Simulated versus Experimental

Differential Power Analysis of an AES Software Implementation on ARM

Ruggero Susella

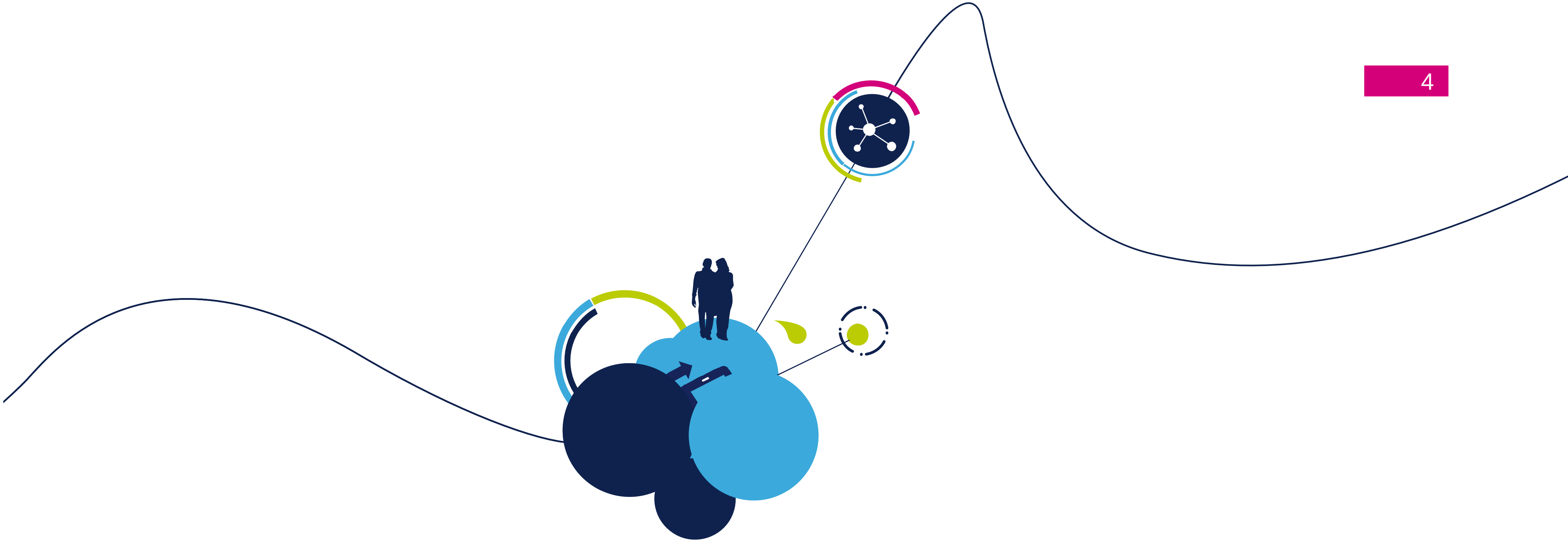
life.augmented

Motivation

- Be able to predict the possibility to attack a software implementation
  - Without needing a real hardware to run it
- Use a simulator and a very simple estimation for the power consumption
  - And see if it reflects reality
- Final goal is to gain confidence that countermeasures tested in simulation will work on the real device

