



Secure Product Lifecycle

Implementation Attacks on High-Security Devices

Christoph Herbst

WHEN YOU NEED TO BE SURE

SGS



PART I – From Security Target to Testplan

WHITEBOX VS. BLACKBOX TESTING

■ Blackbox

- Evaluator gets same (public) information as a consumer
- Unknown algorithm details and countermeasures
- Samples same/similar to product
- No further support from developer



GREY BOX

■ Whitebox

- Evaluator has in-depth knowledge about internals
- Algorithms and countermeasures declared and described in detail
- Samples prepared for testing (opened, added functionality)
- Support from developer
- → More common in high-security domains



WHITEBOX VS. BLACKBOX TESTING

- Scope: Use any measures to break the device
 - Social engineering (e.g. phishing)
 - Attack the DUT in an uncertified way (e.g. configuration)
 - Attack the IT environment of the TOE
 - Attack the development or production environment



- Scope: Asset clearly defined and separated
 - User behaviour is defined
 - Environment is defined
 - Configuration is defined
 - Security functionality is defined
 - Development and production environment is defined and certified



Sonstiges

Chaos Computer Club hackt den Personalausweis 2.0

24. Aug 2010

Chaos Computer Club hacks
German ID Card 2.0





■ What shall be tested?

- TOE (Target of evaluation)
- Secure configuration
- Test depth
- *Example: TD Smartcards and Similar Devices*



■ How is tested?

- Tools & procedures depending on TOE, e.g. for TD Smartcards and Similar Devices



■ When is the device broken?

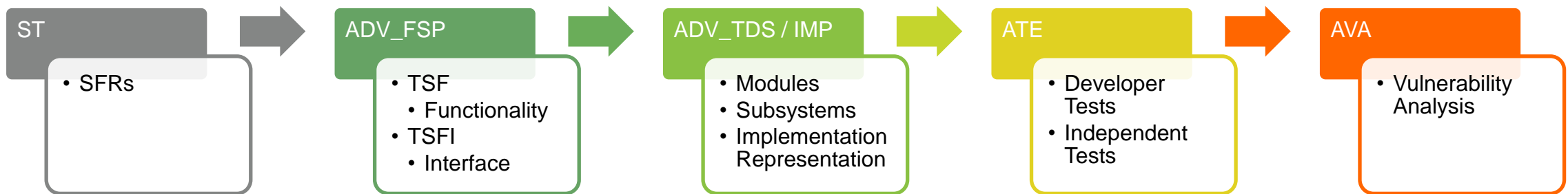
- If the white box attack is successful, is the device secure or not?
- What means secure?
- E.g. CC defines 'secure' as **TOE is resistant against an attacker with different attack potentials: no rating, basic, enhanced-basic, moderate or high**

EXAMPLE COMMON CRITERIA AVA – DUT

■ Security Target

- TOE is defined (Target of Evaluation == Device under Test)
- EAL level is defined
- Security Target: High level requirements (SFR) and services (SF) of the full security functionality
- Cryptographic security functionality is additional reflected in the Cryptographic mechanisms subsection / table

■ Simplified CC mapping between SFRs and AVA test:



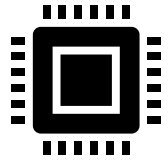
- Developer {
 - Security Functionalities
 - Design description / implementation / Source code / Samples etc.

- Common Criteria {
 - Test requirements
 - CC / CEM (International standards)
 - SOGIS / JIL (European standards)
 - AIS (each CB)

- ➔ Test plan, draft version 0.9
 - Kick-Off with CB, alignment on the test plan

- ➔ Test plan, initial version 1.0

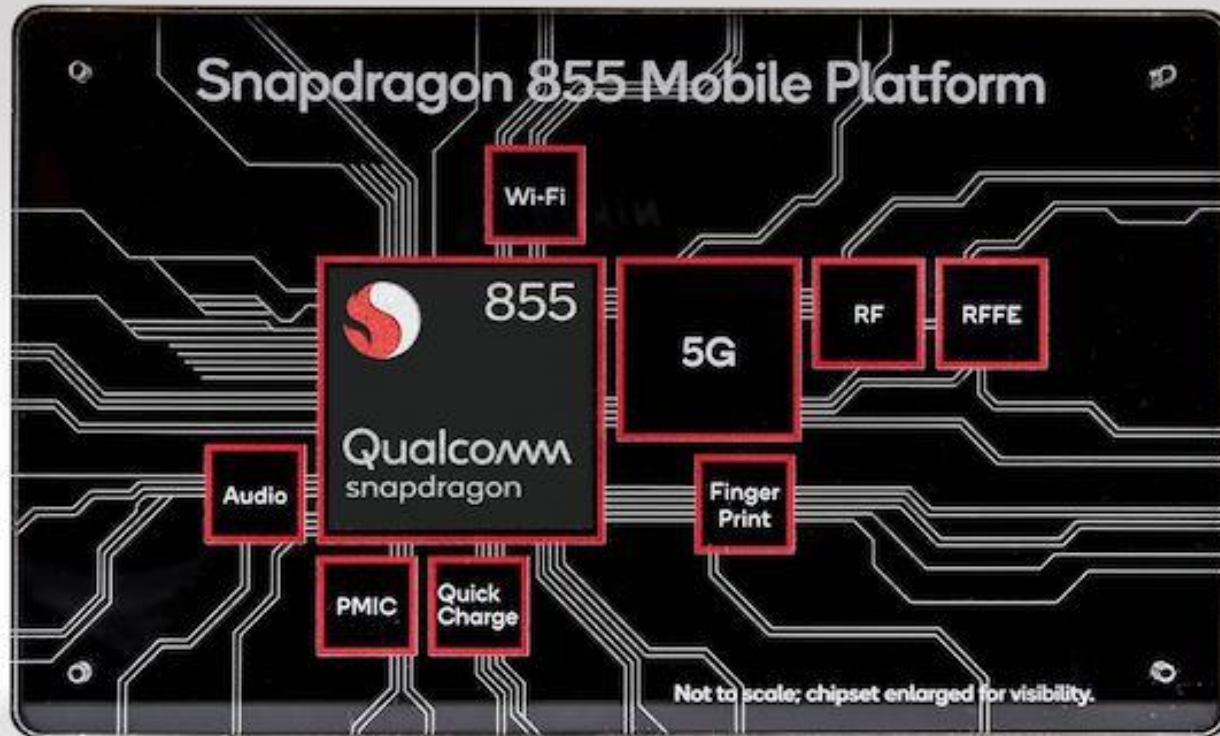
- ➔ Test plan updates while performing AVA testing



- Developer must provide design documentation, developer tests, source code etc. depending on the assurance level.
- Test Guidance for security standards
 - For the German scheme, the test guidance is defined by the German certification body (BSI) additionally to CC framework
 - AIS 46
 - ECC
 - RSA, DSA, Diffie-Hellman
 - Memory Encryption
 - AIS 20, AIS 31
 - RNG including
 - DRG – Deterministic random number generator
 - PTG – True physical random number generator
 - Hybrid random number generator

- Example: Minimum Requirements for Evaluating Side-Channel Attack Resistance of RSA, DSA and Diffie-Hellman Key Exchange Implementations Part of AIS 46 (1)
 - Chapter 2: Side-channel analysis
 - Chapter 3: Modular exponentiation (RSA, DSA, DH) // Chapter 4: RSA etc.

Identifier	Precondition	Category / Target	Short description / Remark	Reference
RC_7 / 'Bellcore Attack'	Modular exponentiation of <ul style="list-style-type: none"> • unknown but constant M • with private d, p, q One correct RSA CRT ($\rightarrow S$) and one faulty RSA CRT with fault on one of S_p or S_q computation ($\rightarrow S'$) or on $q^{-1} \bmod p$ Faulty signature is detectable	Logical SCA; Differential FA / Prime modulus of exponentiation	Without loss of generality, hardware fault occurs during computation of S'_p (i.e. $S_p \neq S'_p \bmod p$) but no fault occurs during computation of S'_q (i.e. $S'_q = S_q \bmod q$). Applying recombination on S'_p and S'_q gives faulty signature S' for M . Then $\gcd(S - S'N) = q$ / Simple countermeasures are not sufficient against practical attacks (see [ABF02]). A faulty S'_p could be also induced by a fault during loading the prime modulus p. A fault on $q^{-1} \bmod p$ also provides a faulty S' after the recombination step.	[BDL96] [BDL01] chapter 2.2 [ABF02]

EXAMPLE TOE
SNAPDRAGON 855 MOBILE PLATFORM (SOC)

SNAPDRAGON 855 MOBILE PLATFORM (SOC) EXAMPLE TOE

**Multigigabit everything**

The world's first commercial 5G mobile platform unleashes transformative 5G experiences with the Snapdragon X50 5G modem while harnessing multi-gigabit 4G connectivity. Combined with a new era of Wi-Fi performance, this is innovation for the next generation.

**Cinema-grade videography**

From bleeding-edge color to a new evolution of cinema-grade videography, this platform sets brilliant new standards for mobile – including the world's first Computer Vision (CV) ISP delivering 4K HDR video capture with Portrait Mode and world's first to capture HDR10+ video.

**Powerful, intuitive experiences**

Snapdragon 855 is powered by new architecture improvements and leading 7nm process technology, a giant leap in CPU smartphone performance, and a Hexagon processor that's purpose built for AI. This platform packs unimaginable performance and power efficiency for the next echelon of 5G, AI and XR. It delivers powerful, intuitive experiences in the tiniest of packages – all while providing what matters most: insane battery life.

**Captivating entertainment**

Go big, even when you're not at home. The Snapdragon 855 steals the show with 4K HDR video playback, your motion pictures will be brilliant, scene-by-scene and frame-by-frame. 5G makes XR real time, so fully immersive interactions happen at the speed of life. Plus, Snapdragon Elite Gaming equates to multi-player experiences that mimic human reflex. Game on.

**AI to the 4th Power**

Our 4th generation on-device AI engine is the ultimate personal assistant for camera, voice, XR and gaming – delivering smarter, faster and more secure experiences. Utilizing all cores, it packs 3 times the power of its predecessor – providing stellar on-device AI capabilities.

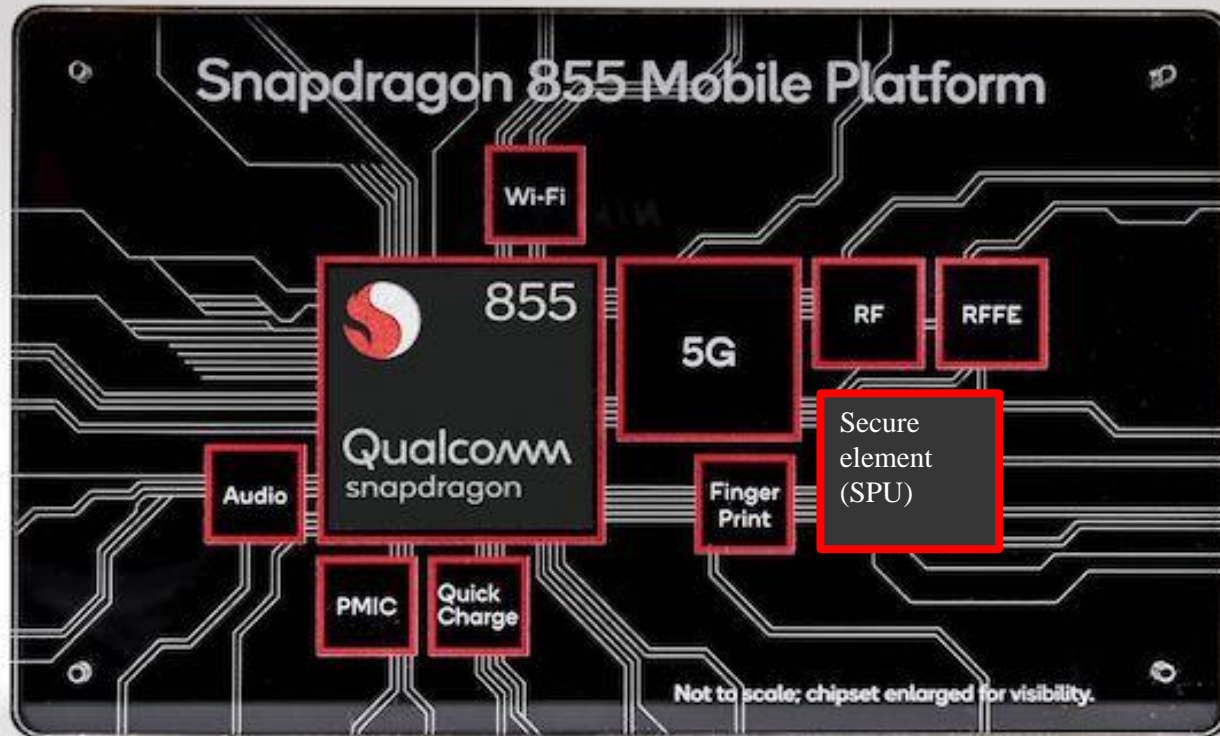
**Cutting edge connectivity**

The Qualcomm® FastConnect™ 6200 subsystem delivers a new era of wireless connectivity with Wi-Fi 6 innovations for new and enhanced mobile experiences, as well as integrated Bluetooth 5.1 with advanced low latency and audio capabilities.

