Proposal: Hardware Random Number Generator

(HWRNG)

Last Revised: 2023-06-20

A hardware random number generator is an electronic device that produces random numbers based on some physical process. Such a generator has the potential to produce much higher quality random numbers than those produced by the deterministic algorithms typically used in computer programs[1]. A hardware random number generator is thus a “true” random number generator as opposed to a pseudo-random number generator.

However, to reach its potential, a hardware random number generator must be designed and implemented carefully or else various sources of bias will corrupt the randomness of its output. There is extensive literature on the subject of hardware random number generator design [1], as well as several commercial products available. In addition, much has been written on the subject of testing random number generators in general and hardware random number generators in particular [2].

Project Description

In this project, we will construct a simple hardware random number generator and develop a library that will make using the generator from applications easier. The expected application of our generator is to provide seed material for cryptographically strong pseudo-random number generators that are then used to produce keys for encryption algorithms in security sensitive applications [3]. A consequence of targeting this particular application domain is that our generator does not need to be fast since the pseudo-random number generator that it seeds can be used to produce most of the output. Instead our generator can focus on producing very high quality random numbers.

To make the generator as useful as possible it will be implemented as a separate unit attached to the computer by way of a standard USB connection. This will allow the hardware to be

[1] A deterministic algorithm can't output truly random numbers at all. If its internal state is known, the output is completely predictable.

connected to many different computers. An application-level library will provide a uniform interface to the generator. Applications must be written to use the library but, once written, they will be source code portable to all supported platforms. Because the command language used by the generator will also be documented, third parties will be able to write their own interface libraries or drivers for platforms that we do not directly support.

Hardware random number generators are subject to degradation over time, or even outright failure. To ensure the quality of the random numbers produced, my generator will contain a statistical monitoring function. An error will be indicated if the monitor detects insufficient entropy in the generator output.

Because the generator will be used in security sensitive applications, attention will be paid to security issues in the design and implementation of the generator. The application-level library will be carefully reviewed for security problems so that programs that use it are not vulnerable because of flaws in the library. In addition, the generator will be fully documented, following a policy of full disclosure, so that concerned third parties can be confident that it contains no loopholes or backdoors.

Similar Products

A device very similar to our proposed project is TrueRNG (version 3) by Ubld.it [4]. Like our proposed project, it uses a physical process to produce random numbers in a USB connected hardware device. Also TrueRNG supports multiple operating systems (Linux, Windows, macOS) although driver and application support appears a spotty at this time. The TrueRNG web site does provide some data on the randomness of the device's output similar in concept to what we propose doing as well. TrueRNG is available for $79.95 from Amazon and thus represents a price point for competitive analysis.

Deliverables

The following are the deliverables of this project.

- USB connected hardware device to generate random numbers.
- Driver software for supported operating system(s), if necessary.
- Application-level library providing a suitable API for supported language(s) that is

uniform across all supported platforms.

- Complete documentation including source code and schematics.
- Results of randomness testing and security review.

References

1: Robert Davies, Hardware Random Number Generators, 2000, http://www.robertnz.net/hwrng.htm

2: National Institute of Standards and Technology, NIST Random Number Generator Project, , https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program/Random-Number-Generators

3: D. Eastlake, J. Schiller, S. Crocker, RFC-4086; Randomness Requirements for Security, 2005

4: Ubld.it, TrueRNG, 2014, https://ubld.it/truerng_v3