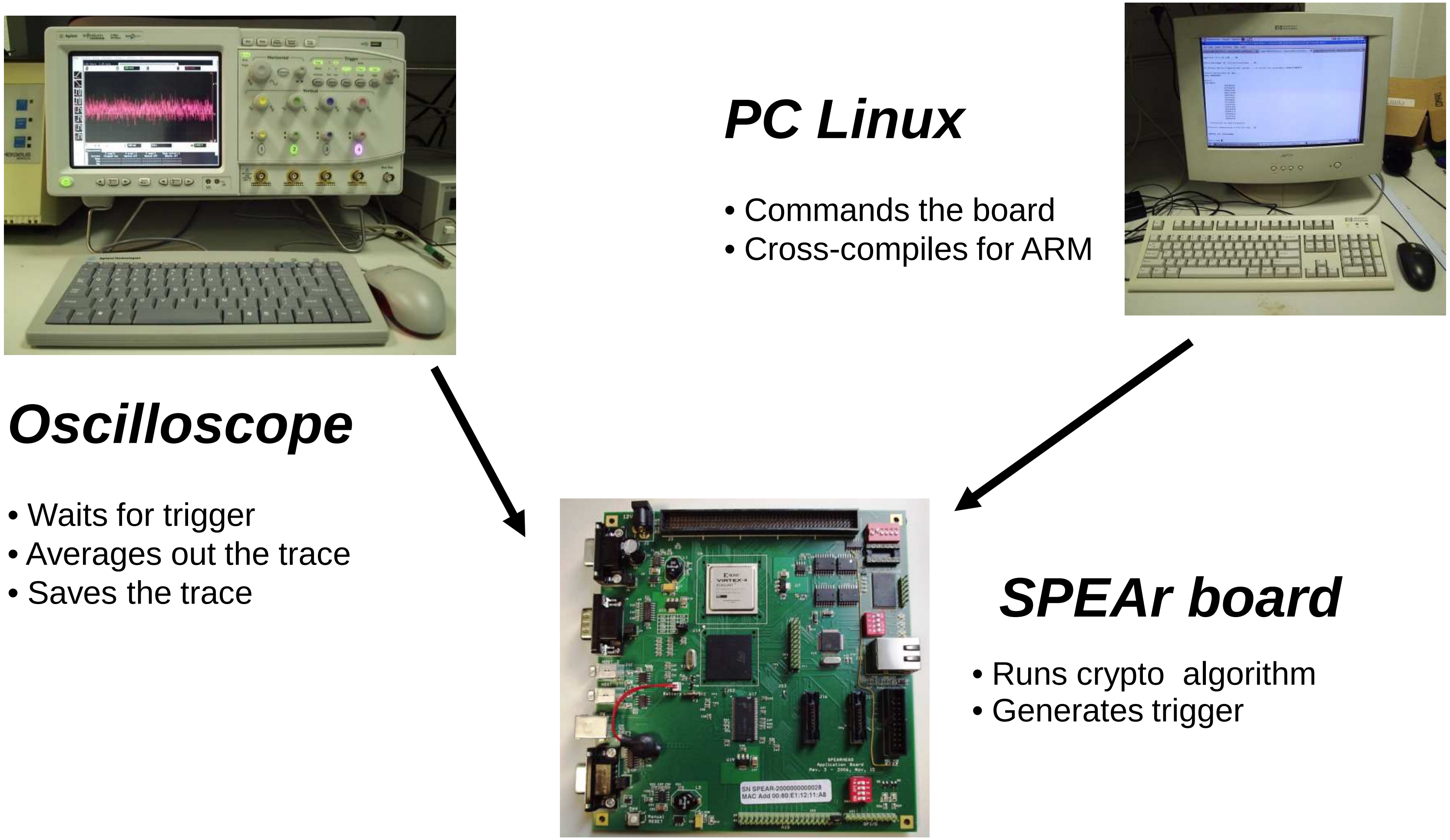
Attacked Algorithm

- C implementation of AES taken from OpenSSL
  - Big Tables (4 T-Tables)
  - Performing Sbox + ShiftRow + Mixcolumns
  - Fully unrolled
  - 9 equal rounds
  - 1, final, different round
- Crosscompiled with gcc for ARM926
  - Disabled all optimizations