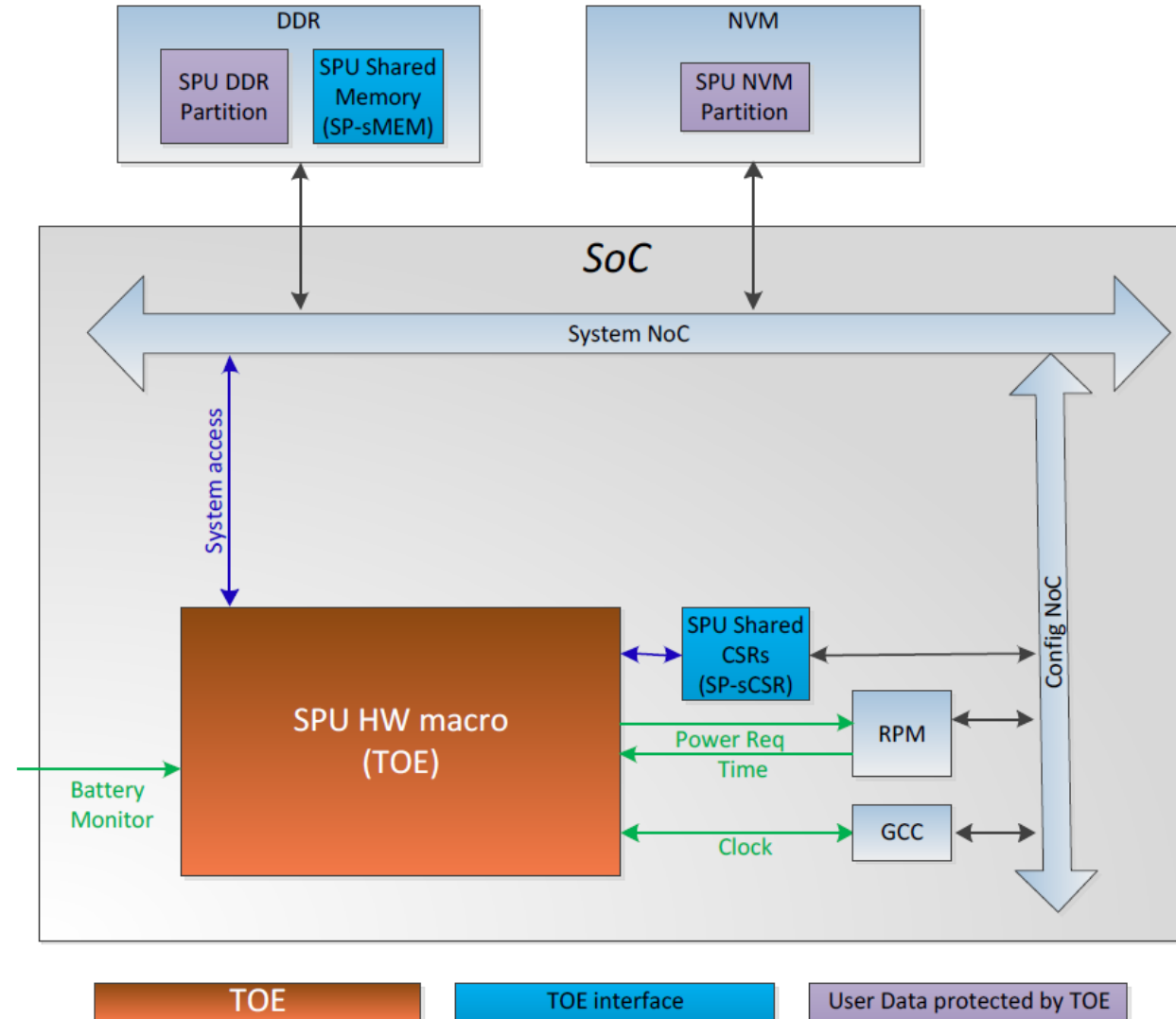
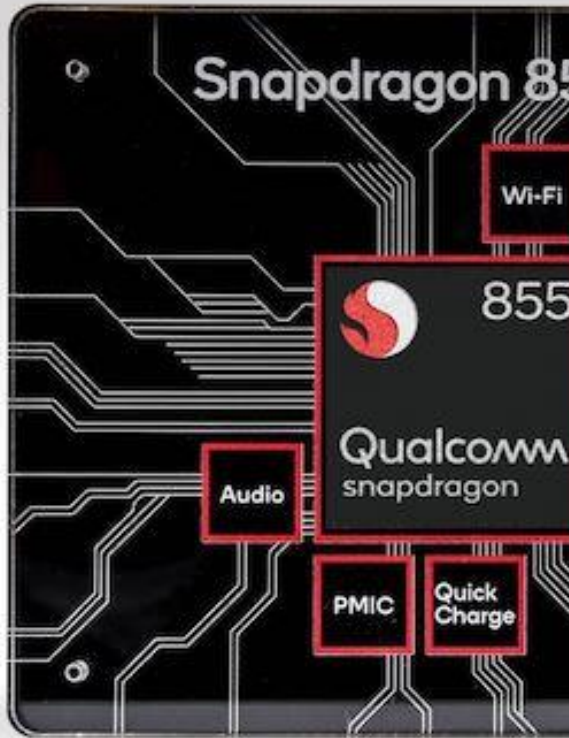


- Security Target: Qualcomm® Secure Processing Unit SPU230 Core Security Target Lite, 80-NU430-6 Rev. B May 3, 2019
- The target of evaluation (TOE) is the secure processing unit (SPU) subsystem serving as a secure element within a package system-on-chip (SoC).
- The TOE is an independent subsystem that is integrated in a system-on-chip (SoC) in a manner that is agnostic to the hardware and software implementation details. The TOE serves as an independent root of trust within the SoC. **It does not rely on any external entity for any security enforcement, allowing it to be evaluated as a separate entity.** It has its own ROM code for secure boot operations.

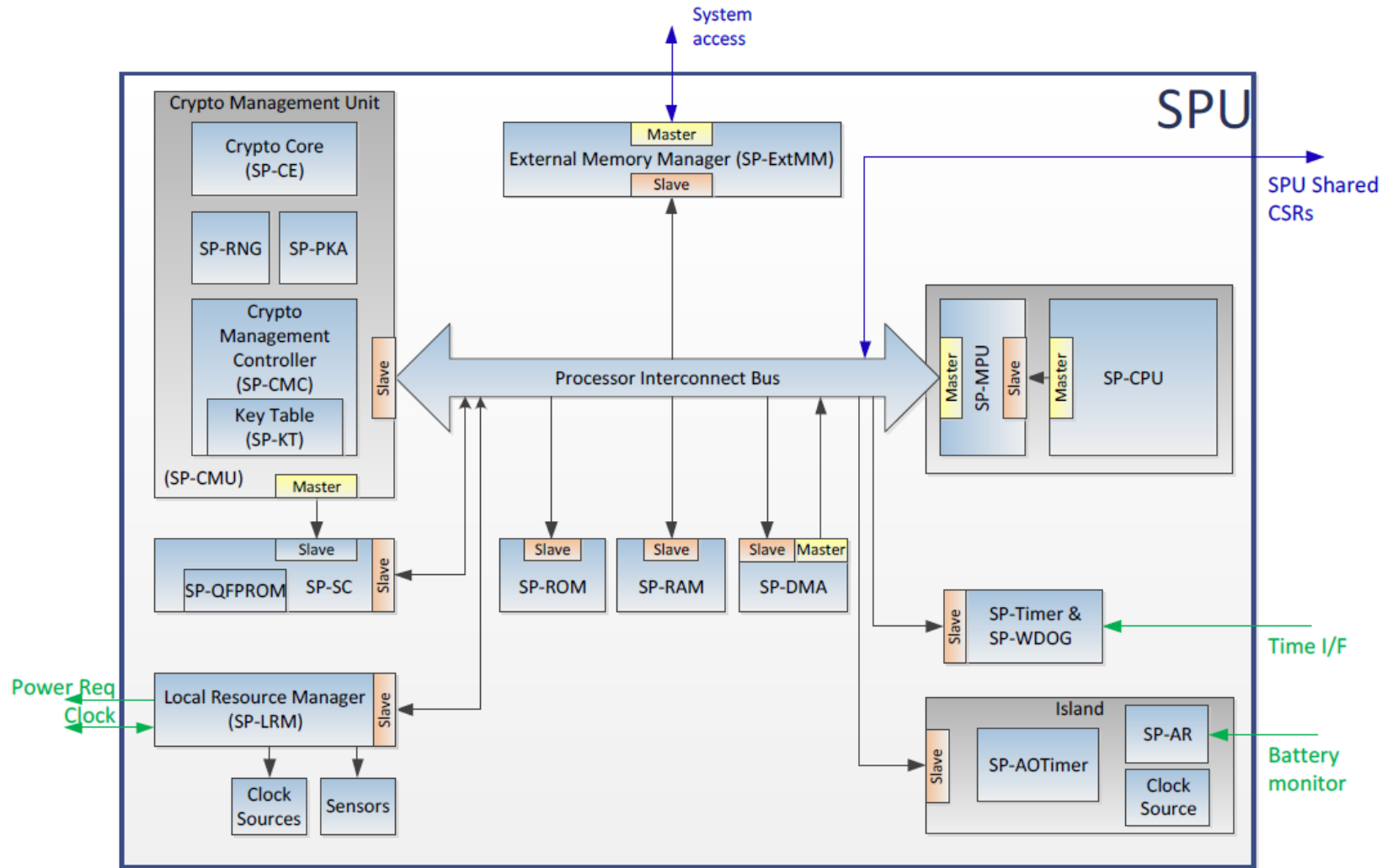
EXAMPLE TOE HARDWARE SEPARATION



EXAMPLE TOE HARDWARE SEPARATION

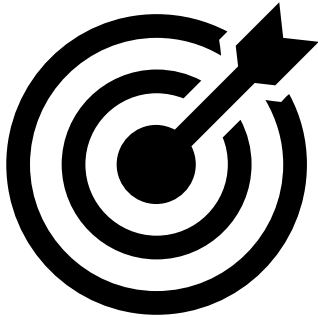


EXAMPLE TOE HARDWARE SEPARATION





- 3 out of 36 SFRs
 - FCS_COP.1/AES – Cryptographic operation – AES
 - The TSF shall perform encryption and decryption and authentication when using CCM mode in accordance with a specified cryptographic algorithm AES in ECB mode, CBC mode, CTR mode, CCM mode and cryptographic key sizes 128-, 256-bit that meet the following: Specification for the ADVANCED ENCRYPTION STANDARD (AES) (FIPS PUB 197), Recommendation for Block Cipher Modes of Operation, Methods and Techniques (NIST SP 800-38A), Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality (NIST SP 800-38C).
 - FCS_CKM.4/AES – Cryptographic key destruction – AES
 - The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting with zeros or overwriting the protecting key encryption key in the key hierarchy with zeros that meets the following: none.
 - FPT_PHP.3 Resistance to physical attacks
 - The TSF shall resist *physical manipulation and physical probing* to the TSF by responding automatically such that the SFRs are always enforced



■ Assessment



- EAL.4 augmented by AVA_VAN.5
- AES, SHA, HASH, RNG
- Side-Channel and fault injection attacks
- Reverse engineering

■ Out-of-scope



- Attacks on any other functionality or with non-certified configuration
- EC cryptographic operations
- No assessment of any feature like 5G, AI or other
- No Pentesting on network interfaces
- ...

