hastim@oregonstate.edu
Mohan Krishna Hasti

Homework 3

1. For this task, we consider two types of devices. i) device of type A implements the AES with just a single S-Box in hardware; ii) device of type B implements the AES with 16 S-Boxes to do a full AES round within just one clock cycle. Both devices are now studied as part of a power analysis attack. Let us consider the behavior of the noise in more detail: (20 pts)

a) In terms of a power analysis, which device is more likely to show higher (algorithmic) noise when assuming randomly distributed plaintext inputs? (3 pts)

Device A is shows less algorithmic noise than device B because of the 16 S-Box computations. So, Device B has more power consumption than device A.

b) Assuming all other device characteristics are the same, which one will be more difficult to attack? Note that difficulty of attack is the number of traces required for a successful key extraction. (7 pts)

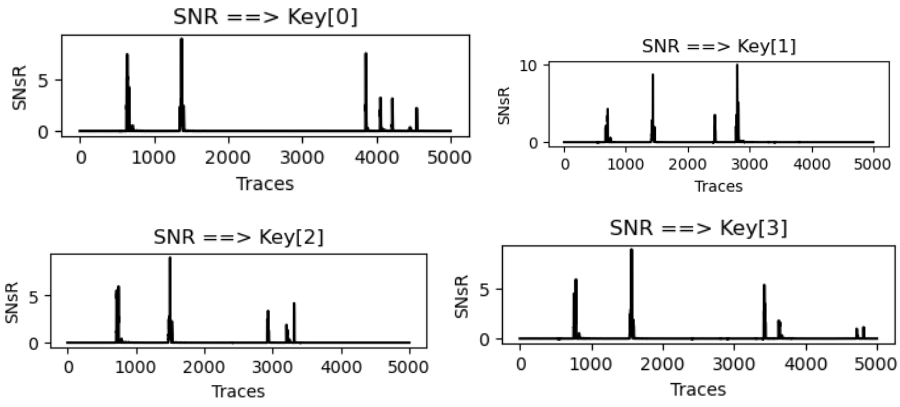
Device B is difficult to attack due to the presence of 16 S-boxes and the power consumption traces is difficult to attack and requires many traces to extract the key. Attacking device A is easier as it has only one S-Box and the power consumption trace is easier to analyse and can be easily attacked.

c) Assuming there are no algorithmic countermeasures, is it possible to adapt the attack on device of type B such that when attacking the first round of AES, the attack would behave similar to device of type A? (hint: the attacker only controls the plaintext input)(10 pts)

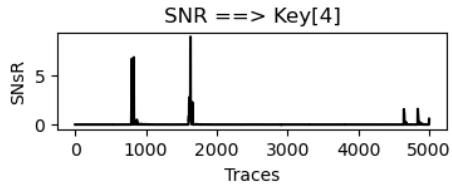
Yes, its possible to replicate the attack on Device B. By considering only one S-Box from the traces, we can replicate the attack, but it might still be difficult because of the noise from other S-Box power consumption.
We can use a CPA to attack the device B by comparing the plaintext traces and having a key-hypothesis to extract the key.

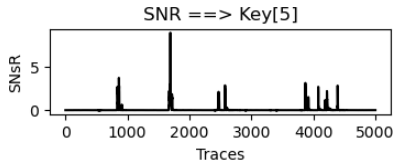
2. In previous assignments, you worked with simulated data that may not necessarily reflect reality. For this task, you will work with actual measurement data of an 8 bit microcontroller running a software implementation of the AES. This will be our first attack on actual measurement data. For this task, use the traces intended for attack, i.e., random plaintext and unknown fixed key. (30 pts)

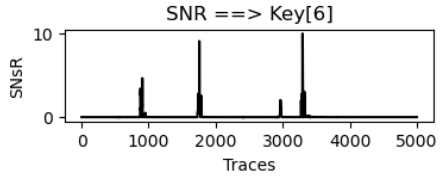
a) Create the SNR plot based on the plaintext input to quickly narrow down the number of points you have to work on. (10 pts).

