Systems and Network Security Laboratory

Homework 2: AES Correlation Power Analysis

Deadline: 2019/5/20 13:59

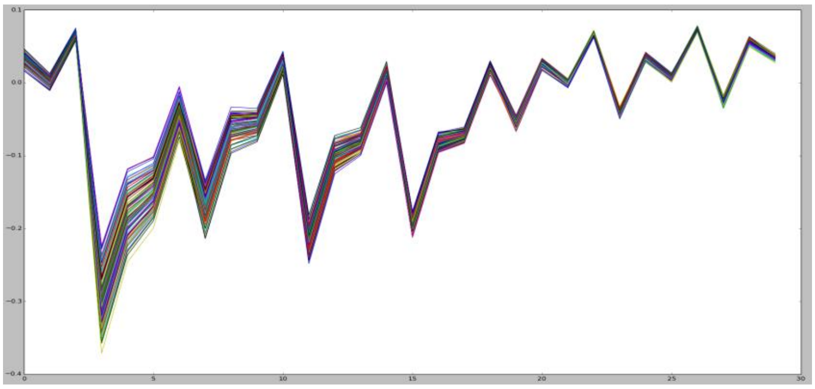
jiunpeng@ntu.edu.tw

1. Problem Statement

Cryptographic devices provide us a way to hide our sensitive information in our daily life. However, it is shown that Side-Channel Analysis such as power analysis can reveal the secret information of a cryptographic device, and compromise its security, even when it is correctly implemented. One of the most popular side-channel analysis against the AES block cipher is the Correlation-based Power Analysis, also known as CPA. This type of attack doesn't require detail knowledge about the targeted device, knowing the encryption algorithm is often enough to launch an attack. In this assignment, **please implement a CPA program** that can recover the encryption key of an AES-128 software implementation **using the side-channel power traces and the Ciphertexts information**.

2. Correlation Power Analysis

The goal of the power analysis attack is to reveal the secret key of a cryptosystem. CPA uses a large number of traces to analyze the power consumption at a fixed moment of time as a function of the processed data. By exploiting the data dependency of the power consumption, the attacker can then use the correlation coefficient value to find the secret key based on the differences in power consumption traces.



Here is the general description of a Correlation Power Analysis:

1. Choosing an Intermediate Value of the Algorithm:

   Find an intermediate value as a function of a known value and part of the key. For example, the 1st round S-Box output in AES, S-Box(Plaintext ⊕ Key). The output of AES S-Box is an 8-bit value, which also contains 8 bits of key information. By dividing the 128-bit key into 16 bytes, attacking 1 byte at a time, we only need 2^8x16 search to recover the key, the search space is a lot smaller than using brute-force on the original 128-bit key.

2. Calculating Hypothetical Intermediate Values:

   Calculate the intermediate value using the known information (e.g plaintext) and all the possible sub-key.

	00	...	3D	...	FF
Sbox(C7,Key)	C6		2D		07
Sbox(2C,Key)	71		82		66
Sbox(D2,Key)	B5		DF		D8
Sbox(89,Key)	A7		8D		38

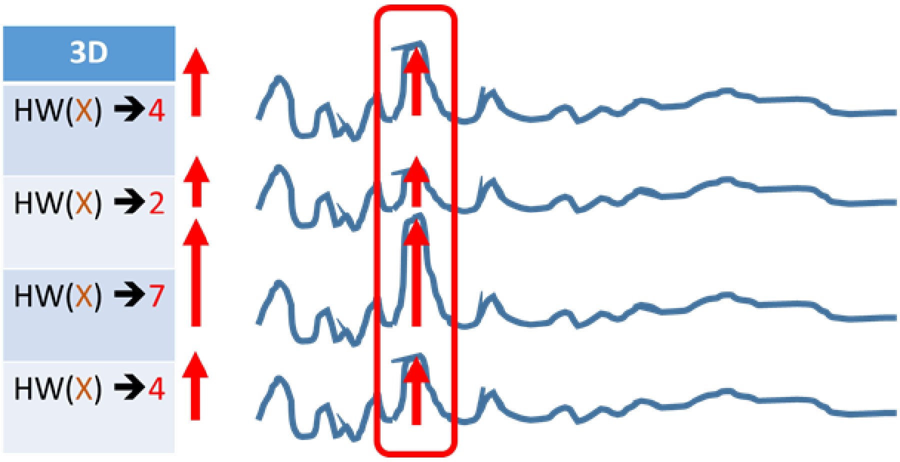
3. Mapping Intermediate Values to Power Consumption:

   Use the power model to map the hypothetical intermediate values to the hypothetical power consumption. In this program, **please use the Hamming Weight power model.**

	00	...	3D	...	FF
HW(Sbox(C7,K))	C6 →4		2D →4		07 →3
HW(Sbox(2C,K))	71 →4		82 →2		66 →4
HW(Sbox(D2,K))	B5 →5		DF →7		D8 →4
HW(Sbox(89,K))	A7 →5		8D →4		38 →3

4. Comparing the Hypothetical Power Consumption Value with the Power Traces:

   Calculate the Pearson correlation coefficient between the hypothetical power consumption and actual power consumption. Do this for every data point in the traces. If the highest value of correlation coefficient should reveal the correct sub-key used in the process, and also the timing where the intermediate value is being processed.



