

RSA[®]CONFERENCE2009

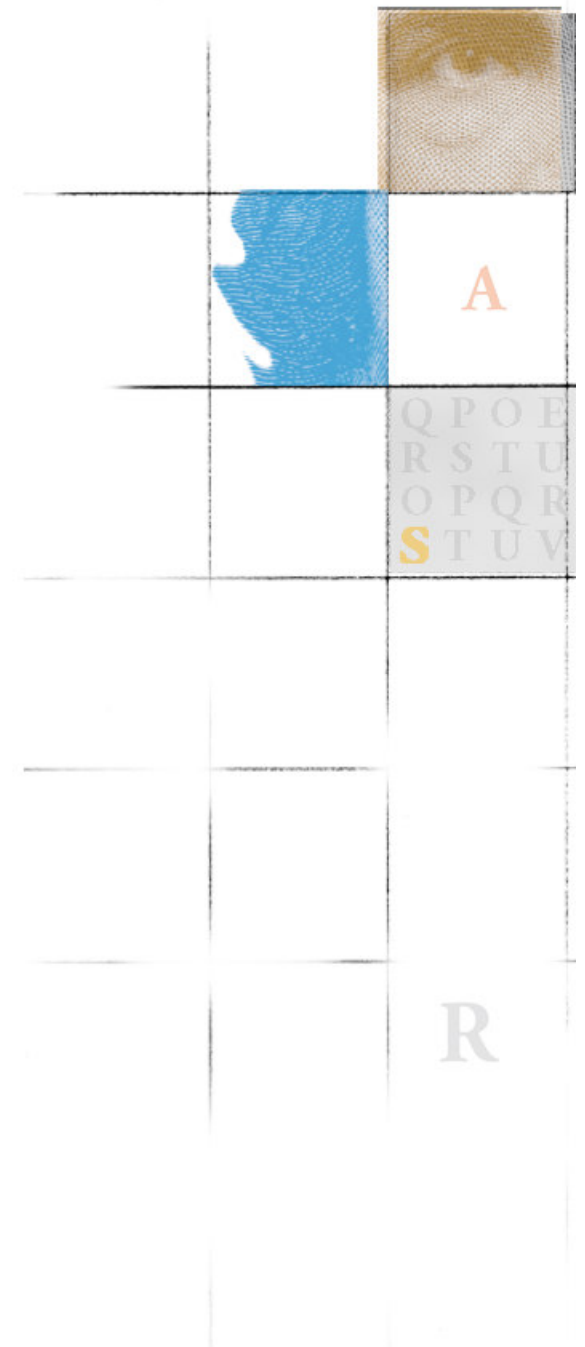
Lessons Learned: Side Channel Analysis on Embedded Systems

Job de Haas

Riscure

22/04/09 | Session ID: HT2-201

Session Classification: Advanced



Agenda

Scope

What is new with SCA on embedded?

How do you test for it in practice?

How to assess the strength of a product?

Scope

- Device: embedded systems with security functions
- Focus on passive side channels
- Goals
 - What is the threat from side channel analysis to embedded systems?
 - How does it compare with attacks on smart cards?
 - What are future developments?
 - Demonstrate side channel analysis.

