

Hardware Security - Side Channel Analysis Assignment 2

Eduardo Hoefel (s1043186)

Bram Thijssen (s4308905)

Group: SCA 16

20 March 2020

Implementation

To implement the power attacks, we decided to use python 3 with external libraries to handle graph plotting (matplotlib) and matrix correlation calculations (numpy and scipy). The code is attached to this report.

Part 1

Part 1 is about implementing an attack against hardware-based cryptography using AES-128. The attack was performed during the last round of encryption. Picture 1 shows the sequence of operations happening.

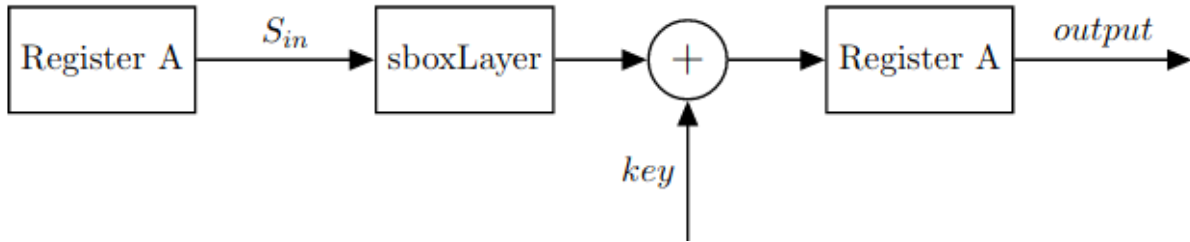


Figure 1: Sequence of operations during the last round of AES-128.

Since this attack is on a hardware implementation, the hamming weight operation was substituted by hamming distance between *old* and *new* Register A values. The procedures applied over the initial data is listed below.

input, traces = *load*() (1)

keys = [0, ..., 255] (2)

matrix = (*input* \oplus *keys*) (3)

sbox = *sboxlayer*(*matrix*) (4)

d = *hd*(*sbox*, *input*) (5)

correlate(*traces*, *d*) (6)

In (3), *matrix* refers to the XOR combination of all input values with all key possibilities. Then on (4) the values are translated according to the reverse *sBoxLayer* table used by AES-128. Then, the resulting table is compared to the original input using the hamming distance operation. Finally, the data is correlated to the power traces provided, and the absolute value is analyzed.

The best correlation was found using key 0x47 (decimal: 71, binary: 1000111), which gave a maximum absolute correlation value of 0.13035009182. The correlation value achieved was about 272% higher than the second best candidate (key 0x5A). Figure 2 has a plot of the maximum correlation value achieved by the key candidates for each power trace time.

