Power Analysis Part II

Cryptographic Hardware for Embedded Systems ECE 3170

Fall 2026

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

©Georgia Institute of Technology, 2018-2026

Reading

- Please read Appendix A of *Power Analysis Attacks: Revealing the Secrets of Smart Cards* by Mangard et al., Springer, 2007, ISBN-13: 978-0-387-30857-9, ISBN-10: 0-387-30857-1, e-ISBN-10: 0-387-38162-7.
- Appendix A is a reprint of the following paper:

  [KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael Wiener, editor, Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings, volume 1666 of Lecture Notes in Computer Science, pages 388-397. Springer, 1999.

©Georgia Institute of Technology, 2018-2026

Simple Power Analysis

- “Simple Power Analysis (SPA) is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations. SPA can yield information about a device's operation as well as key material.” (see Section A.2 of the paper)
- See the next slide; shown is approximately 9 ms (9×10^{-3} seconds) of a *power trace* of microcontroller running DES with its assembly code
  - 16 rounds are clearly visible
  - Oscilloscope used samples at 5 MHz, i.e., 5 million samples per second or one sample each 0.2 μs (0.2×10^{-6} seconds)

©Georgia Institute of Technology, 2018-2026

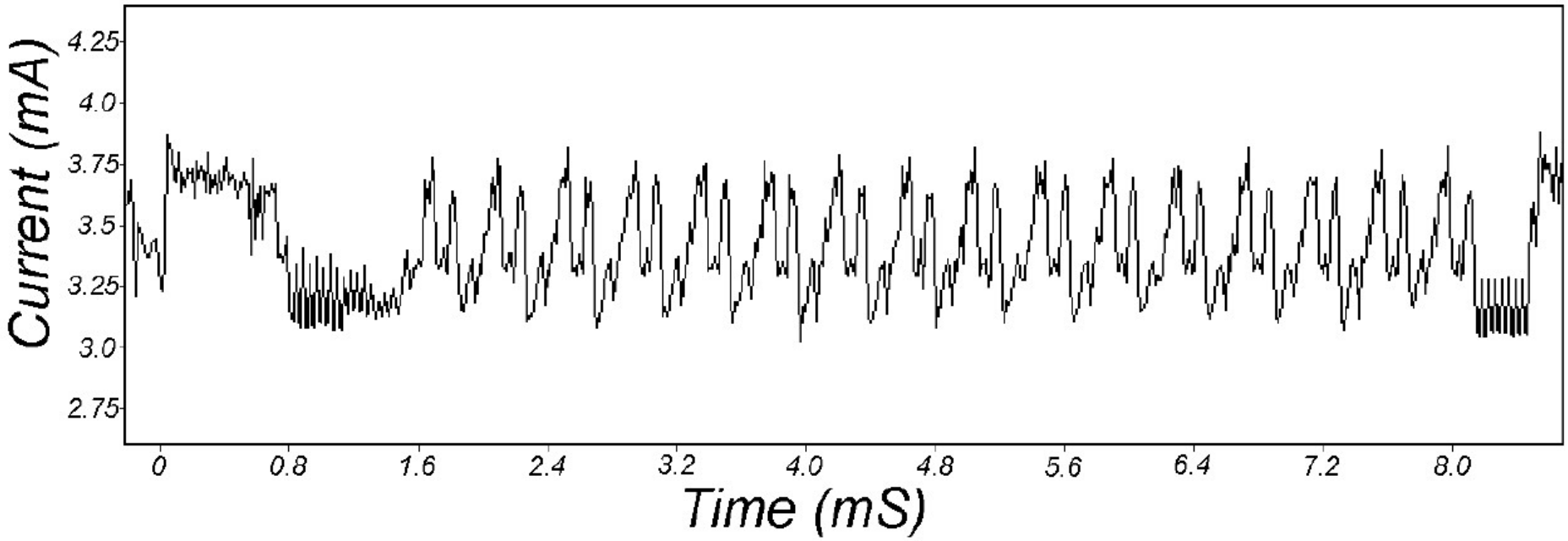


Figure A.1. SPA trace showing an entire DES operation

©Georgia Institute of Technology, 2018-2026

Power Trace of DES Encryption

- See the next slide; shown is approximately 800 µs of the second and third rounds of DES encryption
  - Left arrow shows 28-bit DES key registers C and D rotated once
  - Right hand side arrows show 28-bit DES key registers C and D rotated twice
  - Many small variations in power traces are due to conditional jumps

