Guided Analysis of WS3

DPA Attacks with Windowing on AES Encryptions with Dummy Operations

15 April 2010, Version 1.0

IMPLEMENTATION ATTACKS

Thomas Popp

IAIK – Graz University of Technology
Thomas.Popp@iaik.tugraz.at
www.iaik.tugraz.at

Topics

What do you need?

- Matlab or Octave (software for numerical computations)
- Scripts and working data

Basic hints for Matlab/Octave

Getting started

Tasks 1 – 6

What Do You Need?

Matlab …

- From www.mathworks.com
- Recommended, but no freeware
- Exercise tested with Matlab R2009b 32-bit
- Full-featured GUI

… or GNU Octave

- From www.octave.org
- Freeware (GPL), available for Windows/Linux/Mac OS X/Sun Solaris, basically compatible to Matlab
- Exercise tested with Octave 3.2 32-bit

What Do You Need?

- Only command-line interface
  - Uses gnuplot for plotting diagrams
  - Windows Octave includes Notepad++, which has m-file syntax highlighting
  - On Windows: activate “quick edit mode” of the shell you are using
    - Click on the C:\ icon in the top left of the shell window and select Properties (if you want to set this mode permanently also select Defaults and repeat step 2)
    - Place a check in Quick Edit Mode and click OK
      - Highlight: keep left mouse button pressed
      - Copy: press right mouse button
      - Paste: press right mouse button again

What Do You Need?

Scripts and working data

- “Demo DPA & Windowing Script” (demo_dpa_and_windowing.m)
- “Matlab/Octave multiple plot function” (show_plots.m/octave_plot2.m)
- “Workspace 3 Data” (WS3.zip)
- From www.dpabook.org/onlinematerial/matlabscripts

[optional] The DPA Book

- www.dpabook.org
- Basic DPA is explained in Sections 6.1 and 6.2
- Windowing (integration) is explained in Section 8.2
- AES is explained in Appendix B

Basic Hints for Matlab/Octave

- Everything is an array!
- Array indices start with 1!
- “> help [command]” and “> doc [command]” gives advice
- Command “more” allows to page output to the command window
  - Especially useful for Octave
  - Use “> help more” for details
- Lines ending with “;” → result of variable assignment is NOT printed
- “> pwd” shows current working directory
- Change working directory with GUI buttons (Matlab) or with command: “> cd [drive letter]:\[directory1]\[directory2]\...”

Getting Started

- Unzip WS3.zip to a directory of your choice and also put the three m-scripts there
- Start Matlab/Octave
- Change to the directory where you put the scripts and working data
- Open the script demo_dpa_and_windowing.m in an editor

Task 1

1.1 Get familiar with the first code lines until comment “TASK 2”

- Use the help system if commands are unclear
- Determine what traces are selected for analysis

1.2 Execute the code block until comment “TASK 2”

- Understand roughly what has happened

Task 1 – Answers

For 1.1

- The selected power traces are those where the AES was executed without randomly inserted of dummy operations. Variable name = traces_noDummy.

For 1.2

- Some variables are loaded from WS3.mat
  - HW: vector to calculate the Hamming weight of a byte
  - SubBytes: vector to calculate the AES operation SubBytes of a byte
  - aes_plaintexts, traces_noDummy, traces_withDummy
- Some variables are set
  - samples (number of analyzed traces)
  - analyzed_traces, byte_to_attack, delta

Task 1 – Answers

- Plaintext bytes that correspond to the attacked key byte are prepared

  → column vector D
- Selected traces are prepared

  → matrix “traces”
  - Each row is a power trace
- Row vector K containing all possible key guesses for one byte is prepared

Task 2

2.1 Plot power traces 501 to 505 and zoom-in

- “> help plot”

