Attacks on Implementations of Secure Systems

Chen Doytshman, Igor Alikin, Tomer Wasserman

Assignment 2 Report - Correlation Power Analysis

1 Highest Difficulty Achieved

The highest difficulty we achieved is 2 using 10k traces.

Dear grader: We use `tqdm` for showing progress bar for downloading the traces. If you don't want to use it just comment-out "`from tqdm import tqdm`" and set the flag `USE_TQDM` to `False`.

2 Sample Executions

Below are sample executions that were made using our script. The printouts were made to `stderr`, while only the final printout was made to `stdout`.

Command Line

```
$ python3 ./ex02_M2.py chendoy_traces
    Downloading traces...
    100%|[redacted]| 10000/10000 [09:27<00:00,  6.17it/s]
    Starting power analysis...
    Verifying key...
    chendoy,d60aff3115a72dfc887aa468e2fd2f5c,2
    Total time (CPA): 210.26 seconds
```

Command Line

```
$ python3 ./ex02_M2.py alikin_traces
    Downloading traces...
    100%|[redacted]| 10000/10000 [09:27<00:00,  6.17it/s]
    Starting power analysis...
    Verifying key...
    alikin,0633e729843273b2668a5ff86cff95d4,2
    Total time (CPA): 257.24 seconds
```

Command Line

```
$ python3 ./ex02_M2.py tomerwa_traces
    Downloading traces...
    100%|[redacted]| 10000/10000 [09:27<00:00,  6.17it/s]
    Starting power analysis...
    Verifying key...
    tomerwa,e882be86b35445af3ad66c9612ab02c0,2
    Total time (CPA): 253.64 seconds
```

3 Analysis

3.1 Attack Performance

Finding the key of 128-bit AES cryptosystem with brute-force would take maximun number of 2^{128} requests from the server in the worst case, 2^{127} in average. With our approach using CPA it takes only $10,000$ (to collects the traces, for the highest difficulty) and one extra request to verify the key.

3.2 Finding the Correct Timestamp

The positions in the power trace that were the correct ones to use, are the ones who have the key involved. In order to find them we guessed some key byte, and for that guess we calculated the hypothetical power consumption. Then, we looked for some point in time, which is called the correct time, where the power consumption would indeed be correlated (close to +-1) with this hypothetical power consumption that we guessed and all of the other times will not be correlated (since the DUT is not only processing this byte of the key, but it is processing other bytes of the key, other things, and more).

3.3 Number of Traces To Use

For our attack we have used 10,000 power traces. We could have used less, and we actually have managed to use less yet to guess the correct key, as shown in the graph below (only 6,000 traces will do). For this submission we decided to fix the power traces at 10k for redundancy, trying to optimize the algorithm as much as we can.

Below is a plot that describe the number of correct bytes guessed as a function of the number of traces.

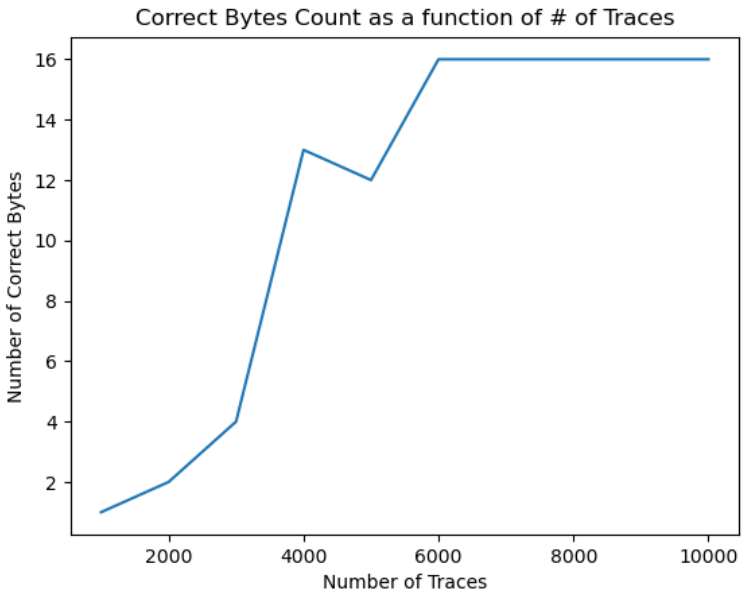

Figure 1: The number of bytes guessed correctly as a function of the number of traces used.