WHITE BOX TEST PLAN



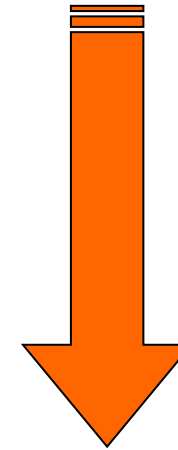
- Security Features and EAL defined by the developer
- Developer provides white box test interfaces
 - Documentation, Samples, Support
- Clear scope of testing including detailed rules
 - CB ensures that those are sufficient and up-to-date
- Rating scheme for vulnerabilities defined
 - Flexible for different attack potentials
- Test plan can be detailed a priori to the assessment
- Test plan can be updated a posteriori during the assessment

PART II – From Testplan to Attack Potential

- Part 2 – Practical evaluation work
 - Starting from an initial version of the testplan
 - Step-by-step through a typical analysis flow
 - Details on certain aspects
 - Attack potential rating

TYPICAL ANALYSIS FLOW

1. Implementation Analysis
2. Modelling of the Attacked Algorithm
3. Identification of Samples
4. Preparation of Samples
5. Establishing a Communication Channel
6. Bypassing Countermeasures
7. Target Identification
8. Worst Case Analysis
9. Attack Paths & Methods
10. Attack Rating



1. IMPLEMENTATION ANALYSIS

■ Inputs:

- Security functional requirements (crypto table)
- Gathered information from document work
 - E.g. HW/SW countermeasures
 - TSFI → interfaces
 - Parameters
 - Options (configuration)
- Implementation representation (source code/design files)
- Workshops with developer
 - Clarify what is not understood well enough
- Guidelines from Standards, papers, etc.
 - What shall be tested? How shall be tested (rather high-level)?
 - → ensure certain quality level, independent of ITSEF lab
- Inputs from other evaluation steps
- Early version of the test plan



1. IMPLEMENTATION ANALYSIS

■ Gathered information

- Targeted algorithm and how to trigger it (efficiently)
- Floor plan (ideally location on chip)
- Implementation details of the algorithm
- Environmental condition boundaries
- Available clock frequencies
- Interrupt or polling behavior
- Configuration options
 - min. security level allowed by guidance
 - max leakage, vulnerability to attacks
 - countermeasures
- etc.



1. IMPLEMENTATION ANALYSIS

- Output → refined testplan
 - For each function in scope (TSF/TSFI, security claim)...
 - Ideally divides a complex function into smaller parts (still need to evaluate and rate as a whole)
 - List of tests covering different aspects (attack paths)
 - And covering different countermeasures
 - Ideas and strategies how to test (e.g. functions to call, bypass countermeasures)





2. MODELLING OF THE ATTACKED ALGORITHM



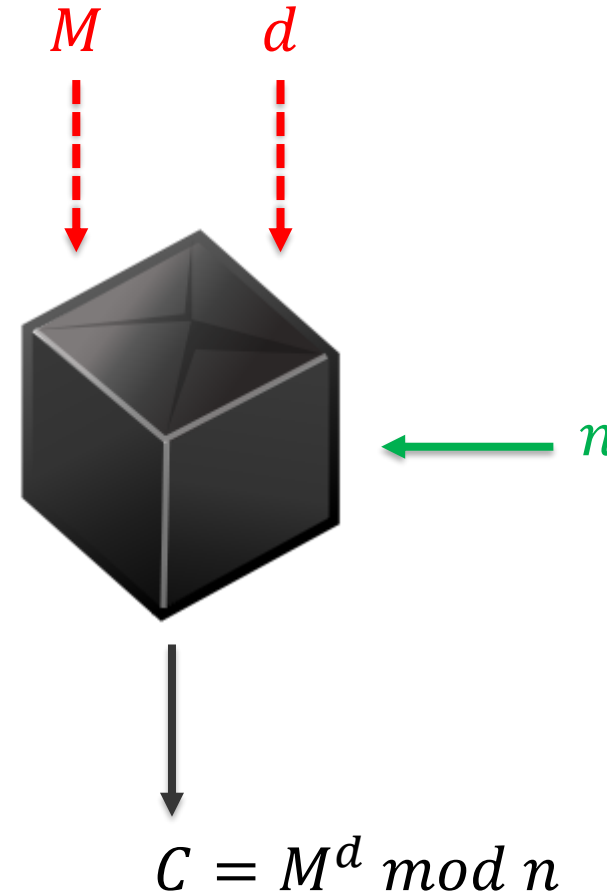
© Photo recreation by Bryan Beasley

- What you read and understand \neq what is implemented (often not even what is written)
- Idea: To implement something you need a certain level of understanding.
- Goals:
 - Gain enough information to analyze and attack algorithm
 - Make decision what parts are most vulnerable
→ select target, e.g. intermediate
 - Calculate required intermediates (SCA/FI)
 - Create a model at an appropriate abstraction level (suitable for attacks)

2. MODELING OF RSA EXPONENTIATION EXAMPLE

■ High-level functional description

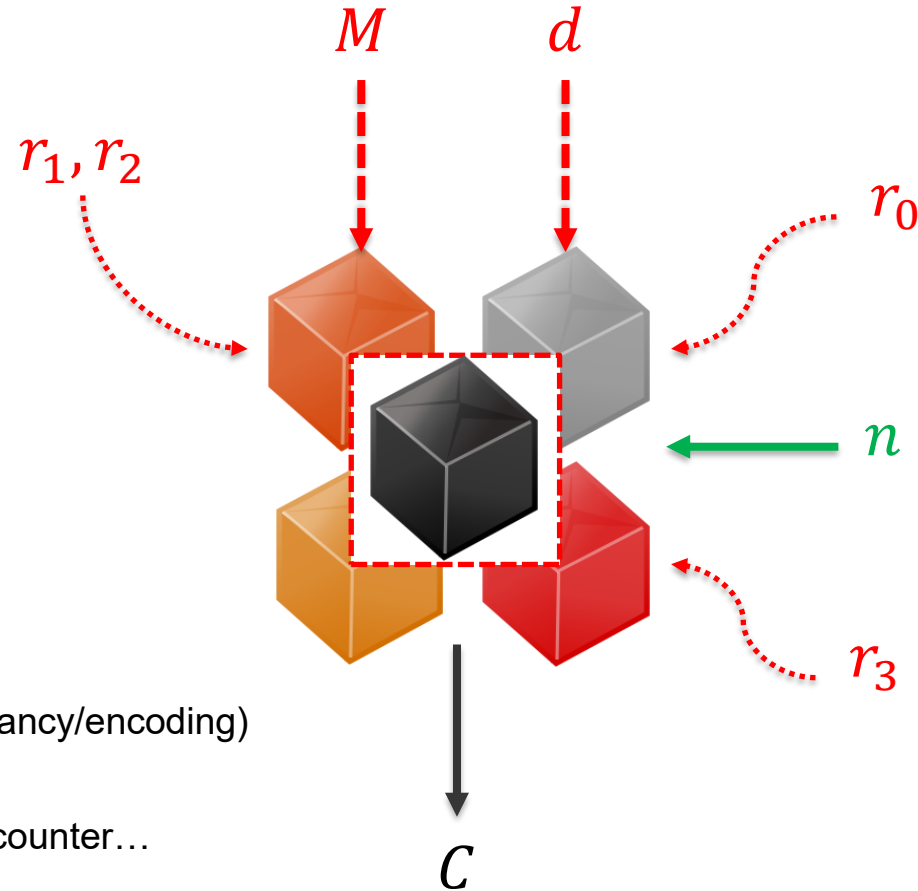
- Inputs
 - M ... plaintext (secret)
 - d ... private exponent (secret)
 - n ... modulus (public)
- Outputs
 - C ... ciphertext (public)
- Implemented functionality
 - $C = M^d \bmod n$
- too inaccurate for any meaningful SCA/FI



2. MODELING OF RSA EXPONENTIATION EXAMPLE

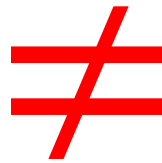
■ In practice

- RSA variants
 - Straight-forward variant
 - Chinese Remainder Theorem (CRT)
- Countermeasures in software
 - Exponent blinding: $d_b = d + r_0 * \varphi(n)$
 - Message blinding:
 - $A_0 = 1 + r_1 n \text{ mod } r_2 n$
 - $A_1 = M + r_1 n \text{ mod } r_2 n$
 - Regular exponentiation (Montgomery Ladder):
 - *foreach*(db_i in d_b):
 - $A_{\overline{db_i}} = A_0 A_1 \text{ mod } n$
 - $A_{db_i} = A_0 A_1 \text{ mod } n$
 - Modulus blinding: $w_b = n \times r_3$
 - ..., Exponent splitting, FI countermeasures (redundancy/encoding)
- Countermeasures in hardware
 - e.g., dummy operations, clock jitter, sensors, error counter...
- Do we need to model all this? → No ☺