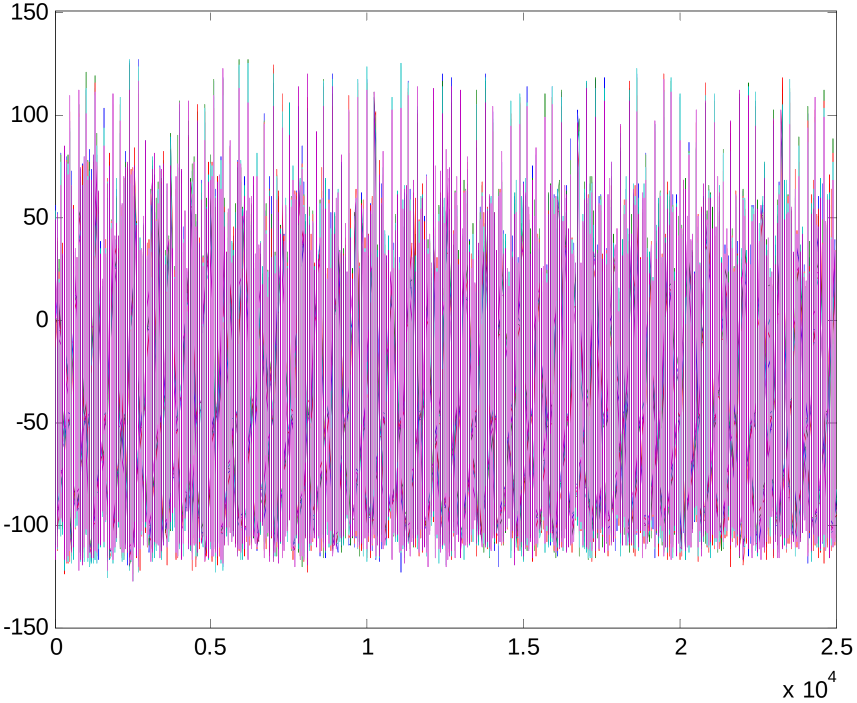
- Note that “plot” prints the COLUMS of a matrix, but the available traces are arranged in ROWS!
  - Transpose a matrix with ‘ (single-quote symbol)

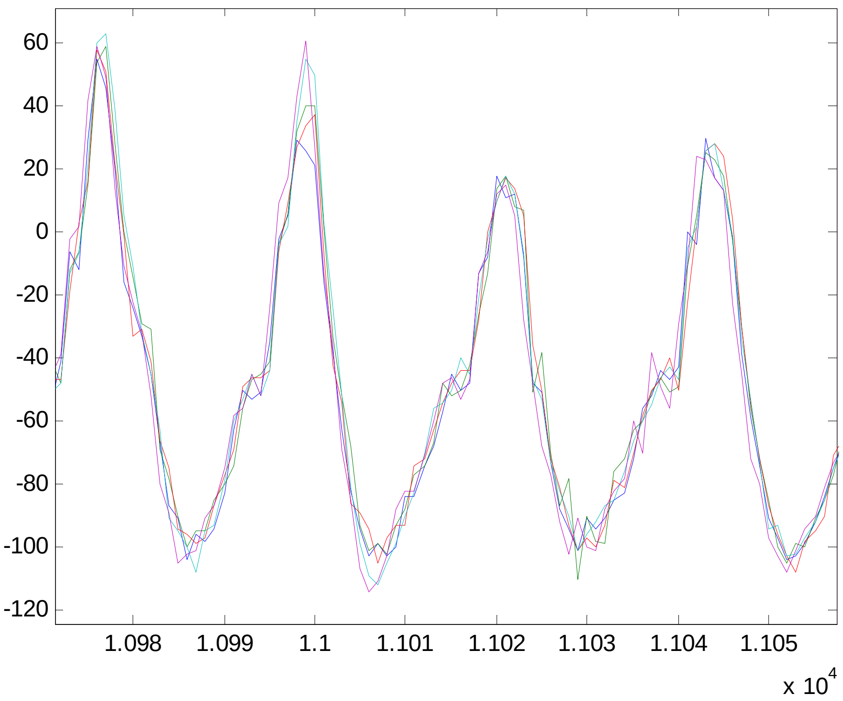
- Also note that the trace values are of data type “int8”
  - Convert traces to “double” before printing!
  - “> double([variable])”

Task 2 – Answers

For 2.1

- “> plot(double(traces(501:505, :)'))”





Task 3

3.1 Try to understand what the intermediate value matrix V and power consumption matrix H contain and how the calculation is done

- “> help repmat”

Task 3 – Answers

For 3.1

- Matrix V
  - Size: samples x key guesses
  - Contains output values of the first AES S-box after initial AddRoundKey for input bytes D (rows) and all possible key bytes K (columns)
- Matrix H
  - Same size as V
  - Contains the Hamming weight of each byte in V
  - We assume our device leaks the HW of a processed data value

Task 4

4.1 Try to understand what happens in the next lines of code until comment “TASK 6”

4.2 Execute the code block between comments “TASK 2” and “TASK 6”

- This may take a while

4.3 View the resulting correlation traces with the show_plots.m or octave_plot2.m function

- What seems to be the correct key byte 1?