©Georgia Institute of Technology, 2018-2026

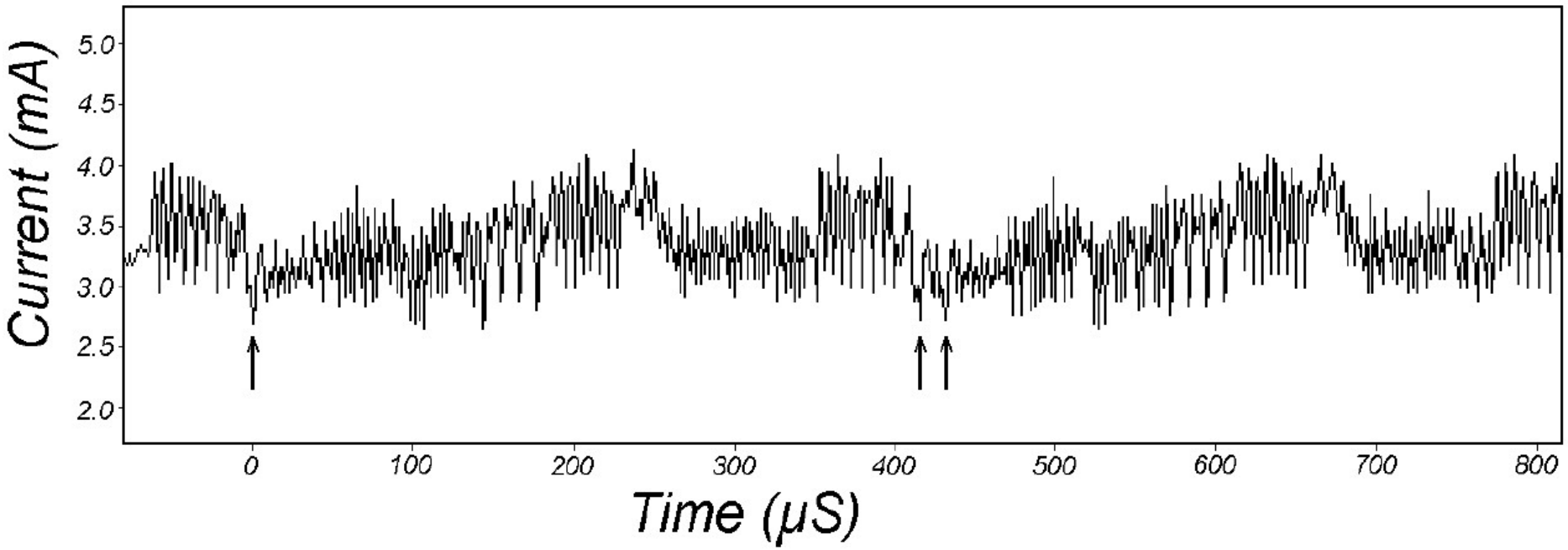


Figure A.2. SPA trace showing DES rounds 2 and 3.

©Georgia Institute of Technology, 2018-2026

Infer the Assembly Execution

- Different assembly code instructions consume different amounts of energy and hence can be inferred from the power trace
- See the next slide; shown are approximately seven clock cycles where a jump is taken or not
  - The two waveforms shown diverge at clock cycle 6
  - Top waveform corresponds to taking the jump
  - Bottom waveform corresponds to the jump not taken