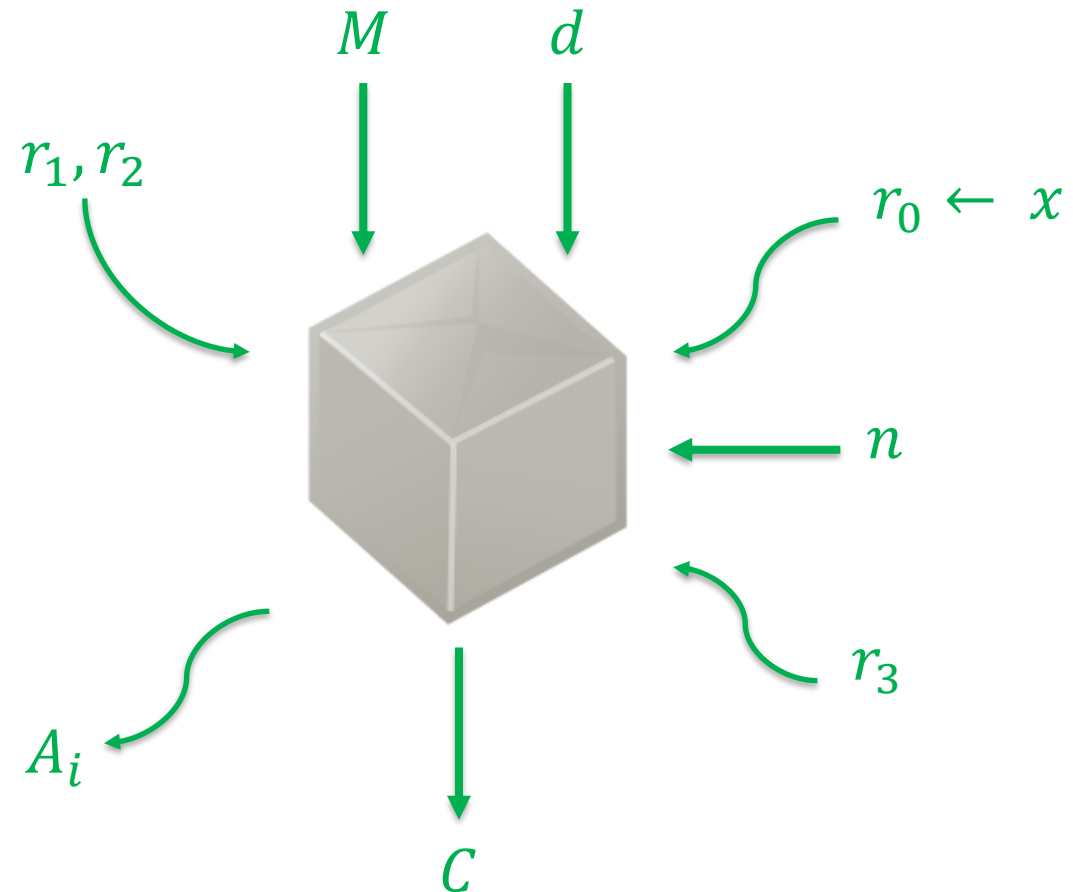
AN EVALUATOR IS NOT AN ATTACKER

- Remember we are in white-box scenario!
- → „Do what only an evaluator can...“



2. MODELING OF RSA EXPONENTIATION EXAMPLE

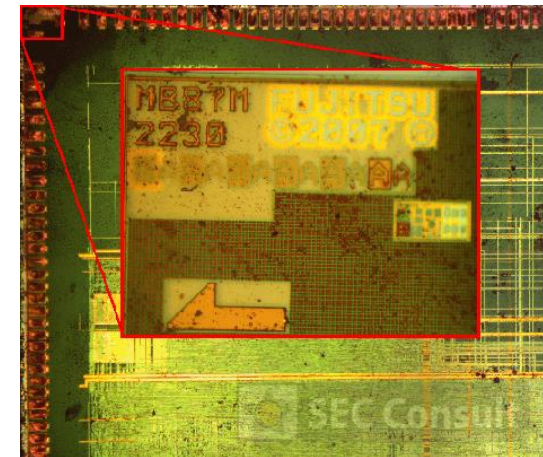
- Ideally a white box evaluator can...
 - Get special interfaces from developer
 - Access & test subcomponents individually
 - E.g. Montgomery Ladder
 - Control and observer
 - Inputs
 - Countermeasures (enable/disable)
 - Randomness
 - Intermediates
 - Outputs
- In practice ...
 - Not everything can be (directly) controlled/disabled



3. IDENTIFICATION OF SAMPLES

- Make sure what you are going to test is what shall actually be tested
- Different TOEs provided different methods of identification
 - See AGD (guidance) → customers need to be provided with a way to ensure they use the correct product and version
- Software TOE
 - Version number return from library function
 - Hashing over SW components
- Hardware TOE
 - Identification plate on chip die
 - or on package

```
# openssl version  
OpenSSL 0.9.8r 8 Feb 2011 (Library: OpenSSL 0.9.8o 01 Jun 2010)  
#
```



<https://sec-consult.com/en/blog/2019/02/reverse-engineering-architecture-pinout-plc/>
→ Interesting article

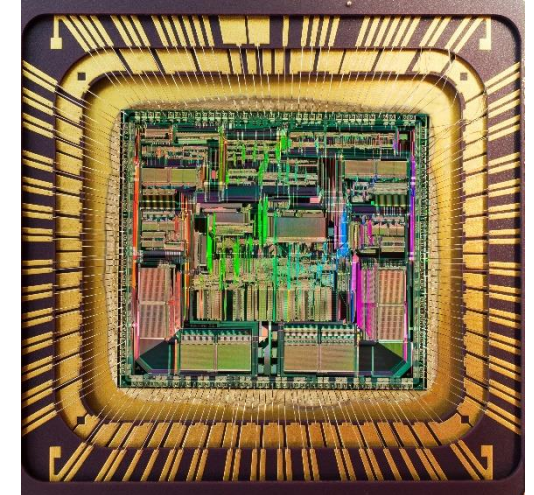
4. PREPARATION OF SAMPLES

- Setup according to guidance!
 - If tests performed on wrong configuration → meaningless
- Sample ready to be tested (efficiently)?
 - Collect > 5 Millions of side-channel traces?
 - Perform multitude of faults?
- White-box evaluation:
 - Triggers in place? → or alternatives?
 - Configuration of countermeasures
 - Chip opened (e.g. access to front or backside)
- Black-box evaluation:
 - Manual deactivation or bypassing of countermeasures
 - Hardware analysis and reverse engineering tasks
 - Decapsulation, thinning, ...



Tools

- Grinding
- Etching
- ASAP
- Microscope
- FIB
- SEM
- ...



- Localization of countermeasures (sensors) and other die features

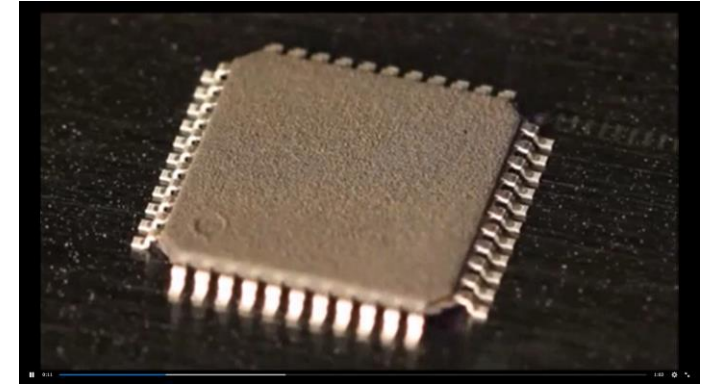
- Deactivation of countermeasures (FIB)

- Sample preparation for FI

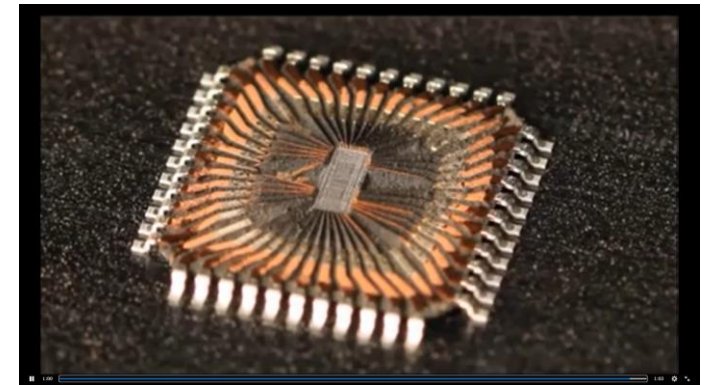
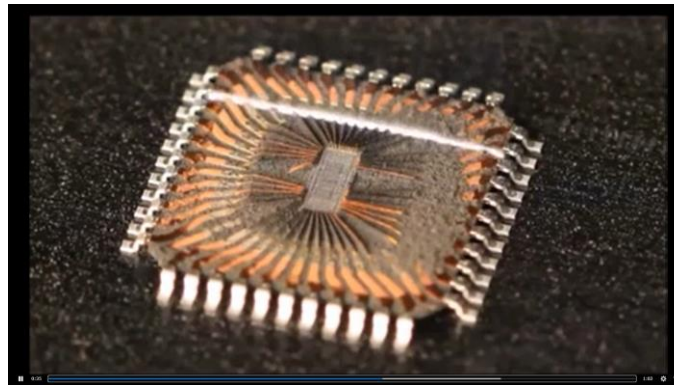
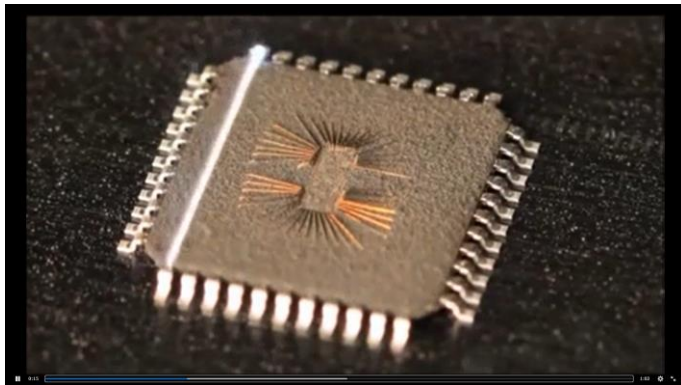
- Decapsulation (remove package where needed)
- Thinning of substrate (backside)
- Polishing of surface



Mostly needed for black-box\grey-box analysis

LASER DECAPING OF CHIP
(FRONT SIDE EXAMPLE)

Source: https://www.reddit.com/r/Damnthatsinteresting/comments/ci08vx/a_semiconductor_chipic_getting_decapped/

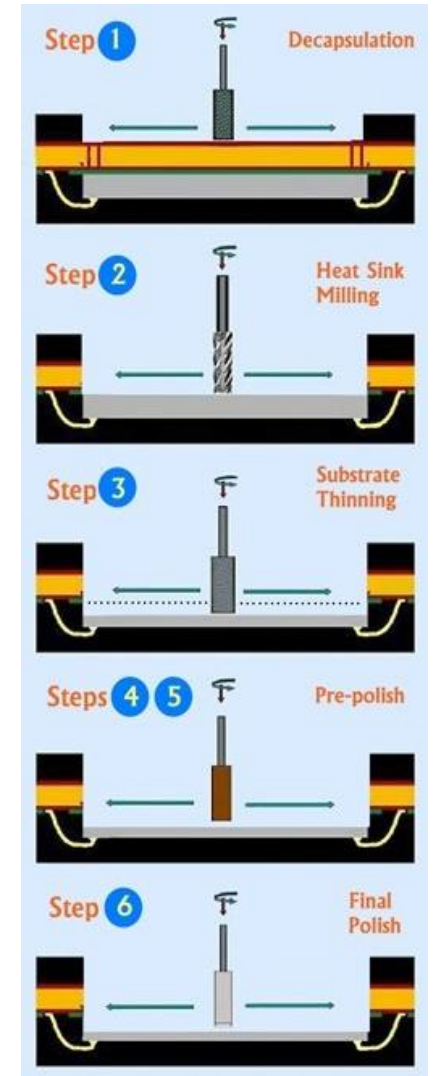
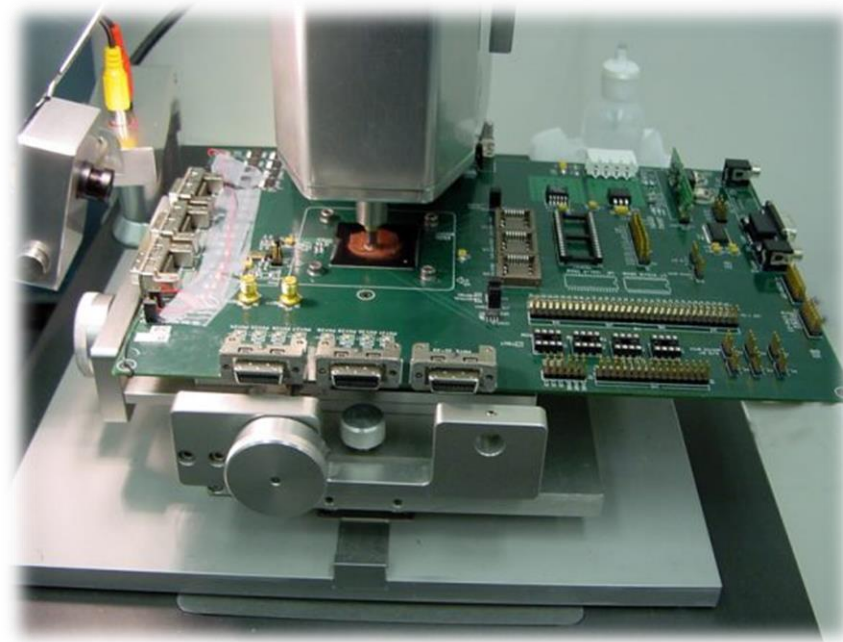


Laser decapsulation makes surface too rough for direct use for LFI!

