The art of cryptography: Heads and tails – Cryptographic random generation summer 2015

Pseudorandom generators

Prof. Dr. Joachim von zur Gathen
Dr. Daniel Loebenberger

A pseudorandom generator is a deterministic algorithm $\mathcal{A}$ with (random) inputs from a small set X and outputs in a large set Y which are "indistinguishable" from random elements of Y.

The most popular pseudorandom generators are the *linear congruential pseudorandom generators*. We have a modulus $m \in \mathbb{N}$, two integers s, t, a random *seed* $x_0 \in \mathbb{N}$, and define

$$x_i = sx_{i-1} + t \text{ in } \mathbb{Z}_m$$

for $i \geq 1$. These are good enough for many purposes, for example in primality testing, computer algebra, and numerical integration, but not for cryptography.

The following *RSA generator* is supposed to be secure. We have $N = pq$ and e, and a random seed $x_0 \in \mathbb{Z}_N^\times$. We define $x_1, x_2, \ldots \in \mathbb{Z}_N^\times$ by $x_{i+1} = x_i^e$ and output the r least significant bits.

For the *Littlewood pseudorandom number generator* , we pick (small) integers $n < d$, which are publicly known, and an n-bit string x as (truly random) seed. We can also consider x as an integer in binary, and $2^{-n}x$ is the rational number with binary representation $0.x$.
Output is the sequence of the dth bits of the binary representation of $\log_2((x+i)2^{-n}) = \log(x+i) - n$ for $i = 0, 1, \ldots$. Thus with $n = 10$, $d = 14$ and seed $x = 0110100111$, the first five pseudorandom bits are 11001, produced according to the entries (all in binary) of the following table.

i	$(x+i)2^{-n}$	$\log_2(x+i) - n$
0	0.0110100111	$-1.01000110100001\mathbf{1}1$
1	0.0110101000	$-1.01000101101001\mathbf{1}1$
2	0.0110101001	$-1.01000100110010\mathbf{0}0$
3	0.0110101010	$-1.01000011111010\mathbf{1}1$
4	0.0110101011	$-1.01000011000011\mathbf{1}0$