Introduction to implementation attacks

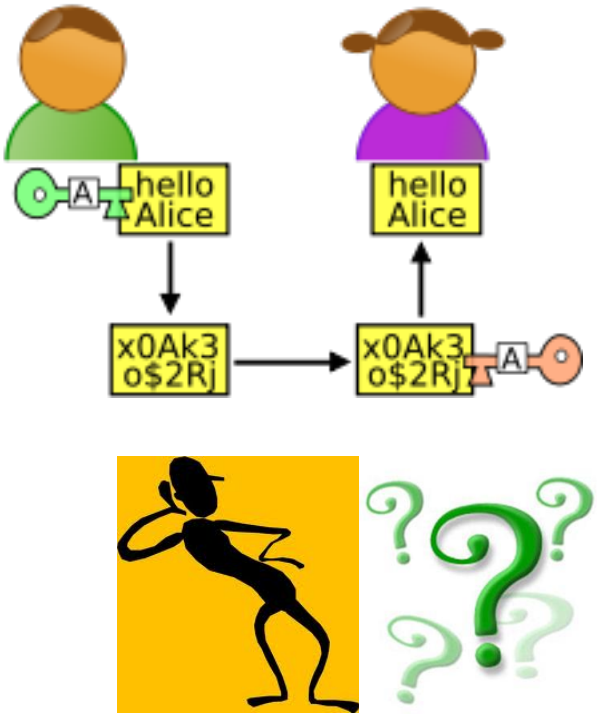
Lejla Batina

Digital Security Group
Institute for Computing and Information Sciences (ICIS)
Radboud University Nijmegen
The Netherlands

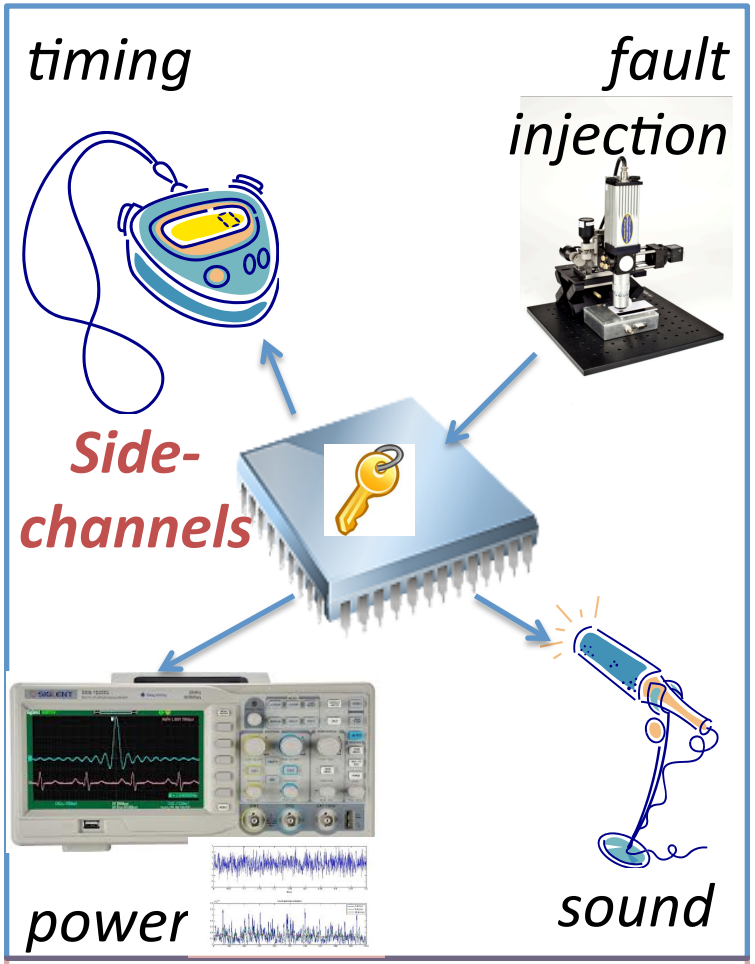
and KU Leuven, Belgium

Summer school on ***Design and Security of Cryptographic Functions and Devices*** **for Real-world applications**
Šibenik, June 3, 2014

Crypto: theory vs physical reality



Algorithms are (supposed to be) theoretically secure



Implementations leak in physical world

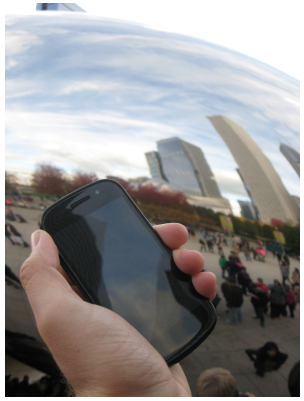
Outline

- Implementation of security vs secure implementations
  - Embedded cryptographic devices
  - Embedded security
- Side-channel analysis basics
- Power analysis attacks
- Other side-channels
- Countermeasures
- Fault analysis
- Recent and future challenges
- Conclusions

Introduction

Embedded cryptographic devices

What do attackers want to achieve?



Embedded security:

- resource limitation
- physical accessibility

(In)security for Embedded Systems

"Researchers have extracted information from nothing more than the reflection of a computer monitor off an eyeball or the sounds emanating from a printer." - Scientific American, May 2009.

Matchstick-sized sensor can record your private chats

› 26 September 2013 by Jim Nash
› Magazine issue 2936. Subscribe and save

A sensor previously used for military operations can now be tuned to secretly locate and record any single conversation on a busy street

EVERYONE knows that to have a private chat in the NSA era, you go outdoors. Phones, the internet, email and your office can all be compromised with ease. But soon even that whispered conversation in the park may no longer be safe from prying ears.

Your Smartphone Spies On What You Type

Written by Mike James
Sunday, 29 September 2013 00:00

All you have to do is place your phone next to your keyboard to provide a direct channel for anyone to read what you are typing - and it's all down to the vibration of the keys.

We all do it - place our phones down on the desk next to the keyboard. This might not be such a good idea if you want to keep your work to yourself.

EMBEDDED DEVICES LEAK AUTHENTICATION DATA VIA SNMP COMMUNITY STRING

by Michael Mimoso Follow @mike_mimoso May 16, 2014 , 1:55 pm

Researchers have discovered previously unreported problems in SNMP on embedded devices where devices such as secondary market home routers and a popular enterprise-grade load balancer are leaking authentication details in plain text.

Our scope: Implementation Attacks

"Remote keyless entry system for cars and buildings is hacked" March 31, 2008
- KeeLoq: eavesdropping from up to 100 m
- www.crypto.rub.de/keeloq

PS3 hack
- ECDSA implementation failed
- resulted in PS3 master key recovery

Recent attacks on real products: contactless smartcards with Mifare Classic, DESFire, Atmel CryptoMemory,…

The goals of attackers

- Secret keys/data
- Unauthorized access
- IP/piracy
- (Location) privacy
- (Theoretical) cryptanalysis [RS01]
- Reverse engineering
- Finding backdoors in chips [SW12]
- …

Physical security: before and today

- Tempest – known since early 1960s that computers generate EM radiation that leaks info about the data being processed
- In 1965, MI5: microphone near the rotor-cipher machine used by the Egyptian Embassy the click-sound the machine produced was analyzed to deduce the core position of the machines rotors
- 1979: effect of cosmic rays on memories (NASA & Boeing)
- First academic publications on SCA by Paul Kocher: 1996 (timing) and 1999 (power)
- Bellcore attack in 1997: Boneh, DeMillo and Lipton
- Many successful attacks published on various platforms and real products e.g. KeeLoq, CryptoMemory, (numerous) contactless cards

Physical security today

- As a research area took off in the late 90's
- CHES workshop since 1999
- Many successful attacks published on various platforms and real products e.g. KeeLoq [EK+08], CryptoMemory [BG+12]
- A good business model for security evaluation labs e.g. Riscure and Brightsight

Concepts of side-channel leakage

- Side-channel leakage is based on (non-intentional) physical information
- Can enable new kind of attack
- Often, optimizations enable leakages
  - Cache: faster memory access
  - Fixed computation patterns
  - Square vs multiply (for PK)