→ Mirror finishing with ASAP (fancy CNC milling machine)

ANALOG SELECTED AREA PREPARATION (ASAP)

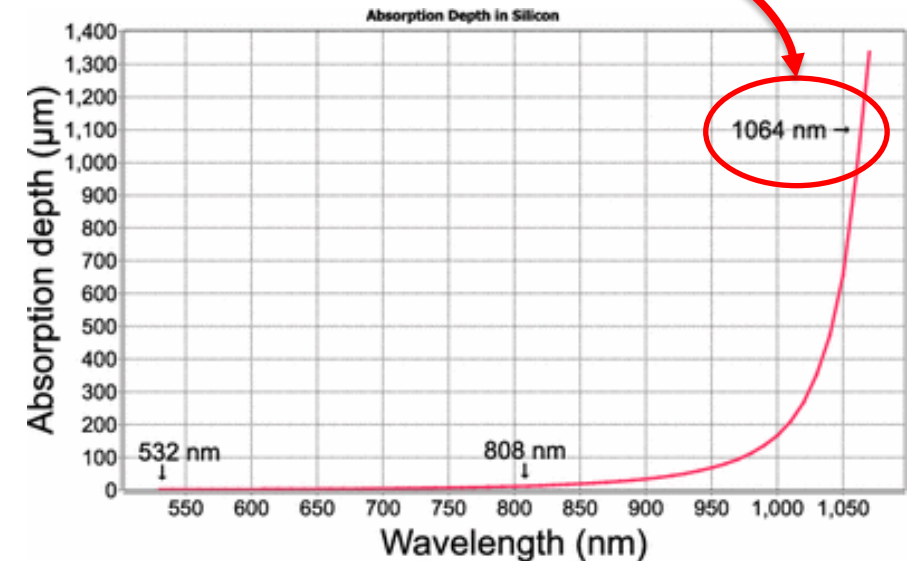
- Decapping and/or polishing
- Goal: thin the substrate from e.g. 925 μm to 50 μm



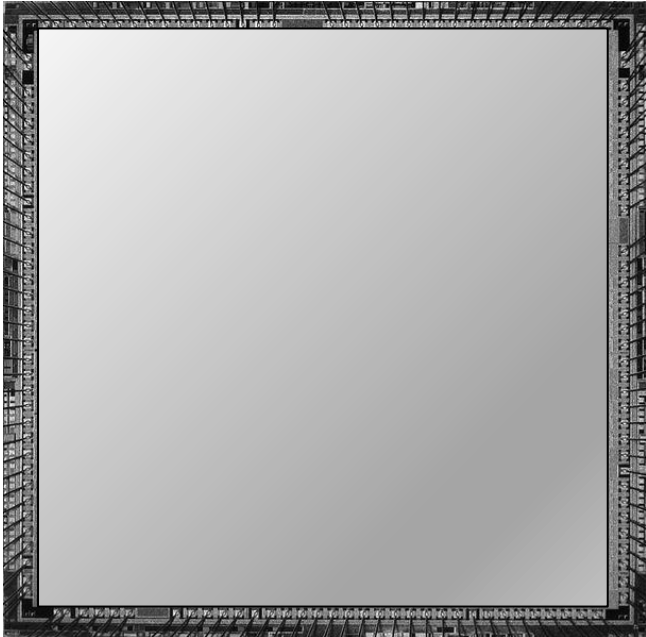
Source: <https://www.ultratecusa.com/product/asap-1/>

WHY THINNING OF CHIP IS REQUIRED?

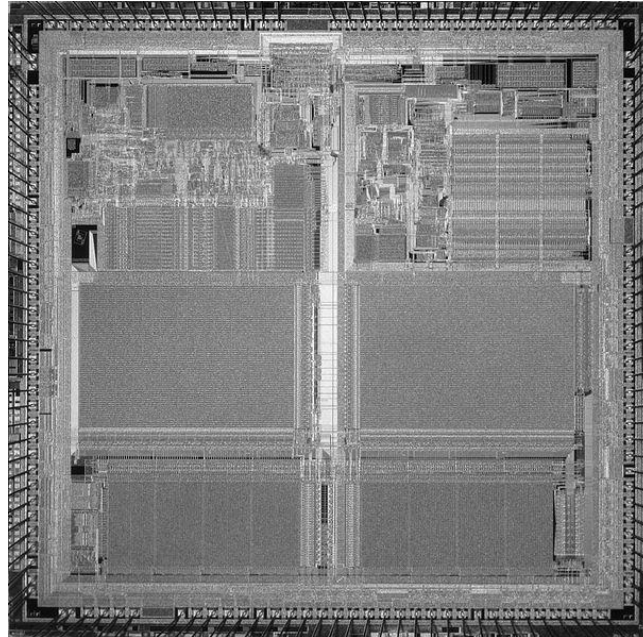
Whole laser beam absorbed at original substrate thickness of 1mm!



He W., Breier J., Bhasin S., Jap D., Ong H.G., Gan C.L. (2016) Comprehensive Laser Sensitivity Profiling and Data Register Bit-Flips for Cryptographic Fault Attacks in 65 Nm FPGA. In: Carlet C., Hasan M., Saraswat V. (eds) Security, Privacy, and Applied Cryptography Engineering. SPACE 2016. Lecture Notes in Computer Science, vol 10076. Springer, Cham. https://doi.org/10.1007/978-3-319-49445-6_3



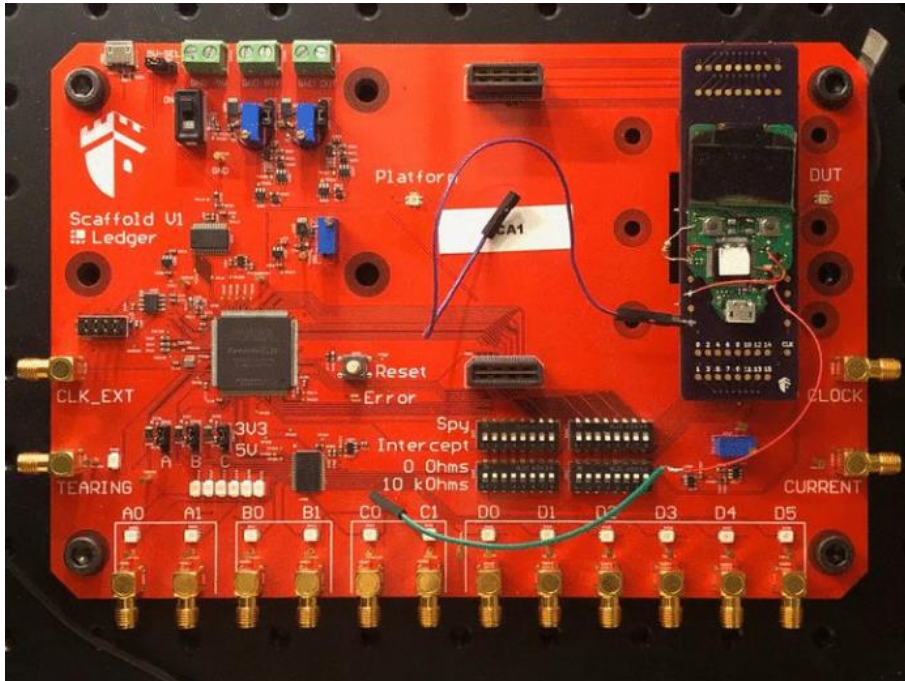
Backside IR image w/o thinning of substrate



Backside IR image with thinning of substrate (50-100 µm)

→ Backside LFI will not work w/o thinning because whole energy is absorbed by the substrate

5. ESTABLISHING A COMMUNICATION CHANNEL



Donjon Scaffold Board by Ledger:
<https://github.com/Ledger-Donjon/scaffold>

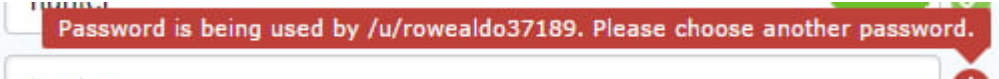
- Tested devices (e.g. packaged chips) often not designed to be used stand-alone → interfaces not very user-friendly
- Or there might exist faster ways to trigger functionality over non-standard interfaces (e.g. debug ports, or dedicated testing interfaces)
- Requires either:
 - Dedicated hardware
 - Versatile hardware (FPGA-based approaches)
- Setting up a communication for a new TOE can take a significant amount of time

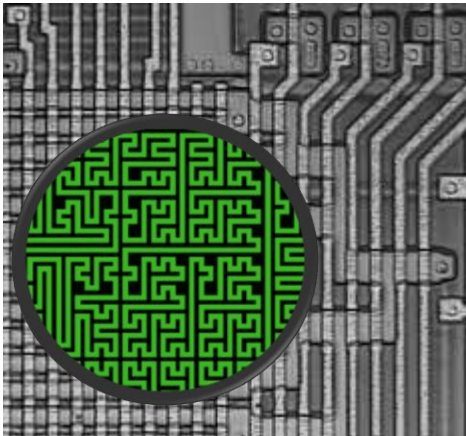
6. BYPASSING COUNTERMEASURES

- Deactivation of optional countermeasures (guidance)
- Configuration of accessible countermeasures
 - e.g. use 64 dummy rounds (min. setting) instead of 256 max. available
- White-box evaluation:
 - Control over countermeasures
 - e.g. set randomness to 0 or read random bits
 - Needs to be considered in attack rating
 - → not always possible
- Black-box evaluation:
 - Manual deactivation/bypassing
→ HW Reverse Engineering



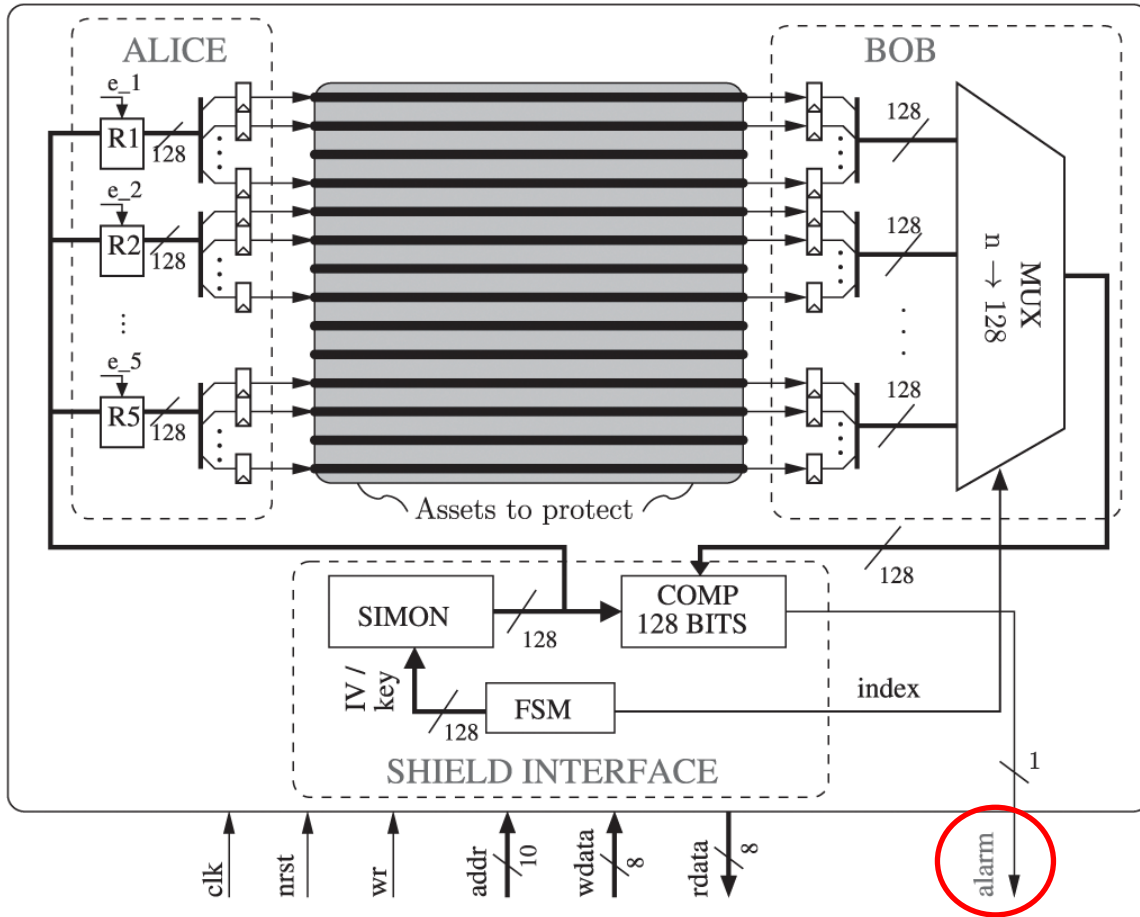
TYPICAL COUNTERMEASURES

- Implemented (ideally) on all/various abstraction levels
 - Architectural
 - Layout/memory scrambling
 - Sensors (rail, light, temperature, etc.)
 - Secure CMOS: WDDL/DPL cells (never perfect, and big)
 - Obfuscation
 - Logical
 - Secure error messages → 
 - Silent after errors
 - Redundancy, e.g. encrypt/decrypt, encoding/decoding, modular redundancy
 - Error counters
 - Algorithm
 - Constant-time algorithms, e.g. Simple Modular Exp. vs Montgomery Ladder
 - Masking /Blinding
 - Hiding: Dummy operations, shuffling, increase noise
 - Physical
 - Active/passive shielding (using the first metal layers to cover front side)
 - Secure routing (sensitive wires short and packed in between)



