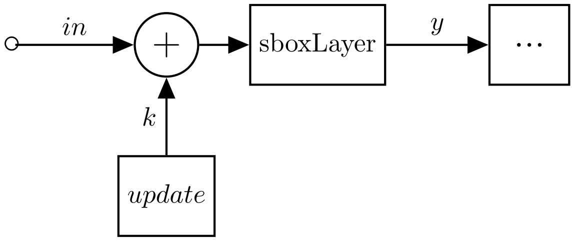
Exercise on Correlation Power Analysis

1 Exercise Description

The goal of this exercise is to code, using the MATLAB/Python environment[1], the CPA attack against a software-based implementation of cryptography. Through a lab experiment, we have measured the electromagnetic emission of the cipher, using a Langer MiniProbe and a Lecroy oscilloscope.

- The cipher under attack is the lightweight PRESENT cipher:
  `http://www.lightweightcrypto.org/present/present_ches2007.pdf`
- The attack targets the 1st round of PRESENT encryption, in particular the value y computed by the sboxLayer:



- All variables in the image above (in, k, y) contain 4-bit values (nibbles)

- ► The exercise goal is to recover k, the 4-bit portion (chunk) of the 1st round key
- ► The attack point will be the intermediate value y, which depends on k and in
- ► Note that the cipher is implemented on an ARM Cortex-M processor. The trace acquisition is performed with an electromagnetic probe placed on top of the chip, which often increases the amount of noise in the signal (in other words correlation peaks will not be very clear!)
- ► The measurement traces required for the attack are available here:
  `https://mega.nz/#!f0kTFYJQ!MEDiR9Fe6Gb9mR4jCcJNFzsunsBJXfuAp56XTisYWHk`
  `https://mega.nz/#!utFhHKJa!RP7N1Ms9nqtIz3nKl1V5Le5F9HsWAX5hPPpPO6e2Z_c`

[1] check `https://datanose.nl/#byod` to use the UvA MATLAB licence

2 Exercise Steps

1. Load the file `in.mat`. It contains 14900 4-bit inputs (variable *in*) that will be used for the attack

2. Using the *in* variable and the structure of the PRESENT cipher, construct the value-prediction matrix on variable y. Analytically, you need to predict all possible values of y, by using *in* and guessing all the values of the 4-bit key chunk, k.

3. Convert the value-prediction matrix into the power-prediction matrix by using the Hamming weight model

4. Load the file `traces.mat`. It contains 14900 aligned power traces, each one with 6990 time samples.

5. For all possible k candidates, compute the column-wise correlation between the traces matrix and the power-prediction matrix (e.g. use the `corr` MATLAB function).

6. Rank the key candidates from best to worst, based on the absolute value of the correlation function. Demonstrate the top candidate based on absolute correlation.

7. Create the following graph: For every time sample, plot the absolute correlation value for every k candidate. Highlight the top candidate (e.g. using a different color). A similar graph is provided in the book "Power Analysis Attacks by S. Mangard et al.", page 126, figure 6.3.

8. Create the following graph: Run the attack with 500, 1k, 2k 4k, 8k and 12k power traces and for every attack, rank the candidates from best to worst (based on the absolute correlation value). Focus on the correct candidate, i.e. the one you recovered previously using 14900 traces. Plot the correct candidate's ranking (e.g. 1st, 2nd etc.) for all these attacks.