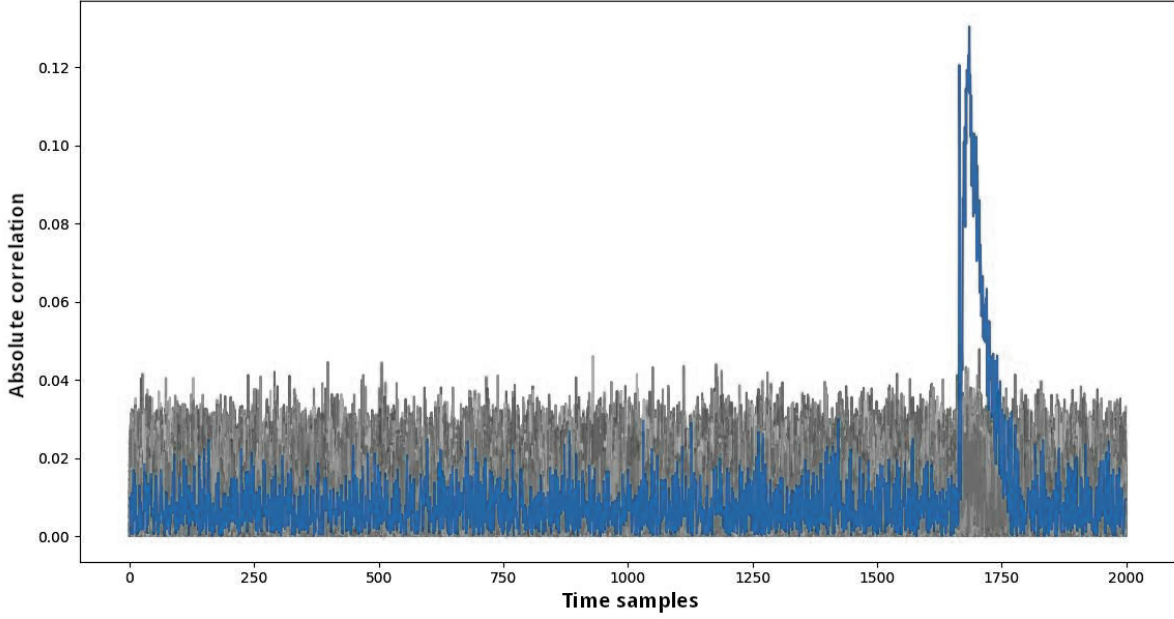


Figure 2: Absolute correlation value per key candidate, with key candidate 0x47 highlighted in blue

Part 2

Part 2 was about running an attack against a protected implementation. Figure 3 has a diagram of operations happening with this cipher.

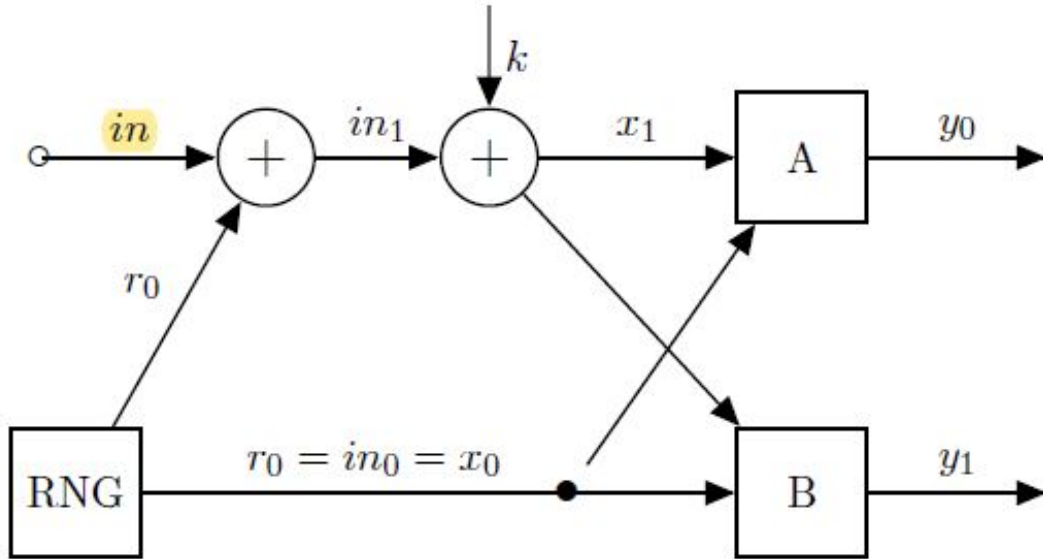


Figure 3: Cipher under attack, where $y = y_0 \oplus y_1$

Looking at the cipher, we first tried to figure out what to correlate. The traceset contains only 10 time samples of the leakage produced after the Substitution layer, for 2k traces. The cipher uses *masking* to randomize key-dependent intermediate values. Therefore, a direct correlation of input with the collected sample data would fail, as we can't distinguish the masks from the original key. To solve this problem, our approach involved pairing power traces, reducing the influence of random values and highlighting the weight of the key during a correlation execution. By combining all possible time samples, we created

a table of 45 power traces.