Task 4 – Answers

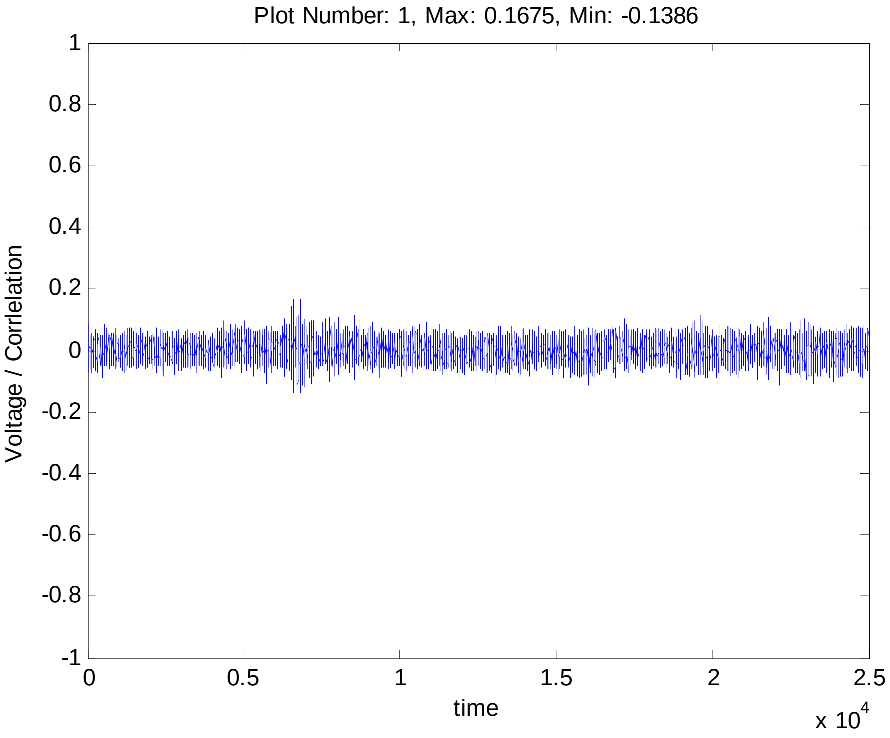
For 4.1

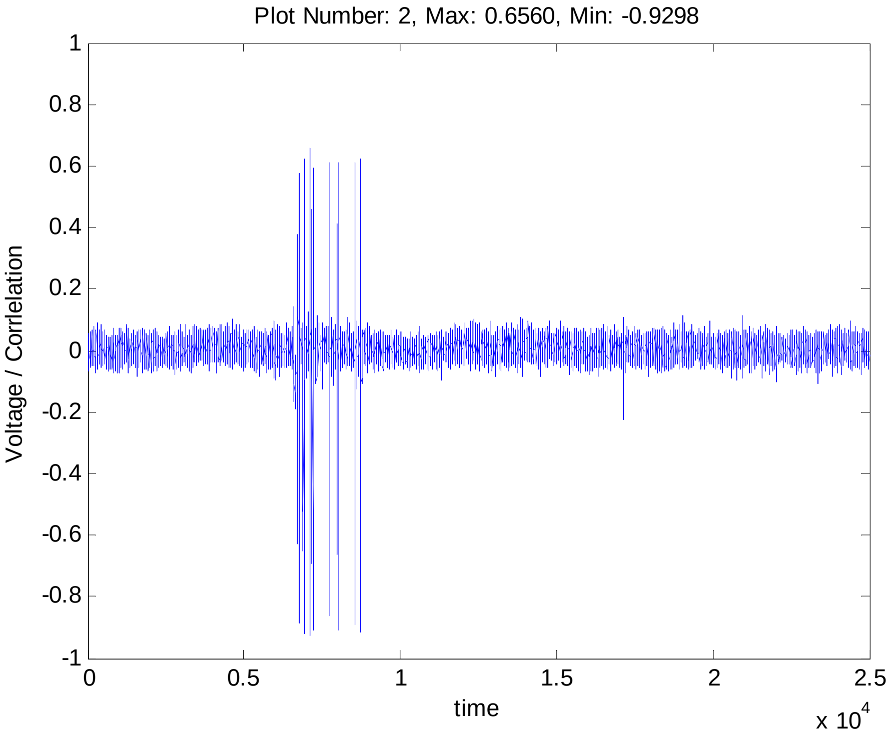
- Matrix R for correlation results is initialized
  - Size: key guesses x trace length
  - Each row vector of R will contain the correlation between the estimated (power model = HW) power consumption of the first SubBytes operation for a specific key guess (= row number – 1) and the measured power consumption at specific points in time (= column number)
- The following for-loop calculates R element-wise with the “corrcoef” function

