Project 3: DPA on AES-128

The goal of the project is to launch Differential Power Analysis attack on AES-128 and recover the round key used in the last round.

In AES-128, the plaintext block is first XORed with the primary key and then goes through 10 rounds of processing. Each of the first 9 rounds consists of four steps: SubByte, ShiftRow, MixColumn, and AddRoundKey. The 10th round is similar but does not have MixColumn.

The data you use are from Differential Power Analysis contest [1], which is held for researchers to compare different attack algorithms in an objective manner. The acquisitions were performed on a SASEBO GII board [2]. The design used for the acquisitions is done in Verilog. In the implementation, the state (128 bits) are stored in a register and updated every round. At the end of the 10th round, ciphertext is stored in the register.

A potential target is the bit transitions when updating the register. For example, at the beginning of the 10th round, eight bits in byte 0 of the state are stored in the left most end of the register. These 8 bits are the input to the first SBox. At the end of the 10th round, byte 0 of the ciphertext is stored at the same location. When the register is updated, some bits are changed and some are not. The number of transitions may be detected from power traces. In this project, you will try to recover the round keys in the 10th round with DPA. Note that the 10th round does not have the MixColumn step.

Since the official dataset is too big, we prepared a smaller set of data for attacking 10th round. The dataset provided includes 20,000 power traces and corresponding ciphertexts. They are available in txt file (for C code) and Matlab format.

For C code, there are two files.

- cipher.txt contains 20,000 lines of cipher text. Each line is 32 hexadecimal digits representing 128 bit values. The first two digits are byte 0, the next two digits are byte 1, and so on.
- pts.txt contains 20,000 lines of power trace. Each line has 700 integers separated by comma. A successful attack can be launched with first 100 points in each power trace.
- pctext.txt contains the plaintext and ciphertext pairs for all 20,000 traces. The 32 hexadecimal digits after ‘m=’ is the plaintext and those after ‘c=’ is the ciphertext.

A template file is included in the package, which has functions to load ciphertext and power traces into memory.

The data also available in Matlab data files.

- Ciphertext.mat contains 20,000 lines of cipher text. Each line is 32 char each represented in hexadecimal. After you load the ciphertext.mat into Matlab, you can see an array of chars. If you type cipher(1,:) into the command line, you will see the first string of cipher text e6a636e30c85f35e980f3546a04daff7. Each string consists of 32 hexadecimal char for 16 internal states of each round. The first two hex ("e6") is byte 0, the next two hex is byte 1 and so on. The hexadecimals need to be converted to binary before being used in the computation.

- Powertraces.mat contains 20,000 lines of power trace or signals as normal decimal numbers. Each power trace has 700 points and corresponds to a line of cipher text in Ciphertext.mat. If you type plot(pts(:,1)) into the Matlab command line, you will see the power trace for the first string of cipher text. Note that Matlab uses column major. A successful attack can be launched with first 100 points in each power trace.

Deliverables:

- You will launch DPA attack in using cipher text and the corresponding power traces provided, and then guess the round keys in 10th round. Here is some information that can help you check your results.

  Considering the power differences at all 8 bit locations gives your better chance to succeed.

  The correct value for byte 0 (index 1 in Matlab) is 0x53.

  The correct value for byte 6 (index 7 in Matlab) is 0x2B.
- Find out the master key from the 10th round key you have recovered. And check if you can generate correct ciphertext blocks from corresponding plaintext blocks.
- Write a report documenting your method and results.

References:

[1] DPA contest v2: http://www.dpacontest.org/v2/download.php

[2] SASEBO: http://staff.aist.go.jp/akashi.satoh/SASEBO/en/index.html