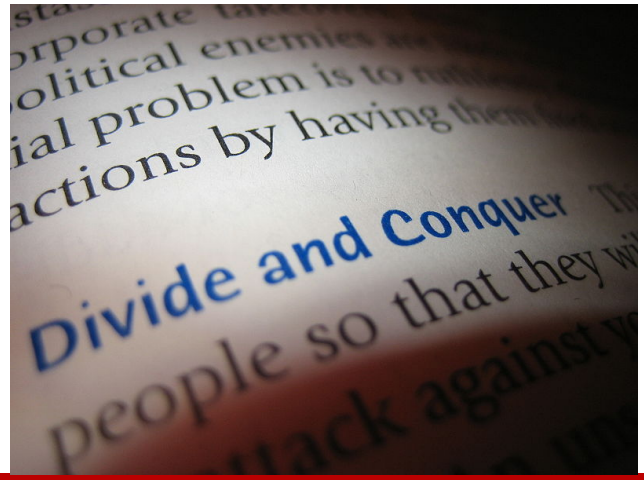
Side-channel attacks basics

Sources of side-channel information

- Timing (Kocher 1996), Power (KJJ 1999), EM (UCL & Gemplus 2001)
- Temperature (Naccache et al.)
  - information about the device's malfunction leaked-out via its temperature
- Light (Kuhn)
  - Reading CRT-displays at a distance
  - Observing high-frequency variations of the light emitted
- Sound (Shamir and Tromer)
  - Distinguishing an idle from a busy CPU
  - Distinguish various patterns of CPU operations and memory access (RSA signatures)
- Photonic emissions (TU Berlin)

Leakage is often explorable

- Due to the (dependency of leakages on) sequences of instructions executed
- Due to the data (even sensitive!) being processed
- Due to other physical effects
- …



Attack categories

- Side-channel attacks
  - use some physical (analog) characteristic and assume access to it
- Faults
  - use abnormal conditions causing malfunctions in the system
- Microprobing
  - accessing the chip surface directly in order to observe, learn and manipulate the device

Taxonomy of Implementation Attacks

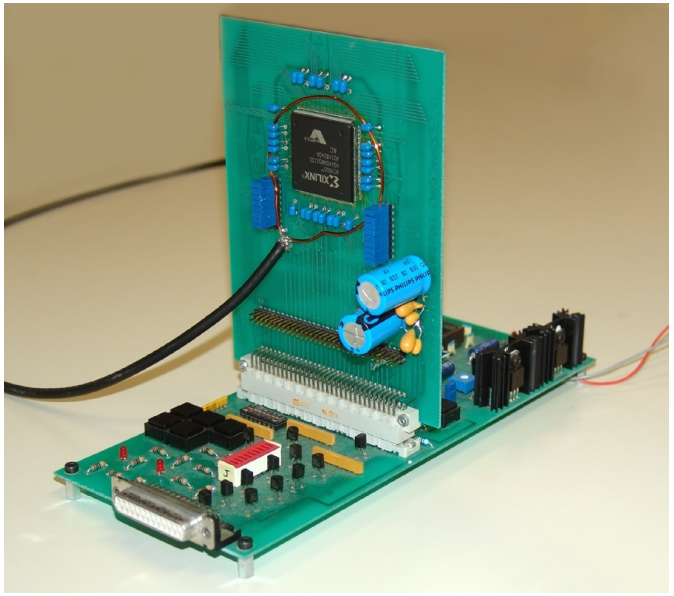
- Active versus passive
  - Active
    - The key is recovered by exploiting some abnormal behavior e.g. power glitches or laser pulses
    - Insertion of signals
  - Passive
    - The device operates within its specification
    - Reading hidden signals
- Invasive versus non-invasive
  - Invasive aka expensive: the strongest type e.g. bus probing
  - Semi-invasive: the device is de-packaged but no contact to the chip e.g. optical attacks that read out memory cells
  - Non-invasive aka low-cost: power/EM measurements
- Side-channel attacks: passive and non-invasive

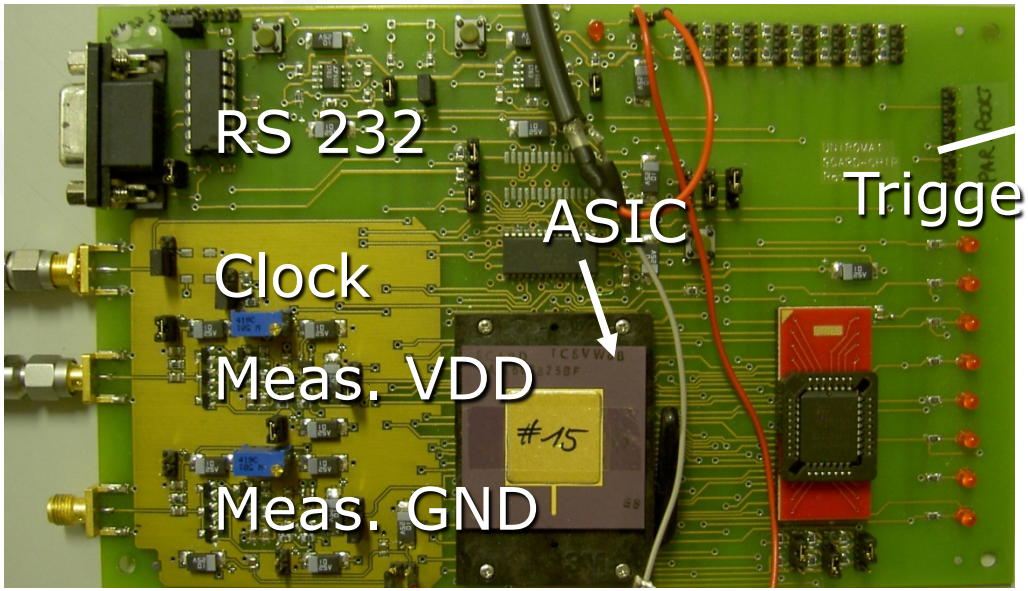
Analysis capabilities

- "Simple" attacks: one or a few measurements - visual inspection
- Differential attacks: multiple measurements
  - Use of statistics, signal processing, etc.
- Higher order attacks: *n*-th order is using *n* different samples
- Combining two or more side-channels
- Combining side-channel attack with theoretical cryptanalysis