What we tried to in our side-channel attack, that gets around the masking protection, is to try calculate all possible y values from the in values that we got in our data set. We soon realized that this would become infeasible. Especially as lookup table A is secret and random, we would need to use a lot of random numbers. So we inspected the cipher a bit closer. In short all that is happening is that the input is XOR'ed with a random number before the key shift is applied and this would afterwards be taken out again by XOR'ing it back with the random number that was used. We also took a closer look in what the tables were doing so we could dissect this baby:

\mathcal{A} receives values:

x_0 and x_1

and produces $y_0 = A(x_0 || x_1)$

\mathcal{B} receives values:

x_0 and x_1

and produces $y_1 = S(x_0 \oplus x_1) \oplus A(x_0 || x_1)$

When we plug these into the following equation we get:

$$y = y_0 \oplus y_1$$

$$y = A(x_0 || x_1) \oplus (S(x_0 \oplus x_1) \oplus A(x_0 || x_1))$$

$$y = A(x_0 || x_1) \oplus S(x_0 \oplus x_1) \oplus A(x_0 || x_1)$$

$$y = S(x_0 \oplus x_1)$$

This shows that we can omit everything happening in table A , when trying to compute the value of y if we remove the XOR earlier.

Similarly we can omit the XOR'ing happening with the random number r_0 .

$$r_0 = in_0 = x_0$$

$$x_1 = in_1 \oplus k$$

$$in_1 = in \oplus x_0$$

$$y = S(x_0 \oplus x_1)$$

unfolding x_1 gives:

$$y = S(x_0 \oplus in_1 \oplus k)$$

unfolding in_1 gives:

$$y = S(x_0 \oplus in \oplus x_0 \oplus k)$$

simplifying:

$$y = S(in \oplus k)$$

Which looks like something we can compute.

After that, we defined the sequence of functions to calculate the correlation as:

$$input, traces = load() \tag{7}$$

$$keys = [0, \dots, 15] \tag{8}$$

$$matrix = (input \oplus keys) \tag{9}$$

$$sbox = sboxlayer(matrix) \tag{10}$$

$$pairs = combinations(traces) \tag{11}$$

$$correlate(traces, sbox) \tag{12}$$

Line (11) represents creating a matrix of all pair combination between the 10 time samples, resulting in a 45×2000 matrix.

We calculated the correlation between the traces and the computed input using each key candidate. On table 1 we can see the maximum value achieved for each key.

Key	Absolute correlation value
0	0.1106371319756917
1	0.0580769122756410
2	0.0523095802839271
3	0.1982485104891907
4	0.0454361770139361
5	0.0824984847962257
6	0.1320831366769909
7	0.0336660427605671
8	0.0789106227989042
9	0.0832624127617274
10	0.0767710160456761
11	0.1255380282268774
12	0.0654640084985294
13	0.0287353027298031
14	0.1053606065666633
15	0.0536736979301641

Table 1: Maximum absolute correlation found for each key candidate.

The best correlation was found using key 3, which gave a maximum absolute correlation value of 0.19824851. The correlation value achieved was about 50% higher than the second best candidate (key 6). Figure 4 has a plot of the correlation value achieved by the key candidates for each power trace time.

