COMP 3704 Computer Security

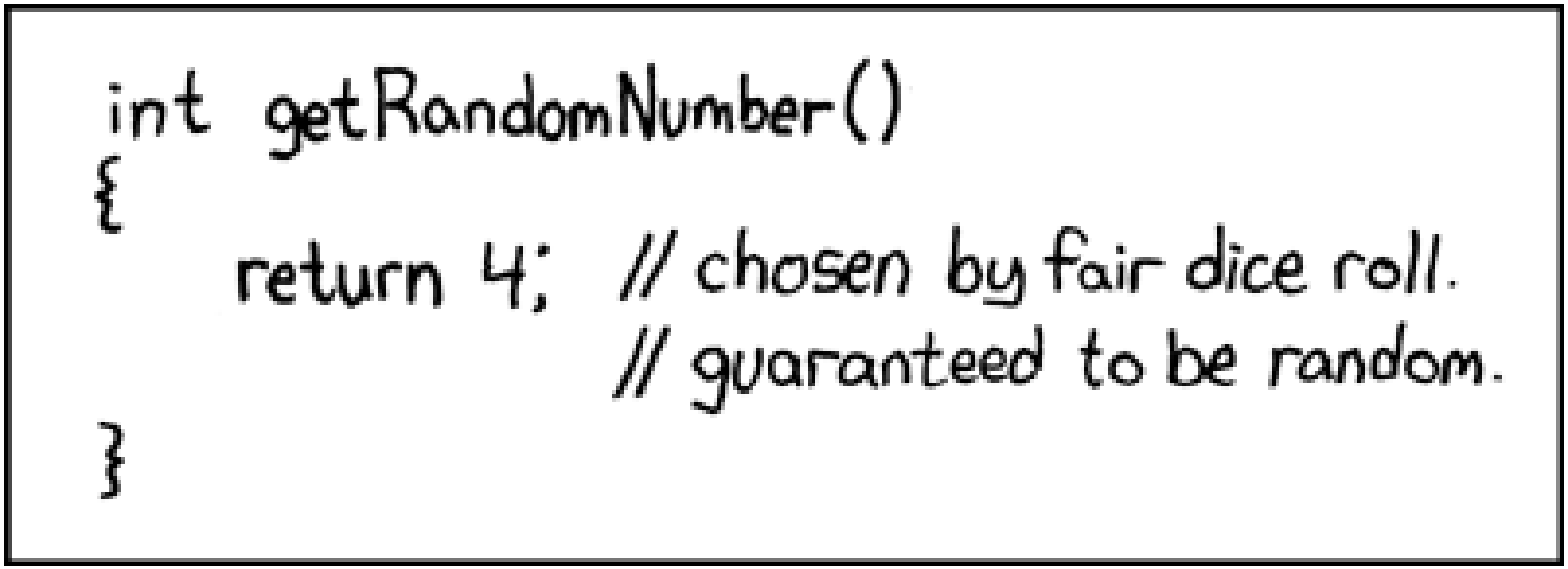
Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

Motivation

- Almost all cryptographic protocols require random numbers
- Random number generation crucial for security
- Example: David Wagner & Ian Goldbergs' break of Netscape SSL in 1995!
- Today: Statistics, Group Theory & PRNG algorithms

Random Numbers



UNIVERSITY OF DENVER

Define Random!

- 10001101001101001111

Define Random!

- 0101010101010101010

Define Random!

- 13

Define Random!

- 1395

Define Random!

- 139541

Define Random!

- 13954139

Define Random!

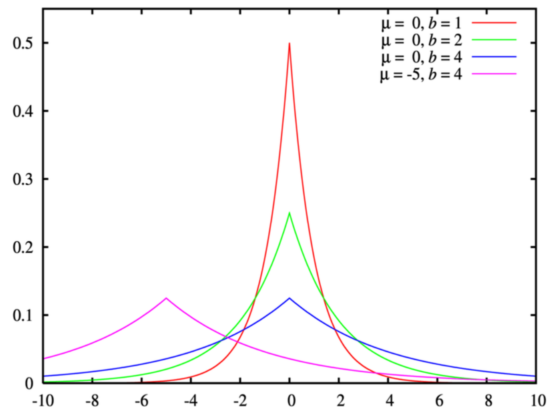
- 1395413954

Define Random!

- 13954139541395413 91

Statistical Expectations

- A probability distribution describes the statistical expectations for a particular probabilistic experiment.
- A $c\%$ confidence interval is an interval in which $c\%$ of all sample runs are expected to fall



Null-hypothesis Testing

- A null hypothesis is a hypothesis set up to be refuted in order to **support** an alternative hypothesis
- Rejecting the null hypothesis may say little about the likelihood that the alternative hypothesis is true
- Null hypotheses are often rejected probabilistically: "it is **unlikely** for the result to occur by chance"

PRNG null hypothesis: "This sequence is not random."

PRNG tests

- Frequency Tests (overall, in M-bit blocks)
- Runs Tests
- Rank of binary matrices
- Compression tests
- N-tuple distribution tests

Compression

- If a bit-sequence can be "significantly" compressed, it is not random.
- Hard to determine in general if a sequence can be compressed: $139541395413954 1391 = n * 3 \mod 11$
- Assuming elements in sequence are independent, coding theory can help!

Entropy

Entropy is a measure of the uncertainty associated with a random variable.

- Average shortest message length, in bits, that can be sent to communicate the true value of the random variable
- Mathematical limit on the best possible lossless data compression

Entropy: Definition

Let the set Φ be the range of the random variable and p_u the probability for choosing $u \in \Phi$. Then

$$S = -\sum_{u \in \Phi} p_u \cdot \log_2(p_u) \tag{1}$$

is the information in each independent choice.