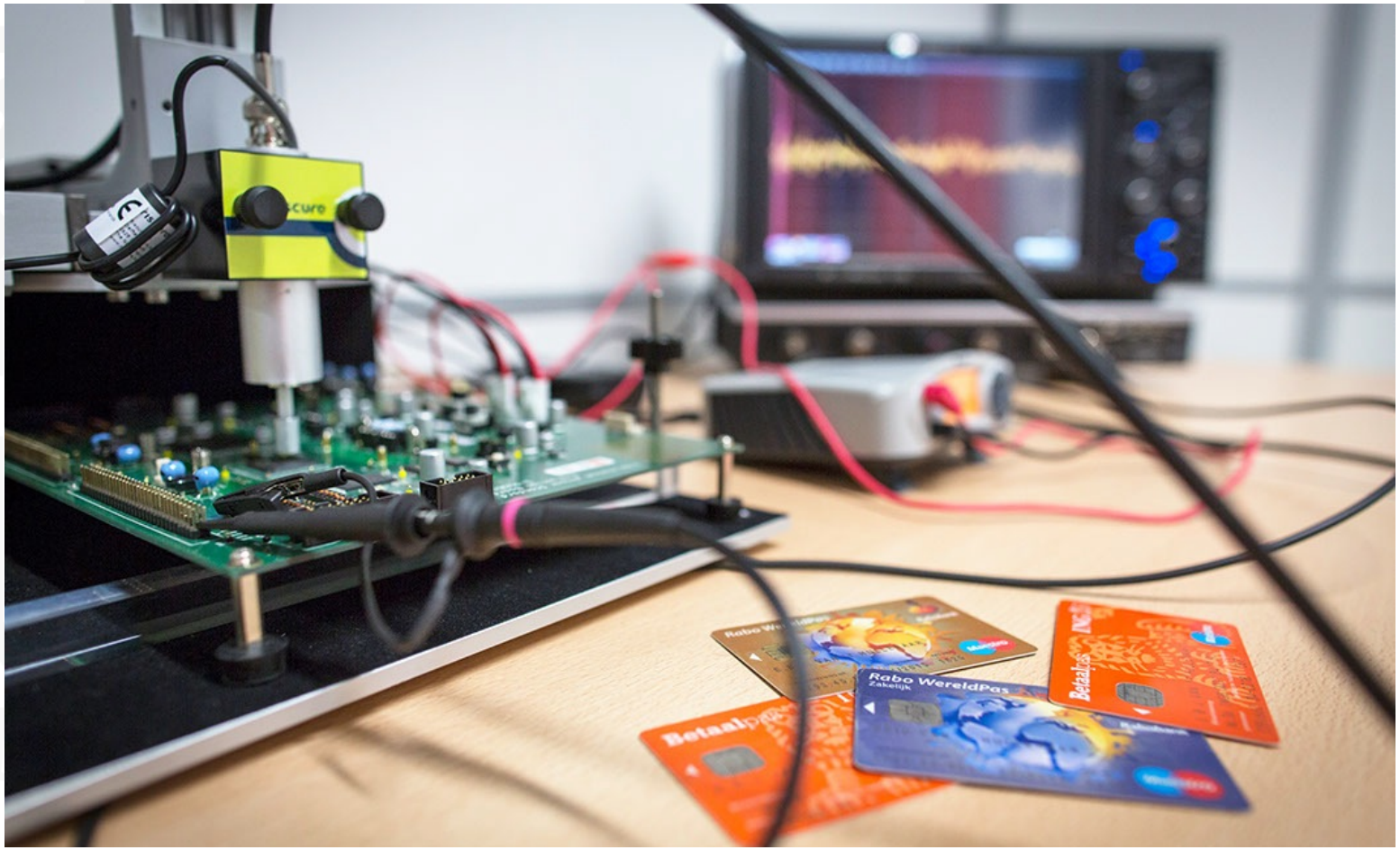
Devices under attack

- Smart card
- FPGA, ASIC
- RFID, PDAs
- Phones, USBs, ...
- Actual products





Implementation attacks - equipment



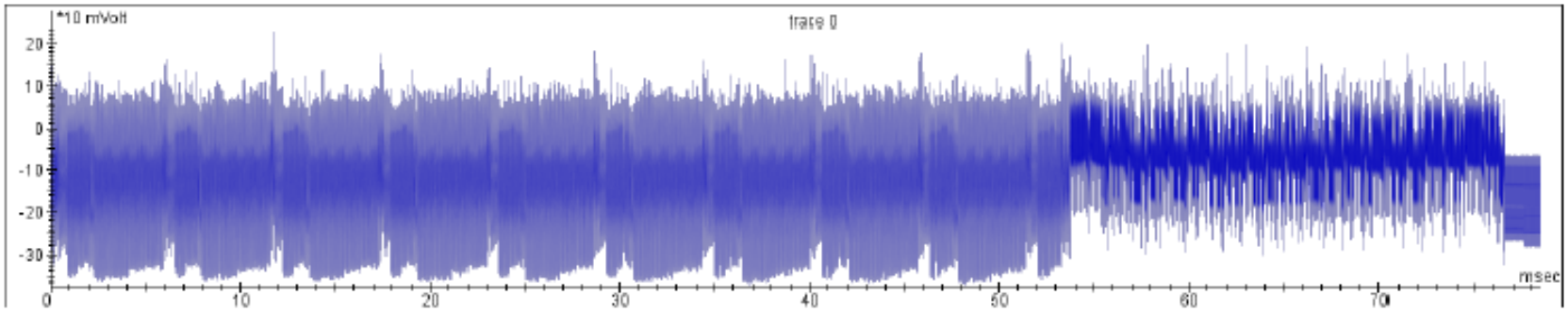
Simple Power Analysis (SPA)

Simple Power Analysis (SPA)

- Based on one or a few measurements
- Mostly discovery of data-(in)dependent but instruction-dependent properties e.g.
  - Symmetric:
    - Number of rounds (resp. key length)
    - Memory accesses (usually higher power consumption)
  - Asymmetric:
    - The key (if badly implemented, e.g. RSA / ECC)
    - Key length
    - Implementation details: for example RSA w/wo CRT
- Search for repetitive patterns