Workbench for Experimental

Workbench



PC Linux

- Commands the board
- Cross-compiles for ARM

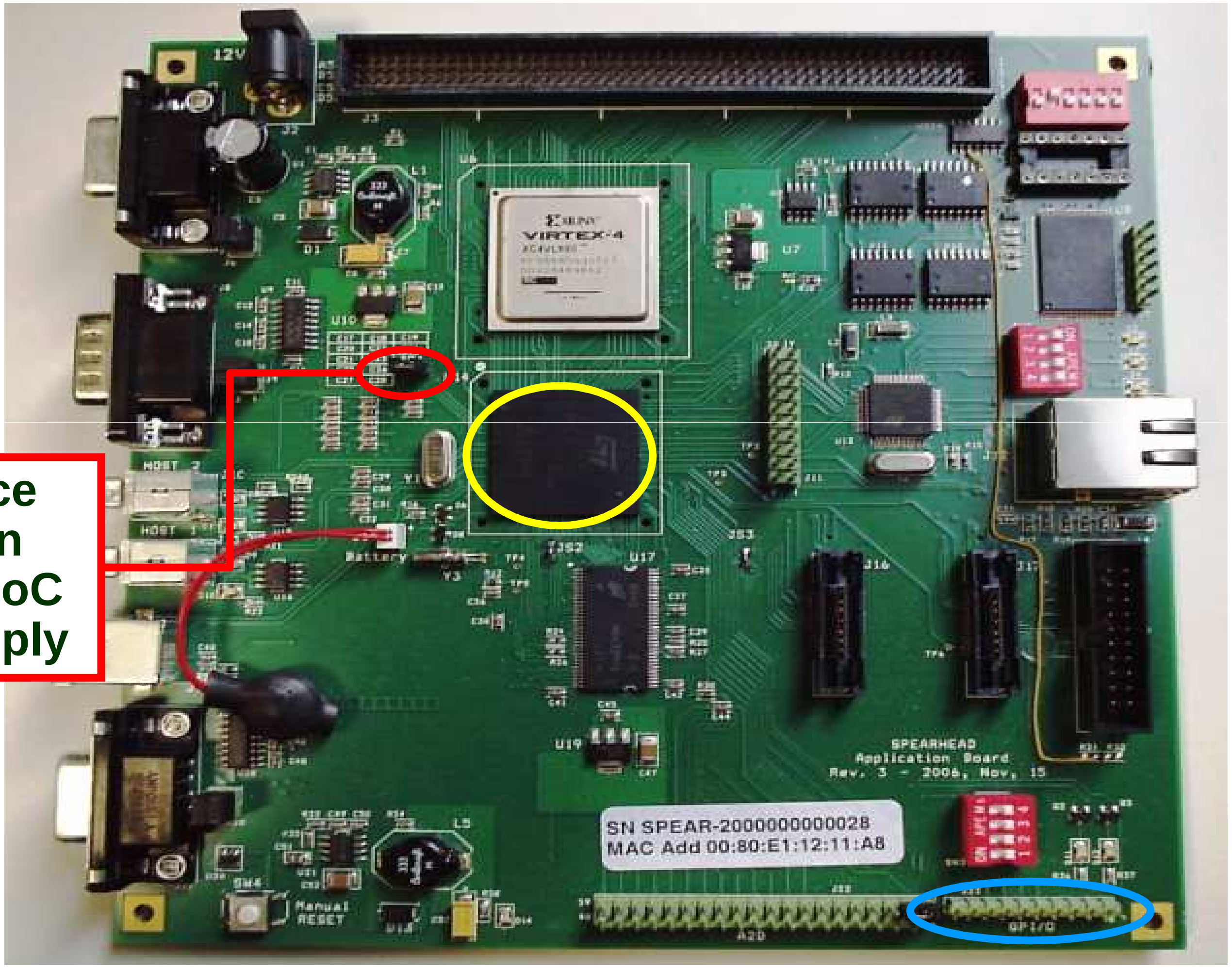
Oscilloscope

- Waits for trigger
- Averages out the trace
- Saves the trace

SPEAr board

- Runs crypto algorithm
- Generates trigger

SPEAr board



Resistance applied in series to SoC Power Supply

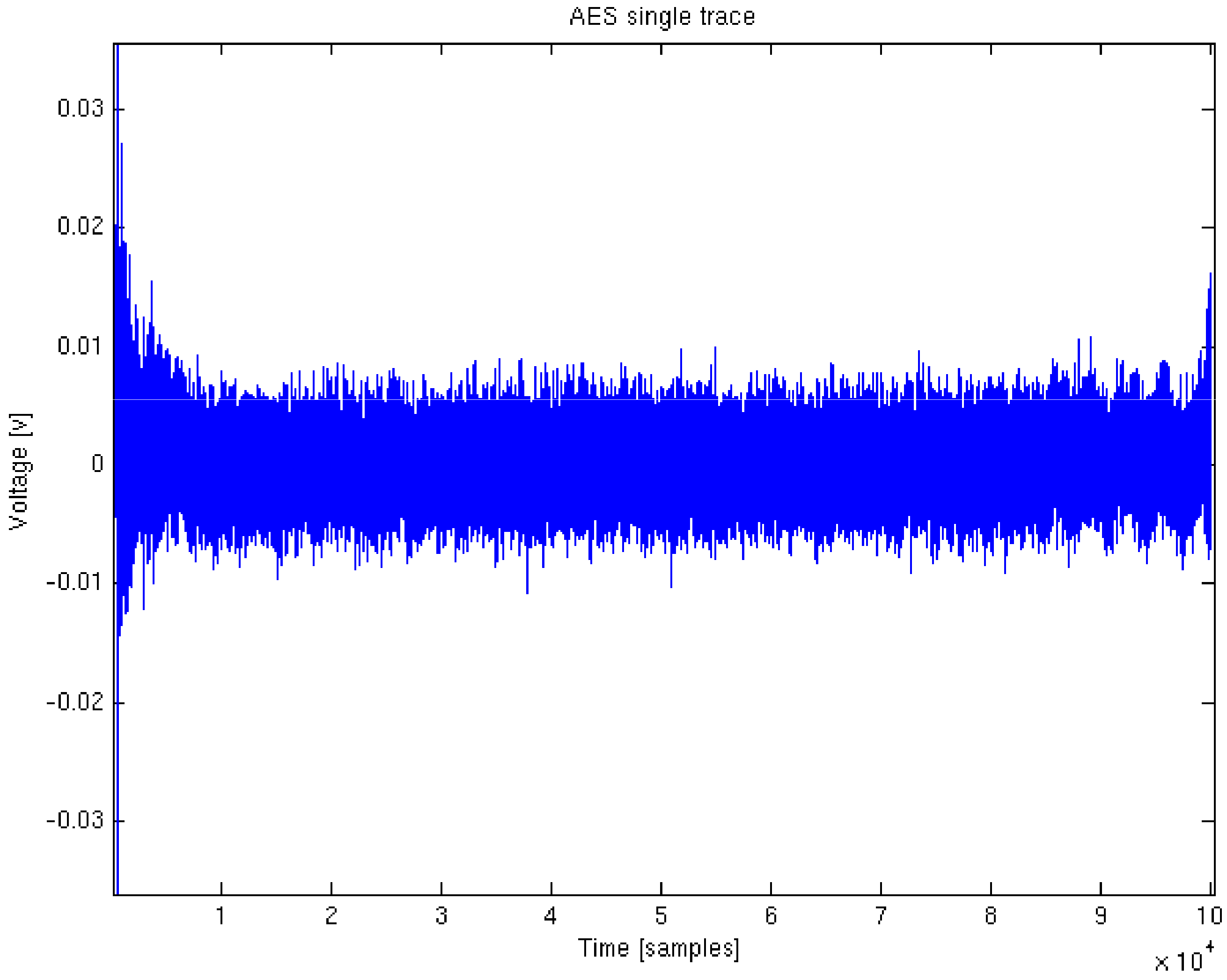
Oscilloscope

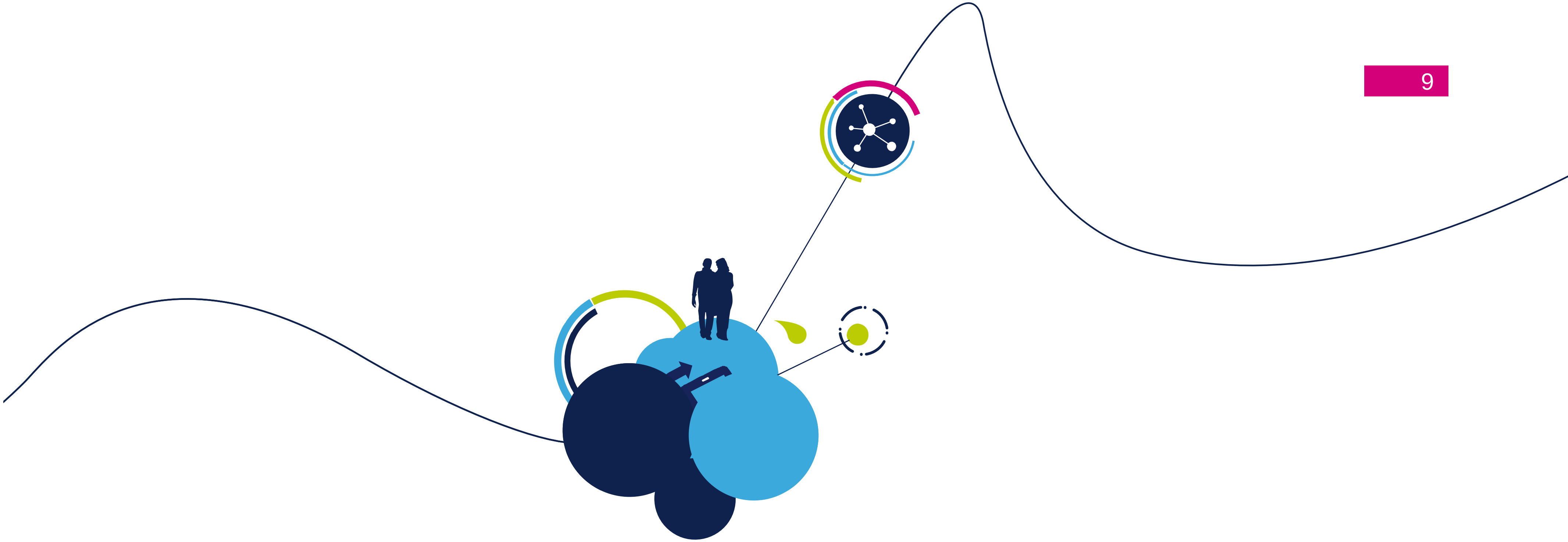
- Agilent Infiniium
- Features:
  - Windows XP
  - Max 40 Gsa/s
  - Max 2M samples
  - 4 Channels