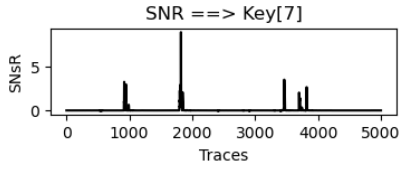


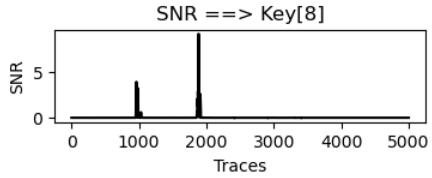
hastim@oregonstate.edu
Mohan Krishna Hasti

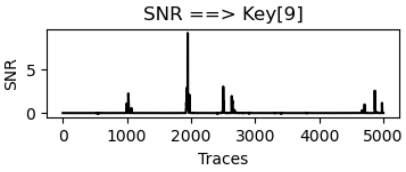


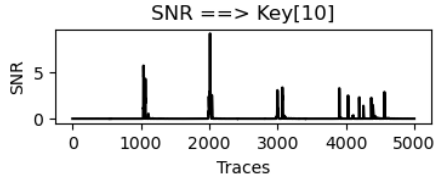


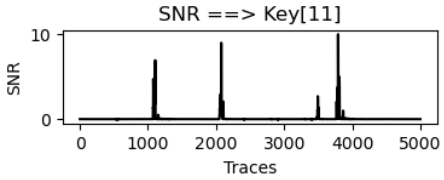


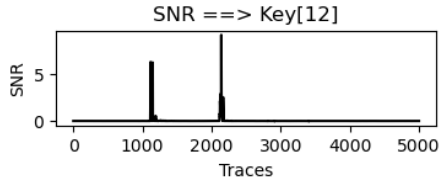


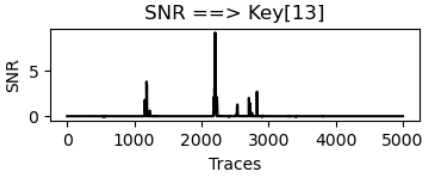


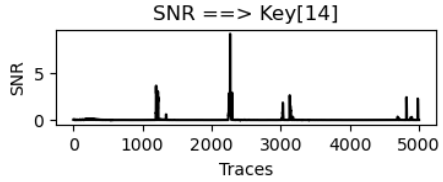


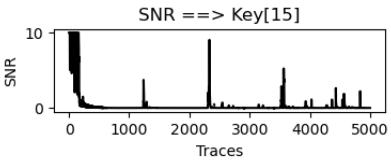






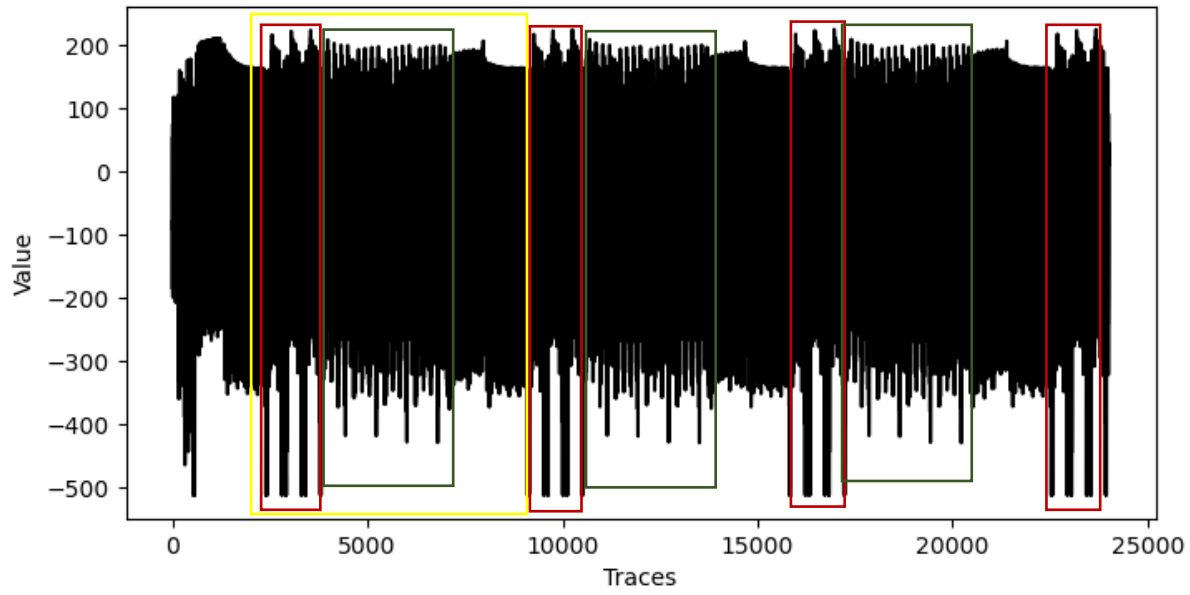






hastim@oregonstate.edu
Mohan Krishna Hasti

b) Create a plot where multiple traces with the same input data are averaged to remove the noise. Annotate the plot with at least: points in time where KeyAdd and SubBytes of the first round are performed. Identify the whole AES round. (5 pts)



KeyAdd operation **SubBytes operation** **AES round**

c) Run a CPA to recover the key. Try different power models such as Hamming Distance/Weight, on different intermediate values such as KeyAdd and SubBytes. What do you observe? What works best and why? (15 pts)

Key = **42, 127, 20, 23, 41, 175, 210, 166, 170, 246, 21, 137, 8, 207, 78, 61**

Total execution time = **129.236 secs = 2.15 mins** for 7000 data (S-Box with Hamming Weight)