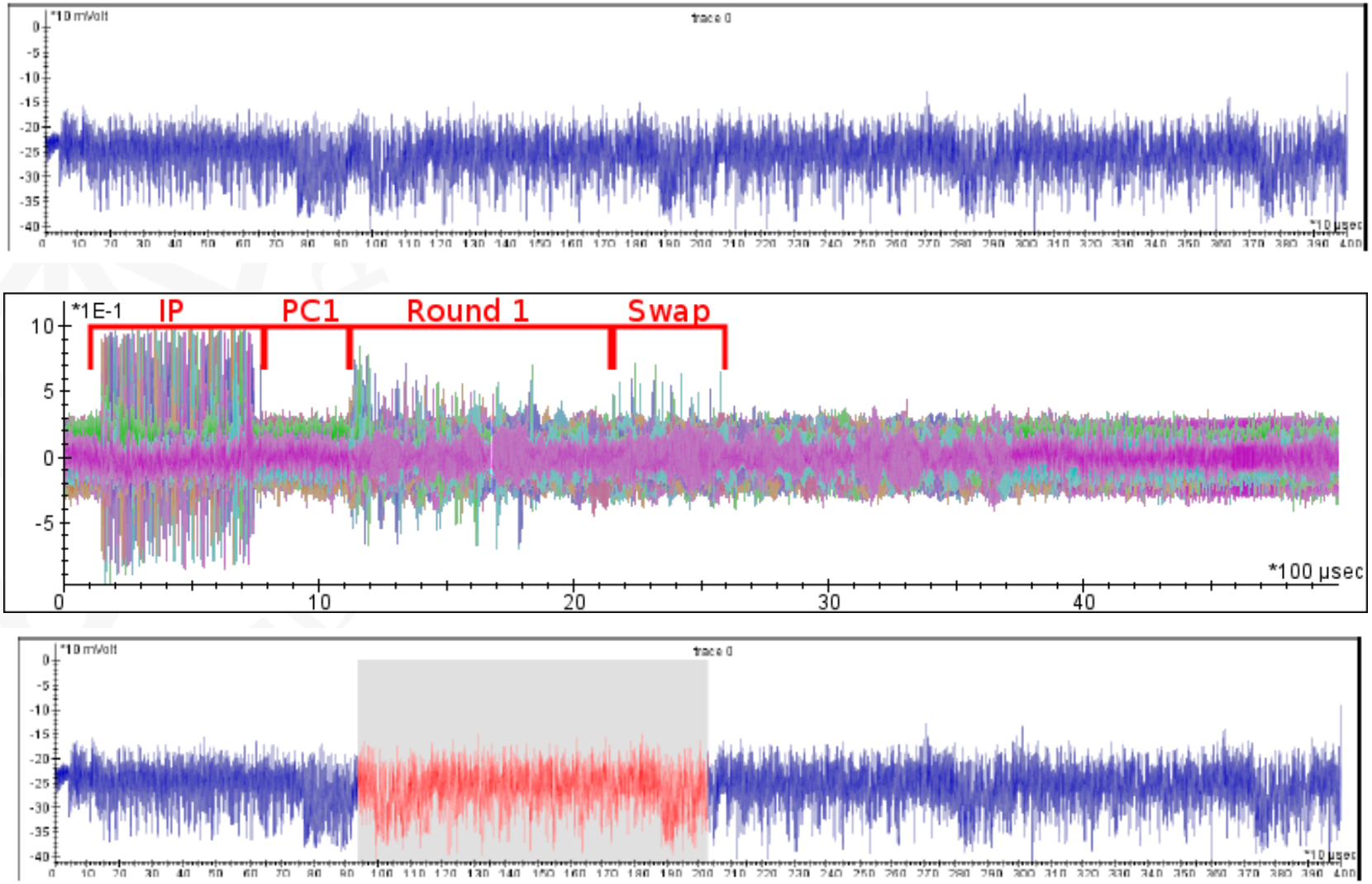
conditional operation

Simple Power Analysis

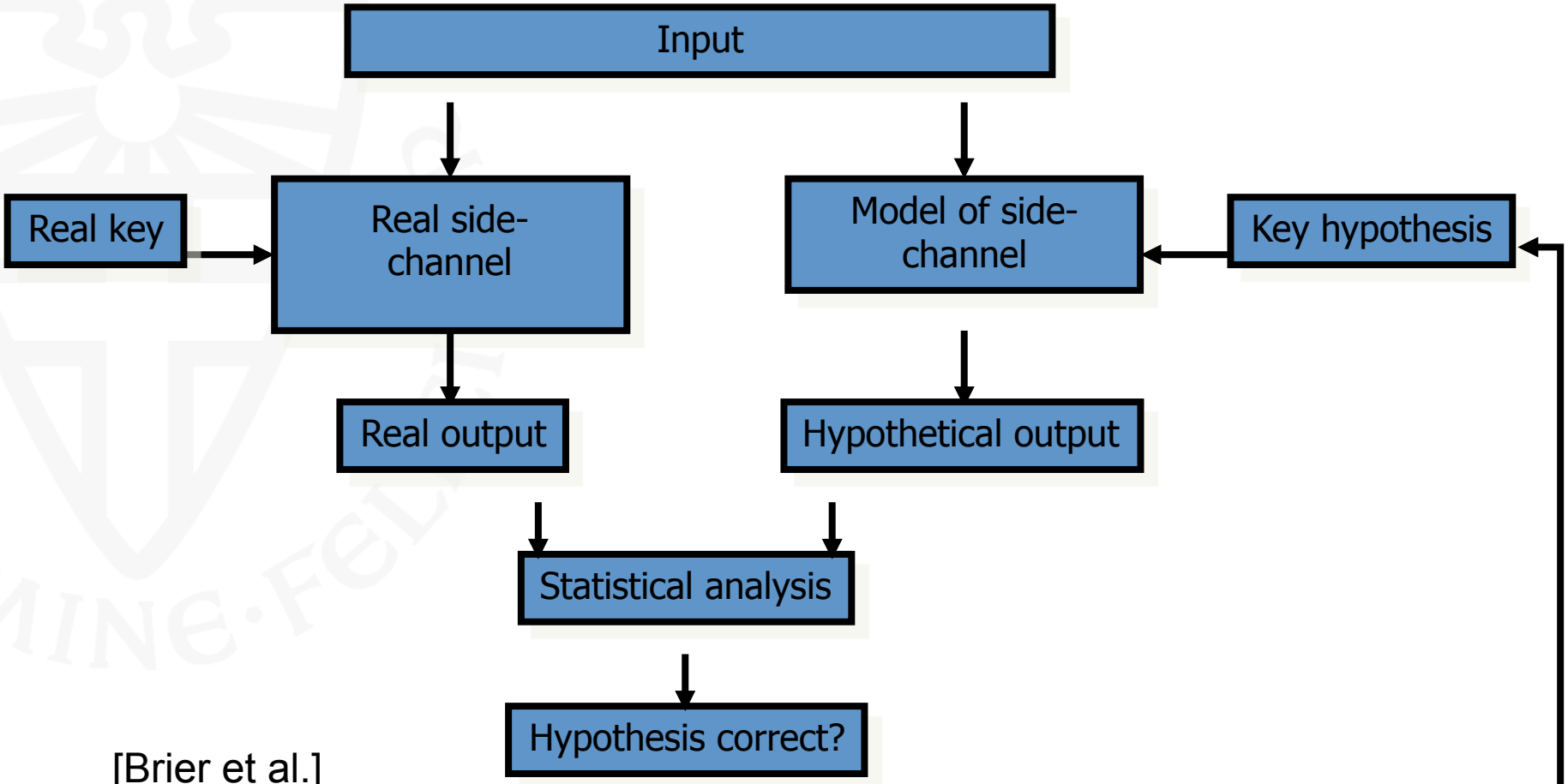


time axis

Using SPA to find a good place to attack



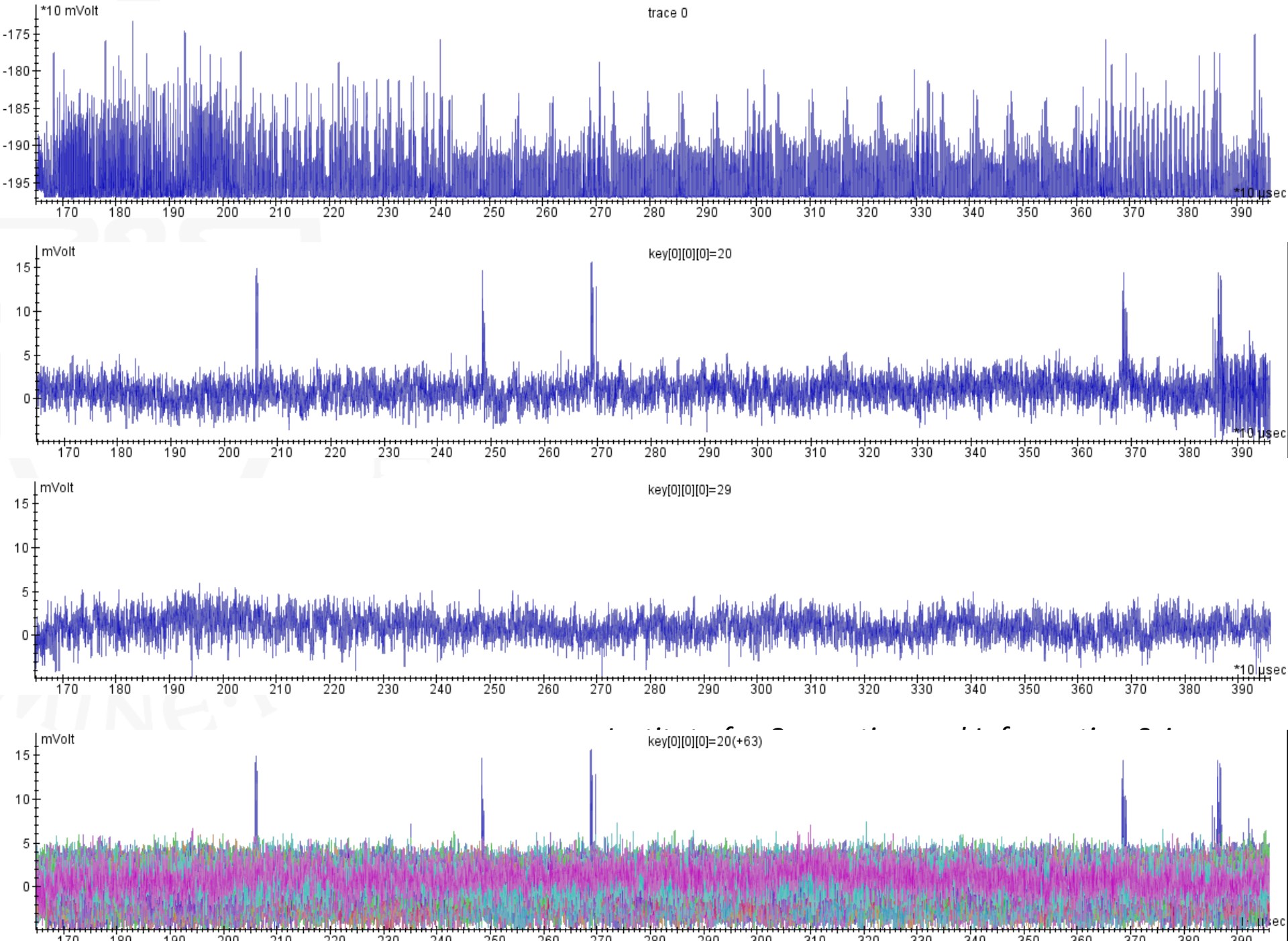
Differential Power Analysis (DPA)