3. Input/Output Specification

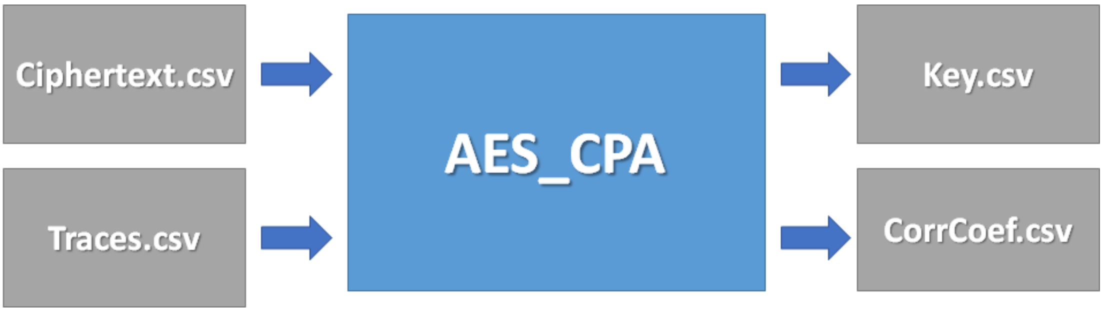
The program should take in 2 files:

1) Ciphertext.csv: the encrypted data

2) Traces.csv: the traces of the encryption process.

After the Correlation Power Analysis, the program should generate 3 output files:

1) Key.csv: include the last round key (CPA result) and encryption key (master key).
2) Result.csv: containing the correlation coefficient and leakage point information.



Ciphertext.csv

Traces.csv

AES_CPA

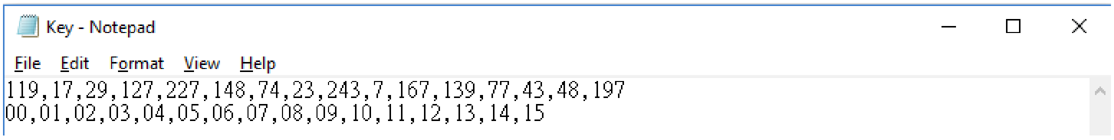
Key.csv

CorrCoef.csv

The Ciphertext.csv and Traces.csv format are the same as the last assignment.

Key.csv / Roundkey.csv format:

The first line last round key used in the encryption process, which is also the direct result of the correlation power analysis. The second line is the actual key used for the encryption, which can be calculated using the inverse of the key schedule and the last round key.



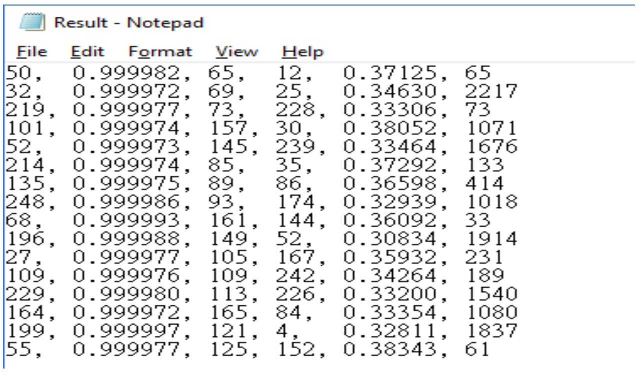
```
119,17,29,127,227,148,74,23,243,7,167,139,77,43,48,197
00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15
```

Result.csv format:

A 16-line .csv file, showing the result of the 16 bytes.

Please show the following information in each line:

<highest ranking key>, <highest correlation coefficient>, <Leakage point>, <2nd highest ranking key>, <2nd highest correlation coefficient>, <Leakage point>



```
50,   0.999982, 65,   12,   0.37125, 65
32,   0.999972, 69,   25,   0.34630, 2217
219,  0.999977, 73,   228,  0.33306, 73
101,  0.999974, 157,  30,   0.38052, 1071
52,   0.999973, 145,  239,  0.33464, 1676
214,  0.999974, 85,   35,   0.37292, 133
135,  0.999975, 89,   86,   0.36598, 414
248,  0.999986, 93,   174,  0.32939, 1018
68,   0.999993, 161,  144,  0.36092, 33
196,  0.999988, 149,  52,   0.30834, 1914
27,   0.999977, 105,  167,  0.35932, 231
109,  0.999976, 109,  242,  0.34264, 189
229,  0.999980, 113,  226,  0.33200, 1540
164,  0.999972, 165,  84,   0.33354, 1080
199,  0.999997, 121,  4,    0.32811, 1837
55,   0.999977, 125,  152,  0.38343, 61
```

4. Command-line Parameters

In order to test your program, you are asked to add the following command-line parameters to your program:

[executable file name] [Traces.csv] [Ciphertext.csv]

```
./AES_CPA Traces.csv Ciphertext.csv
```

5. Language

Language: C/C++, or python

6. Submission

Please submit a .zip file named **HW2_${StudentID}.zip** (e.g. HW2_r06921000.zip) including the following materials:

1) source code
2) a text README file named **README.txt**
3) a report named **report.pdf**

```
HW2_r06921000
|-- README.txt
|-- report.pdf
|-- src/
```

README file:

Please provide the following information in your README file:

1. Student ID
2. Compiler / Interpreter version
3. How to build the program
4. How to run the program

Report:

1. Which intermediate value do you choose in your CPA program, and how do you calculate it? (e.g. S-box(plaintext $\oplus$ key))
2. How to recover the encryption key from the last round key?
3. Do you use any techniques to speed up your analysis? Please describe it.

Late submission penalty: 20% per day.