Problem 15

Ryan Burmeister

January 26, 2016

1 What is Bull Mountain?

From a security perspective, random numbers are vitale to ensure the generation of secure keys. The more random the production of keys is the harder it is for an advesary to beat the encryption scheme. There have been methods implemented in software and hardware, but their are limitations with both. Software implementations are inherenetly flawed as a computer is always in a well-defined state which only changes when processes tell it to. Hardware implementations are better as they can rely on the chaotic world around us. However, they often cannot produce random numbers a rate fast enough to keep up with the demand of several processes. Intel set out on a project "Bull Mountain" to make a random number generator implemented in hardware which could overcome the typical shortcomins of previous hardware implementations.

2 How does it work?

Like previously mentioned, "Bull Mountain" is a hardware implementation created to ease the generation of random numbers while still ensuring the quality of numbers produced (very low bias and correlation). The design is dependent on the use of an entropy source, just like other hardware implementations. The entropy souce used in Bull Mountain is an all digital implementation which relies on inverters. In order to capture the random component, two transistors are added to the circuit between the inverters which forces the state to be the same for the inverters (without the transistors, if the input to one inverter was 1 then the output of that inverter was 0 which creates a cycle between the two inverters). Temporarily, the inverters are at the same state until random thermal noise sends the circuit back into one of its two stable states which produces a single bit which can be used. To repeat this process, the transistors are hooked up to a clock which forces the cycle to repeat based off the clock's period.

3 What components in a processor are necessary?

To make all of this possible, additional hardware components are necessary. The three components are a hardware entropy source (the circuit I described above), a conditioner, a deterministic random bit generator, and an enhanced non-determnistic random number generator. The conditioner groups 256 bits from the entropy source and combines them using an encryption scheme known as AES-CBC-MAC. The output is also 256 bits and is passed to the determnistic random bit generator (DRBG) to be used as a seed value. This allows for the random number generated by the conditioner to be used to generate addiitonal random

numbers faster than can be produced in the hardware. The DRBG decides when it needs to be reseeded with an upper bound of 1022 sequential random values used with any one seed. The enhanced non-deterministic random number generator offers a way for software based DRBGs to obtain seeds. As opposed to producing random numbers directly in hardware, this method allows for the generation of random numbers in software with a hardware based seed.

4 How does it compare?

The first requirement that Intel engineers had was to produce a random number generator compliant with NIST standards. Furthermore, they build in methods to ensure bits were not too biased. The two methods used are called Online Health Tests (OHTs) and Built-In Self Tests (BISTs). OHTs compare bit patterns against the expected pattern based on the model of the ES and look at the sample health over many samples to ensure that the samples remain above a predetermined threshold of health. In addition to complying with the NIST standards, engineers also wanted to produce a method which could produce random numbers at a much faster rate than previous methods. They were able to obtain a rate of 3GHz for a ranodm stream of bits. Furthermore, while other methods can consume substantial amounts of power, Bull Mountain requires no additional external power supply to run and is design to function over the wide range of operating conditions.