Embedded systems to consider

Microcontroller based

- USB sticks
- Car locks
- Remote access tokens

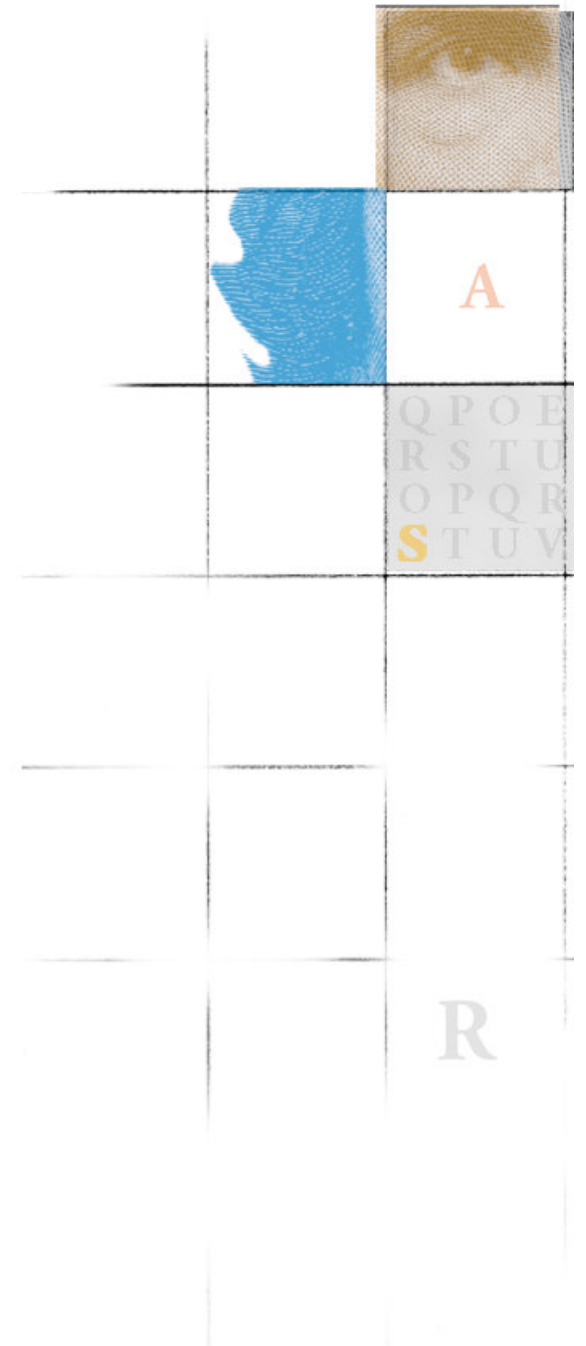


'Complex' processor based

- Mobile devices
- Game consoles
- Multi-media chipsets for pay-TV



What is new with SCA on embedded?



Purpose of SCA on embedded

Retrieve secrets

- Key
- PIN
- Unlock code

Reverse engineer

- Program flow
- Crypto protocol
- Algorithm

Not much changes ...

When does SCA become interesting?

If side channel threats apply, depends on

- Physical access?
- Access time window?
- Interfacing and control?
- Exploitation equipment \$?

A device becomes interesting when

- It contains a secret
- It contains a feature that can be unlocked
- Logical or physical access to internals is hard