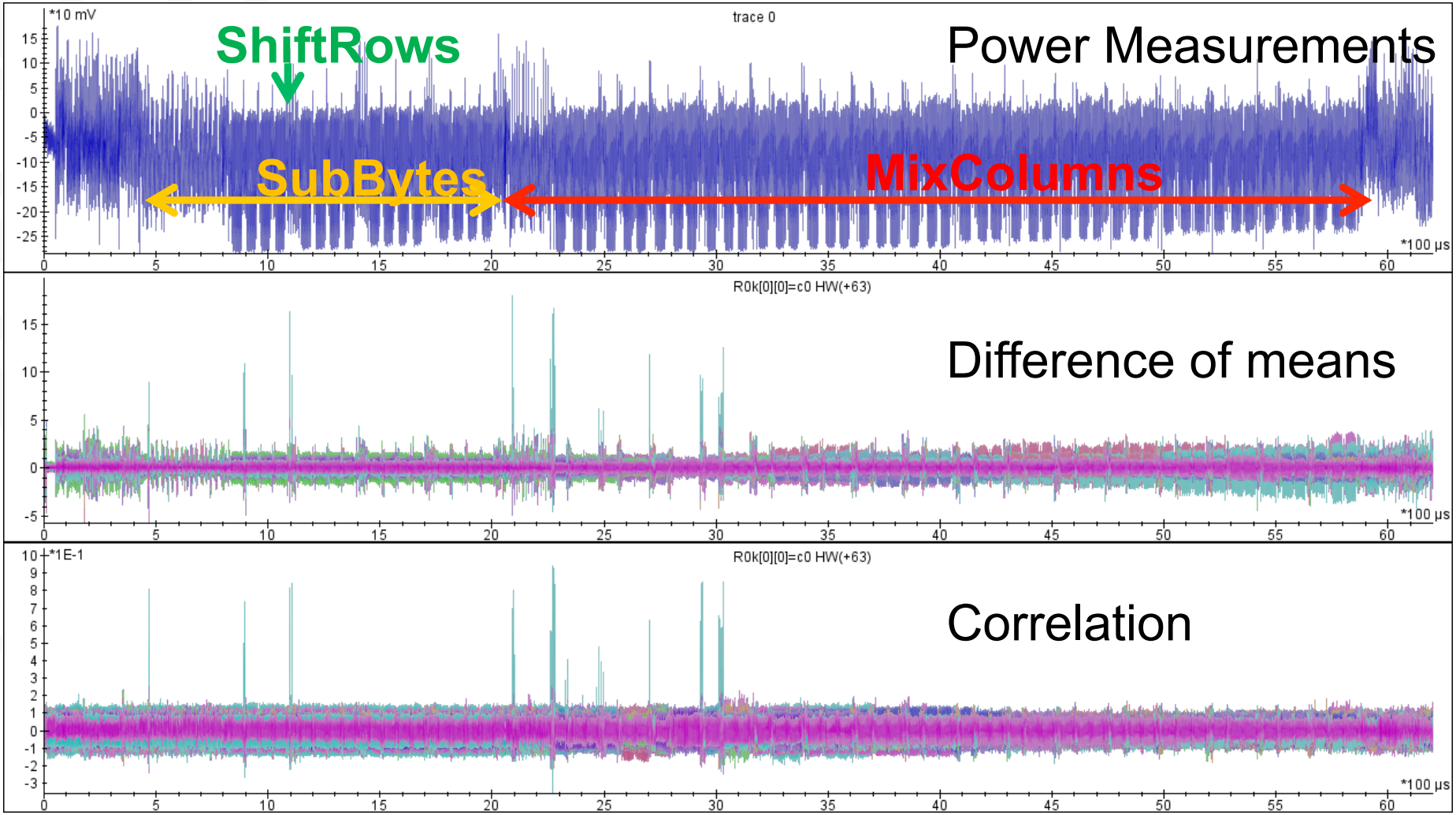
[Brier et al.]

Power Analysis

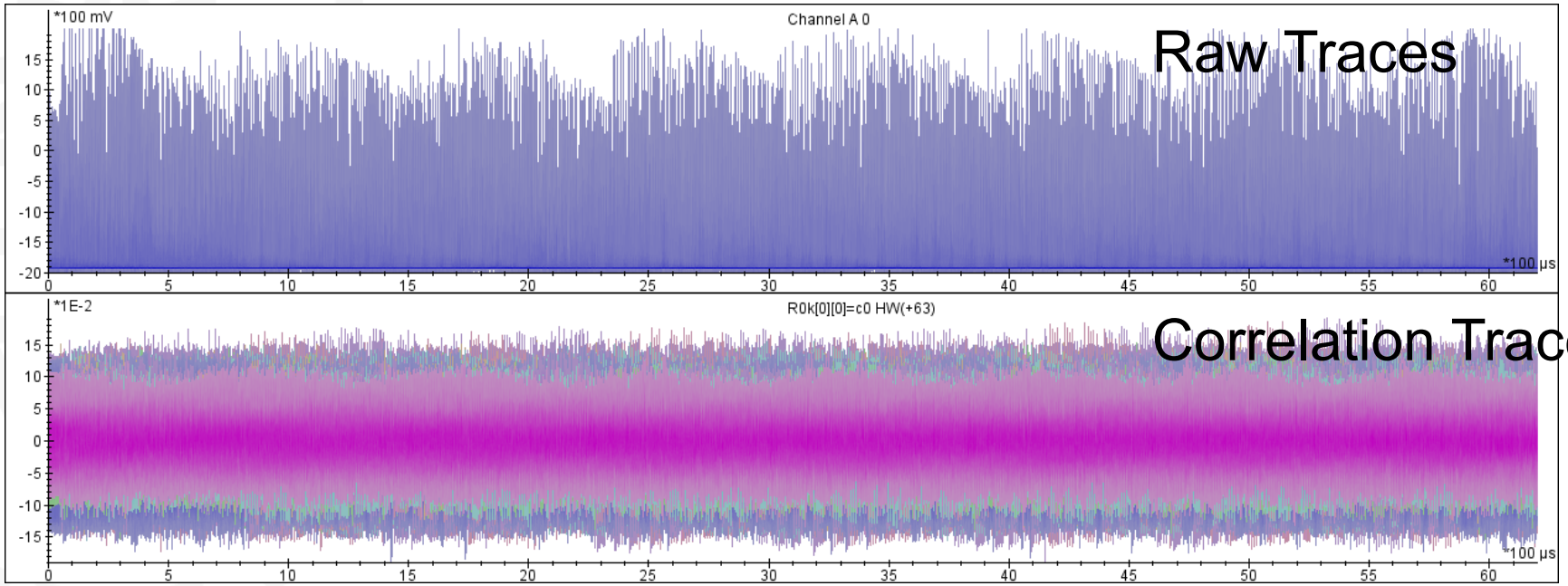
- Direct attacks
  - Simple Power Analysis (1999)
  - Differential Power Analysis (1999)
  - Correlation Power Analysis (2004)
  - Collision Attacks (2003)
- Two-stage attacks
  - Template Attacks (2002)
  - Stochastic Models (2005)
  - Linear Regression Analysis (LRA)
- Advanced attacks: Mutual Information Analysis - MIA (2008), Diff. cluster analysis (2009), PCA (2011), ...


*10 mVolt
trace 0
-175
-180
-185
-190
-195
170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390
*10 usec
mVolt
key[0][0][0]=20
15
10
5
0
170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390
*10 usec
mVolt
key[0][0][0]=29
15
10
5
0
170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390
*10 usec
mVolt
key[0][0][0]=20(+63)
15
10
5
0
170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 360 370 380 390
usec

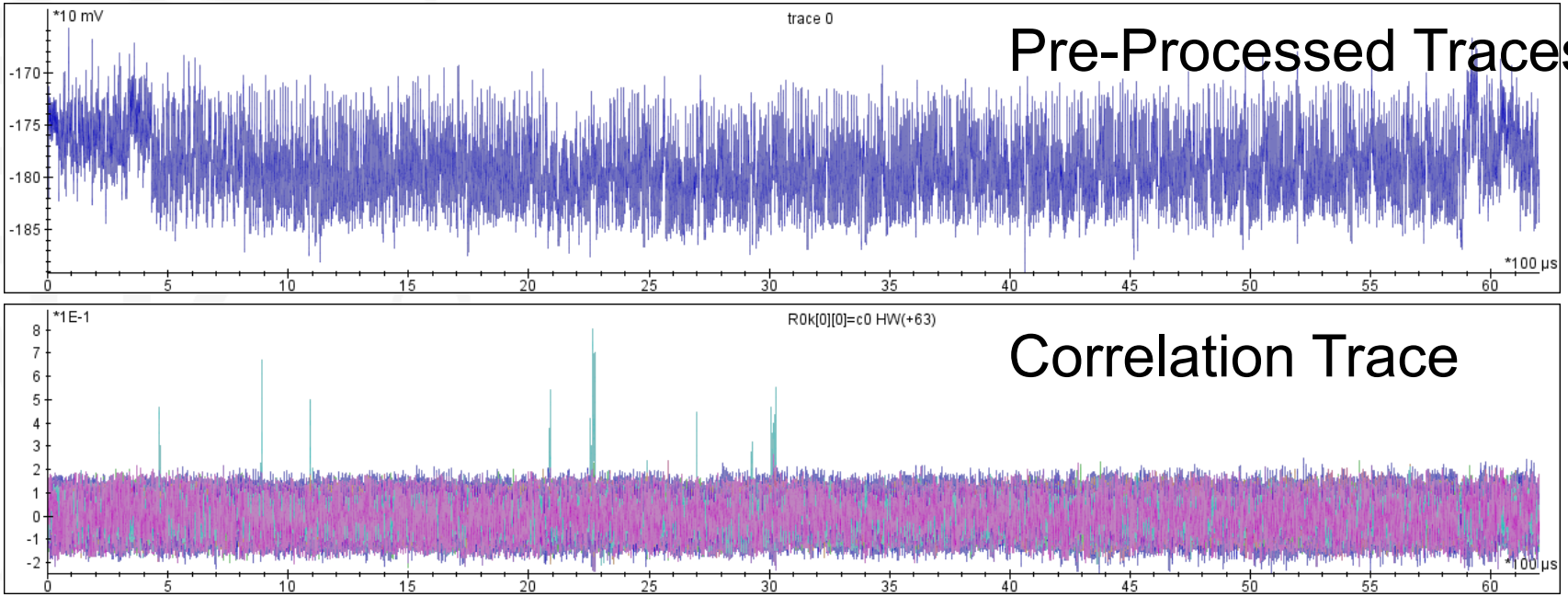
Attacking AES 1st round



Noisy Traces



Preprocessing



Intro to Static CMOS

- Most popular circuit style!
- Power consumed when an output signal switches is much higher (than when no switch happens)

0->0: static (low)
0->1: static + dynamic (high)
1->0: static + dynamic (high)
1->1: static (low)

"We don't understand electricity. We use it."
- Maya Angelou

=> Dynamic power consumption is the dominant factor in the total power consumption and it is data dependent!

