# Hiding Countermeasures in Cryptographic Hardware: Part I

## *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

- This lecture covers a portion of Chapter 7 of *Power Analysis Attacks: Revealing the Secrets of Smart Cards* by Mangard et al., 2007, ISBN-13: 978-0-387-30857-9, ISBN-10: 0-387-30857-1, e-ISBN-10: 0-387-38162-7.

©Georgia Institute of Technology, 2018-2026

# Goal of Hiding Measures in Cryptographic Hardware

- Sever relationship between energy consumption (i.e., power) and the calculations being performed
  - Sever the relationship between power and the operation (for software, assembly instruction) being performed
  - Sever the relationship between power the data value(s) being used
- There is no exploitable relationship between energy consumption and calculations being performed if either of the following two conditions hold true
  - A random amount of energy is consumed each clock cycle
  - A constant amount of energy is consumed each clock cycle

©Georgia Institute of Technology, 2018-2026

# Time Dimension

- Power analysis attacks often rely on the same operation being performed at the same point in time (same relative to the start of the cryptographic computation) in each of a collection of power traces
- A countermeasure then is to randomize the execution sequence of operations necessary for the cryptographic computation
  - Dummy operation insertion at random
    - Each time a cryptographic computation, e.g., encryption, is performed, random numbers are generated based on which dummy operations are inserted at random code locations
  - Shuffling
    - Change the sequence of actual cryptographic operations each time, e.g., execute the substitution box (S-BOX) operations in a different order (in AES there are 16 S-BOX operations each round which can occur in any order within the round)
    - Advantageous over dummy insertions as execution time not increased

©Georgia Institute of Technology, 2018-2026

# Amplitude Dimension

- Goal: reduce SNR
- Option 1: increase the noise
  - Method 1.1: increase the architecture bit-width
    - E.g., change an AES architecture from 32 bits to 128 bit operations in parallel
  - Method 1.2: add dedicated noise engines
    - E.g., amplify thermal noise with an operational amplifier
- Option 2: decrease the signal
  - Method 2.1: design transistor logic with flat energy consumption characteristics
  - Method 2.2: filter the power rails
    - E.g., add an on-chip filter to the metal lines carrying the power and ground supplies

©Georgia Institute of Technology, 2018-2026

*Table 7.1.* Hiding countermeasures to make the power consumption of cryptographic devices random or equal during all clock cycles.

| | Equal power consumption | Random power consumption |
|---|---|---|
| Time dimension | - | Dummy operations, shuffling |
| Amplitude dimension | Reduction of signal | Increase of noise |

©Georgia Institute of Technology, 2018-2026

# Architectural Level Hiding in Software

- Time Dimension
  - Randomized insertion of dummy assembly instructions
  - Randomized execution of portions of cryptographic algorithm where order does not matter
    - Note that both of the above require high quality random numbers to be generated
- Amplitude Dimension
  - Choose instructions with lower information leakage
  - Avoid conditional jumps, especially when related to the key value
  - Try to limit memory addresses based on key values; when necessary, choose addresses with similar or the same Hamming Weights
  - Activate coprocessors or communications interfaces in parallel with the execution of cryptographic algorithms

©Georgia Institute of Technology, 2018-2026

# Architecture Level in Hardware

- Time dimension
  - Randomized insertion of dummy logic operations
  - Randomized execution of portions of the cryptographic algorithm where order does not matter
  - Randomized insertion of dummy clock cycles
    - Can alternate with dummy logic operations, e.g., with a duplicated set of registers
  - Randomly skip clock pulses / edges
  - Randomly change the clock frequency
  - Maintain several different clock domains and randomly switch among them
- Note that all of the above require high quality random numbers and require that the techniques themselves not be detectable

©Georgia Institute of Technology, 2018-2026

# Architecture Level in Hardware (cont'd)

- Amplitude dimension
  - Place a filter between the power and ground distribution network (metal layers) on-chip and the input and output pads connected to the power supply
    - A variety of approaches exist including switched capacitors, constant current sources and other known techniques
  - Generate noise in parallel with (at the same time as) the cryptographic algorithm
    - Uses random numbers
    - Typically requires a network of large capacitors which are charged and discharged based on the random numbers
- Note that not all techniques work well against all power measurement setups/equipment
  - E.g., measurement of electromagnetic emanation from many locations on-chip may not be significantly impacted by the placement of filters at the I/O pads

©Georgia Institute of Technology, 2018-2026