For 4.3

- Only plot 2 shows significant correlation peaks
  → key byte 1 is most likely 1 (true: the complete AES key is 1, 2, 3, …, 14, 15, 16)

Task 4 – Answers





Task 5

5.1 Change variable “analyzed_traces” to the traces with dummy cycles and re-run complete script until comment “TASK 6”

- Previously, save the R-trace for the correct key and no dummy operations with the command:
  - “> R_noDummy = R(2, :);”
- This may again take a while

5.2 View the resulting correlation traces with the show_plots.m or octave_plot2.m function

- What happened with the correlation values for the correct key?
- Detail: randomly, 0 or 1 dummy operations have been inserted in this AES execution

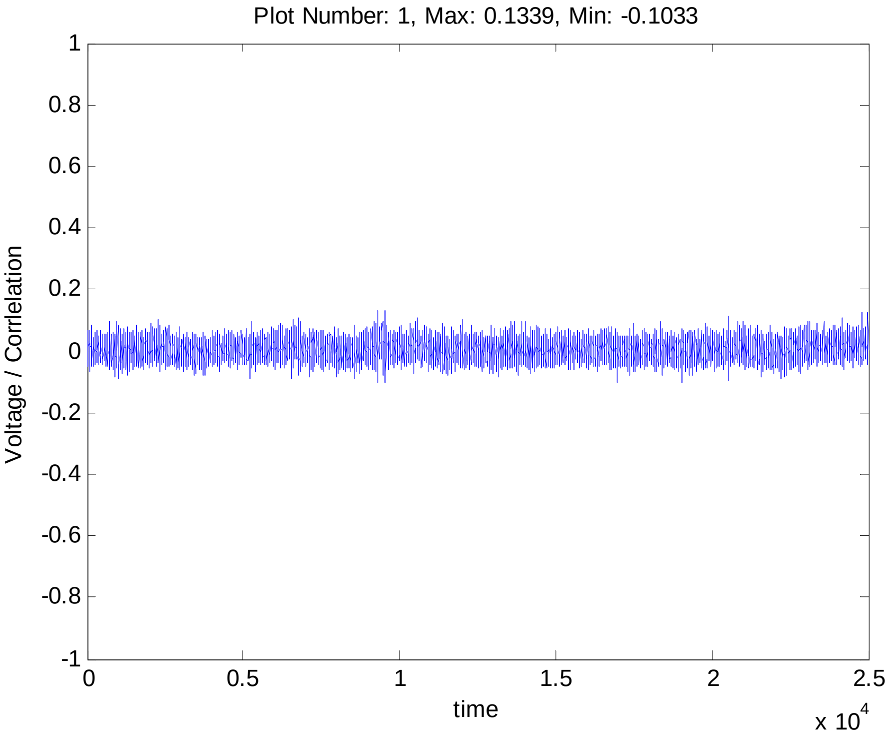
Task 5 – Answers

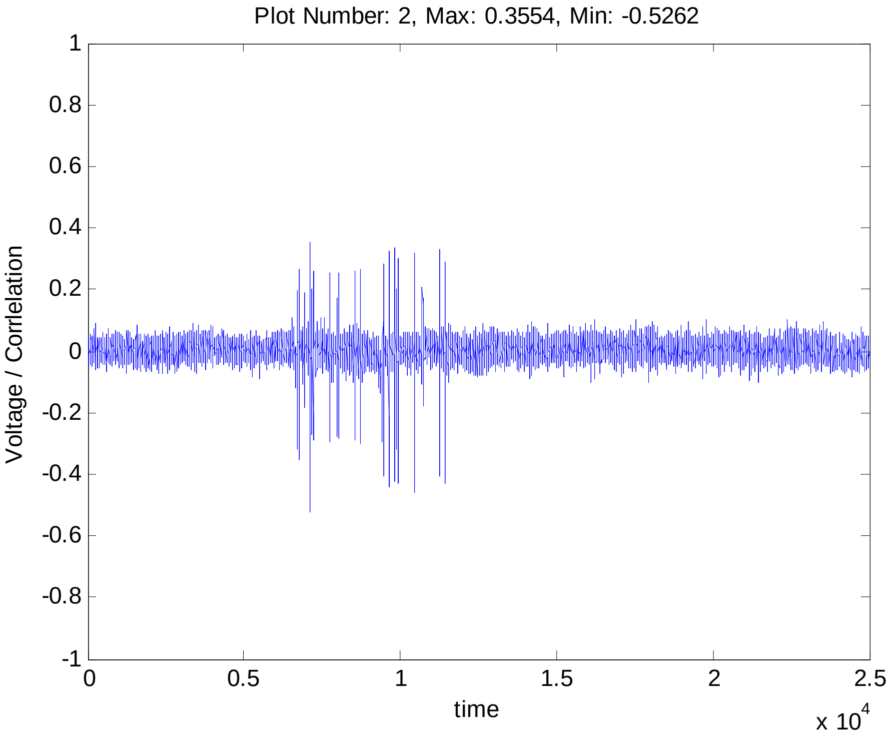
For 5.2

- The correlation peaks for the correct key value 1 are significantly smaller. This is caused by the misalignment of the power traces due to the random insertion of a dummy operation.

- The correlation peaks are approximately halved. This corresponds nicely to theory: Due to the randomly inserted dummy operation, the attacked AES operation can occur at 2 positions in time → the correlation value is approx. reduced by this factor 2 → an attacker needs approx. 4 (=2^2) times more traces to get the same unambiguous result as in case of the AES execution without dummy operations.

