Selected Topics on Hardware for Security (NWI-IMC065)

Assignment 2 (2023-2024)

IMPORTANT: This is a group assignment.

Grading: This assignment has a total of ten (10) points obtained by answering the questions. To pass this assignment, you must obtain at least five (5) points. Please check the questions for a breakdown of how the points are awarded.

Assignment	**Points**
Random Numbers	4 points
PUFs	3 point
Mystery board	3 points

Deadline: Wednesday, December 20, 2023, 23:59 sharp!
Before you start: To complete this assignment, your teacher will provide each group with a custom PCB for *Mystery Board*. You will provided with the PCB on December 5, 2023.
Handing in your answers:

- You can hand in your solutions via the assignment module in Brightspace.
- Make sure that your name and student numbers for both team members are on top of the first page!
- For submitting the answers, please submit a `.pdf` typeset solution and other required files to support your submission.

Your teacher will grade your assignment digitally in Brightspace. Although this is a group assignment, all group members should be able to answer any questions about this assignment during the oral follow-up.

Please return the mystery board you received, to your teacher, during the oral discussion scheduled for the 22nd of December or earlier.

1 Testing Random Number Sequences

If you have not done so during the tutorial, download and install the Dieharder battery of statistical tests[1]. Using the built-in PRNGs of the Dieharder test suite, pick a gener-

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

Number	Name
0	Diehard Birthdays Test
1	Diehard Bitstream
2	Diehard 6x8 Binary Rank
3	Diehard DNA Test
4	RGB Kolmogorov-Smirnov Test
5	Diehard Parking Lot Test
6	Diehard Squeeze Test

Table 1: Selection of Statistical Tests in Dieharder

ator. The typical output of a Dieharder generator is a file of 32-bit integers. To answer the questions, generate a file with at least 20000 values.

Q1 - 0.5 points. Visual investigation of the random-byte sequence.

1. Plot the histogram[2] of the byte values in the sequences. To answer this question, add the plot and comment on the uniformity of the output.
2. What information can you find about the generator you picked? (A few sentences are enough).

Q2 - 0.5 points. Entropy estimation:

1. What is the min-entropy $H_\infty(X)$ for the sequence of random numbers in the file?
2. What is the Shannon entropy $H(X)$ for the sequence of random numbers in the file?
3. Compare the value you obtained for $H(X)$ and $H_\infty(X)$. What does the comparison tell you about the sequence provided to you?

Q3 - 1 point. In this part of the assignment, we analyze the statistical quality of the sequence of random numbers in the file.

1. Pick a test in the selected statistical tests in Table 1 and describe how it works (a paragraph should be enough). Copying the test description is not considered an answer to this question. You must explain the test in your own words and use an example to illustrate the test.

   You will use your student numbers to choose the statistical test. Let us take the following s-numbers: s1005614 belongs to T_1[3] *and s1235676 belongs to T_2*[4]*. You*

[2] The histogram describes the number of occurrences for each of the possible 256 values

[3] team member 1

[4] team member 2

determine your statistical test by computing the mod 7 sums of your last digits. In this case: $(4+6)\%7 = 3$*. This means that the target test is Dieharder DNA tests.* IMPORTANT: Please pay attention to how you choose your test. Making an incorrect choice means that this question will not be graded.

2. Run the test on your random number file. What is the result of the test? What can you say about the output of the test?

Q4 - 2 points. Pick two tests in the Dieharder battery of tests (one of these should be the one you used for Q3, the second you can freely choose from *any* in the Dieharder battery of tests):

1. Create a binary sequence that fails one of the tests and passes the other. Please add a screenshot of the test results in the report.
2. Explain how you crafted the sequence (why it fails).
3. Upload the file with the generated sequence and the code you used to create it with this report.
4. Be prepared to reproduce your results during the discussion session.

2 Physically Unclonable Functions

Physically Unclonable Functions (PUFs) are specialized circuits that generate unique and pseudorandom digital identifiers for circuits. In this assignment, you will simulate some arbiter-based PUF constructions and evaluate their statistical metrics. For simulation, we will use the `pypuf` package.

Q1 - 1 point. Simulation of XOR Arbiter PUF.

1. Simulate 10 different instances of 64-bit k-XOR Arbiter PUF (choose k from the set $\{k = 2, 3, 4, 5, 6, 7, 8\}$) using your student numbers. Choose noise rate (`noisiness` parameter) to 0.1. *Add the last digit of your student numbers and compute mod 7. For example if the student numbers are s1005614 and s1235676. Then,* $k = ((4+6)\%7) + 2 = 5$.
   IMPORTANT: Please pay attention to how you choose your test. Making an incorrect choice means that this question will not be graded.
2. Generate $20,000$ random challenges, each of length 64-bits. Repeat the response generation for 20 measurements for each instance for the same set of challenges.
3. Save the responses for each instance as a .npy/.npz file.

Q2 - 1 point. In this part of the assignment, we analyze the statistical quality of the responses produced by the PUF instances. The statistical quality of PUF responses is computed using three performance metrics: *uniqueness, uniformity* and *reliability.* Report the performance metrics of the responses generated in the previous question. For this, you need to perform the following steps:

1. Compute the golden PUF response by performing majority voting over 15 measurements.
2. Compute the uniformity of the PUF responses using the golden response and the *bias* metric in `pypuf` tool. Report the bias of each instance along with its aggregated value. Mention in the report how is the bias metric related to the uniformity of PUF response.
3. Compute the uniqueness of the chosen PUF design from the *uniqueness* metric in the `pypuf` tool. Plot the frequency distribution (histogram) of the Hamming distance (HD) of the golden response obtained from each pair of instances. For 10 instances, the number of instance pairs will be $\binom{10}{2} = 45$. You must compute the HD of all 45 pairs and plot their frequency distribution.
4. Compute the reliability of each of the instances. For this, you need to compute the HD between each measurement of the responses with the golden response. Plot the histogram of the HD as well.

[For plots you can use `matplotlib` package, as shown in the tutorial.]

Q3 - 1 point. Explain the contribution of the number of measurements on the reliability of the PUF response.

1. Generate the temporal majority-voted response for each PUF instance from 5, 15, and 25 measurements.
2. Compute the HD of the golden response (generated in **Q2**) from the majority-voted responses obtained from the previous step. Explain how the discrepancy in the responses changes with different number of measurements.

3 Mystery Board

Searching for information is a key skill required in hardware hacking. This assignment aims to uncover as much information as possible about the ToE provided to you.

Q1 - 1 point. Identify one microcontroller on the ToE. Specifically:

- Provide a picture of the TOE which highlights the microcontroller;
- Describe the model and manufacturer;
- Which PIN(s) are GND? Add a picture with the chip pin-out corresponding to this information;

- Describe the size and type of memory on-chip in the microcontroller;

Q2 - 1 point. Describe and identify a component present on the ToE other than the microcontroller you chose above (it can be another microcontroller).

- Add a figure of the TOE which highlights your chosen component;
- Describe the designator, the markings, and a figure of the component;
- Describe the main function of the target component as specified by the datasheet.

Q3 - 1 point. Can you identify any debug interfaces? If yes, please specify:

- Is there a UART interface present on the board? (add a figure of the pinout and pin type.)
- Is there a flash chip present? (add a figure of the pinout and pin type.)
- Is there an I2C interface on the board? (add a figure of the pinout and pin type.)