Shielding

ACTIVE SHIELDING EXAMPLE



- Protects against
 - Probing from frontside
 - LFI from frontside
 - Circuit manipulation (FIB)
- Usually combined with passive shielding
 - → visually block frontside
- Only front-side protection

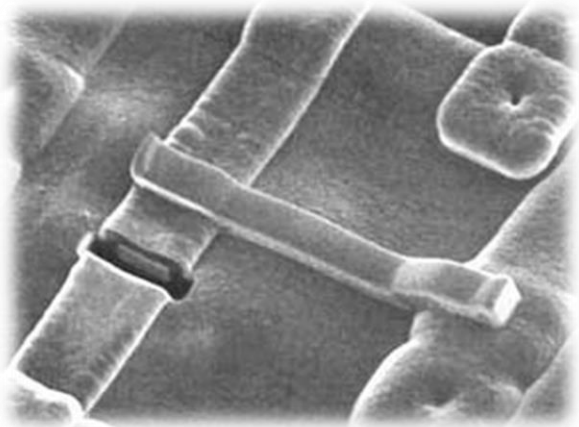


Xuan Thuy Ngo, Jean-Luc Danger, Sylvain Guilley, Tarik Graba, Yves Mathieu, Zakaria Najm, Shivam Bhasin:
Cryptographically Secure Shield for Security IPs Protection. IEEE Trans. Computers 66(2): 354-360 (2017)

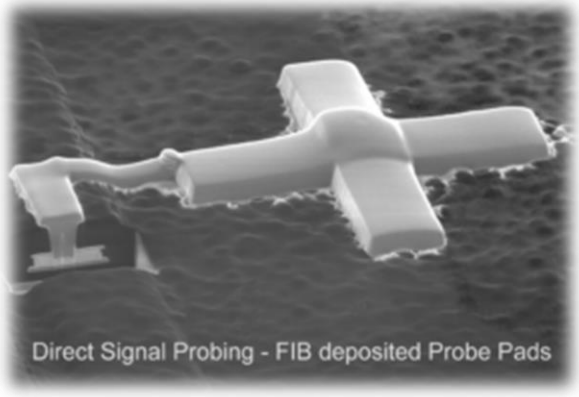
Chip probing: <http://cas.ee.ic.ac.uk/research/photos5.htm>

CHIP PROBING EXAMPLE FROM OUR LAB





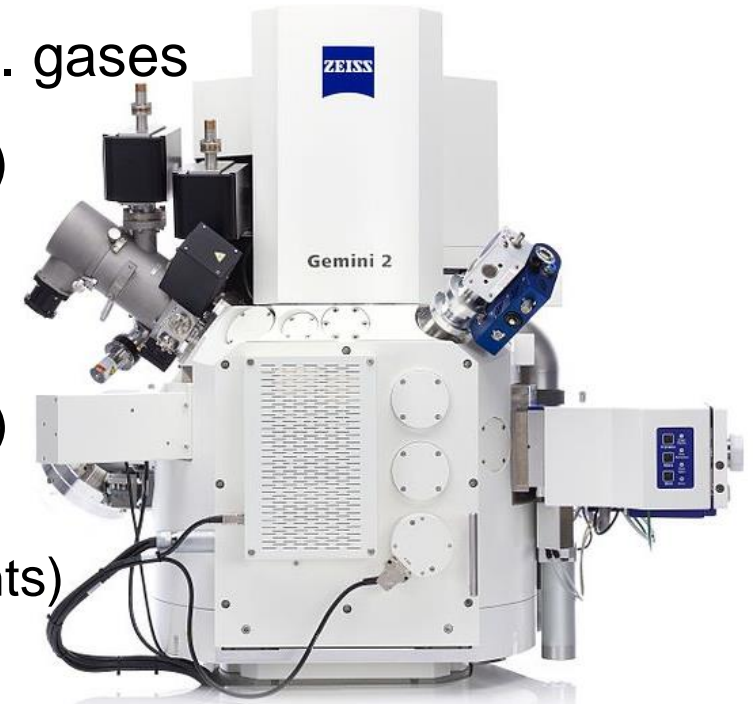
<http://www.spade.ust.hk/introduction/FIB.html>



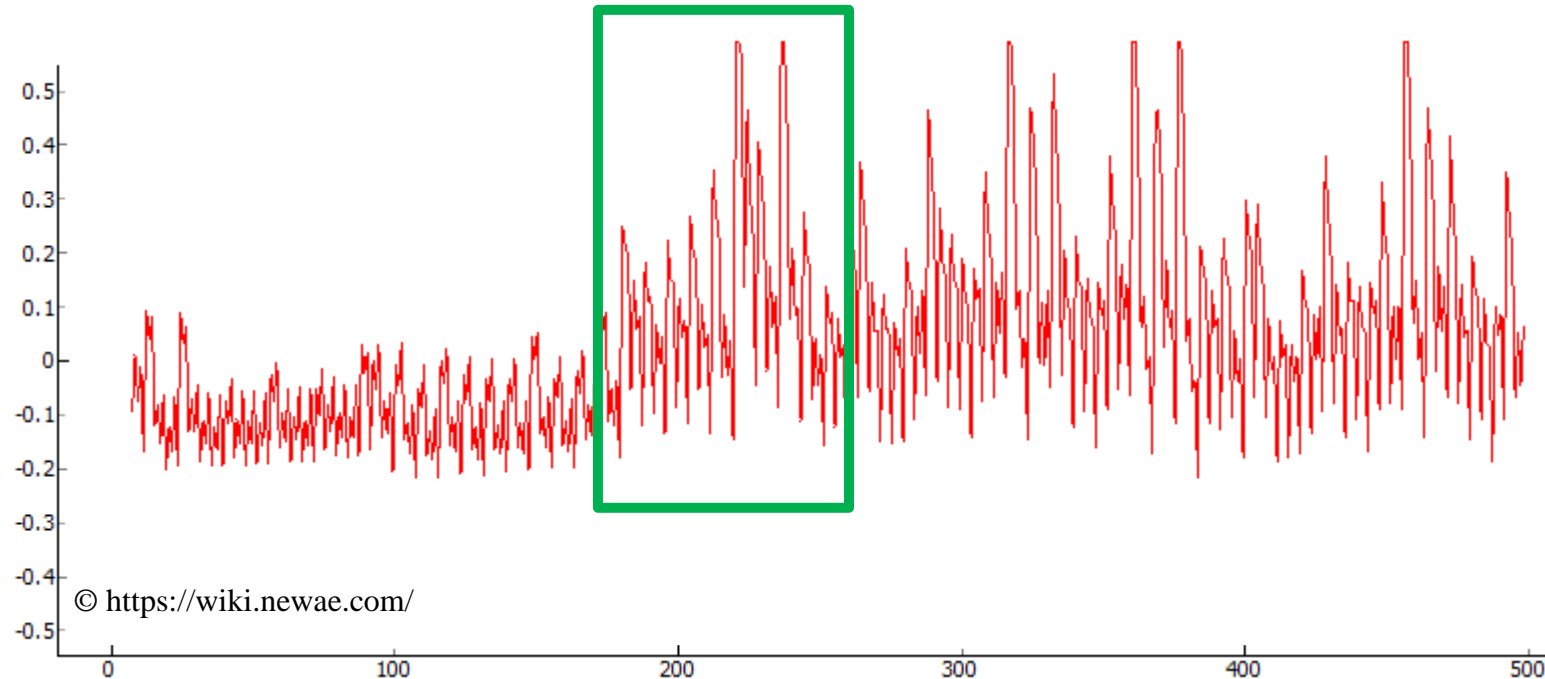
Direct Signal Probing - FIB deposited Probe Pads

<http://www.ic-crack.com/...>

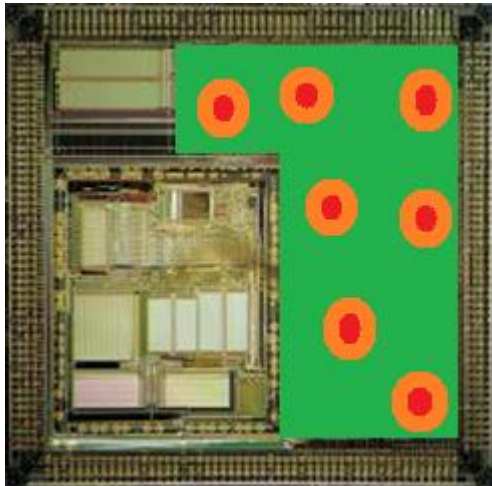
- Uses e.g. Gallium (Ga^+) ions + div. gases
- Combined with visualization (SEM)
- Abilities
 - Nanometer precision
 - Cut through material (wires/gates)
 - Remove material
 - Add material (rewire, probing points)
- Used for:
 - Deactivating countermeasures (sensors, shields, RNGs)
 - Create probing points
 - Rewire circuit,



- Example: Sensors (temperature, light, rail...) combined with error counters
- Fault counter needs to update non-volatile memory (e.g. FLASH)
→ requires higher voltages/energy to write
- React on certain EM/power pattern → Rapidly drain voltage supply



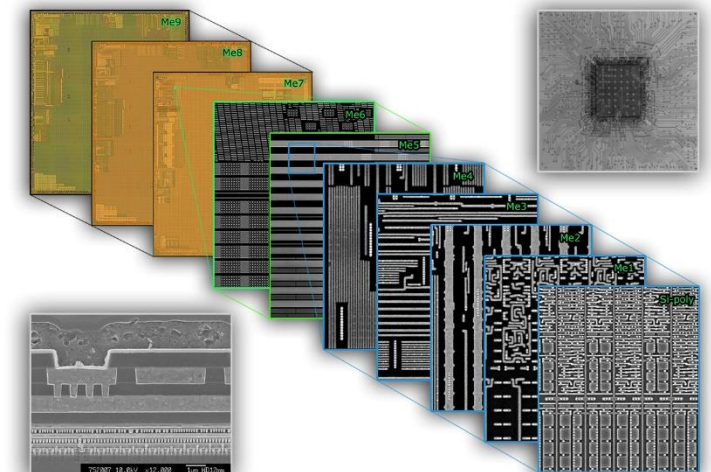
- Alternative approach: Find out where not to shoot 😊
 - a) Sacrifice some samples (LFI)
 - b) Passive imaging technique (OBIC)
 - c) Reverse engineering (delayering)



a) LFI Approach



b) OBIC approach¹



c) Delayering

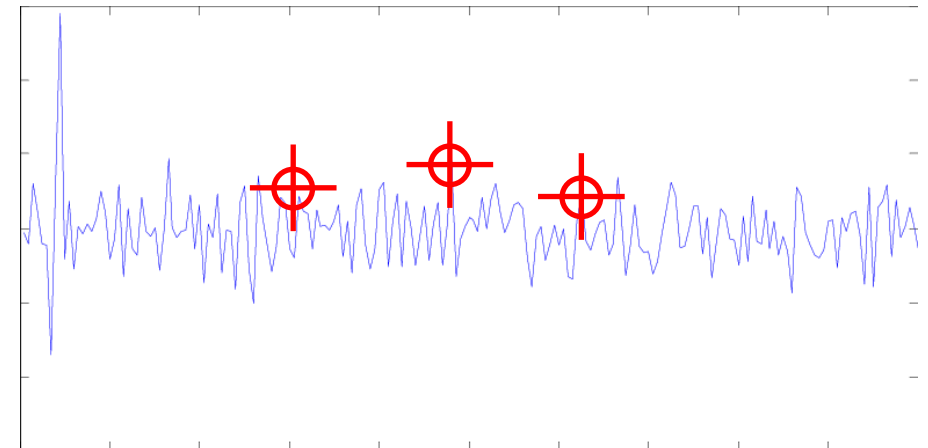
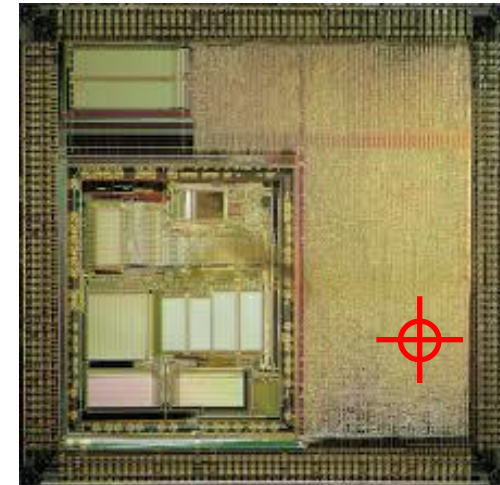
© <http://www.ic-crack.com/>

1) Picture taken from "On the Complexity Reduction of Laser Fault Injection Campaigns Using OBIC Measurements", Schellenberg et al., FDTIC 2015: 14-27.