Task 5 – Answers





Task 6

6.1 Try to understand how the traces are preprocessed with windowing

- What would an attacker have to do who doesn’t know the value “delta” (= the usual case)

6.2 Perform the attack with the preprocessed traces

- Execute the code block after comment “TASK 6”
- This may again take a while

6.3 View the resulting correlation traces with the show_plots.m or octave_plot2.m function

- What happened with the correlation values for the correct key this time?

Task 6 – Answers

For 6.1

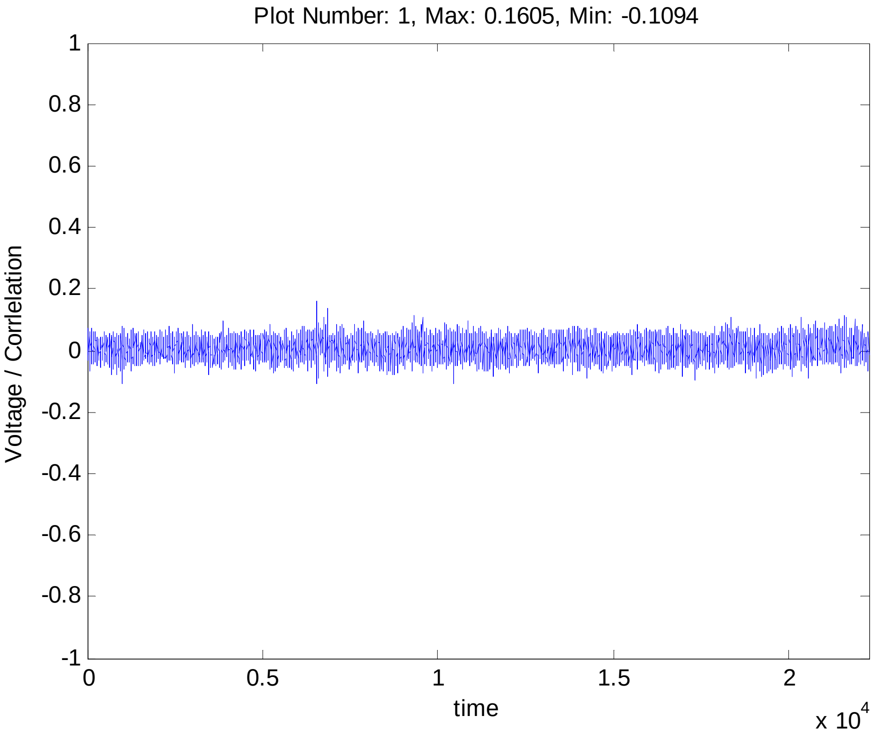
- A copy of the traces is shifted by "– delta" and added to the original traces ("delta" columns at the end of the original traces matrix are cut off)
- An attacker who doesn't know the value "delta" (= distance between the two positions where the attacked operation can occur) has to search for the correct value (it is usually a multiple of the clock period).
- Furthermore, an attacker would have to guess the number of differently shifted trace copies that have to be summed up. It is usually unknown to him how much the number of inserted dummy operations can vary (0 or 1 dummy operations as in our scenario is a rather artificial case).