Total Time = **207.689 = 3.46 mins** (KeyAdd with Hamming distance)

Using hamming distance was slower than hamming weights.

Attacking at the KeyAdd operation resulted in duplicate keys and the window to attack KeyAdd is very less and is not vulnerable in most devices as it involves only a simple addition of key and a plaintext.

Attacking at the SubBytes is easier and more vulnerable due to the large power consumption and a larger window of attack and it resulted in unique keys.

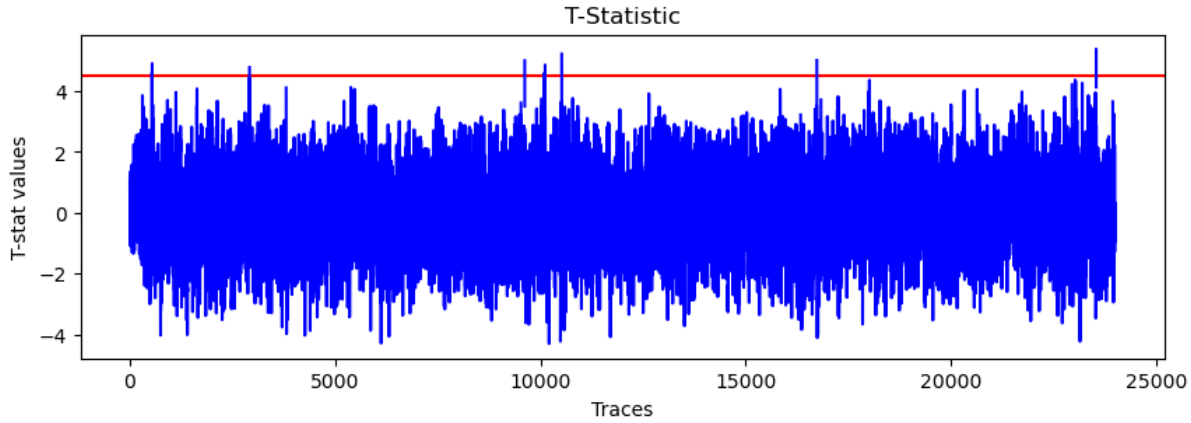
Attacking at the SubBytes is better due to a larger window.

d) Bonus: Create "measurements to disclosure" (MTD) plots for the different hypotheses that you tested in c) (5 pts)

3. In the following, you are tasked to implement one of the following methods. (150 pts)

Welch's *t*-test (1st order)
– Implement the leakage test and plot the result.

hastim@oregonstate.edu
Mohan Krishna Hasti



– Does the implementation show leakage? If so, where?

Yes, the t-test shows some leakage, and it crosses the 4.5 threshold at couple of places, so the test fails.

Points that cross 4.5 threshold: **535, 543, 2915, 9619, 10099, 10115, 10519, 16735, 23539**