Leakage models

- Transition = Hamming distance model
  - Counts number of 0->1 and 1->0 transitions
  - Assuming same power consumed for both, ignores static power consumption
  - Typically for register outputs in ASIC's
  - HD(v_0, v_1)=HW(v_0 xor v_1)
  - Requires knowledge of preceding or succeeding v_i
- Hamming weight model
  - Typical for pre-charged busses
- Weighted Hamming weight/distance model
- Signed Hamming distance (0->1 neq 1->0)
- Dedicated models for combinational circuits

Practical attacks on all platforms

- In the beginning mainly in-house made set-ups
- **Attacks on actual products:**
  - 2008: products employing KeeLoq: Remote Keyless Entry (RKE) systems (chip embedded in RFID transponders) [EK+08]
  - 2009-2011: real-world contactless payment applications based on MIFARE Classic cards, MIFARE DESFire cards (public transport etc.) [KK+09, KS+10]
  - 2012: Atmel CryptoMemory devices (used for printers, gaming, laundromats, parkings etc.) [BG+09]
  - 2013: system 3060 manufactured and marketed by SimonsVoss (wireless door openers)

Template attacks [CRR02]

- Consist of 2 phases:
  - Characterization or Building templates
  - Template matching or Key recovery
- Assumption that the same device (as the one under attack) is available
- Find templates for certain sequences of instructions
- Obtaining a template for every pair of data and key
- Maximum-likelihood rule finds the right key

Combining with theoretical cryptanalysis

Classical vs side-channel cryptanalysis

- Knowledge:
  - Input/output pairs
  - Input/output pairs + some leakage
- Applicability
  - Generally applicable
  - Limited to certain implementation

Combining both could be beneficial when when access to side-channel info is restricted!

Algebraic side-channel attacks [RS09]

- Phase 1 – on-line:
  - Adversary selects as many intermediate computations in the target algorithm as possible and measures their physical leakage represented e.g. by Hamming weight
- Phase 2 – off-line:
  - Adversary writes the algorithm as a system of equations and adds the previously defined functions with known outputs to the system and uses e.g. SAT solvers to find the solutions

Algebraic side-channel attacks on block ciphers

- Applied on PRESENT and AES
  - Single encrypted plaintext was enough
  - Experiments on an 8-bit microcontroller
  - AES example: 18 000 equations in 10 000 variables
    - Exploring SubBytes implemented as a 256-byte table lookup
    - Exploring MixColumn implemented as four 256-byte table lookups and 9 XOR operations (giving 13 potential leakage points)
- Extended beyond the Hamming weight model
- Generalization of collision-based attacks

Side-channel attacks: Countermeasures

Countermeasures

Purpose: destroy the link between intermediate values and power consumption

- Masking
  - A random mask concealing every intermediate value
  - Can be on all levels (arithmetic -> gate level)
- Hiding
  - Making power consumption independent of the intermediate values and of the operations
  - Special logic styles, randomizing in time domain, lowering SNR ratio

Software Countermeasures

- Time randomization: the operations are randomly shifted in time
  - use of NOP operations
  - add random delays
  - use of dummy variables and instructions (sequence scrambling)
  - data balancing (a data element is represented redundantly to make H.w. constant)
- Permuted execution
  - rearranged instructions e.g. S-boxes
- Masking techniques

Hardware countermeasures

- Noise generation
  - hw noise generator would include the use of RNG
  - total power is increased (problem for handheld devices)
- Power signal filtering
  - ex.: RLC filter (R-resistor, C-capacitor, L-inductor) smoothing the pow. cons. signal by removing high frequency components
  - one should use active comp. (transistors) in order to keep pow. cons. relatively constant - problem for mob. phones
  - detached power supplies - Shamir
- Novel circuit designs
  - special logic styles

Masking

- Random masks used to hide the correlation between the power consumption and the secret data
- Two types of masking
  - Boolean masking- use $\oplus$, $x' = x \oplus r_x$
  - Arithmetic masking - use addition and subtraction modulo 2^w (where w is the digit size), e.g. $x' = (x - r_x) \bmod 2^w$
  - The conversion from one type to another
- Costs for an example platform
  - Software e.g. 32-bit ARM processor: cycle count - factor 1.96; RAM - 6.27, ROM - 1.36 [Mes00]
- Hardware, ASIC: overhead for masking triples the size of the S-box, from 234 gates (NAND equivalents) to 700 gates [CB08]

Masking AES

- A masking function: $f(x, m) = x * m$
  - * additive or multiplicative masking
- AES includes all linear transformations except S-boxes

$$S(x + m) = S(x) + m' \neq S(x) + S(m)$$

- several solutions:
  - Re-computation of masked S-box s.t.

$$Masked\ S(x + m) = S(x) + m$$

  - Multiplicative masking $S(x) = A \times x^{-1} + b$
  - Masking in tower fields: in GF(2^2) inversion is linear

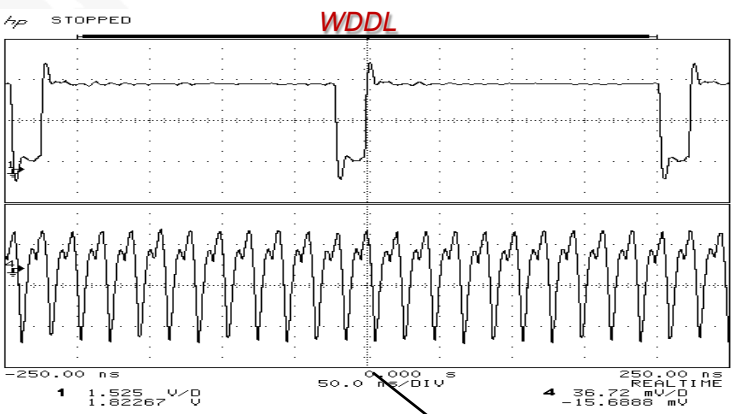
Issues with masking

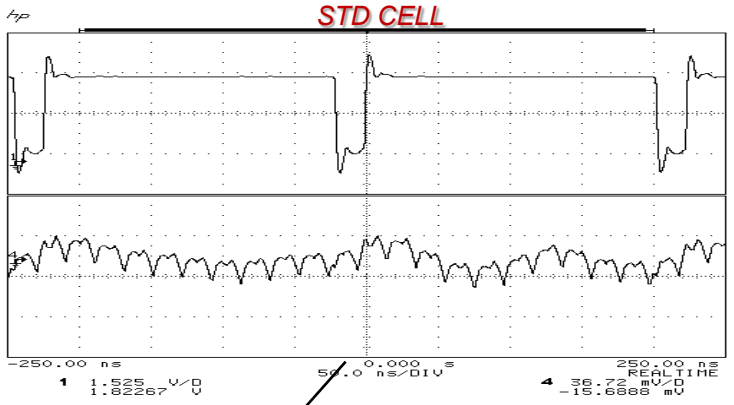
- A TRNG is required
- Masked implementation leak due to glitches
  - More in the talk of Svetla
- Masking public-key algorithms
  - Many algorithmic/arithmetic options