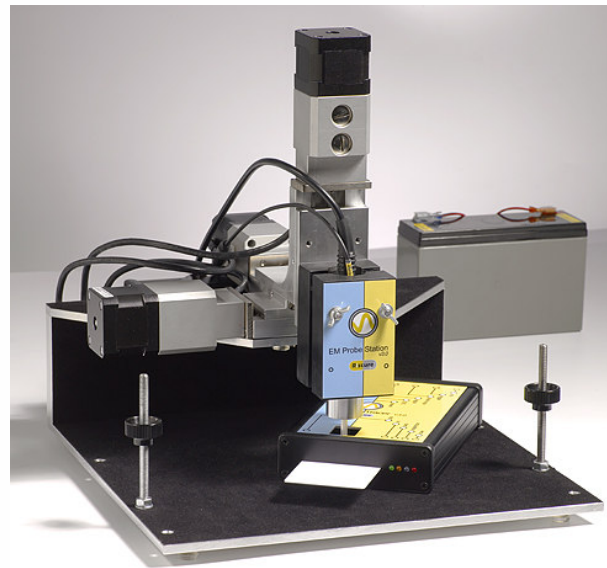
Typical SCA set up



Configure / Retrieve



**Commands /
data**



**Signal +
Trigger**

Typical prerequisites

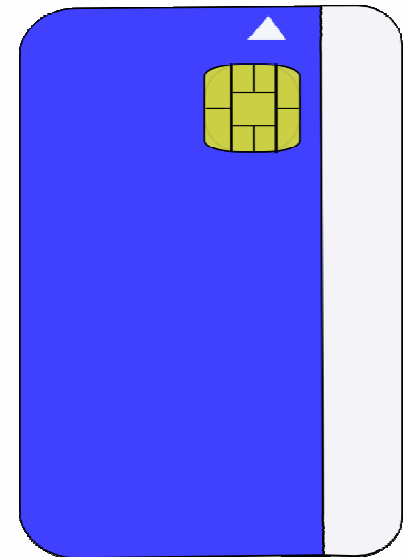
- ✓ Access to side channel
- ✓ Access to input or output data
- ✓ Minimize noise in side channel
- ✓ Time measurement of operation (trigger)
- ✓ Link data to operation

Comparing to smart cards

So far SCA testing centered on smart cards

A smart card:

- Standardized device
- Focus of SCA since its conception
- The benchmark of how SCA is rated



A smart card is an embedded system

... But a very well defined one

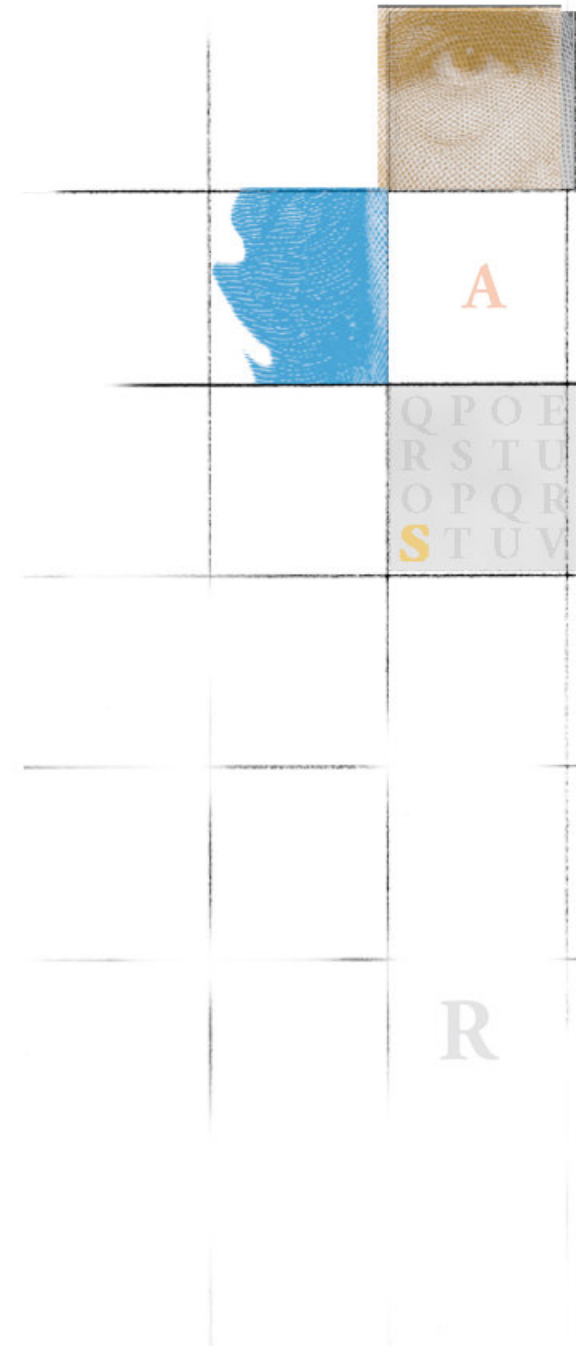
Processor comparison

	Smart card	Embedded
Processor complexity	Simple CPU next to crypto core	Complex processor with lots of peripheral next to crypto core(s)
Crypto core size	Significant compared to overall chip	tiny compared to overall chip
No. of crypto engines	One core per crypto operation	>10 cores for different purposes
SW or HW engine	Few SW implementations	Both HW and SW implementations
Countermeasures	Hardware and software countermeasures against leaking of both CPU and crypto core	No countermeasures and CPU leaks significantly

