3	9	10
4	11	10
5	10	10
6	10	10

Calcul Chi-Square

Calcul Chi-Square

$$\chi^2 = \frac{(8-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(9-10)^2}{10} + \frac{(11-10)^2}{10} = 0.4 + 0.4 + 0.1 + 0.1 = \mathbf{1.0}$$

(ultimele două valori sunt 0, deci nu contribuie)

Interpretare

Comparăm $\chi^2 = 1.0$ cu valoarea critică $\chi^2_{crit} = 11.07$ (pentru $df = 5$, $\alpha = 0.05$). **Concluzie:** Nu respingem ipoteza de uniformitate $\Rightarrow$ distribuția pare uniformă.

Linear Congruential Generator (LCG)

Formula LCG

$$X_{n+1} = (aX_n + c) \bmod m$$

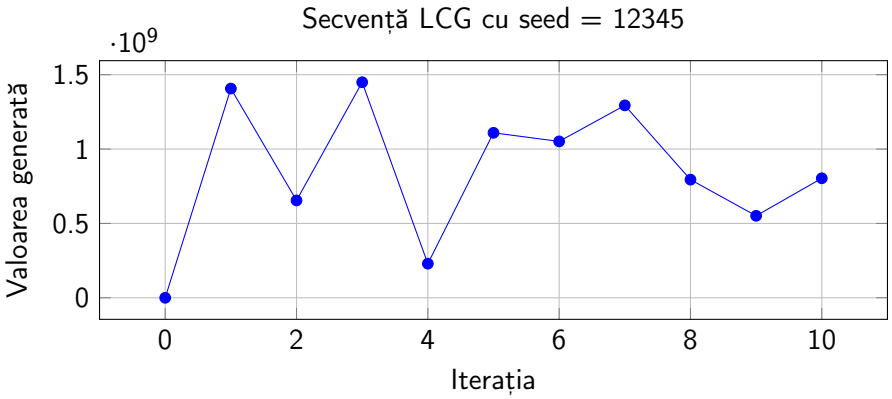
- a = multiplicator
- c = increment
- m = modulus
- X_0 = seed (valoarea inițială)

Example

Pentru $a = 1664525$, $c = 1013904223$, $m = 2^{32}$:

$$X_1 = (1664525 \cdot X_0 + 1013904223) \bmod 2^{32}$$

Exemplu Vizual - LCG



Problema

Dacă cunoști seed-ul și parametrii, poți prezice întreaga secvență!

Exemplu Pas cu Pas - LCG

Parametrii aleși

$a = 1664525$, $c = 1013904223$,
$m = 2^{32} = 4294967296$(*alegem o putere a lui 2 pentru calcul rapid pe computer*), $X_0 = 12345$

Calculele pas cu pas

$$
\begin{aligned}
X_1 &= (1664525 \times 12345 + 1013904223) \bmod 2^{32} && (1)\\
&= (20556463725 + 1013904223) \bmod 4294967296 && (2)\\
&= 21570367948 \bmod 4294967296 && (3)\\
&= 1406932606 && (4)\\
X_2 &= (1664525 \times 1406932606 + 1013904223) \bmod 2^{32} && (5)\\
&= 654583775 && (6)
\end{aligned}
$$

Observație

Fiecare valoare este complet determinată de cea anterioară!

Mersenne Twister

Avantaje

- Perioadă foarte lungă: $2^{19937} - 1$
- Distribuție uniformă în 623 dimensiuni
- Rapid și eficient

Principiul de funcționare

$$X_{k+n} = X_{k+m} \oplus ((X_k | X_{k+1\,\text{upper}}) \cdot A) \tag{7}$$

$$\text{unde } A = \begin{pmatrix} 0 & I_{w-1} \\ a_{w-1} & (a_{w-2}, \ldots, a_0) \end{pmatrix} \tag{8}$$

Limitare

Încă este deterministic - nu este cryptographically secure!

Cum funcționează Mersenne Twister (MT19937)

Structura generală

MT menține un vector de stare $X_0, X_1, \ldots, X_{623}$.
La fiecare pas, creează un nou element folosind:

$$Y = (X_k \& 0\text{x}80000000) + (X_{k+1} \& 0\text{x}7\text{FFFFFFF})$$
$$X_{k+624} = X_{k+397} \oplus (Y \gg 1) \oplus (\text{dacă LSB(Y)} = 1 \text{ atunci } A)$$

Explicația variabilelor

- X_k — elementul curent din vectorul de stare.
- X_{k+1} — următorul element (circular).
- & — operația AND pe biți.
- | — operația OR pe biți.
- $\oplus$ — operația XOR (exclusiv SAU).
- A — o constantă (matrice sau valoare fixă), folosită pentru „twist".
- $Y \gg 1$ — deplasare la dreapta cu 1 bit (echivalent împărțire la 2).
- $LSB(Y)$ — cel mai puțin semnificativ bit.

