ECE 592 / CSC 591: Cryptographic Engineering and Hardware Security

Assignment 2: Breaking AES through DPA

Instructor: Dr. Aydin Aysu
Email: aaysu@ncsu.edu

1 Introduction and Goals

The purpose of this assignment is to provide a hands-on introduction to power-based side-channel analysis. You are given a set of power traces taken during AES-ECB executions with the associated input/output values of AES, and you are asked to extract the 128-bit secret key via Differential Power Analysis (DPA).

The assignment is provided to you with four files in addition to this .pdf instructions file. These are:

1) `input_plaintext.txt`

2) `output_ciphertext.txt`

3) `traces.csv`

4) `traces2.csv`

5) `aes128_table_ecb.v`

These files correspond to 10000 AES-ECB executions with random input and a fixed, secret key. Hence, there are 10000 distinct input values, corresponding output, and power measurements. `input_plaintext.txt` is the set of 128-bit input plaintexts of AES, `output_plaintext.txt` is the set of 128-bit output ciphertext for AES, and `traces.csv` and `traces2.csv` contain the power measurement for each input-output pair for two slightly different hardware implementations. For your reference, the `aes128_table_ecb.v` file is the verilog for the AES hardware implementation used to capture `traces.csv`. (I made minor modifications to integrate it to a platform to run measurements.)

1.1 Graded Items

You will turn in a soft copy report of your results as well as any code to the TA on Moodle. Make sure to include answers to every question in the report.

1. **Figure 1** Plot of the first power trace.
2. **Figure 2** Plot for all key guesses.
3. **Figure 3** Plot of two best key guesses.
4. **Figure 4** Plot the "evolution" of all key hypothesis for 10000 measurements at this particular time instance.

Each question is worth 12.5 points. The last two questions, therefore, can be treated as bonus. Lab reports must be in *readable* English and not raw dumps of log files. Your lab reports must be typeset and must not exceed 6 pages. You will be required to use LaTeXto generate the reports. You can use overleaf to generate the LaTeXfile. Please submit your lab report (in .pdf format) and all of your code in a .zip file on Moodle as lab1_YOUR_UNITY_ID_HERE.zip

You can use any software language you prefer to write the programs required to complete for this assignment. If you have any questions about the assignment, please first refer to the TA of the course (Anuj Dubey, aanjudu@ncsu.edu).

2 Problem 1: Analyze power traces

To start the analysis use `traces.csv` and omit `traces2.csv` for now.

Figure 1 Plot the first power trace, corresponding to the first encryption operation. **Question 1:** Analyze the power trace. What are the peaks in the trace? How is AES implemented?

Before we proceed to DPA, think of a good pre-processing on the acquired power process to make the attack more successful. Hint: Plot the first few power traces and think how to better align them.

3 Problem 2: Perform the DPA Attack

Let's execute the DPA attack using Pearson's correlation test. First, think of an easy target operation for the DPA and, second, consider the number of bits to be estimated by DPA at a time.

Question 2: What is a good power model for the target operation? Justify it. Feel free to try out different power models before you commit to one.

Apply the DPA on the first byte of the key, plot the results **Figure 2** for all key guesses.

Find the maximum correlation (consider both positive and negative peaks) and plot the two best key guesses **Figure 3** that have the maximum likelihood.

Question 3: Which one is more likely to be the correct key guess? Why do you specifically see these two results?

Question 4: What is the maximum leak point (in time domain)? Plot the "evolution" of all key hypothesis for 10000 measurements at this particular time instance **Figure 4**.

Question 5: Starting at how many traces, does the key guess with maximum correlation show highest correlation?

Question 6: Apply the DPA on the entire key. What is the 128-bit AES secret key? Note that not all key values could be 100% detected depending on your targeted computation, i.e. there can be false positives. If this is the case, you may need to take the first n possible results and apply a brute-force search within these reduced search space. Hint: Targeting the initialization may especially cause this behavior, targeting computations after the initialization might be a better idea. You can also try the partition-based tests (With difference of means) to extract the key.

4 Problem 3: Evaluating DPA

Question 7: What is the mean time to disclosure, i.e. average number of traces required to extract the key for your DPA attack? To find this average, you need to estimate starting at how many measurement the targeted secret key portion leaks and average this value for all parts. If the key cannot be correctly estimated for a particular key part, you can use the maximum number of measurements (i.e. 10000) for that part.

Question 8: Compare this with the theoretical cryptanalysis (i.e. brute-force attack) of AES, what is the reduction ratio in the number of traces required to break AES? Why is DPA so powerful?

Question 9: How much time it took you to perform the experiment and complete questions 1–8?

5 Extending the DPA Attack

Now use the traces2.csv, which is a set of measurements obtained from a slightly different implementation. Apply the same DPA attack using these traces. **Question 10:** Why is it failing now? What is the difference in hardware implementation causing this and how would you change the attack to break the implementation?

Question 11: Assume that you do not have access to input plaintext but only to output ciphertext (i.e. ciphertext-only cryptanalysis), how would you modify the attack? Hint: You can search papers on DPA of AES and see where/how they perform the attack.