Problem 13

Ryan Burmeister

January 25, 2016

The article is about the testing and application of random number generators. The author, Robert Davies, delves into how to evaluate the effectiveness of random generators and why anyone should care. He defines a truly random number generator as one which can produce a series of bit which are entirely independent. The higher the correlation between bits, the worse the generator.

He begins by introducing hardware and pseudo-random generators. He favors hardware generators as they are more apt to produce statistically independent bits and more naturally tend to fit the for requirements he gives for random number generation. However, he does indicate that pseudo-random number generators are sufficient in most cases. The cases in which they are not have to deal with long term dependicies between data. In a lottery example, he demonstrates why hardware, if done correctly, can eliminate the necessity to verify that the probability numbers drawn by computers are the same as those which would have been drawn from an urn. With a pseudo-random number generator, it would be hard to prove that the probability of the balls drawn have the same probability of those drawn from a perfectly random scenario, as with the urn. The hardware generator allows for the focus on the evaluation of the other three requirements for computer based draws.

The author then goes into his evaluation and analysis of hardware random number generators. He began by evaluating the a disk of random number captured and was able to determine an error in the programming of the individual capturing the numbers. I believe the author was attempting to demonstrate the ease by which the correlation between bits can be driven up due to a programming error even with an effective hardware generator. In evaluating the Canadian, German, and Californian generators, get measure the percentage of 1s experience in each of the bits while also checking the cross-correlation between bits, sampling rate, and long term drift and perodicities. He then delves into determing the how good a random number generator needs to be by giving using lotteries, statistical simulations, and encryption ase examples.

While the paper is interesting, I ended the paper not having a good feel as to why anyone would need truly random numbers to such a degree as examined in this article. For example, in the encryption example, I question ones ability to predict the key when the range of keys is of sufficient length. Even if there is a slight correlation between bits, does that necessarily mean that I generate some algorithm that can generate the correct keys? With each of the examples given, it appears the correlation between bits only drives up probabilistic likelihood by which you could predict some random outcome. However, how much it improves this likelihood in comparison to other methods I do not know. To improve my own understanding, I guess I would need to better understand the ease by which someone trying to break some encryption scheme with pseudo-random keys would be able to do so.