Surse de Entropie Fizică

Zgomotul Termic (Johnson-Nyquist)

$$\langle V^2 \rangle = 4 k_B T R \Delta f$$

- k_B = constanta Boltzmann (1.38×10^{-23} J/K)
- T = temperatura (K)
- R = rezistența ()
- Δf = lățimea de bandă (Hz)

Sursa de randomness: fluctuațiile aleatoare ale tensiunii cauzate de mișcarea termică a electronilor.

Zgomotul Shot (Poisson)

$$\sigma^2 = 2 q I \Delta f$$

unde q = sarcina electronului, I = curentul mediu

Sursa de randomness: discrepanțele întâmplătoare în fluxul de electroni printr-o barieră (ex: în diode sau tuneluri cuantice).

Exemplu Numeric - Zgomotul Termic

Scenariul: Rezistor la temperatura camerei

- $T = 300$ K (temperatura camerei)
- $R = 10$ k (rezistența)
- $\Delta f = 1$ MHz (lățimea de bandă)
- $k_B = 1.38 \times 10^{-23}$ J/K

Calculul tensiunii de zgomot

$\langle V^2 \rangle = 4k_B TR\Delta f$
$= 4 \times 1.38 \times 10^{-23} \times 300 \times 10^4 \times 10^6$
$= 1.656 \times 10^{-10}$ V^2

$$V_{rms} = \sqrt{1.656 \times 10^{-10}} = 12.87 \text{ V}$$

Aplicație

Această tensiune mică poate fi amplificată și digitizată pentru a genera biți aleatori!

Exemplu Numeric - Zgomotul Shot

Scenariul: Fotodiodă

- $I = 1$ A (curentul mediu)
- $q = 1.6 \times 10^{-19}$ C (sarcina electronului)
- $\Delta f = 1$ MHz (lățimea de bandă)

Calculul curentului de zgomot

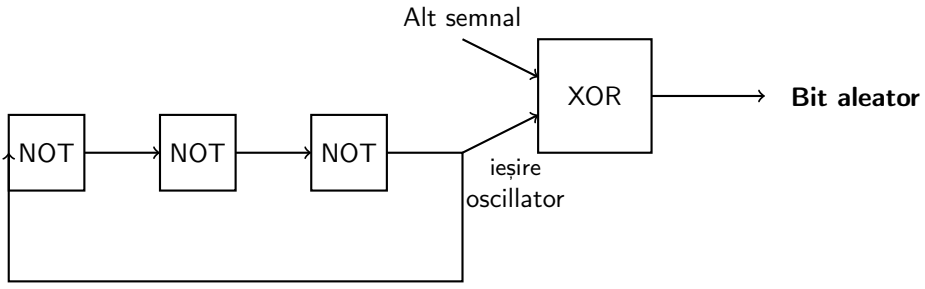
$\sigma^2 = 2qI\Delta f$
$= 2 \times 1.6 \times 10^{-19} \times 10^{-6} \times 10^{6}$
$= 3.2 \times 10^{-19}$ A^2

$$i_{rms} = \sqrt{3.2 \times 10^{-19}} = 0.566 \text{ pA}$$

Rata fotonilor

Numărul mediu de fotoni pe secundă: $\frac{I}{q} = \frac{10^{-6}}{1.6 \times 10^{-19}} = 6.25 \times 10^{12}$ fotoni/s

Ring Oscillator TRNG



Oscilator în inel (instabil)

Principiul

Variațiile în întârzierea propagării prin porți logice duc la jitter temporal impredictibil.

Quantum Random Number Generators

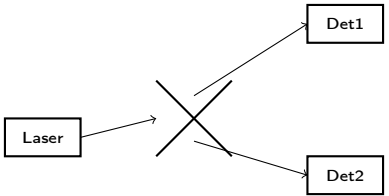
Principiul cuantic

Măsurarea stării cuantice este fundamental probabilistică conform interpretării de la Copenhaga.

Implementare cu fotoni

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Probabilitatea măsurării stării $|0\rangle$ sau $|1\rangle$ este exact $\frac{1}{2}$.



50/50 Beam Splitter

Generarea Cheilor Criptografice

Pentru o cheie RSA de 2048 biți, avem nevoie de:

$$p, q \text{ prime cu } p \cdot q = n \text{ și } |p - q| > 2^{1016}$$

One-Time Pad

Cheia trebuie să fie:

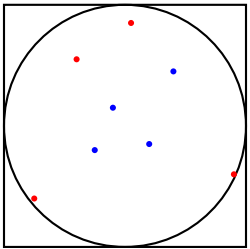
- La fel de lungă ca mesajul
- Perfect random
- Folosită o singură dată

$$C = M \oplus K$$

unde C = text cifrat, M = mesaj, K = cheie random

Calcularea lui π

$$\pi \approx 4 \cdot \frac{\text{numărul punctelor în cerc}}{\text{numărul total de puncte}}$$



Simulare Monte Carlo pentru π

Precizia

Eroarea scade cu $\frac{1}{\sqrt{N}}$ unde $N =$ numărul de puncte.

