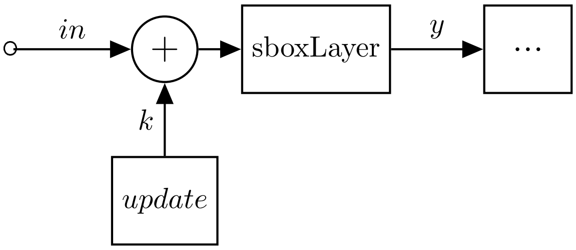
Side-Channel Analysis Assignment 1

March 11, 2019

1 Cryptographic Implementation Description

The purpose of this assignment is to implement in detail the Correlation Power Analysis attack (CPA) against software-based cryptographic implementations. For this assignment, the encryption algorithm used is the PRESENT cipher, linked below.
`http://lightweightcrypto.org/present/present_ches2007.pdf`.
The attack will be performed during the 1st round of encryption, as shown below.



All variables in the image above (in, k, y) contain 4-bit values (nibbles). The CPA will be performed in this reduced version of PRESENT cipher, i.e. you need 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. 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!).

2 Attack Description

The full theory with respect to the CPA attack is provided in Chapters 6.1, 6.2.1 of the Power Analysis Attacks book.

The data files required for the attack can be downloaded here:
`https://mega.nz/#!f0kTFYJQ!MEDiR9Fe6Gb9mR4jCcJNFzsunsBJXfuAp56XTisYWHk`

`https://mega.nz/#!utFhHKJa!RP7N1Ms9nqtIz3nKl1V5Le5F9HsWAX5hPPpPO6e2Z_c`

We describe the intermediate steps here:

1. Open the "in" Matlab file. It contains 14900 4-bit inputs that will be used for the attack (i.e. the *in* variable).
2. Based on 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. Open the "traces" Matlab file. 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 Power Analysis Attacks book, 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.

3 Submission

Please submit your code in MATLAB or Python together with a small report(pdf) with the results.

4 Contact

For questions, meetings or anything related, mail us at *nsamwel@cs.ru.nl*.