©Georgia Institute of Technology, 2018-2026

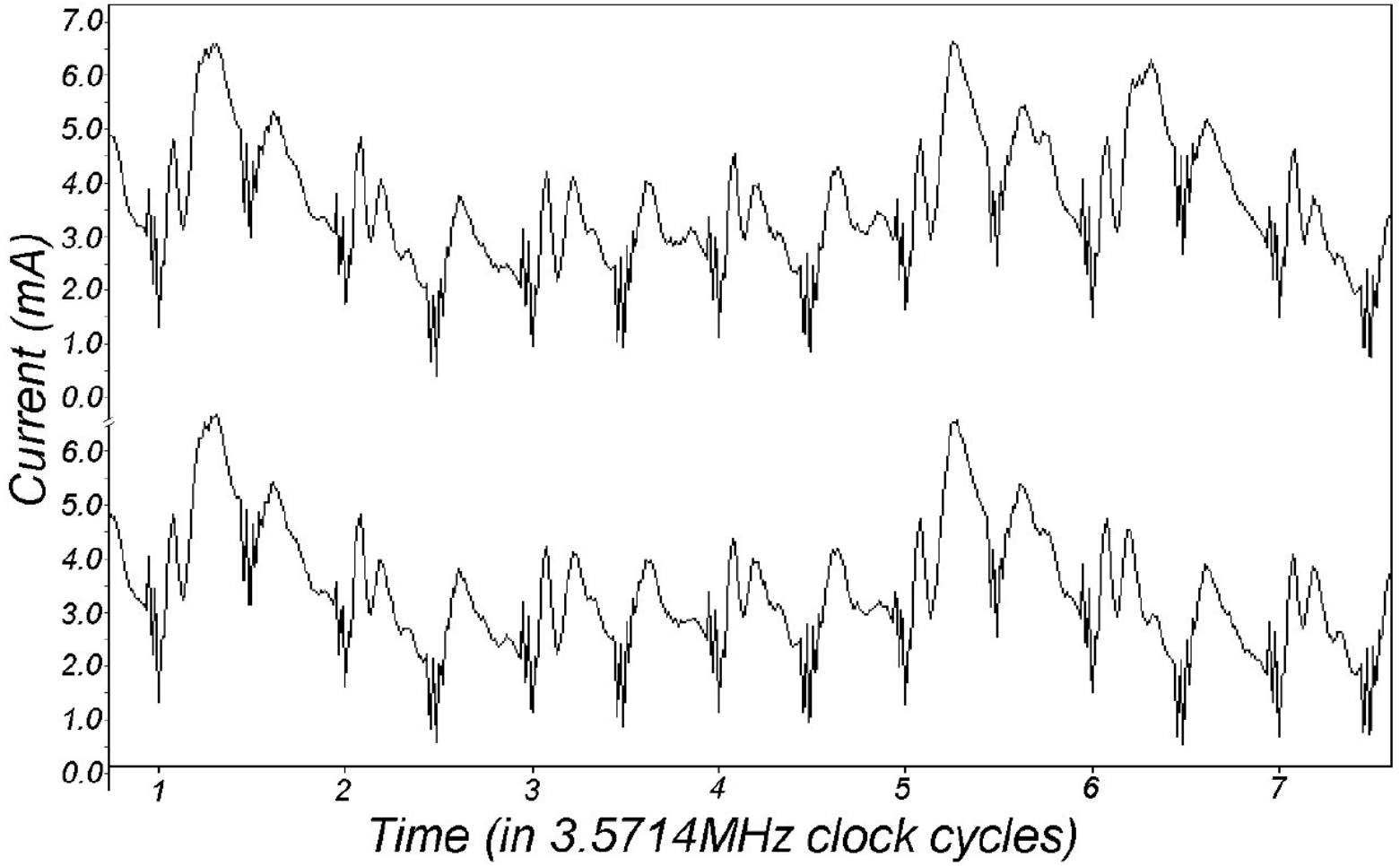


Figure A.3. SPA trace showing individual clock cycles.

Simple Power Analysis: Section A.2 at the End

Because SPA can reveal the sequence of instructions executed, it can be used to break cryptographic implementations in which the execution path depends on the data being processed. For example:

DES key schedule: The DES key schedule computation involves rotating 28-bit key registers. A conditional branch is commonly used to check the bit shifted off the end so that "1" bits can be wrapped around. The resulting power consumption traces for a "1" bit and a "O" bit will contain different SPA features if the execution paths take different branches for each.

DES permutations: DES implementations perform a variety of bit permutations. Conditional branching in software or microcode can cause significant power consumption differences for "O" and "1" bits.

Comparisons: String or memory comparison operations typically perform a conditional branch when a mismatch is found. This conditional branching causes large SPA (and sometimes timing) characteristics.

©Georgia Institute of Technology, 2018-2026

Multipliers: Modular multiplication circuits tend to leak a great deal of information about the data they process. The leakage functions depend on the multiplier design, but are often strongly correlated to operand values and Hamming weights.

Exponentiators: A simple modular exponentiation function scans across the exponent, performing a squaring operation in every iteration with an additional multiplication operation for each exponent bit that is equal to "1". The exponent can be compromised if squaring and multiplication operations have different power consumption characteristics, take different amounts of time, or are separated by different code. Modular exponentiation functions that operate on two or more exponent bits at a time may have more complex leakage functions.

©Georgia Institute of Technology, 2018-2026