Acquisition comparison

	Smart card	Embedded
Power interface	Standard interface	Implemented on PCB with dedicated power supply
Triggering of acquisition	Standard interface allows controlled trigger	Trigger may be difficult without control over CPU
Flexibility of interfacing	Interface restricted	Control over CPU can often be gained through reverse engineering
Power consumption	Low power device (few mA)	Low to High power device (0.5A to 4A)
Clocks	Moderate clocks speeds (<50MHz), limited number	Moderate to high clock speeds, single or multiple clock domains
Sample preparation	Attacks are often noninvasive	Attacks mostly require invasive action

**How do you test
for it in practice?**



Test versus attack

An attacker needs to turn a vulnerability into an exploit

A tester needs to gain insight in attacker cost efficiently

➤ How to create the **optimal environment** to discover a vulnerability?

General aspects

Controlling the crypto

Linking data with measurements

Efficiency of acquisition

Increased speed versus increased complexity

Timing analysis

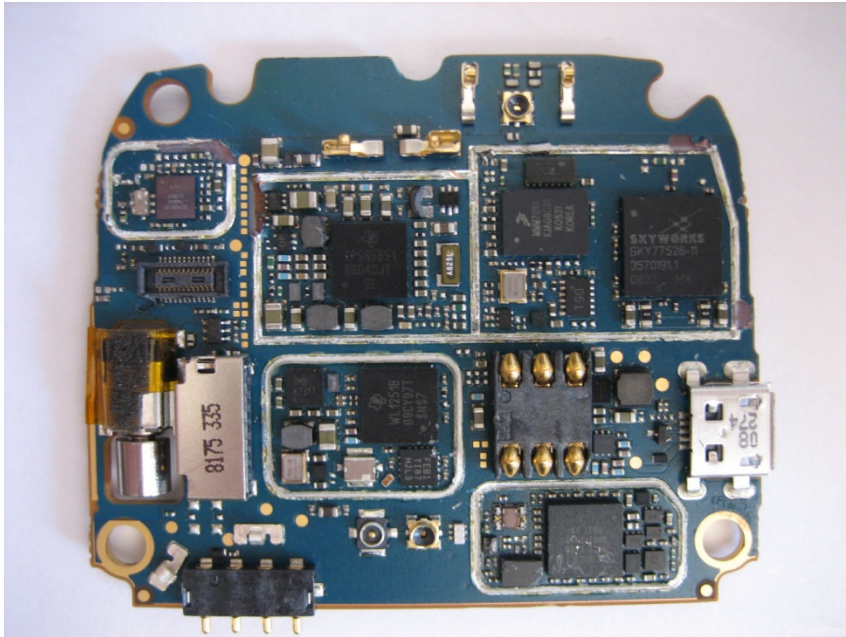
Peripheral outputs assist (example XBOX 360)

Exploiting runtime access (cache)

Increasing accuracy with EM and power

- Timing is a risk in many software implementations: both crypto and comparisons

Power analysis



<http://www.phonewreck.com>

- Tapping power or supplying it
- Reaching rails
- Identifying the correct supply rail
- Disabling power domains
- Disabling peripherals

