BI-BIT-18 Generátory náhodných čísel (PRNG, TRNG). Testy prvočíselnosti (Fermatův test, Rabinův-Millerův test).

Obsah

Generátory náhodných čísel (RNG)

Náhodné číslo - čislo vygenerované procesem, který má nepředpovídatelný výsledek a jehož průběh nelze přesně reprodukovat. Tomuto procesu říkáme **generátor náhodných čísel** (RNG).

Od náhodných posloupností očekáváme dobré statistické vlastnosti:

- **rovnoměrné rozdělení** - všechny hodnoty jsou generovány stejnou pravděpodobností
- **nezávislost** - jednotlivé generované hodnoty jsou nezávislé (není mezi nimi žádná korelace)

Veličina **entropie** popisuje míru nahodilosti - jak obtížné je hodnotu (náhodné číslo, náhodnou posloupnout) uhodnout. Entropie generátoru je maximální, pokud generuje všechny možné hodnoty se stejnou pravděpodobností.

Pseudonáhodné generátory (PRNG)

Algoritmus, jehož výstupem je posloupnost, která sice ve skutečnosti není náhodná, ale která se zdá být náhodná, pokud útočníkovi nejsou známy některé parametry generátoru.

- + algoritmické ⇒ snadno realizovatelné

- + obvykle rychlé
- + zpravidla mají dobré statistické vlastnosti
- - výstup je předvídatelný

Lineární kongruenční generátor

Není kryptograficky bezpečný.

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

Kryptograficky bezpečné pseudonáhodné generátory

Jedná se o algoritmicky generovaná "náhodná" čísla (deterministickým počítačem). G

Kryptograficky bezpečný PRNG musí splňovat:

- **next-bit test**: pokud se zná prvních n bitů náhodné posloupnost, nesmí existovat algoritmus, který v polynomiálním čase dokáže předpovědět další bit, s pravděpodobnstí větší jak 1/2.
- **state compromise**: i když je znám vnitřní stav generátoru, nelze zpětně zrekonstruovat dosavadní vygenerovanou posloupnost. Navíc, pokud do generátoru za běhu vstupuje entropie, nemělo by být možné ze znalosti stavu předpovědět stav v dalších iteracích.

Tyto generátory obvykle vyžadují náhodný a tajný seed. Entropie je získána seedem, samotný generátor žádnou entropii nepřidává. Kvalita generátoru se tak odvíjí od kvality seedu.

Blum-Blum-Shub

Jedná se o PRNG, který by měl být kryptograficky bezpečný.

$x_{n+1} = x_{n-1}^2 mod m$

- x_0 je definováno seedem a musí být větší 1 (jinak by nefungovalo to umocňování)
- modul $m = q * r$, kde q i r jsou prvočísla
- pro q i r musí platit, že $q / r = 3 (mod 4)$
- při znalosti x_0 lze dopočítat pomocí rovnice jakýkoliv člen, proto musí zůstat utajen
- pokud x_{n+1} vyjde sudé, jde na výstup 0, jinak 1

Skutečně náhodné generátory (TRNG)

Využívají zdroj entropie, kterým je zpravidla nějaký fyzikální jev.

- radioaktivní rozpad
- teplotní šum (např. na analogových součástkách)
- chování uživatele (pohyb myší, prodlevy při psaní na klávesnici)

Není předvídatelný i když známe všechny parametry.

Má horší statistické vlastnosti, je nutné následné zpracování (post-processing).

Post-processing

Účelem je vylepšit statistické vlastnosti TRNG, zejména:

- Odstranění nevyváženosti jedniček a nul (bias) a zajištění rovnoměrného rozdělení.
- Extrakce entropie – zvýšení entropie výstupních bit□u za cenu snížení rychlosti jejich generování (bitrate).

Metody post-processingu

- **John von Neumannův dekorelátor** je schopen eliminovat nevyváženost a snížit korelovanost výstupu. Bity se odebírají po dvou.

Vstup	Výstup
00, 11	(vstup se zahodí)
01	0
10	1

- XOR s výstupem kryptograficky silného PRNG
- XOR výstupů různých TRNG
- Hashování výstupu kryptograficky kvalitní hashovací funkcí

Testování náhodných generátorů

K ověření vlastností náhodných generátorů se používají **statistické testy**. Testy ověřují, zda generovaná posloupnost splňuje některé vlastnosti náhodné posloupnosti.

Statistické testy mohou ukázat, že generátor NENÍ kvalitní. Nelze však prokázat, že JE kvalitní. Generátor může obsahovat slabinu, kterou testy neodhalily.

Příklady testů náhodnosti

- frekvenční test
- "runs" test
- test hodnosti matic
- spektrální test
- Maurerův univezální statistický test

Známé testovací sady

- Diehard - 12 testů
- Dieharder - reimplementace a rozšíření Diehard testů

- NIST - 16 testů

Tyto sady byly nicméně vyvinuty převážně pro testování PRNG. Při testování TRNG je třeba důkladně analyzovat zdroj entropie a navrhnout a provést cílené testy, které by odhalily případné slabiny specifické pro tento zdroj entropie.

Testy prvočíselnosti

Hledání prvočísel: náhodné vygenerování čísla, které následně testujeme, zda-li je prvočíslem.

