Random Number Generation

Utku Bozdoğan

November 8th, 2016

Randomness

- Unpredictable, has no order, no pattern
- Uncertain
- There are three accepted random behavior in systems:

  1) Randomness from the environment (entropy)

  2)Randomness from the initial conditions (dice)

  3)Randomness generated by the system (pseudorandom)

True Random Number Generators

Measures a chosen, randomly occurring, unpredictable phenomenon

Then removes biases in measurement

Examples include random.org using atmospheric noise, any atomic or subatomic phenomena that is random by nature, cosmic background radiation, radioactive decay

Pseudo Random Number Generators

A chosen fixed number called key or seed is used to generate sequences with the help of computational algorithms.

Appears random to the user and also under analysis, but the entire sequence can be produced again if the key is known.

Blocking problem

It takes a while to measure and process the data for a true random number, so when random numbers are demanded faster than they are produced, the program is blocked and does not give an output until the data is processed.

This is not the case for pseudo random numbers since they can be produced almost immediately compared to the true random numbers. And usually, they are sufficiently random for the job.

There is a hybrid method, where the data is processed to create true random numbers but there is also an algorithm based pseudo number generator if the demand is more frequent than the true number generator can supply.

Reseeding

If we use only one key to generate pseudo random numbers, the key will be compromised after a while since the outputs will follow a distinguishable pattern.

So a good pseudo random number generator needs to change keys frequently in order to be secure. Changing the key sent to the generator is called reseeding. It can be done periodically or have rules to send a different key.

I will briefly talk about two PRNGs that use a hybrid method now.

1) YARROW Algorithm
2) FORTUNA

YARROW ALGORITHM

Unpatented, open source algorithm devised by John Kelsey, Bruce Schneier and Niels Ferguson.

Takes its name from Chinese sign reading-fortune telling technique which depends on randomness.

Simple, effective, has 2 pools.

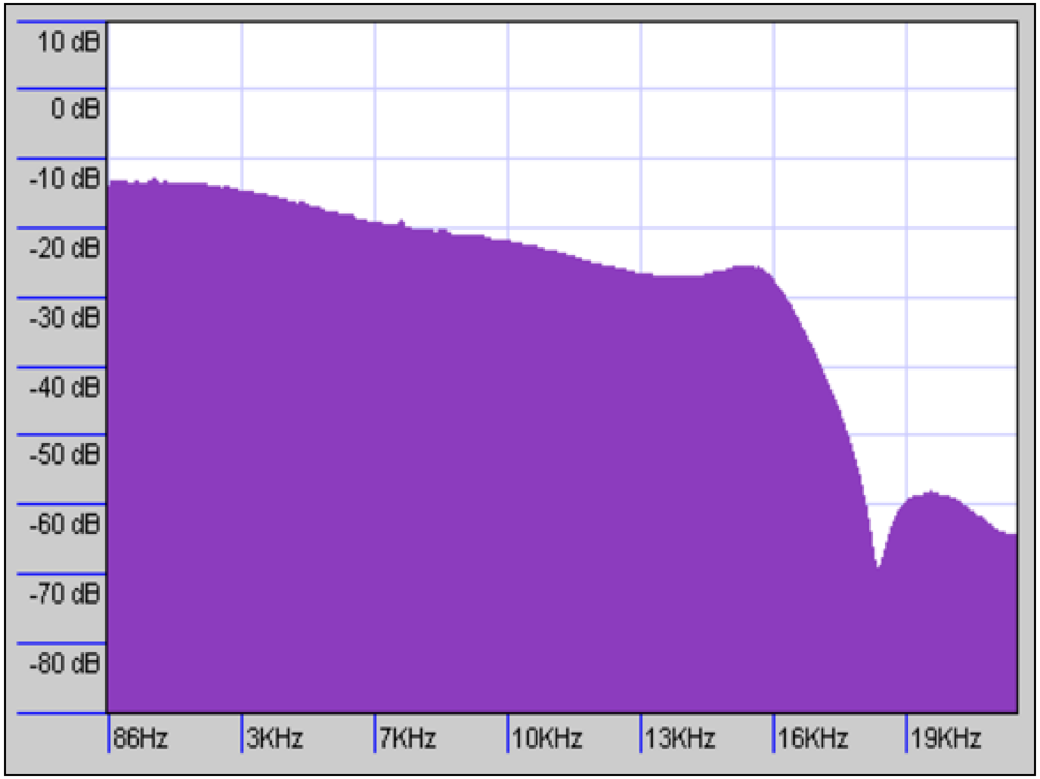
FORTUNA

Devised by Bruce Schneier and Niels Ferguson

Takes its name from Fortuna, the Roman goddess of chance.

More detailed, has 32 pools.

The figure below shows the frequency distribution for a sound file containing atmospheric noise. In fact, this file was recorded directly from one of the RANDOM.ORG radios using a sampling rate of 44.1 kHz, a sample size of 16 bits and using a single channel, i.e., as a mono signal. The graph was made using the free audio tool Audacity.



Frequency analysis of raw atmospheric noise

[play 1MB wav file]

As you can see, the spectral density is not particularly regular. Atmospheric noise can vary a good deal in its characteristics, and there is probably no accurate colour (at least not in general use) to describe this particular shape. We can observe a general decrease in power for higher frequencies, which is one of the characteristics of pink noise, but the sharp drop at 16 kHz and the small rise around 19 kHz are not typical for pink noise.

Because raw atmospheric noise cannot be expected to have a uniform spectral density, RANDOM.ORG performs processing on it. This is

Sources:

https://en.wikipedia.org/wiki/Random_number_generation

https://en.wikipedia.org/wiki/Randomness#Generation

https://en.wikipedia.org/wiki/Yarrow_algorithm

https://en.wikipedia.org/wiki/I_Ching_divination#Yarrow_stalks

https://en.wikipedia.org/wiki/Fortuna_(PRNG)

https://en.wikipedia.org/wiki/Key_generation

https://www.random.org/audio-noise/description/