Načrt (1. apr. 99): 12. poglavje

Generatorji psevdo naključnih števil

- uporaba in primeri
- kaj je naključno število
- definicije
- primeri

Uporaba in primeri

En izmed konceptov popolne varnosti, ki smo ga študirali v poglavju o entropiji, je bil **enkratni ščit**, kjer sta čistopis in ključ dve zaporedji bitov, ki ju zašifriramo tako da ju seštejemo z XORjem.

Praktičen problem pa je generiranje, izmenjava in dolžina ključa, tj. naključnega zaporedja bitov.

- LFSR stopnje k za dano k-bitno seme generira dodatnih $2^k - k - 1$ bitov predno se začne ponavljati.

- Zaporedno potenciranje (DLP), zaporedna uporaba simetričnega sistema, zaporedna uporaba zgoščevalne funkcije itd.

-majhne, enostavne in poceni elektronske naprave (Intel načrtuje vgraditi v Pentium III microprocessor hardwarski RNG, glej našo domačo stran)

Intel starts preaching about security
EE Times Print, By Craig Matsumoto
(01/21/99, 3:27 p.m. EDT)

Intel: We won't track ID chips PC chip giant says it's walking on glass over the privacy considerations of new processor ID scheme
ZDNN, By Robert Lemos
(January 21, 1999 5:49 PM PT)

... For starters, Intel will burn a unique, secret identification number into every Pentium III that will ship.

...

Because the ID number also could be a privacy threat, Intel plans to allow end users to block transmission of the number, reportedly through a software patch. ...

For companies that sell into corporate networking environments, the ID number is a long-awaited relief. "We had dreamed of having a 'serial number' on the motherboard," ...

... Intel plans to provide a hardware-based random-number generator in every PC. The flaw in computer-generated pseudorandom numbers is that they fall in deterministic sequence; each "random" number is calculated based on its predecessor, making cycles and subtle patterns inevitable. Truly random numbers can only be gathered through physical phenomena, such as radioactive decay or, in Intel's case, thermal noise.

... Chances are, the hardware random-number generator will be used to select a "seed", or starting point, for an application's pseudorandom generator. ...

M. Kac (*Amer. Scientist* **71** (1981), 405-406)

Kaj je naključno število?

Na to vprašanje ni absolutnega odgovora (teorija informacij, teorija števil, teorija kompleksnosti, fizika).

Za začetek moramo ločiti med naključnim zaporedjem števil in generiranjem naključnega zaporedja.

Najbolj pogost primer naključnega procesa je **metanje kovanca** (idealno). Če ga vržemo n-krat zaporedoma, potem je očitno, da lahko dobimo vsakega izmed 2^n zaporedij grbov ali cifr, tj. da ima vsako od 2^n zaporedij grbov ali cifr enako verjetnost.

Mi se bomo ukvarjali s psevdo-naključnim zaporedjem števil, tj. zaporedjem, ki "izgleda naključno" oziroma demonstrira *neurejenost/kaos*.

Knuth (*The art of Computer Programming*, 2nd ed., Addison-Wesley, Reading (1981), 689 pp.) je predstavil številne **statistične teste**, ki merijo neurejenost.

-v zaporedju se pojavi z enako frekvenco vsako podzaporedje dolžine $1, 2, \ldots$

Chaitin in Kolmogorov pa pravita, da (dolgo) končno zaporedje bitov, ki se ga dobiti iz programa, ki je precej krajši kot dano zaporedje, ni naključno.

1. Set i equal to 1.
2. Print "1".
3. If $i = n$, then stop.
4. Add 1 to i.
5. Go back to Step 2.

Ne glede na to ali je število n veliko, ima ta program le fiksno število več bitov kot jih je v binarni reprezentaciji števil i in n, ki ne presega $2\log_2 n$ (na binarnem računalniku).

Če pa je zaporedje dovolj neurejeno, potem tudi program, ki ga izpiše ne bo dosti krajši.

Običajno preštevanje pa nam zagotavlja, da bo veliko manj programov, katerih dolžina bo občutno manjša od števila n.

Chaitin pa uporabi **problem zaustavljanja**, da pokaže, da če je dano zaporedje tako dolgo, da je njegova kompleksnost večja kot kompleksnost sistema aritmetike, potem je v splošnem nemogoče dokazati, da je naključno zaporedje.

Definicija zaporedja naključnih števil

-algoritmično naključno število ... ???????

Namesto nereda, statistike, je za kriptografijo bolje študirati (ne)napovedovanje naslednjega člena.

Kdo skuša napovedovati?

Ali lahko kdorkoli (z izjemo generatorja) v
polinomskem času poišče polinomski algoritem
za napoved naslednjega bita?
(dodatno dovolimo še poznavanje podzaporedja bitov)

Kriptografsko krepak generator psevdo-naključnih bitov

Splošen način so predstavili Blum in Micali ter Blum, Blum in Shub.