- Differential Probe
  - Voltage difference measurement on a resistor

- Simple probe
  - Trigger detection

Single Power Trace





Workbench for Simulation

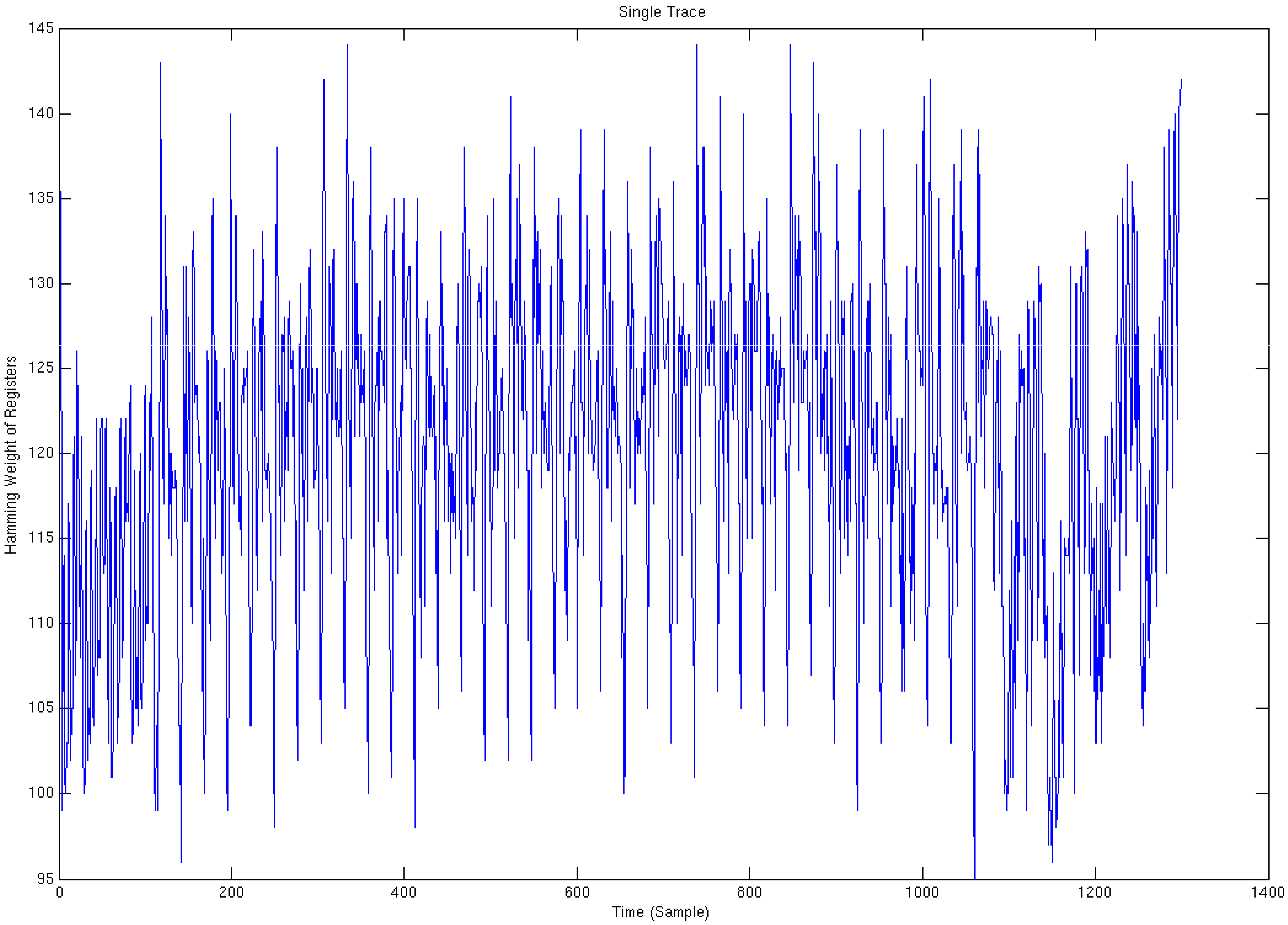
Simulator

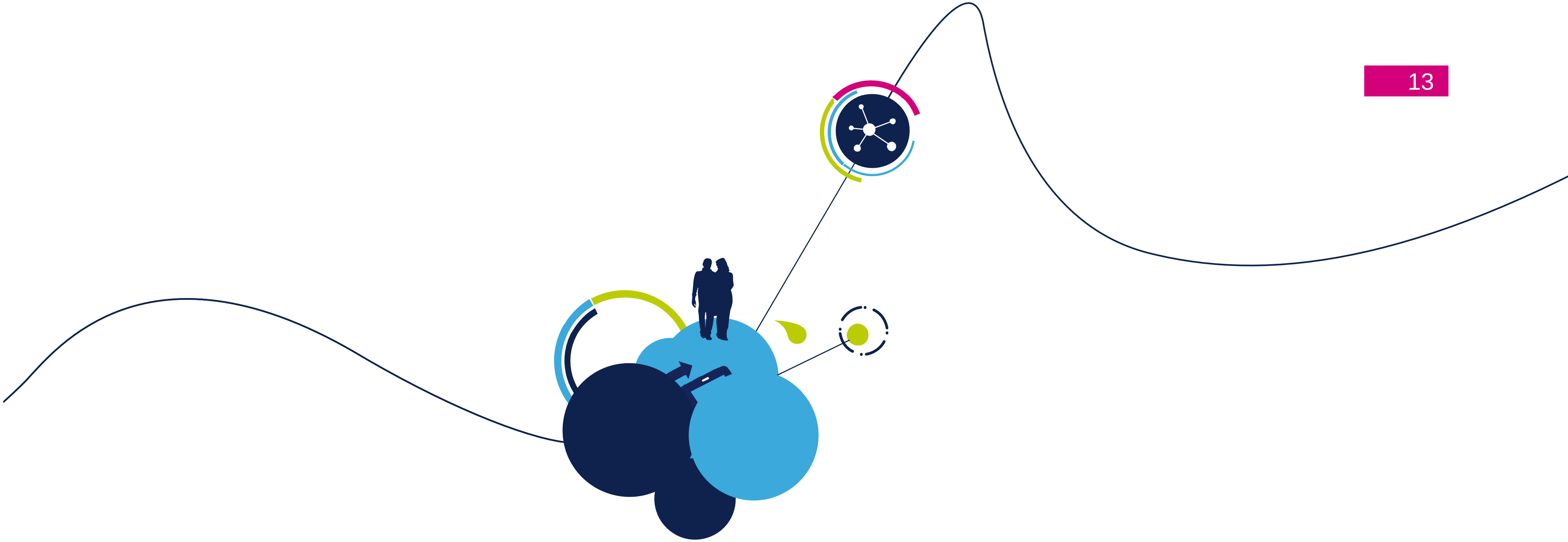
- Execution is simulated in a software environment
  - At assembler level
- Simulator supports ARMv5 instructions
  - No specific knowledge of the hardware is required
- Execution results in a txt file
- Each row contains the value of all registers after the execution of a single line of asm code