➤ All require (more detailed) knowledge on target

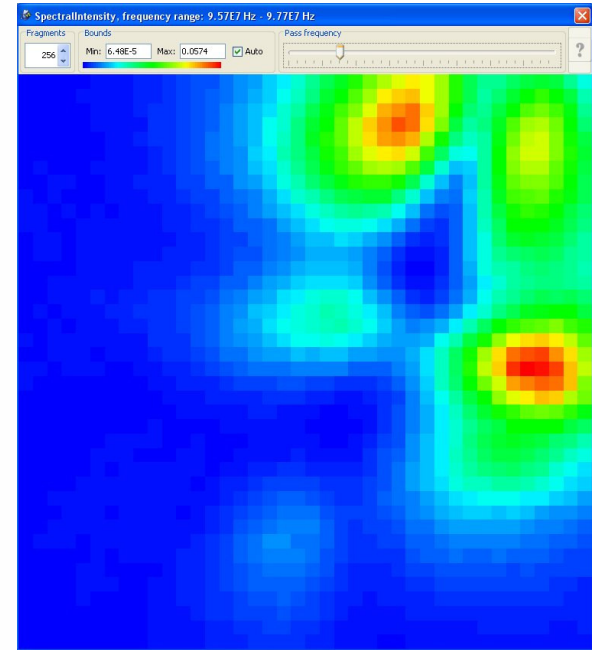
EM Analysis

EM signal adds dimension

How to locate?

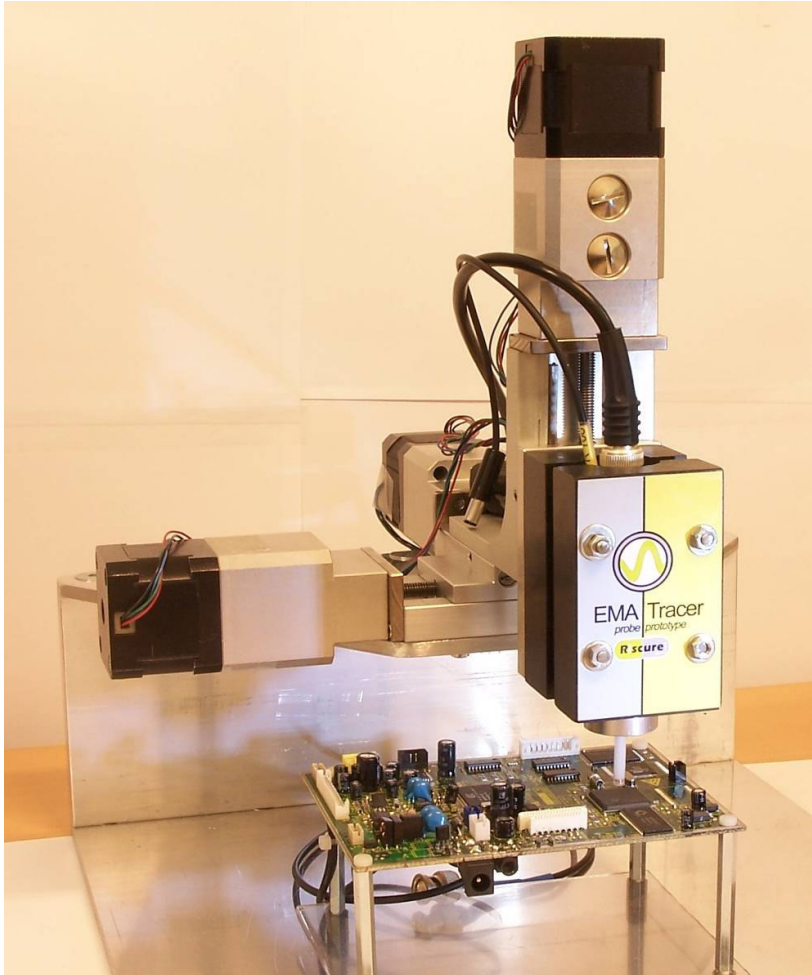
When can EM be better?