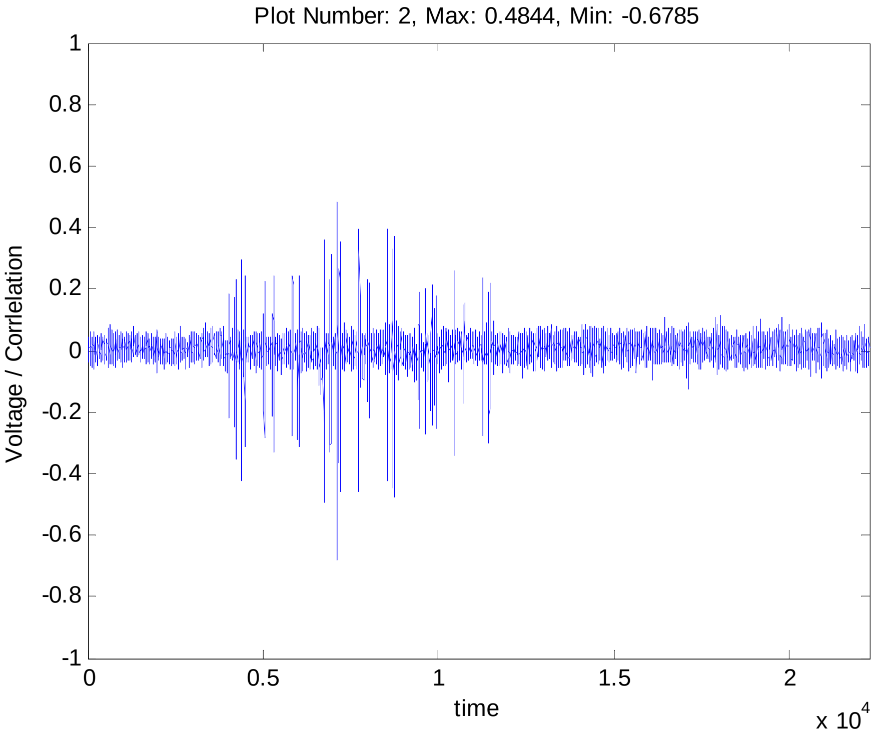
Task 6 – Answers

For 6.3

- The correlation peaks for the correct key value 1 are higher than without windowing. Successful windowing eliminates the misalignment of the power traces due to the random insertion of a dummy operation but the summing of power values increases the noise. Fortunately, the negative effect of the increased noise is smaller than the negative effect of the misalignment.
- The correlation peaks are approximately reduced by the factor sqrt(2). This again corresponds nicely to theory: Due to the randomly inserted dummy operation, the attacked AES operation can occur at 2 positions in time → the correlation value is approx. reduced by the factor sqrt(2) because windowing of the two positions is used → an attacker needs approx. 2 (=sqrt(2)^2) times more traces to get the same unambiguous result as in case of the AES execution without dummy operations.

Task 6 – Answers


Plot Number: 1, Max: 0.1605, Min: -0.1094
Voltage / Corrlelation
time
x 10^4


Plot Number: 2, Max: 0.4844, Min: -0.6785
Voltage / Corrlelation
time
x 10^4