Post processing & Final Trace

- A post processing replaces each row with its Hamming Weight
  - We wanted to test the simplest possible leakage model
  - With more information about the hardware better models are possible
- Each simulated traces consists in 1299 HW values
  - One for each asm line executed
  - Each value can vary between 0 and 512 (16 registers of 32 bit)

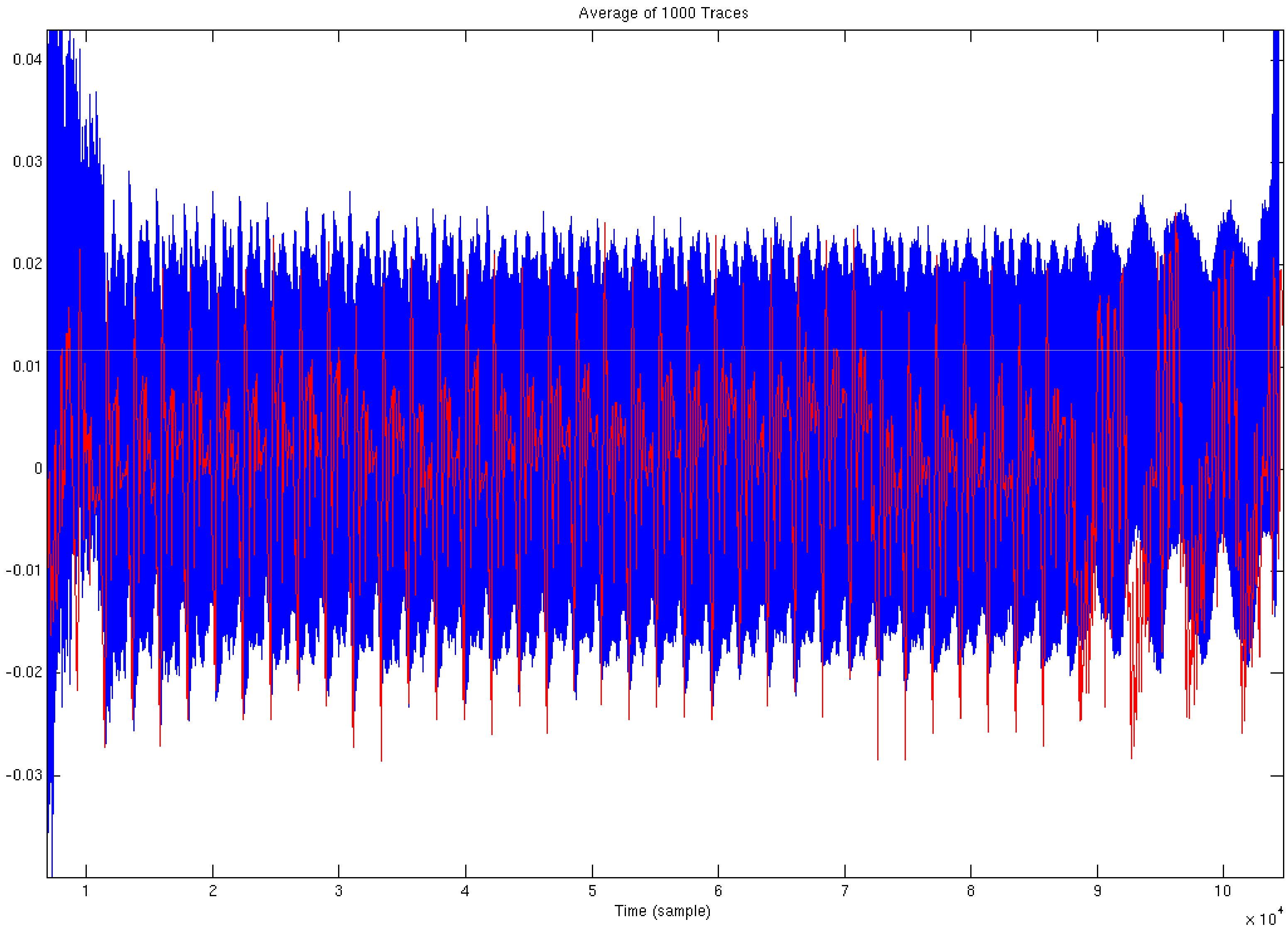
Single Simulated Trace



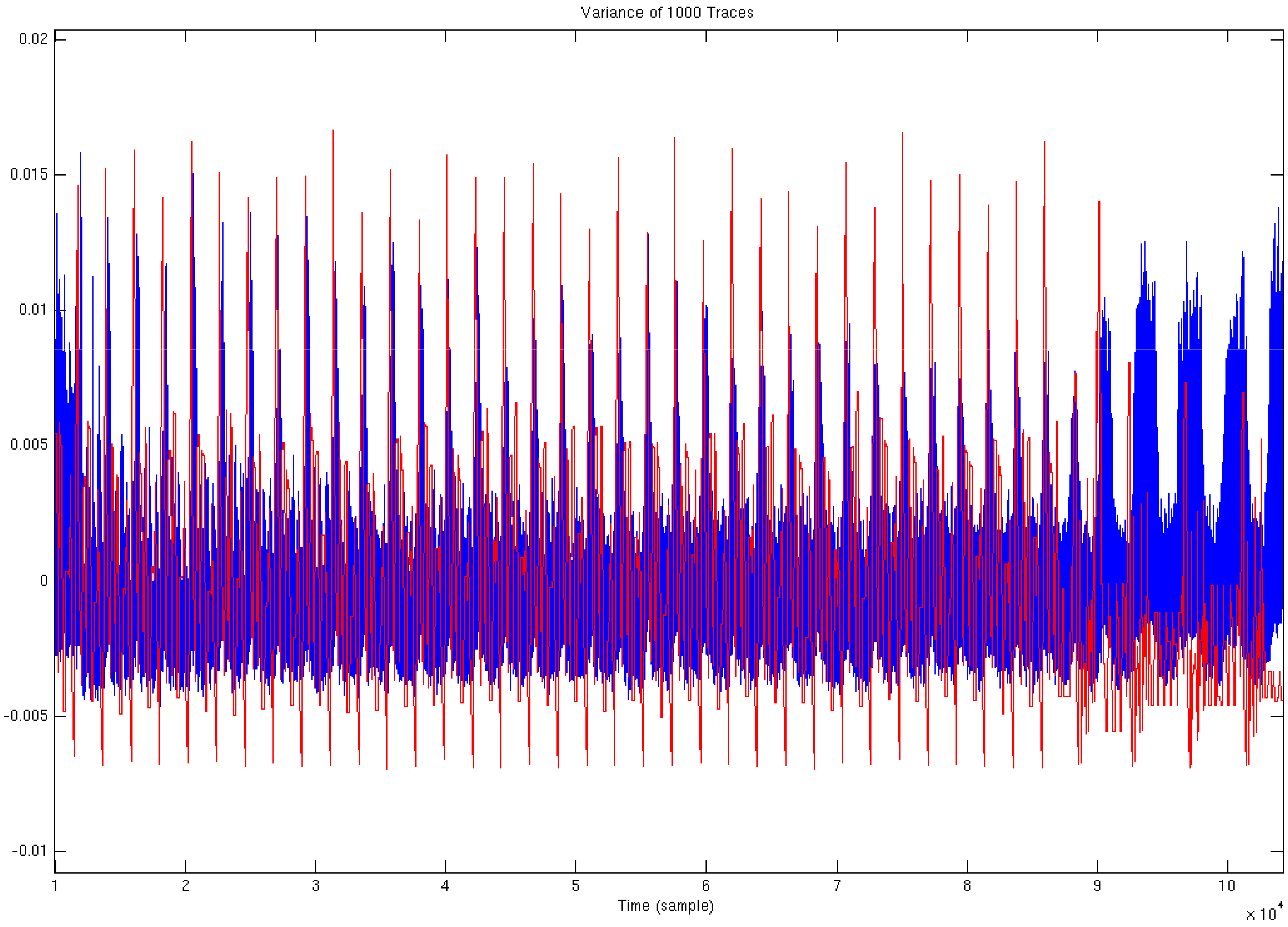


Results

Mean of 1000 Traces

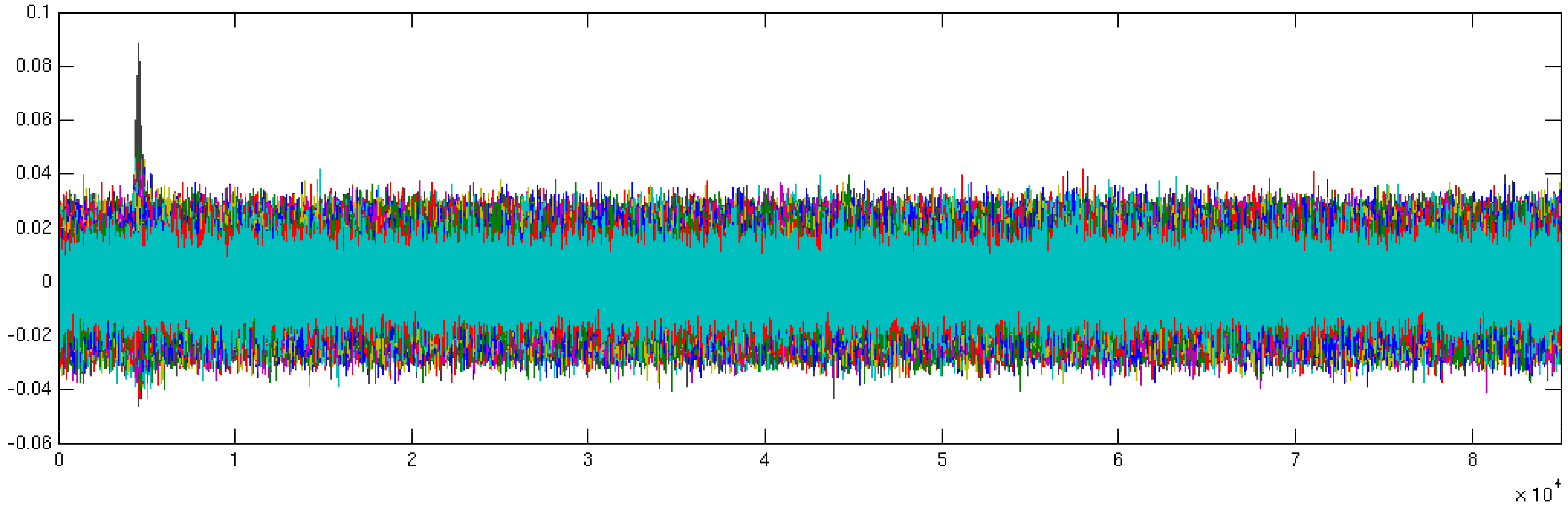


Variance of 1000 Traces

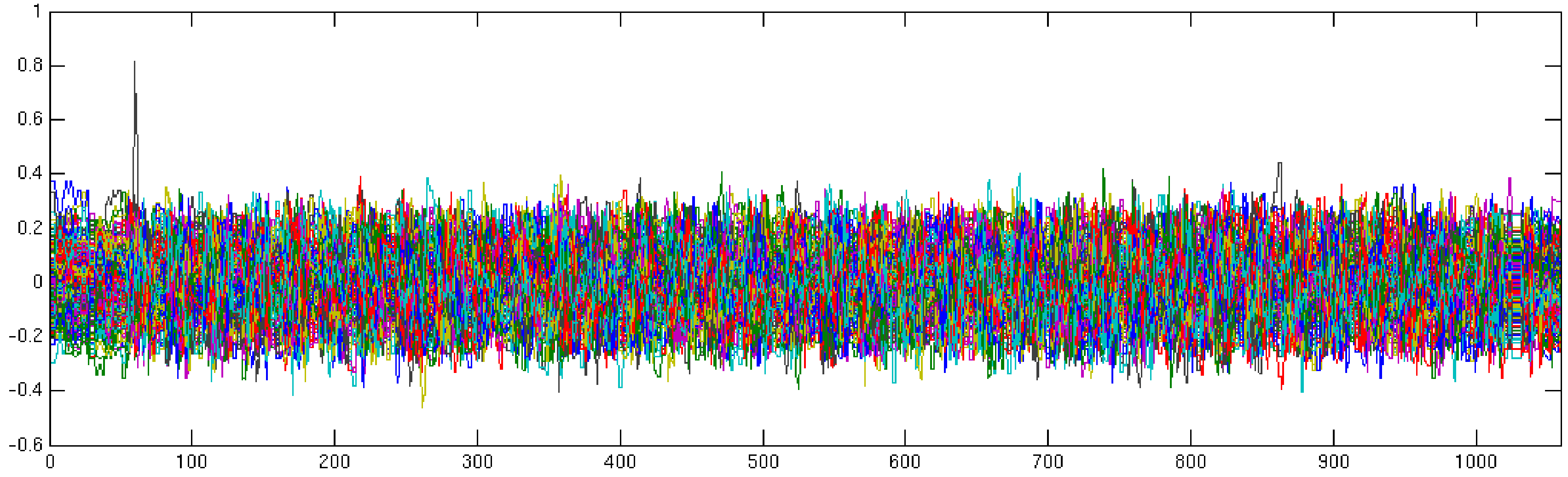


First Round Attack (1/3)

