University of Maryland Baltimore County
CMPE/ENEE 691
Hardware Security
Spring 2023

HW8: Power Analysis Attack on AES-128
Due Date: 04/21/2023

The goal of the Homework is to launch 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 attack in a lab. 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 Power Analysis Attacks. Note that the 10th round does not have the MixColumn step.

Since the dataset is too big, we prepared a smaller set of data for attacking 10th round. The dataset provided includes 7500 power traces and corresponding ciphertexts. They are available in txt file.

You can find the following files in the package provided.

- cipher.txt contains 7500*16=120000 lines of cipher text. Each line is 1 decimal number representing 1 byte of the AES cipher text. The first number is byte 0 for cipher trace 1, the next number is byte 1 for cipher trace 1, and so on up to 16th line that includes byte15 of cipher trace 1. 17th line includes the byte 0 for cipher trace 2, 18th line refers to the byte 1 of cipher trace 2 and so on. So, for 7500 traces we have 7500*16=120000 lines.

- power.txt contains 7500 lines of power trace. Each line has 2500 integers separated by a gap. It is the power trace for only round 9 – 10 of AES. So, use the whole traces.

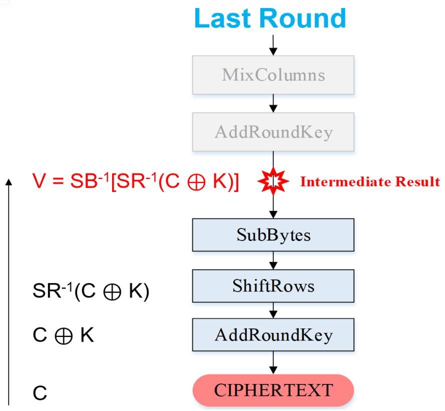
- InvSubBytes.mat: It is a MATLAB matrix. After loading, if you give a byte in decimal of its index, it will return you corresponding InvSubBytes in decimal. For example if you want to have the inverse sub byte of 25, you can have it in MATLAB via InvSubBytes(25+1) = 142.
  inverse sub byte of N = InvSubBytes(N+1)
- HW.mat: It is a MATLAB matrix. After loading if you give a byte as a decimal of its index, it will return you corresponding Hamming Weight as a decimal.
  Hamming Distance of N = HW(N+1)

Suggestions:

- Use Matlab to design your attack. You can use any other coding but it is easier to implement in Matlab (Use UMBC id for Matlab account which gives you free license)
- Use HD model for hypothetical data. It represents the best power model.
- Use Matlab function “load” for loading the files provided
- Use Matlab function “corrcoef ( )” for implementing the “r” matrix
- You can use Matlab function “bitxor( )” for doing XOR between two numbers
- AES has 256 key guesses for each byte of key, So, for R matrix you will have 256 rows.
- AES of this design has been implemented as a block cipher as below:

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

- Note that you have the cipher text so to find the 10th round key (which has max correlation) you need to guess the intermediate values inversely like the figure below.



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 of byte 0 is 19 = 0x13.
The correct value of byte 3 is 127 = 0x7f.
The correct value of byte 6 is 74 = 0x4a.
The correct value of byte 9 is 7 = 0x07
The correct value of byte 12 is 77 = 0x4d

Deliverables:

- You will launch attack by using cipher text and the corresponding power traces provided, and then guess the keys of the 10th round
- Find out the whole 16-byte key for the 10th round you have recovered and put it in a text file “round10key.txt”
- Submit all the files as single zip file.

Useful Hint:

1) If you are going to implement the algorithm in MATLAB, please consider that all indexes of arrays start form 1 not zero. However, in programming languages such as C/C++ and python the index of arrays starts from 0. In case of using Inverse SBox and Hamming Distance, and other variable in the code consider this hint.

2) When you are going to calculate hypothetical intermediate value, you need to apply Inverse Shift Row and then Inverse SBox. It is recommended just use the first 16 byte from cipher.txt file and assume Key = 0 to test your code and make sure that you did not do mistake in inverse SBox and Inverse Shift Row.

3) In case of using MATLAB, function **corrcoef()** will help you to compute the correlation, you should pick one column from hypothetic power matrix and one column from power trace matrix.

   $corr\_out = corrcoef(h_i, t_j)$
   $i = 1,2,3 \dots 256$
   j = 1,2,3,…2500

   The output of corrcoef() is a 2×2 matrix. Please use corr_out(1,2) as the output . Now you have 256×2500 matrix that each cell is the output of correlation function. Find the row index and column index of maximum value of this matrix. The row index is key. (Do not forget that you need to decrease one unit from key value as the index in MATLAB start from 1). Repeat point 3 for all other 15 bytes of the cipher text to find all other 15 bytes of key.