Exemplu Numeric - Monte Carlo

Simularea cu 10,000 de puncte

Iterația	Puncte în cerc	Total puncte	Estimarea
1,000	785	1,000	$4 \times \frac{785}{1000} = 3.140$
5,000	3,927	5,000	$4 \times \frac{3927}{5000} = 3.142$
10,000	7,854	10,000	$4 \times \frac{7854}{10000} = 3.142$

Eroarea teoretică

Pentru $N = 10,000$ puncte:

$$\text{Eroarea} \approx \frac{1}{\sqrt{N}} = \frac{1}{\sqrt{10000}} = \frac{1}{100} = 0.01$$

Eroarea noastră reală: $|3.14159 - 3.142| = 0.00041$

Concluzie

Cu mai multe puncte random, estimarea devine mai precisă!

Gaming și Lottery

Cerințe pentru Gaming

- Unpredictability - imposibil de prezis
- Fairness - distribuție uniformă
- Reproducibility - pentru audit
- Performance - generare rapidă

Testele NIST pentru RNG

- Frequency Test
- Runs Test
- Longest Run of Ones Test
- Spectral Test
- Universal Statistical Test

Atacuri asupra PRNG

State Recovery Attack

Dacă un atacator observă suficient de multe ieșiri, poate reconstrui starea internă.

Exemplu - LCG Attack

Pentru LCG: $X_{n+1} = (aX_n + c) \bmod m$
Dacă cunoști 3 valori consecutive X_i, X_{i+1}, X_{i+2}:

$$a = \frac{X_{i+2} - X_{i+1}}{X_{i+1} - X_i} \bmod m$$

$$c = X_{i+1} - aX_i \bmod m$$

Defensa

- State compromization resistance
- Forward secrecy
- Backward secrecy

Exemplu Numeric - Atacul asupra LCG

Observăm 3 valori consecutive

- $X_i = 1406932606$
- $X_{i+1} = 654583775$
- $X_{i+2} = 1449466924$

Reconstruirea parametrilor

$$a = \frac{X_{i+2} - X_{i+1}}{X_{i+1} - X_i} \bmod 2^{32} = \frac{794883149}{-752348831} \bmod 2^{32} = 1664525$$

$$c = X_{i+1} - aX_i \bmod 2^{32} = 654583775 - 1664525 \cdot 1406932606 \bmod 2^{32} = 1013904223$$

Rezultat

Atacatorul poate acum prezice toate valorile viitoare!

Entropia și Post-Processing

Măsurarea Entropiei

Entropia Shannon: $H(X) = -\sum_i p_i \log_2 p_i$
Pentru distribuție uniformă cu n valori: $H(X) = \log_2 n$

Von Neumann Extractor

Pentru o sursă cu bias, dar cu biți independenți:

Citește două biți: b_1, b_2
if $b_1 = 0$ și $b_2 = 1$ **then**
  Output: 0
else if $b_1 = 1$ și $b_2 = 0$ **then**
  Output: 1
else
  Repetă procesul
end if

Exemplu Practic - Von Neumann Extractor

Sursă cu bias: $P(1) = 0.7$, $P(0) = 0.3$

Secvența de intrare: 1101001110100111010

Aplicarea algoritmului

Pereche	Acțiune	Ieșire
11	Ignoră	-
01	Outputează 0	**0**
00	Ignoră	-
11	Ignoră	-
10	Outputează 1	**1**
10	Outputează 1	**1**
01	Outputează 0	**0**
11	Ignoră	-
01	Outputează 0	**0**

Rezultat

Din 18 biți de intrare cu bias $\rightarrow$ 5 biți fără bias: **01100**
Eficiența: $\frac{5}{18} = 27.8\%$ (teoretic: $2 \times 0.7 \times 0.3 = 42\%$)

Sumar

Ce am învățat

- Oamenii sunt predictabili în alegerea numerelor "random"
- PRNG sunt rapide dar deterministe
- TRNG oferă randomness real dar sunt mai lente
- Aplicațiile diferite au cerințe diferite
- Testarea și post-procesarea sunt cruciale

Viitorul

- Quantum RNG devin mai accesibile
- Hybrid approaches (PRNG + TRNG)
- AI-resistant randomness
- Post-quantum cryptography

Mulțumesc!

Întrebări?

True randomness matters!