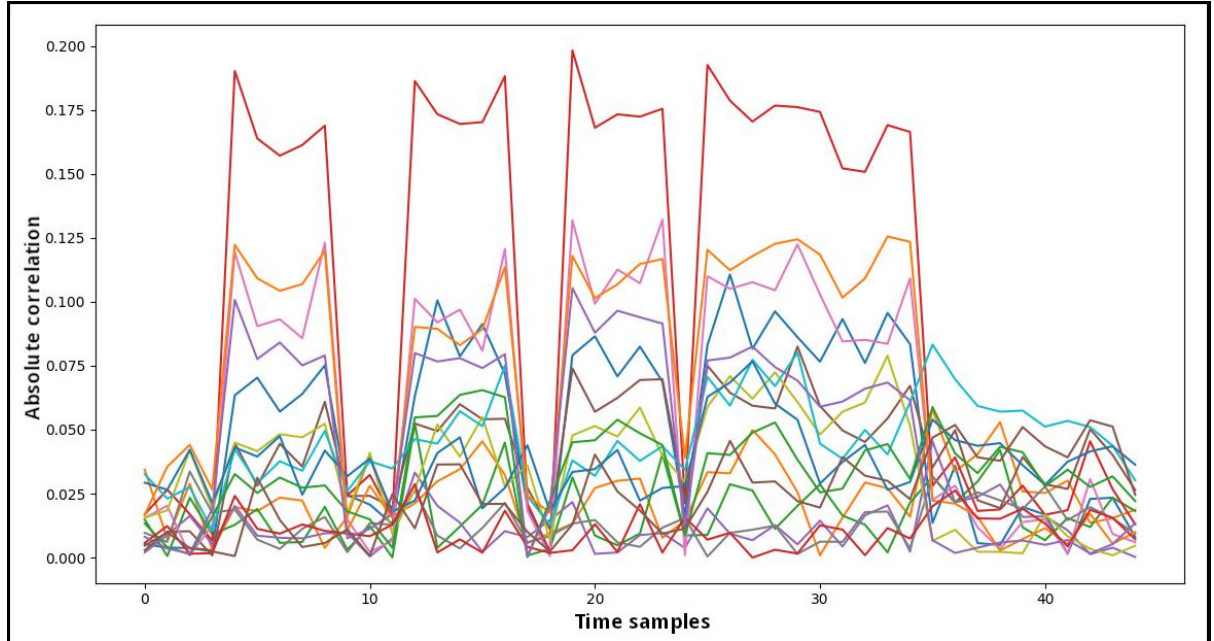


Figure 4: Absolute correlation value per key candidate per time sample, with key candidate 3 highlighted in red

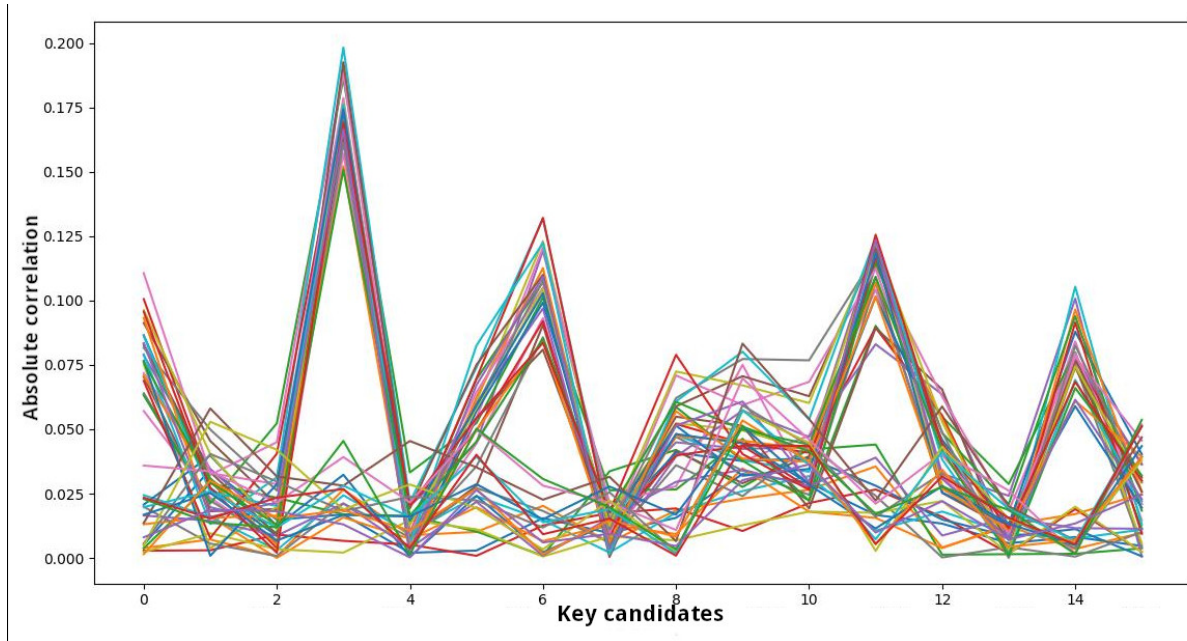


Figure 5: Comparison of highest correlations achieved on each power trace, grouped by key candidates

Figure 5 groups the correlations achieved by key candidate, highlighting the amount of traces with high correlation achieved by the key candidates.