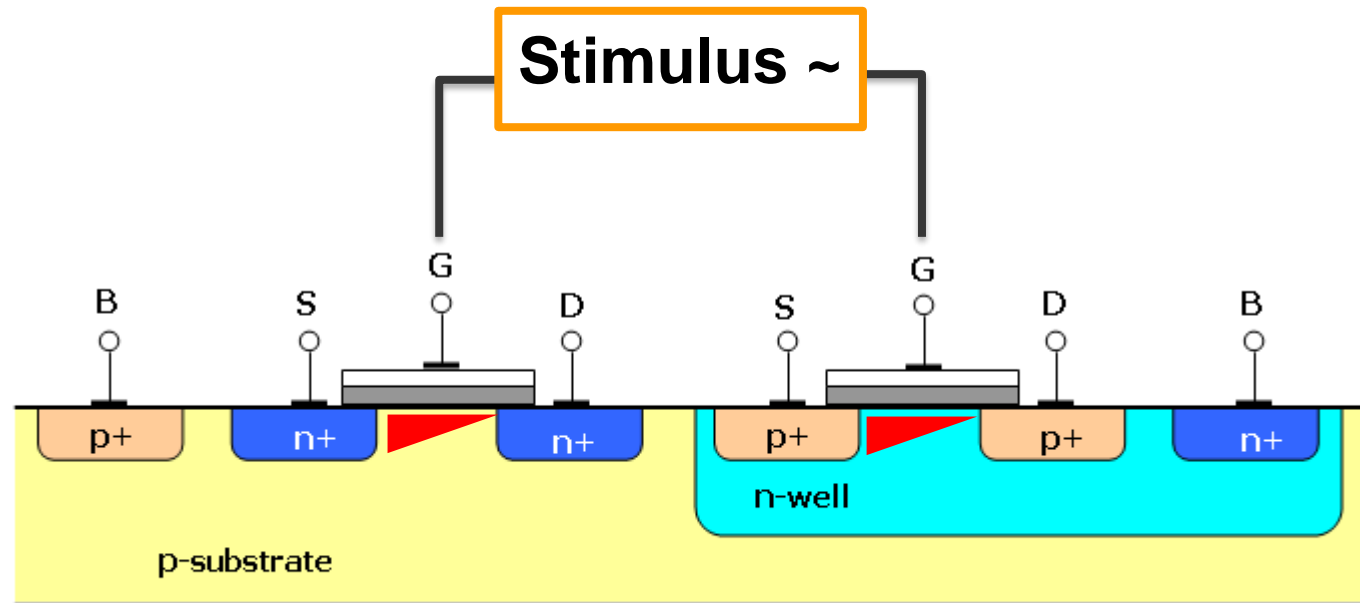
7. TARGET IDENTIFICATION

- Spatial / Area of Interest
 - Input from documentation (floor plan)
 - HW reverse engineering
 - Frequency analysis (EM)
 - Photon Emission Analysis

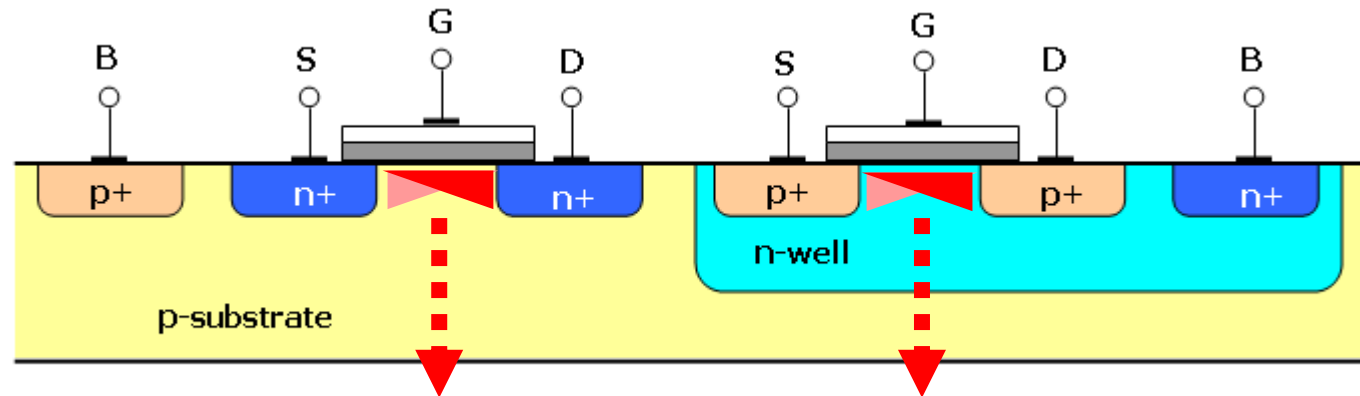
- Time / Time of Interest
 - a) Overview trace (SCA)
 - b) I/O signals
 - c) Input/output correlation
 - d) Leakage analysis
 - e) Points of interest selection



- Idea: Stressing transistors emits photons



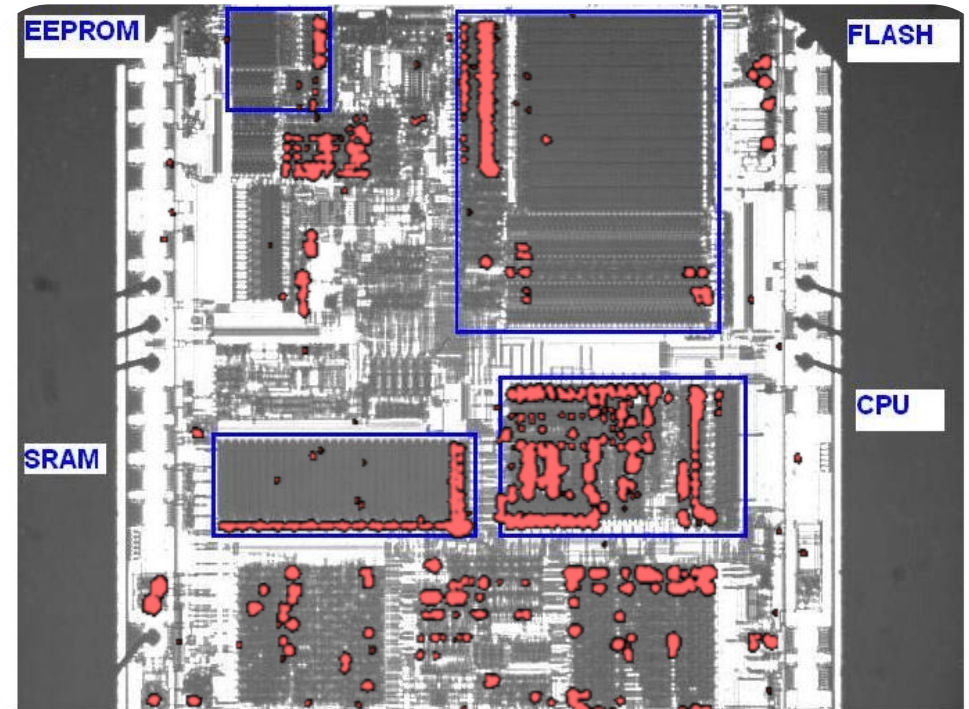
- Idea: Stressing transistors emits photons
- Photons leaking through the substrate can be detected with a very sensitive cooled camera



InGaAs Detector (cooled camera)

LOCALIZATION WITH PHOTON EMISSION ANALYSIS

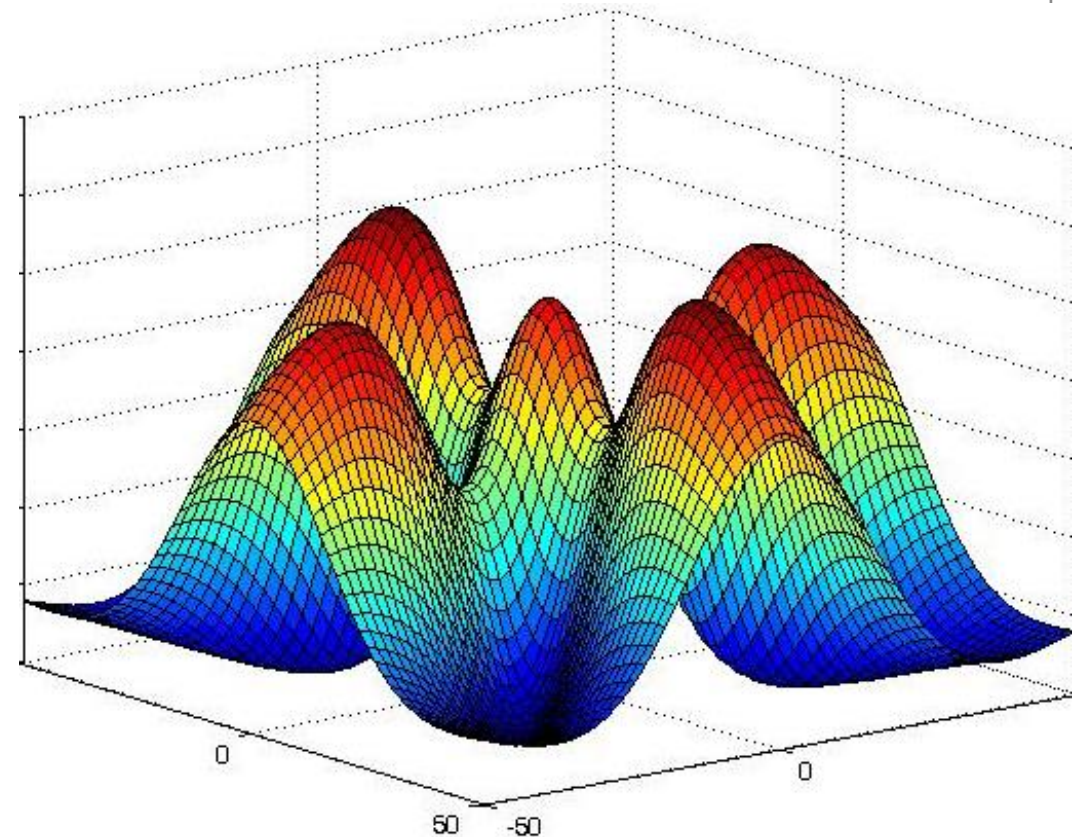
- Idea: Stressing transistors emits photons
- Photons leaking through the substrate can be detected with a very sensitive cooled camera
- Resulting image:
(IR overlay with PHEMA)



Skorobogatov, S., „Using Optical Emission Analysis
for Estimating Contribution to Power Analysis.“,
FDTC 2009.