EXP



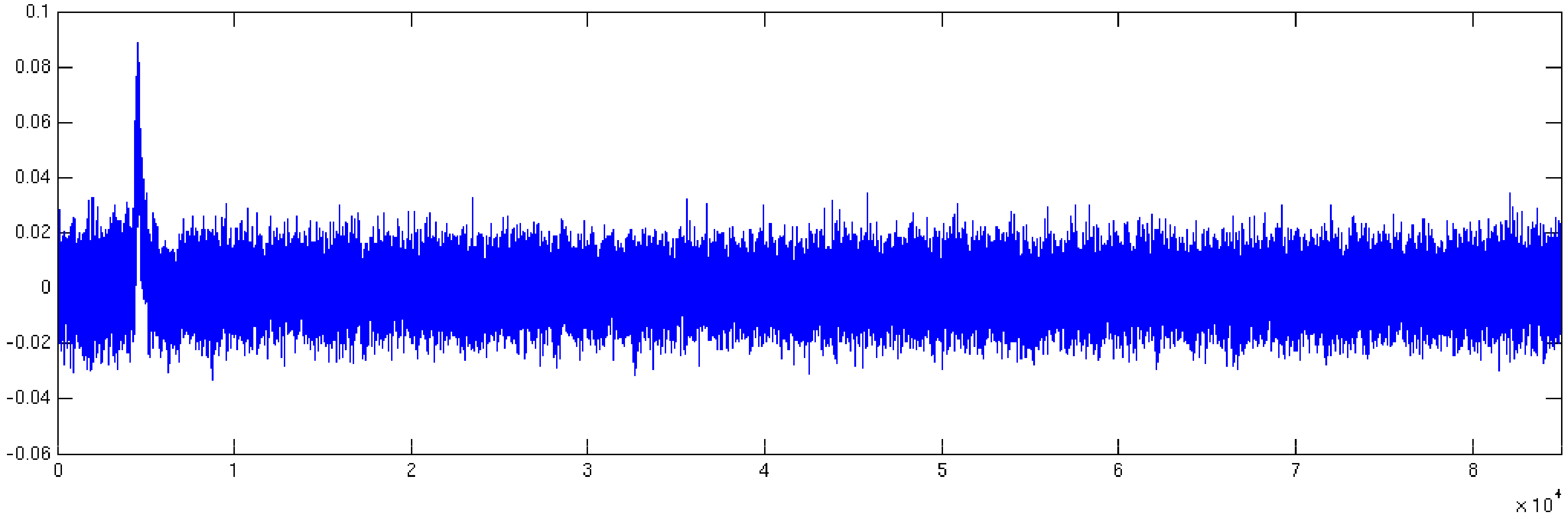
SIM



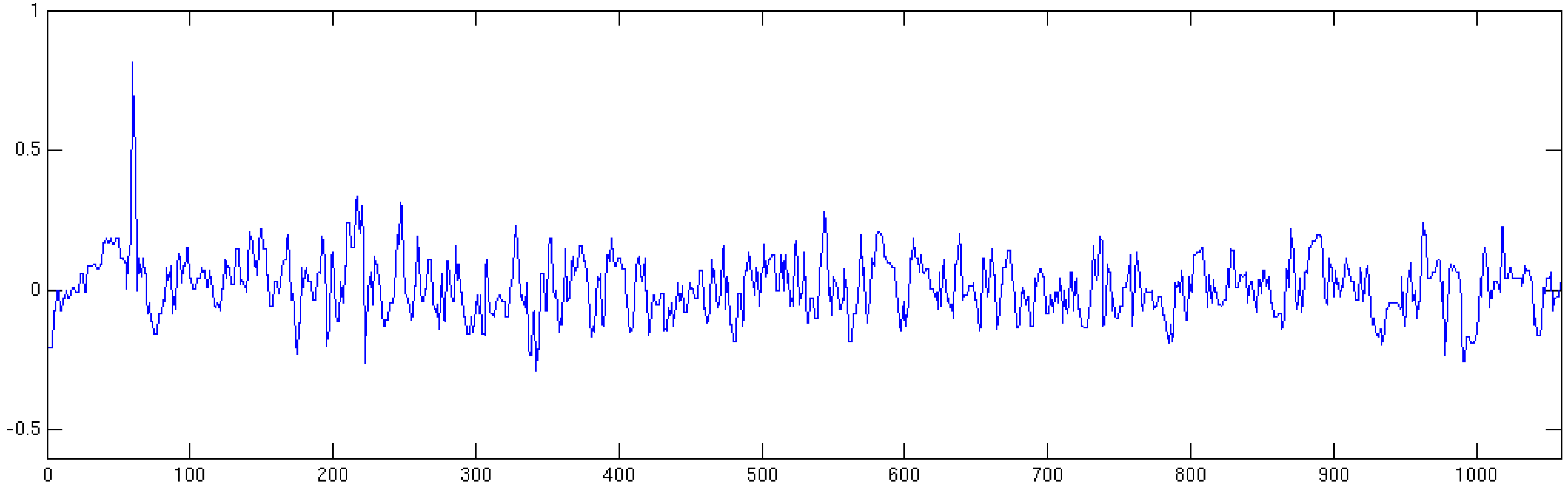
life.augmented

First Round Attack (2/3)

