

SORGENTE CASUALE BINARIA

genera una sequenza di bit tale che

1) $p(0) = p(1) = 1/2$

2) la generazione di un bit e' indipendente dai bit precedenti

la condizione 1) puo' essere sostituita da

$$p(0) > 0, \quad p(1) > 0$$

immutabili durante il processo

Questo e' possibile perché si puo' trasformare una sorgente binaria

stabilmente su uno dei due bit e renderla bilanciata raggruppando i bit in coppie

e.g. 01|10|01|10|01|01|...
~~01|10|01|10|01|01|...~~

$$\begin{array}{l} 01 \rightarrow 0, \quad 10 \rightarrow 1 \\ \curvearrowright 0110010 \end{array}$$

GENERATORI di NUMERI PSEUDOCASUALI

L'idea e' quella di usare un algoritmo per generare casualita', la si ricerca all'interno di processi matematici:

Input: sequenza breve ("seme")

Output: flusso di bit arbitrariamente lungo

(ha un "periodo", il periodo e' una sequenza casuale. Il flusso si ripete quando viene rigenerato il seme)

In concreto, c'è un amplificatore di causalità: espande la causalità
del segnale

E' tanto migliore quant. più lungo è il periodo rispetto al segnale.

S. segnale $1.5^t - s$ bit

Il generatore genera il più 2^s sequenze diverse

lunghe quanto il periodo

Risulta $2^s \ll 2^n$

GENERATORE LINEARE

$$x_i = (ax_{i-1} + b) \bmod m$$

con $a, b, m \in \mathbb{N}$ parametri del generatore

Risulta periodo $\leq m$

Se i parametri sono scelti bene allora il generatore

produce una permutazione degli interi $[0; m-1]$

CRITERI $\text{MCD}(b, m) = 1$

$a-1$ divisibile per ogni fattore primo di m

$$4|m \Rightarrow 4|a-1 \quad (\text{leg. "4 divide } m\text{"})$$

Per generare sequenze binarie si prende la parità della
prima cifra di $r = x_i/m$.

Si generalizza nel generatore polinomiale

$$x_i = (a_1 x_{i-1}^t + a_2 x_{i-1}^{t-1} + \dots + a_t x_{i-1} + a_{t+1}) \bmod m$$

Una buona scelta e'

$$a = 3141592653, b = 2718281829, m = 2^{32}$$

Per giudicare una sequenza casuale si ricorre a dei test statistici:

- 1) test di frequenze: verifica se i diversi elementi della sequenza appaiono con la stessa probabilità
- 2) poker test: verifica la equidistribuzione di sottosequenze di lunghezza arbitraria prefissata
- 3) test di autocorrelazione: verifica che non accada che a distanza prefissata ricorra sempre lo stesso elemento
- 4) run test: verifica che le sottosequenze minimamente contenenti elementi tutti uguali abbiano distribuzione esponenziale negativa

Si richiede in crittografia che si superi un test detto "TEST AL PROSSIMO BIT" (superare questo implica superare i test statistici elencati sopra)

TEST AL PROSSIMO BIT: un generatore binario supera il test se non esiste un algoritmo polinomiale in grado di prevedere l' $i+1$ -esimo bit generato a partire dalla conoscenza degli i bit precedentemente generati con probabilità maggior di $1/2$.

Se un generatore supera il test di prossimo bit allow si definisce crittograficamente sicuro.

Si ricorre a funzioni one-way: calcolare $y = f(x)$ e' facile
calcolare $x = f^{-1}(y)$ e' difficile

$$x \quad f(x) \quad f(f(x)) \quad f(f(f(x))) \quad \dots$$

Viene consumata nell'ordine inverso

GENERATORE BINARIO CRIPTOGRAFICAMENTE SICURO

Definisco $b(x)$ predicato hard-core di una funzione one-way f se

$b(x)$ e' facile da calcolare conoscendo x

$b(x)$ e' difficile da prevedere se si conosce $f(x)$

e.g. $f(x) = x^2 \bmod 77$

$$b(x) = \text{parity}(x)$$

$$x = 10, \quad f(x) = 100 \bmod 77 = 23$$

$$\text{Data } x, b(x) = 1$$

Non conoscendo x , $b(x)$ sapendo che $f(x) = 23$ e' difficile

Il generatore BBS funziona in questo modo.

GENERATORE BBS

E' crittograficamente sicuro

$$n = p \cdot q, \quad p, q \text{ numeri primi (grandi)}$$

$$p \bmod 4 = 3, \quad q \bmod 4 = 3, \quad \text{MCD}(2_{\lfloor p/4 \rfloor + 1}, 2_{\lfloor q/4 \rfloor + 1}) = 1$$

$$\text{MCD}(g, n) = 1$$

Si calcola $x_0 = y^2 \bmod n$

Si genera una successione di numeri interi

$$x_i = (x_{i-1})^2 \bmod n \quad i > 1$$

In corrispondenza viene generata la successione binaria

$$b_i = 1 \text{ se } x_{m-i} \text{ e' dispari}$$

E.g. $p=11, q=19$

$$11 \bmod 4 = 3 \quad 19 \bmod 4 = 3$$

$$2 \lfloor \frac{11}{4} \rfloor + 1 = 7$$

$$2 \lfloor \frac{19}{4} \rfloor + 1 = 9 \quad n = 209$$

$$y = 30 \quad \text{MCD}(30, 209) = 1$$

$$x_0 = 30^2 \bmod 209 = 64 \Rightarrow 0$$

$$x_1 = 64^2 \bmod 209 = 125 \Rightarrow 1$$

$$x_2 = 125 \bmod 209 = 159 \Rightarrow 1$$

$$x_3 = 159 \bmod 209 = 201 \Rightarrow 1$$

Questo generatore, seppure utile algoritmi polinomiali, e' molto lento
(elevamento al quadrato, divisione intera etc.)

Generazione di numeri pseudocasuali usando cifrari simmetrici

I cifrari simmetrici lavorano cifrando un blocco alla volta

Definisce r : numero di bit delle parole probabili

(ie lunghezza di un blocco)

$s :=$ sequenza casuale lungo r bit

$m :=$ numero delle parole da r bit probabili

$k :=$ chiave segreta

Per fatto che sia critograficamente sicura e' garantito dalle proprietà del cifrario

GENERATORE (s, m):

$d =$ rappresentazione su r bit di data e ora

$y = C(d, k)$

$z = s$

for $i = 1$ to m do:

$x_i = C(y \text{ xor } z, k)$

$z = C(y \text{ xor } x_i, k)$

comunica x_i all'esterno

$C :=$ funzione di cifratura simmetrica

$\text{xor} :=$ funzione che fa xor bit per bit