Hardware Security (NWI-IMC065)

Tutorial: Random Number Generators

Goals: This is a guided assignment. After completing these exercises, you should be able to:

- Generate random numbers using the PRNG of your choice and, optionally, the Infinite Noise TRNG.
- Estimate the amount of entropy in a given source.
- Run the Dieharder battery of tests to evaluate the output of a random number generator.

Handing in the tutorial report:

- You can email your report to senna.vanhoek@ru.nl.
- Make sure that your names and student numbers for both team members are mentioned.
- There are two REPORT QUESTIONS marked as such in this tutorial. It is okay if you did not complete all the exercises; we want a best-effort approach.
- This report is NOT graded.

Exercise 1 (Dieharder): For checking the quality of the output of a random number generator, we can use statistical tests. Dieharder, NIST800-22, and SEMB GM/T0005-2012 are well-known randomness test suites. Due to its availability on multiple platforms and ease of use, this tutorial uses the Dieharder battery of tests. Next to many statistical tests, the Dieharder battery of tests comes with various built-in pseudorandom number generators.

1. Download and install the Dieharder battery of statistical tests[1].
2. Get familiar with Dieharder. In the terminal window, try the following commands:
   (a) `dieharder -g -1` What is the result of this command?

[1], `https://webhome.phy.duke.edu/~rgb/General/dieharder.php`

(b) `dieharder -l` What is the result of this command?

(c) `dieharder -d 14` What is the result of this command?

(d) `dieharder -g name generator -t number of samples -o -f file.txt` What is the result of this command?

(e) `dieharder -g name generator -t number of samples -o -B -f file.bin` What is the result of this command?

With your newly acquired skills:

- generate a `.txt` file with 20k samples using `AES OFB`(205) PRNG.
- generate a `.txt` file with 20k samples using `randu` (041) PRNG.
- save these files in a known location.

Table 1: Files to download for exercise 2.

File name	**Description**
`PRNG_visualize.ipynb`	[CODE] `jupyter` notebook, which you can use to visualize the .txt files generated with Dieharder
`INFNOISE_visualize.ipynb`	[CODE] `jupyter` notebook, which you can use to visualize the .txt files generated with the INFNOISE TRNG
`infnoise_raw.bin`	[DATA] raw output of the INFNOISE TRNG
`infnoise_raw_white.bin`	[DATA] whitened output of the INFNOISE TRNG.
`test_20k_AES_OFB.txt`	[DATA] txt file which contains 20000 32-bit random numbers generated by the `AES_OFB` generator with seed = 2164768560.
`test_20k_randu.txt`	[DATA] txt file which contains 20000 32-bit random numbers generated by the `randu` generator with seed = 1667924845

Exercise 2 (Testing random numbers): Download the files for this exercise. The files are described in Table 1.

1. (REPORT QUESTION) Open `PRNG_visualize.ipynb`. There are several functions implemented in this file. In the table below, fill in what each function does.

Function name	Role
`extract_bytes`	
`initialize_byte_dictionary`	
`process_data_in_file`	

Table 2: Functions in `PRNG_visualize.ipynb` file.

2. Visualize data:

   (a) Use the `PRNG_visualize.ipynb` notebook and run the code on the `test_20k_AES_OFB.txt` and `test_20k_randu.txt` files. You can also use the files you have generated in Exercise 1. Save the resulting figures in two separate files.

   (b) Use `INFNOISE_visualize.ipynb` notebook and run the code on the `infnoise_raw.bin` and `infnoise_raw_white.bin` files. You can also use the files you have generated in Exercise 1. Save the resulting figures in two separate files.

3. Analyze the plots:

   (a) Open the two saved plots produced with the `PRNG_visualize.ipynb` side by side. What difference do you observe between `randu` and `AES_OFB`?

   (b) Open the two saved plots produced with the `INFNOISE_visualize.ipynb` side by side. What difference do you observe between the raw and whitened sequence of random numbers?

4. Estimate the entropy of the bytes in the random sequence.

   (a) Find the formulae for of *min-entropy* and *Shannon entropy* using your favorite search engine.

   (b) Implement these formulae and calculate the *min-entropy* and *Shannon entropy* for the bytes in the files by the `test_20k_randu.txt` and `test_20k_AES_OFB.txt`

   Hint: For the values in the file, the following values were calculated.

File	$H(X)$	$H_\infty(X)$
`test_20k_AES_OFB.txt`	0.99	0.97
`test_20k_randu.txt`	0.97	0.86

Table 3: Entropy values for the two files. The values represent the entropy/bit.

   (c) Although close, the two values are slightly different. What is your interpretation of the difference between these two values?

5. Run the Dieharder battery for the tests on a generator of your choice.

(a) Run the command `dieharder -g 205 -a`. Give it some time to produce some output.

(b) In a different terminal window, use the command `man dieharder` to figure out what these parameters mean.

(c) Choose one statistical test to run on your generator of choice. What command do you use?

(d) (REPORT QUESTION) Reflection:

- Did you have enough time for doing the exercises?
- Any other topics we should have covered?
- What is your opinion of such guided assignments?

OPTIONAL Exercise 3 (TRNG). Generating random numbers using the *Infinite Noise* TRNG[2]

1. Install the driver for the Infinite Noise TRNG from here: `https://github.com/waywardgeek/infnoise`.

2. Get familiar with generating random numbers using the infinite noise TRNG; Try out the following commands[3] and determine what they do:

   - `./infnoise --debug --no-output`
   - `./infnoise --raw --debug > file.bin`
   - `./infnoise --debug > file.bin`

   You will need a physical implementation of the infinite noise TRNG for this step. When you are comfortable with your setup, try the following:

   - Generate a .bin file with some samples
   - You can also download the .bin files provided in the resource folder
   - If all you want to do is verify the output using the dieharder tests, you can use: `sudo ./infnoise | dieharder -g 200 -a`, but practise patience as it takes a while to see some output.

[2] `https://www.crowdsupply.com/leetronics/infinite-noise-trng`

[3] Two observations: 1)login to the software folder in the infnoise directory and 2) although not specified, I had to be root to run the commands