8. WORST-CASE ANALYSIS

- Finding a (near) optimal setup for a fault or side-channel analysis is a multidimensional optimization problem
- Example: Laser Fault Injection
 - Front side or back side
 - Laser source (wavelength, single/multimode, technology)
 - Single or double laser
 - Optical magnification (spot size)
 - Laser power
 - Glitch length
 - Position (x, y, z)
 - Timing
- Argumentation needed why chosen parameter set is the most suitable for attacking



9. ATTACK PATHS AND METHODS

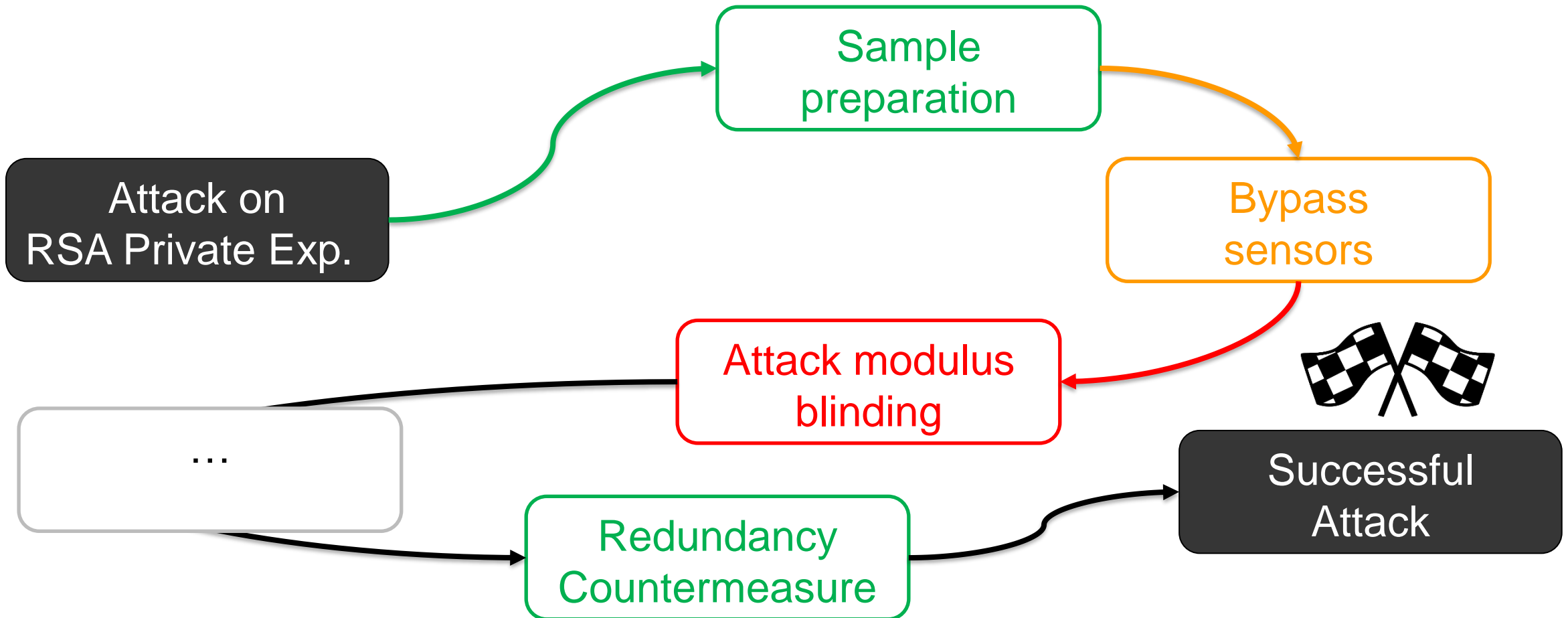
- Lots of ways to attack a function
 - → keep scope, testplan, CB recommendations/requirments in mind

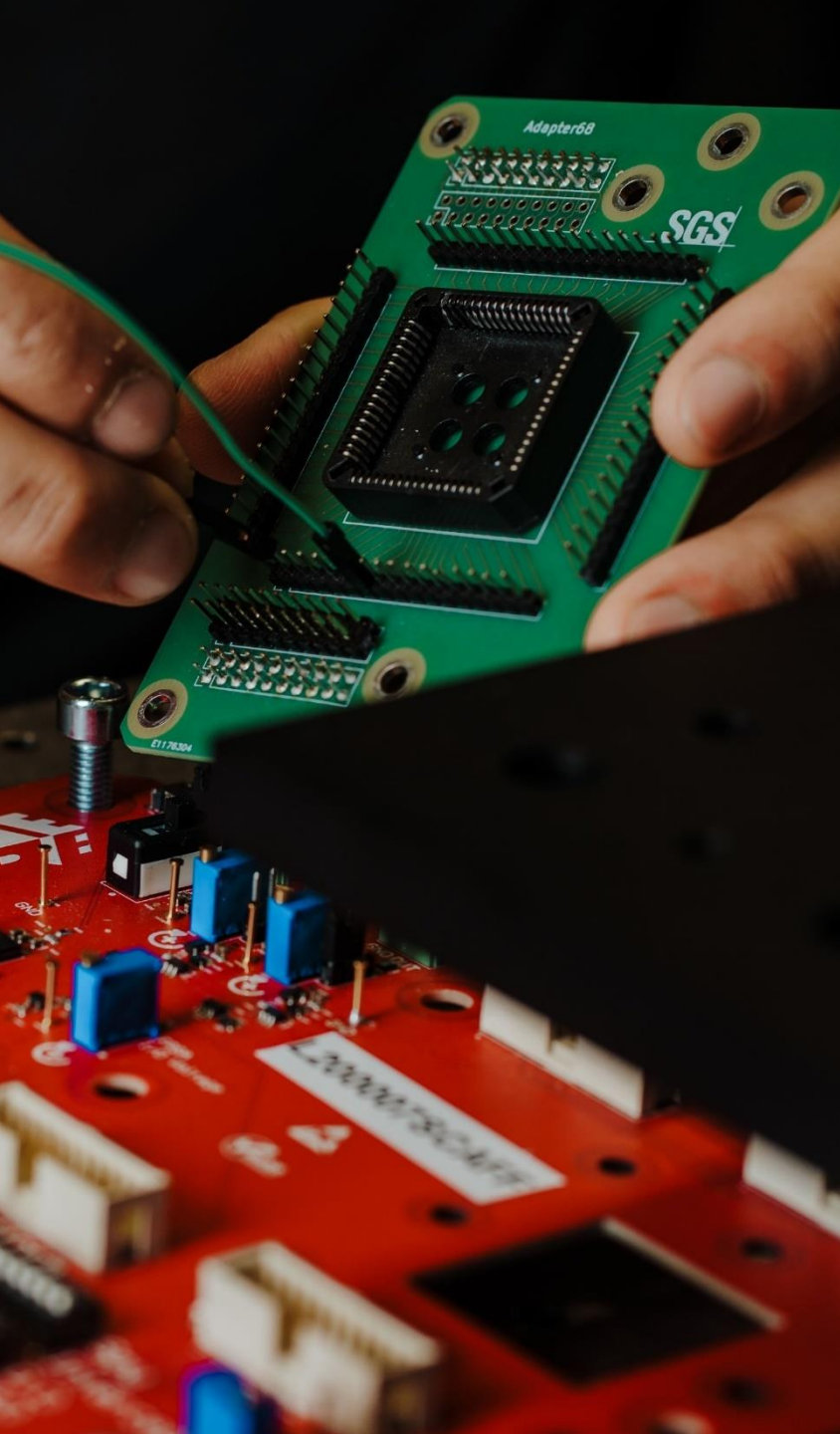
- Attack path examples
 - Cryptanalysis
 - Side-channel analysis
 - Timing, power, EM, sound, photon emission...
 - Template attacks, Simple power analysis, Differential power analysis
 - ...
 - Fault injection
 - Laser fault injection (LFI), body-bias injection (BBI), EMFI ...
 - ...

- Going into details is beyond this course

9. ATTACK PATHS AND METHODS

- A long and winding road...





How to Rate Attacks Example from Common Criteria

CALCULATION OF ATTACK POTENTIAL (SMART CARD)

- Each evaluated attack path is subdivided into
 - A) Identification → how long does it take to find the attack path
 - B) Exploitation → how long does it take to perform the attack
 - Final attack potential = identification + exploitation rating

- Rating depends on:
 - Elapsed time: one hour – not practical (>> 4 months)
 - Expertise: layman – multiple experts
 - Knowledge of TOE: public – not practical
 - Access to TOE: <10 samples – not practical (>> 100 samples)
 - Equipment: None – multiple bespoke
 - Open samples: public – critical (very few open samples/very strong control)

ASSESSMENT OF THE OUTCOME JIL RATING

- The evaluator:
 - Identified and attack claimed security functionality of a TOE
 - The attack was performed within a white box scenario
 - The attack was successful, e.g. the evaluator could read out a cryptographic key
- Can this attack be repeated by an attacker in the field?
 - This depends on the potential of the attacker

Range of values*	TOE resistant to attackers with attack potential of:
0-15	No rating
16-20	Basic
21-24	Enhanced-Basic
25-30	Moderate
31 and above	High

Table 13: Rating of vulnerabilities and TOE resistance

*final attack potential = identification + exploitation.

Factors	Identification	Exploitation
Elapsed time		
< one hour	0	0
< one day	1	3
< one week	2	4
< one month	3	6
> one month	5	8
> four months ³	6	10
Not practical	*	*
Expertise		
Layman	0	0
Proficient	2	2
Expert	5	4
Multiple Expert	7	6
Knowledge of the TOE		
Public	0	0
Restricted	2	2
Sensitive	4	3
Critical	6	5
Very critical	9	*
Not practical	*	*
Access to TOE		
< 10 samples	0	0
< 30 samples	1	2
< 100 samples	2	4
> 100 samples	3	6
Not practical	*	*
Equipment		
None	0	0
Standard	1	2
Specialized ⁽¹⁾	3	4
Bespoke	5	6
Multiple Bespoke	7	8
Open samples (rated according to access to open samples)		
Public	0	NA
Restricted	2	NA
Sensitive	4	NA
Critical	6	NA

CALCULATION OF ATTACK POTENTIAL (SMART CARD)

Factors	Identification	Exploitation
Elapsed time		
< one hour	0	0
< one day	1	3
< one week	2 ✓	4
< one month	3	6 ✓
> one month	5	8
> four months ³	6	10
Not practical	*	*
Expertise		
Layman	0	0
Proficient	2 ✓	2
Expert	5	4 ✓
Multiple Expert	7	6
Knowledge of the TOE		
Public	0 ✓	0
Restricted	2	2
Sensitive	4	3
Critical	6	5 ✓
Very critical	9	*
Not practical	*	*
Access to TOE		
< 10 samples	0	0
< 30 samples	1	2 ✓
< 100 samples	2	4
> 100 samples	3 ✓	6
Not practical	*	*
Equipment		
None	0	0
Standard	1 ✓	2
Specialized ⁽¹⁾	3	4 ✓
Bespoke	5	6
Multiple Bespoke	7	8
Open samples (rated according to access to open samples)		
Public	0	NA
Restricted	2 ✓	NA
Sensitive	4	NA
Critical	6	NA

- Identification → 2+2+0+3+1+2 = 9
- Exploitation → 6+4+5+2+4 = 21
- Final attack potential → 9 + 21 = 30

Range of values*	TOE resistant to attackers with attack potential of:
0-15	No rating ✓
16-20	Basic ✓
21-24	Enhanced-Basic ✓
25-30	Moderate ✓
31 and above	High ✗

- Outcome depends on AVA_VAN level
- Fail?
 - Discussion with developer/sponsor
 - Change guidance (force formerly optional countermeasures)
 - Change TOE
 - Reduce security claim