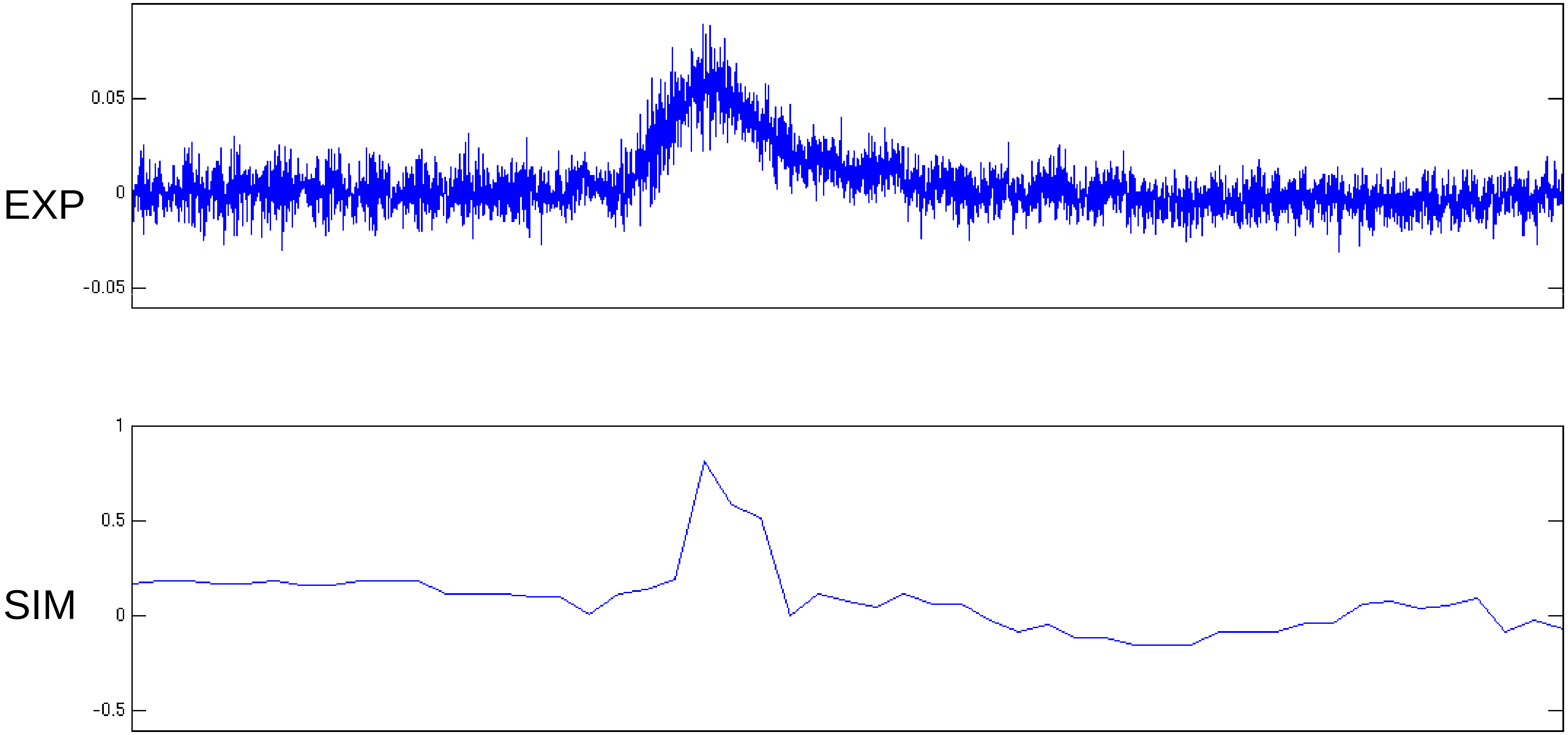
EXP



SIM

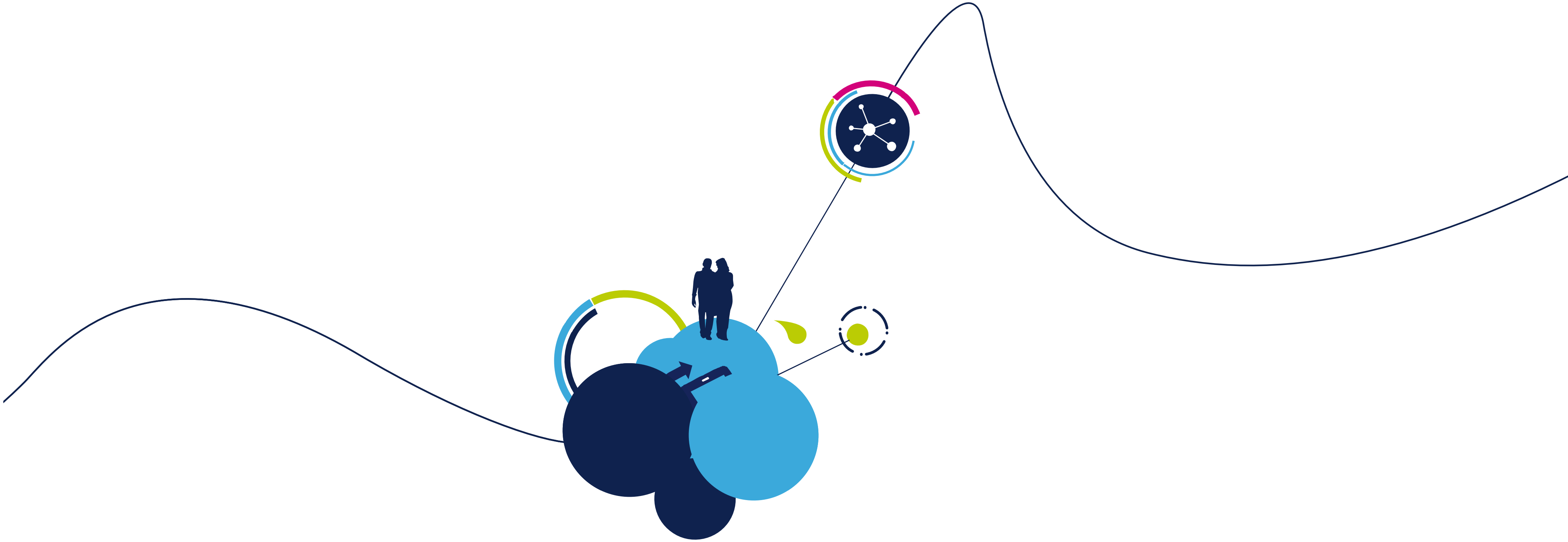


First Round Attack (3/3)



Conclusions

- In our setup 100 simulated traces provides comparable result as 16000 experimental traces
- Traces have common behavior
  - Mean
  - Variance
  - Attack's peak location and shape
- Hamming Weight of all registers is a good approximation of power consumption



Thank you! Questions ?

life.augmented