Randomness

Qcamp 2018, Experimental Session 01/06/2018

1 Introduction

A main ingredient of a secure QKD system is a random number generator. In this exercise, we will explore some aspect about the randomness behaviour.

1.1 In Our Experiment

In our experiment, the random numbers (for polarisation choices) were generated with the "Entropy" library [1]. This library derives random numbers from the (entropic) behaviour in the timing (jitter) of the different timings in Arduino: one generated by the crystal oscillator (clock), and another one from an RC circuit. Figure 1 shows the scatter plot for the generated random numbers.

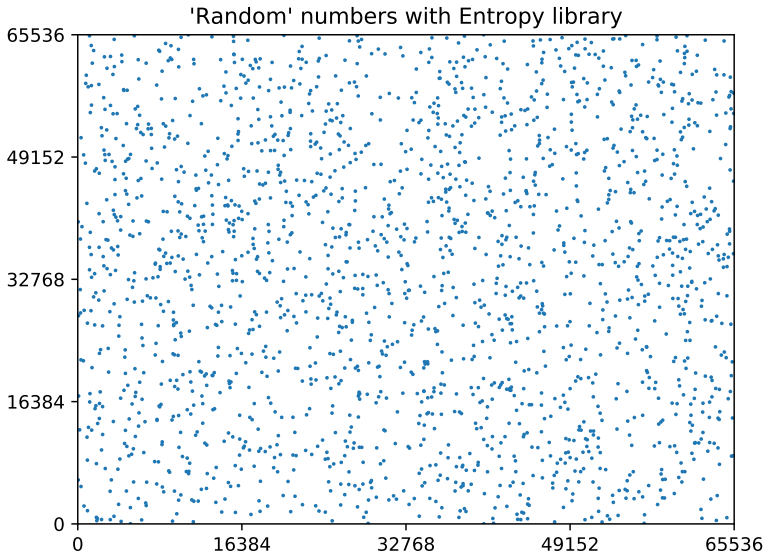

Figure 1: Scatter plot of the random numbers generated with the Entropy library.

One can see that the values generated with the Entropy library are pretty random, but is it? How do you quantify it? How to compare the randomness between, say the Entropy library and the random numbers generated by humans? This question is very hard to answer, and this exercise is designed to give a bit of flavour and perspective in this problem.

1.2 Randomness And Entropy

The concept of randomness is probably related to the concept of disorder, and as physicists (and also computational scientists) we have a quantity for the degree of disorder: entropy [2]. There are many proposed

[1] https://sites.google.com/site/astudyofentropy/

[2] https://en.wikipedia.org/wiki/Entropy_(information_theory)

methods on how to measure entropy and the degree of disorder.

In this exercise, we are going to look into a particular one: BiEntropy [3]. In the paper, they propose two (similar) measures to quantify the degree of entropy: BiEn and TBiEn, which values can range from 0 to 1, with 0 indicating no randomess, and 1 indicating maximum randomness.

There is still another interesting perspective to look into this randomness problem: compression algorithm (i.e. the one used in zip or rar files). Highly random numbers or sequences contain high degree of disorder and thus very hard to be compressed. Thus, one can run a compression algorithm to a sequence of random numbers, and see how much compression is done.

2 Assignment

Task 1 [3 pts] Generate 50 random numbers, with each number ranging from 0 to 9! Every member in the group has to contribute at least 5 random numbers.

Task 2 [2 pts] Run the Arduino random number generator program "gimmeRandom.ino" (in the folder "Random") which produces random numbers on the serial monitor. Write down the random numbers!

[3] https://arxiv.org/pdf/1305.0954.pdf

Task 3 [2 pts] When asked to generate random numbers, studies [4] shows that humans tend to prefer certain numbers or configuration. In this case, we'll see whether humans have some preference to even/odd numbers. Complete the table below, and provide some comments regarding your observation!

	Total odd numbers	Total even numbers	Odd %	Even %
Human (Qcampers)				
Arduino (Entropy)				

Task 4 [3 pts] Now, we look at the randomness estimation result from the "BiEntropy" library and the compression capability with "zlib" program. These results can be obtained by running "analysis.py" (in the folder "Random") [5]. For control, we will use the sequence "0123456789" repeated 5 times. Complete the table below, and provide some comments regarding your observation!

	BiEn	TBiEn	Compression ratio
Human (Qcampers)			
Arduino (Entropy)			
Control			

[4] http://www.cs.cmu.edu/ jblocki/HumanRandomness.htm

[5] For simplicity reason, the input to the program is the ASCII representation of the numbers.