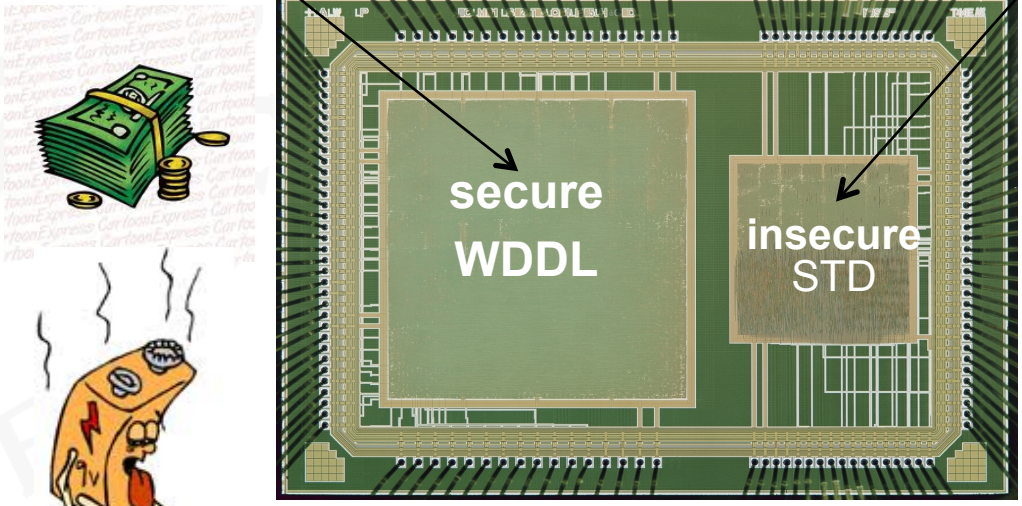
Hardware countermeasures – details in the talk of Ingrid

- Dynamic and differential logic (pre-charged dual rail)
  - Duplicate logic
  - Bits are encoded as pairs, e.g. 0 = (1,0) and 1 = (0,1)
  - Circuit is pre-charged, e.g. to all zero (0,0)
  - Each DRP gate toggles exactly once per evaluation
  - The number of bit flips is constant and data independent

CMOS vs. WDDL (Tiri, Verbauwhede 2004)







Doesn't work for small devices!

Fault Analysis: Introduction, Basic concepts & History

History

- 1978: one of the first examples fault injection was unintentional, discovered by May and Woods (radioactive particles)
- 1979: effect of cosmic rays on memories (NASA & Boeing)
- 1992: use of laser beam to charge particles on microprocessors, discovered by Habing
- 1997: 1st academic pub. by Boneh, DeMillo, and Lipton showing what's possible with a single fault [BDL97]
- 1997: differential fault analysis on secret-key cryptosystems by Biham and Shamir [BS97]
- 2002: 1st pub. implementing Bellcore attack [AB+12]
- 2003: 1st FDTC workshop

Methods

- Variation in supply voltage i.e. glitching
  - Can cause a processor skip instruction
  - Actively investigated by smartcard industry
- Variation in the external clock
  - May cause data misread or an instruction miss
- Change in temperature
  - Change in RAM content
  - Write operations work better
- White light – photons induce faults
- X-rays and ion beams

Goals

- Insert computational fault
  - Null key
  - Wrong crypto result (Differential Fault Analysis - DFA)
- Change software decision
  - Force approval of false PIN
  - Reverse life cycle state
  - Enforce access rights
- …

Types

- Non-invasive
  - Glitching: clock, power supply
  - Countermeasures: sensors for V, T, f
- Semi-invasive attacks
  - Using UV light, laser, imaging, optical fault injection
- Invasive attacks
  - Microprobing, FIB probing, optical reverse engineering
  - Require a long preparation phase

Fault attacks in theory and practice

Facts

- Injecting exploitable faults is very hard
  - Reproducibility, accuracy
  - Regardless the target device (hardware accelerator, microprocessor, Java card, etc.)
- Cheap techniques e.g. glitching can be very effective

Expensive tools for physical attacks

- Microscope
  - optical or scanning electron microscope (SEM)
- Probe station
  - to probe wires on the chip
- Focused Ion Beam (FIB)
  - uses ions instead of electrons
  - not only for observing, but also making changes:
    - removing or adding wires, insulators,...
- Laser cutter
  - to cut holes through passivation layer, expose lower levels for probing,…, for much lower cost than FIB

Fault analysis: What is now possible

- To flip bits in SRAM cell, by targeting one of its transistors (light pulses, laser, FIB…)
  - However, using light single cell cannot be hit!
- Laser can inject multiple faults within the same execution of a cryptographic algorithm
- FIB enables an attacker to:
  - arbitrarily modify the structure of a circuit (i.e. reconstruct missing buses, cut existing wires,…)
  - debug and patch chip prototypes
  - reverse engineer by adding probing wires to parts of the circuit that are not commonly accessible

Low-cost fault injection techniques

- Attacker collects **a large number** of faulty computations and selects exploitable faults
- Examples:
  - under-powering of a computing device (can cause a single-bit error and no knowledge of the implementation details of the platform is needed!)
  - injection of well-timed power spikes on the supply line of a circuit (possible to skip the execution of a single instruction of microprocessor code)
  - tampering with the clock (shorten the length of a single cycle or overclocking the device)
  - Increasing temperature
  - EM pulses (Eddy current)

Fault injection by light

- UV lamp or a camera flash can be used
- Can cause the erasure of EEPROM and Flash memory cells (usually constants are kept there!)
- It is possible also to selectively wipe out only a part of the stored data
- Optical attacks by S. Skorobogatov [SA02]

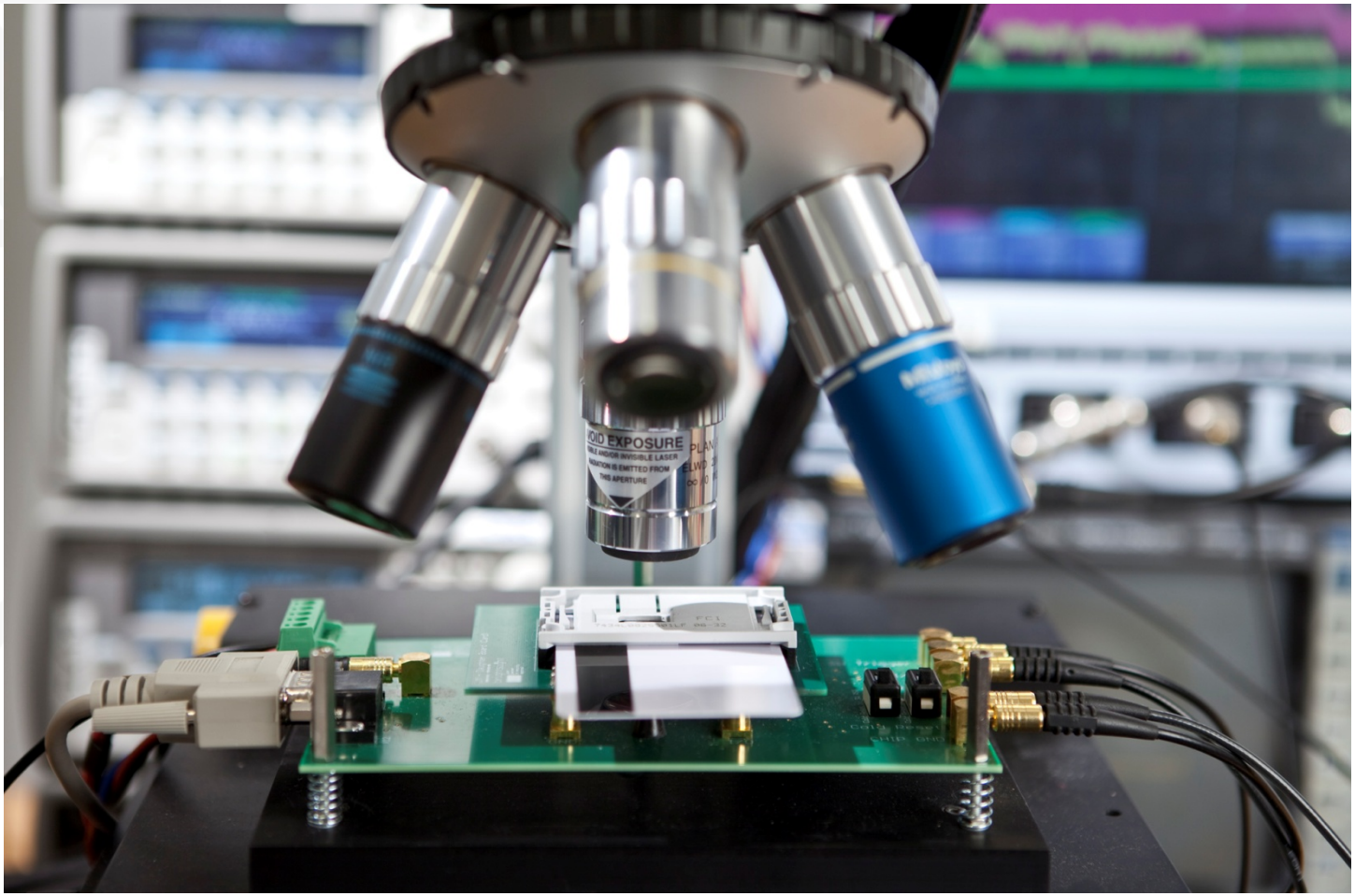
Set-up with laser

Camera used to get images of laser spots on the surface

<u>Goals:</u>

- removing polymer layer from a chip surface
- local removal of a passivation layer for microprobing attacks
- cutting metal wires inside a chip
- can access the second metal layer at most