Def: To je generator bitov, ki razširi naključno izbrano zaporedje dolžine n do polinomsko dolgega psevdo-naključnega zaporedja v polinomskem času.

Krepko psevdo-naključno zaporedje bitov $\{b_i\}$ ima lastnost, da ne obstaja polinomski algoritem, ki bi iz zaporedja $b_j b_{j+1} \ldots b_{j+m-1}$ napovedal bit b_{j-1}.

Od tod pa sledi, da noben polinomski algoritem ne loči krepko psevdo-naključnega zaporedja bitov od resnično naključnih bitov.

Generator $\frac{1}{P}$

kjer je P dano praštevilo, b pa baza številskega sistema, $1 < b < P$, ki je primitiven koren po modulu P.

To zaporedje je generirano s števkami števila $1/P$ v številskem sistemu z osnovo b. Če preskočimo prvo števko, potem je vsaka naslednja števka r "izbrana" na intervalu $(0, P)$, nato pa nadaljujemo z r/P.

Čeprav zaporedje "izgleda naključno" zaradi periode $P-1$ (če je P recimo 50 mestno) in ima dobre statisti'cne lastnosti, se izkaže, da se ta generator da napovedati.

Izrek. *Iz* $\lceil \log_b 2P^2 \rceil$*-bitnega podzaporedja lahko opazovalec v polinomskem času od* $\log_b P$ *določi naslednji člen zaporedja.*

Primer: naj bo $b = 10$, dodatno pa predpostavimo, da ne poznamo P.

Generator vprašamo za 3 števke in dobimo: 407.

Število $0.407 = 407/1000$ zapišemo v obliki verižnega ulomka:

$$\frac{407}{1000} = 0 + \cfrac{1}{2 + \frac{1}{2+\frac{1}{5+\frac{1}{3+\frac{1}{5+\frac{1}{2}}}}}}$$

$$= [0, 2, 2, 5, 3, 5, 2],$$

Zaporedje: 0, .5, .4, .4074, .406, .4070, .407=407/1000.

Prvi člen, ki ujame .407 je 11/27 napove, da bo naslednja števka 4.

Generator pa nam da 3.

Ponovimo proces za 4073/10000 in dobimo 145/356 = .40730..., ki ujame .4073 in napove 0.

Generator pa nam da 3.

Ponovimo proces za 40733/100000 in dobimo

$300/491 = .407331975560081466395112016293279 02...$

Ker se generatorjevih naslednjih 30 števk ujema z našimi sprejmemo za $P = 491$.

Algoritem za psevdo-naključne bite

Naj bo $N = pq$, kjer sta p in q praštevili, in naj bo C_0 tako naravno število, da je $(C_0/p) = (C_0/q) = 1$, kjer je $(-/-)$ Legendrov simbol.

1. izberi $X_0 \in \mathbb{Z}_n^*$ tako, da je $D(X_0 + C_0, N) = 1$
2. za začetni vrednosti X_0 in C_0 obnovi X_i in C_i $(i \geq 0)$ z $X_{i+1} \equiv X_i - C_i x_i^{-1} \pmod{N}$ in $C_{i+1} \equiv 4C_i \pmod{N}$
3. za $i \geq 0$ izračunaj in izpiši zaporedje $\{b_i\}$:

$$b_i = \begin{cases} 1, & \text{če je } X_i > -C_i X_i^{-1}; \\ 0, & \text{sicer,} \end{cases}$$

Problem C-kvadratnih ostankov

Naj bo $N = pq$, kjer sta p in q praštevili, in naj bo $C \in \mathbb{Z}_N^*$. Potem določi ali je za $X \in \mathbb{Z}_N^*$ število $X^2 + C$ kvadratni ostanek po modulu N.

(Kurosawa and Matsu) ... SASA

Reference

[GW] I. Goldberg and D. Wagner, Randomness and the Netscape Browser, *Dr. Dobb's Journal*, January 1996, 66-70.

[Ba] S. Bassein, A Sampler of Randomness, *American Mathematical Monthly* **103** (1996), 483-490.

[KSWH] J. Kelsey, B. Schneier, D. Wagner and C. Hall, Cryptoanalytic Attacks on Pseudorandom Number Generators, probably from the Internet, 21 pages.

[P] C. Plumb, Truly Random Numbers, *Dr. Dobb's Journal*, November 1994, 113-114, 137-139.

[NGGLG] J. No, S. W. Solomon, G. Gong, H. Lee and P. Gaal, Binary Pseudorandom Sequences of Period $2^n - 1$ with Ideal Autocorrelation, *IEEE Transactions of Information Th.* **44**, March 1998, 814-817

[NGGLG] A. Chang, P. Gaal, S. W. Golomb, G. Gong and P. V. Kumar, Some Results Relating to a Sequence Conjectured to have Ideal Autocorrelation, submitted to *IEEE Transactions of Information Th.* in July 1998.

[Ch] G. J. Chaitin, Randomness in Arithmetic, *Scientific American*, July 1988, 52-57.