Testy prvočíslenosti můžeme dělit na:

- testy, které nám s jistotou řeknou, zda-li je číslo prvočíslem nebo ne (jsou pomalé)
- testy, které nám s jistotou řeknou, že číslo není prvočíslem a nebo že číslo je prvočíslem s určitou pravděpodobností (jsou rychlé)

Test hrubou silou

Dělíme všemy prvočísly, která jsou menší nebo rovné $\sqrt{N}$

Fermatův test

Používá Malou Fermatovu větu.

$a^{p-1} \equiv 1 \bmod p$

Testujeme, zda-li n je prvočíslo (např. $n = 100$):

1. náhodně zvolíme celé číslo a (tzv. svědek)
   - např. $a_1 = 3$
2. dosadíme do MFV
   - $3^{100-1} \equiv ? \bmod 100$
     - pokud vyjde 1 → může být prvočíslem
     - pokud nevyjde 1 → není prvočíslem

Existují bohužel tzv. *Carmichaelova čísla*, která jsou složená, ale vždy vyjde výsledek 1.

Lehmanův test

Používá Malou Fermatovu větu.

Testujeme, zda-li n je prvočíslo (např. $n = 101$):

- pokud n je sudé → není prvočíslo (pouze 2)
- pokud n je liché → tak $(n-1)$ je sudé (lze zapsat jako $2k$)

Obecné úpravy (předpokládáme n liché):

$a^{n-1} \equiv 1 \bmod n$ (Malá Fermatova věta)

$a^{n-1} - 1 \equiv 0 \bmod n$

$a^{2k} - 1 \equiv 0 \bmod n$ (sudé číslo zapíšeme jako $2k$)

$(a^k + 1)(a^k - 1) \equiv 0 \bmod n$ (středoškolský vzorec pro $a^2 - b^2$)

1. případ: $(a^k + 1) = 0$ nebo $(a^k - 1) = 0$ → $\boldsymbol{n}$ může být prvočíslo
2. případ: obě závorky jsou nenulové → $\boldsymbol{n}$ není prvočíslo

Může se stát, že nám test s náhodně vybraným svědkem nepravdivě řekne "$\boldsymbol{n}$ může být prvočíslo". Existuje však důkaz, že tato chyba může nastat maximálně v 50% případů. Pravděpodobnost, že nám tak např. 6 testů dá pokaždé tuto špatnou odpověď je $0.5^6 \approx 0.0016$.

Příklad:

1. testujeme, zda-li např. $n = 101$ je prvočíslo
2. náhodně zvolíme *svědka* např. $a_1 = 3$
3. dosadíme do $(a^k + 1)(a^k - 1) \equiv 0 \bmod n$
   - $n - 1 = 2k = 100$ → $\mathbf{k = 50}$
   - $(3^{50} + 1)(3^{50} - 1) \equiv 0 \bmod 101$
     - počítáme první závorku:
       - $(3^{50} + 1) \bmod 101 = (100 + 1) \bmod 101 = \mathbf{0}$
       - vyšla 0 → číslo $\mathbf{n}$ může být prvočíslo, dále nepokračujeme

Rabinův-Millerův test

Používá Malou Fermatovu větu.

Testujeme, zda-li n je prvočíslo (např. $n = 101$):

- pokud n je sudé → není prvočíslo (pouze 2)
- pokud n je liché → tak $(n - 1)$ je sudé (lze zapsat jako $2k$)

Obecné úpravy (předpokládáme n liché):

$a^{n-1} \equiv 1 \bmod n$ (Malá Fermatova věta)

$a^{n-1} - 1 \equiv 0 \bmod n$

$a^{2k} - 1 \equiv 0 \bmod n$ (sudé číslo zapíšeme jako $2k$)

$(a^k + 1)(a^k - 1) \equiv 0 \bmod n$ (středoškolský vzorec pro $a^2 - b^2$)

$(a^{2l} + 1)(a^l + 1)(a^l - 1) \equiv 0 \bmod n$ (pokud k je sudé, zapíšeme jako $2l$ a opakujeme předchozí krok)

$(a^{4m}+1)(a^{2m}+1)(a^m+1)(a^m-1) \equiv 0 \bmod n$ (pokud l je sudé, zapíšeme jako $2m$ a opakujeme předchozí krok)

1. případ: pokud alespoň jedna závorka je nulová → $\boldsymbol{n}$ může být prvočíslo
2. případ: žádná závorka není nulová → $\boldsymbol{n}$ není prvočíslo

Může se stát, že nám test s náhodně vybraným svědkem nepravdivě řekne "$\boldsymbol{n}$ může být prvočíslo". Existuje však důkaz, že tato chyba může nastat maximálně v 25% případů. Pravděpodobnost, že nám tak např. 3 testy dají pokaždé tuto špatnou odpověď je $0.25^3 \approx 0.0156$.

Příklad:

1. testujeme, zda-li např. $n = 101$ je prvočíslo
2. náhodně zvolíme *svědka* např. $a_1 = 3$
3. dosadíme do $(a^k+1)(a^k-1) \equiv 0 \bmod n$
    - $n-1 = 2k = 100$ → $\mathbf{k = 50}$
    - $(3^{50}+1)(3^{50}-1) \equiv 0 \bmod 101$
        - počítáme první závorku:
            - $(3^{50}+1) \bmod 101 = (100+1) \bmod 101 = \mathbf{0}$
            - vyšla 0 → číslo $\mathbf{n}$ může být prvočíslo, dále nepokračujeme