Laser-cutter set-up



[Source: Brightsight]

Typical problems

- Inaccurate timing of fault injection
- Card breaks down after fault injection test
- Too many parameters that have to be fixed

Differential Fault Analysis

- Bellcore attack in 1995
  - Differential faults on RSA-CRT signatures
  - Requires 1 correct and 1 wrong signature
- Attack on DES in 1997
  - Biham and Shamir
  - Attacks on last rounds of DES
- Special attacks on AES, RC5, ECC etc.
- Fault attacks on key transfer

DFA on symmetric-key crypto

- Basic DFA scenario:
  - adversary obtains a pair of ciphertexts that are derived by encrypting the same plaintext (one is correct value and the other is faulty)
  - two encryptions are identical up to the point where the fault occurred

  => two ciphertexts can be regarded as the outputs of a reduced-round iterated block cipher where the inputs are unknown but show a small (and possibly known) differential

DFA on DES

- The original attack of Biham and Shamir
  - exploits computational errors occurring in the final rounds of the cipher
  - assumes that one bit of the right half of the DES internal state is flipped at a random position
- Attacks that exploit faults occurring in middle rounds of DES also possible
- A DFA technique that targets the early rounds of the cipher is based on internal collisions

Countermeasures on symmetric-key cryptosystems

Hardware countermeasures

- Light detectors
- Supply voltage, frequency detectors
- Active shields
- Redundancy: duplication of hardware blocks
- Dual rail implementations
- *(m-of-n) encoding:* each bit is represented by *n* wires, from which exactly *m* carry a 1

Generic countermeasures

- Correctness check: encrypt twice
- Random delays: limits the precision
- Masking:
  - Linear secret sharing complicates probing wires of the device
  - Adversary cannot predict the effect of the injected fault

Countermeasures for symm.-key ciphers

- Introducing redundancy is harder than for PKC
  - modular redundancy i.e. the algorithm is executed several times – expensive
- Using the inverse
- Loop invariant:
  - 2nd variable counting in the opposite way prevents tampering the counter of a loop
  - add a signature that is updated in every run of the loop (checksum)
- To ensure the integrity of the stored data, Cyclic Redundancy Check (CRC) can be added

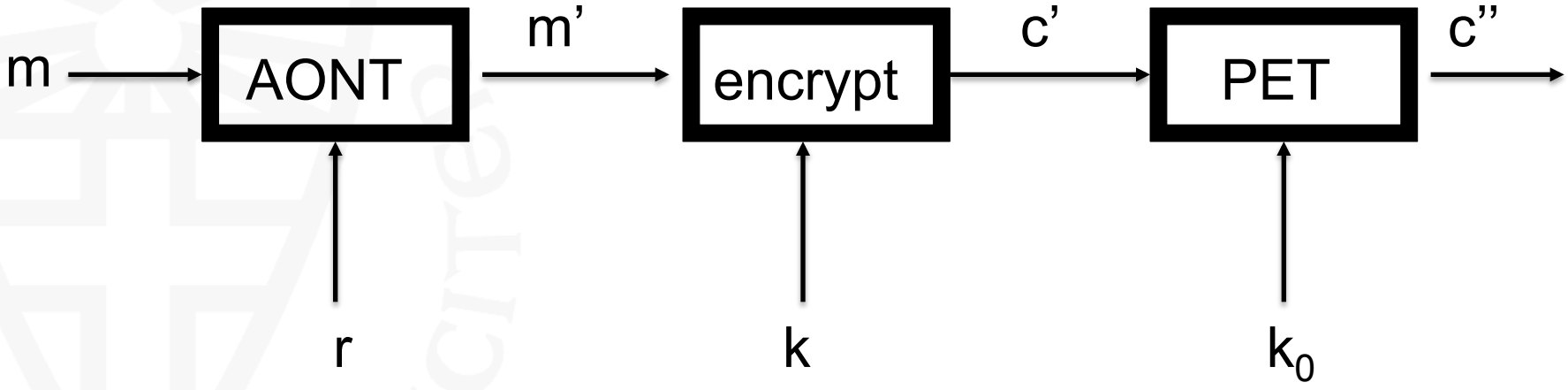
Protocol-level countermeasures

- Based on the fact that differential attacks need at least two encryptions with the same key and plaintext => randomize the key or the plaintext – works for DPA as well
- Disadvantages:
  - changes in the protocol are needed
  - only one party within a 2-party communication can be protected

Protocol-level countermeasures: 3 approaches

- All-Or-Nothing transforms
  - Originally intended as DPA countermeasure
- Message randomization
- Re-keying (helps also against DPA)

All-Or-Nothing Transform (AONT)



PET – post encryption transformation with pre-shared key k_0

Message randomization

Encrypt

$$m' = m \oplus r$$

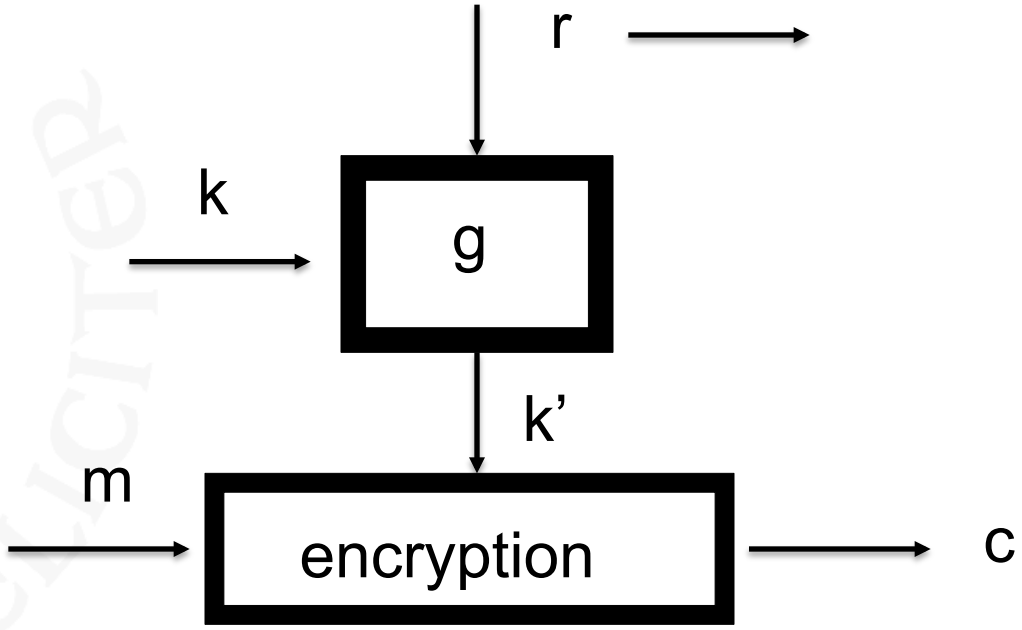
instead of m

DPA attacks are possible (because m' is known)

r – random but made public together with c

Fresh re-keying

- randomizing the key before every encryption



k'=g(k, r) – session key

SCA: Recent developments

- Theory
  - Framework for side-channel analysis
  - Leakage resilient crypto
- Theory and Practice
  - Even more advances in attacks: algorithm specific (combined with cryptanalysis)
  - Machine learning methods
  - Similar techniques apply to traffic analysis
  - New countermeasures
  - New models (going sub-micron)

Conclusions and open problems

- Physical access allows many attack paths
- Trade-offs between assumptions and computational complexity
- Requires knowledge in many different areas
- Combining SCA with theoretical cryptanalysis

Questions?