UNIVERSITY OF DENVER

The χ^2 method

Given n independent observations falling into k categories and p_s being the probability that each observation falls into category s and with Y_s being the number of observations that actually do fall into category s, define:

$$\chi^2 = \sum_{s=1}^{k} \frac{(Y_s - np_s)^2}{np_s} \tag{2}$$

UNIVERSITY OF DENVER

Expected Distributions

The probability density function for χ^2 is

$$f(x) = \frac{x^{k/2-1}e^{-x/2}}{2^{k/2} \cdot \Gamma(k/2)} \tag{3}$$

for k independent, normally distributed random variables with mean 0 and variance 1.

Creating Tuples!

- 010101010 is perfectly random if χ^2 is applied to individual bits
- Idea: build t-tuples and see if their frequencies are (still) as expected!
- $\Rightarrow$ Project 1, Part 1

Critical Values

Probability is probability of exceeding the given critical value for k degrees of freedom.

k	0.10	0.05	0.01
1	2.706	3.841	6.635
3	6.251	7.815	11.345
7	12.017	14.067	18.475

Questions

?

One Approach...

1. Given 10-digit number X, set $Y = \lfloor X/10^9 \rfloor$.
2. Set $Z = \lfloor X/10^8 \rfloor$. Goto step $3 + Z$.
3. If $X < 5000000000$, set $X = X + 5000000000$.
4. Set $X = \lfloor X^2/10^5 \rfloor$.
5. Set $X = (X \cdot 1001001001) mod 10^{10}$.
6. If $X < 100000000$, set $X = X + 9814055677$, otherwise set $X = 10^1 0 - X$.
7. ...
8. ...
9. ...
10. ...
11. ...
12. ...
13. If $Y > 0$, decrease Y by 1 and return to step 2, otherwise terminate with random value X.

UNIVERSITY OF DENVER

...that does not work!

- Algorithm can quickly converge to 6065038420.
- For other inputs, period maybe 3178.

⇒ "Random numbers should not be generated with a method chosen at random." – Knuth.

Common PRNG construction

- Maximize period – using discrete mathematics!
- Pass statistical tests
- Pass more statistical tests
- Consider computational efficiency

A bit of Group Theory

A monoid $(G, \oplus, n)$ is a set of elements G with a binary associative operation $\oplus : G \times G \rightarrow G$ and a neutral element $n \in G$ such that $g \oplus n = n \oplus g = g$ for all $g \in G$.

A group is a monoid where for each element $a \in G$ the set contains an element $a^{-1} \in G$ such that $a \oplus a^{-1} = a^{-1} \oplus a = n$.

Generators

- A set of generators is a set of group elements such that possibly repeated application of the generators on themselves and each other is capable of producing all the elements in the group.

- Cyclic groups $\mathcal{C}_n$ can be generated as powers of a single generator.

- That generator X satisfies $X^n = 1$ where 1 is the neutral element.

- $(\mathbb{Z}_n, 0, +)$ is a cyclic group.

Euler's Totient Function

$\phi(n)$ is the number of positive integers $\leq n$ that are relatively prime to (i.e., do not contain any factor in common with) n.

Example: $\phi(24) = 8$ because totatives of 24 are 1, 5, 7, 11, 13, 17, 19 and 23.

Pure Multiplicative Generators

$$X_{n+1} = aX_n \mod m \tag{4}$$

- If $X_n = 0$, sequence degenerates to zero.
- If d is a divisor of m and X_n, all succeeding elements X_{n+i} will be multiples of d.
- The maximum period of a pure multiplicative generator is $\phi(m)$.

The Linear Congruent Method

$$X_{n+1} = aX_n + c \mod m \tag{5}$$

Choice of modulus m

- Pick a large value since period cannot be bigger than m
- Orient at machine word-size $w = 2^e$ for efficiency
- Good choices are w, $w \pm 1$ and p where p is the largest prime with $p < w$.
- For $m = w$, the lowest bits in X_n are less random (for any divisor d of m and $Y_n := X_n \mod d$ the equation $Y_{n+1} = (aY_n + c) \mod d$ will hold).

Choice of multiplier

- Choose multiplier to maximize period length
- However, $a = c = 1$ is obviously not a good choice
- $\Rightarrow$ pick "large" multiplier to make modulo operation almost always meaningful

Theorem A

The linear congruential sequence defined by m, a, c and X_0 has period length m if and only if

1. c is relatively prime to m;
2. $b = a - 1$ is a multiple of p for every prime p dividing m;
3. b is a multiple of 4, if m is a multiple of 4.

Proof: Knuth, Volume II, pages 17-19.

Other Good Methods

- $X_{n+1} = (dX_n^2 + aX_n + c) \mod m$
- $X_n = (X_{n-24} + X_{n-55}) \mod m$, m even, $X_0, \ldots, X_{54}$ not even – period $2^{e-1} \cdot (2^{55} - 1)$ for $m = 2^e$
- $X_n = (a_1 X_{n-1} + \ldots + a_k X_{n-k}) \mod p$
- $\Rightarrow$ Project 2, Part 2

Mapping to Desired Domain

In order to get a random integer r in $[0 : n]$ use

$$r = \left\lfloor \frac{X_n}{m} \cdot n \right\rfloor \tag{6}$$

to avoid using low-order bits. Note that this only works if $n << m$.

Questions

?

Problem

Alice and Bob want to generate a random number in the interval of $[0 : 2^{32} - 1]$. Both have a good random number generator, however neither trusts the other to use it correctly. Design a protocol that allows them to generate a random number jointly where both are certain that the resulting number is completely random.