EMA is an active research topic

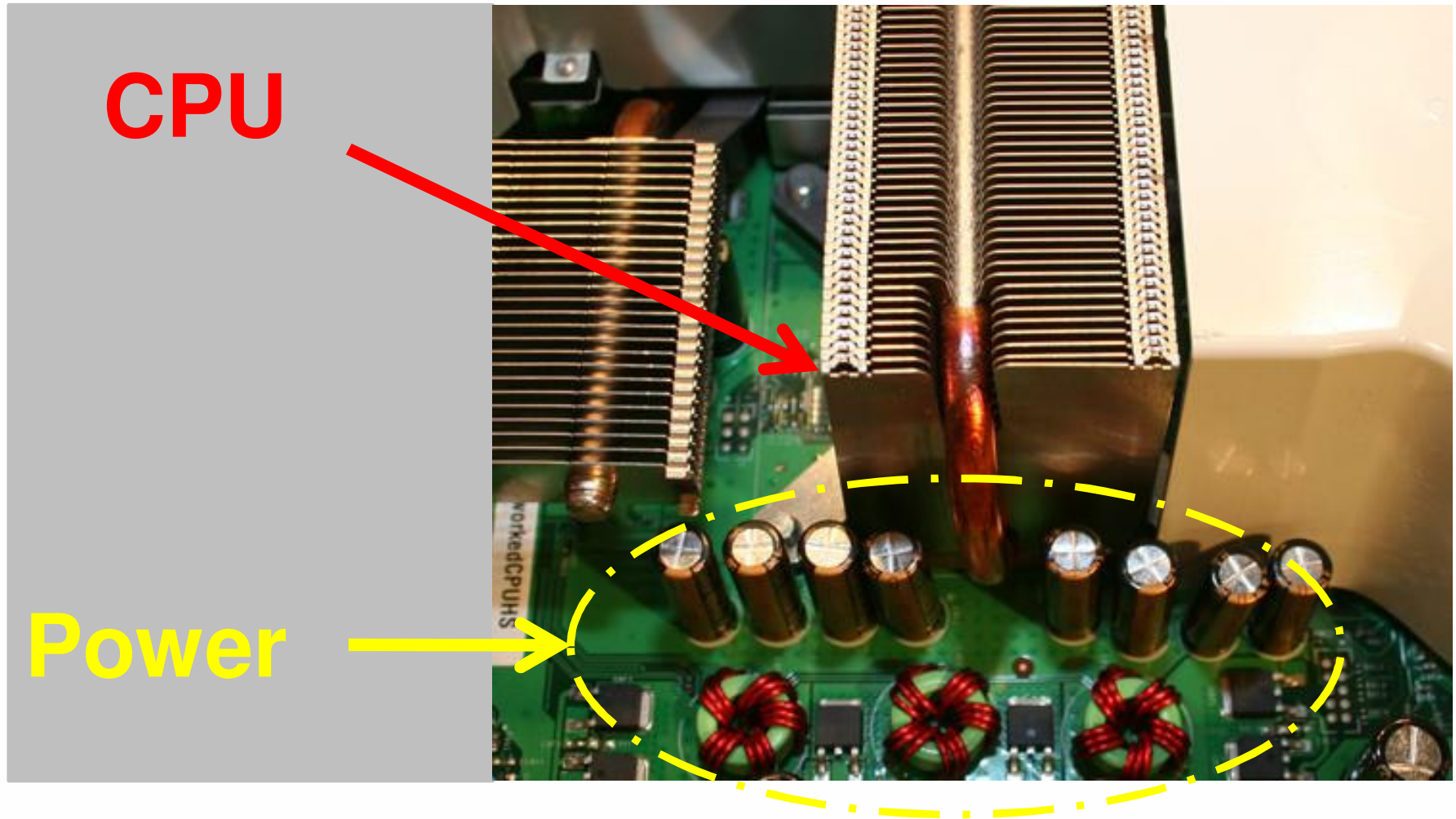


- EM seems to add most when target operation is small relative to overall chip

EM scanning DEMO

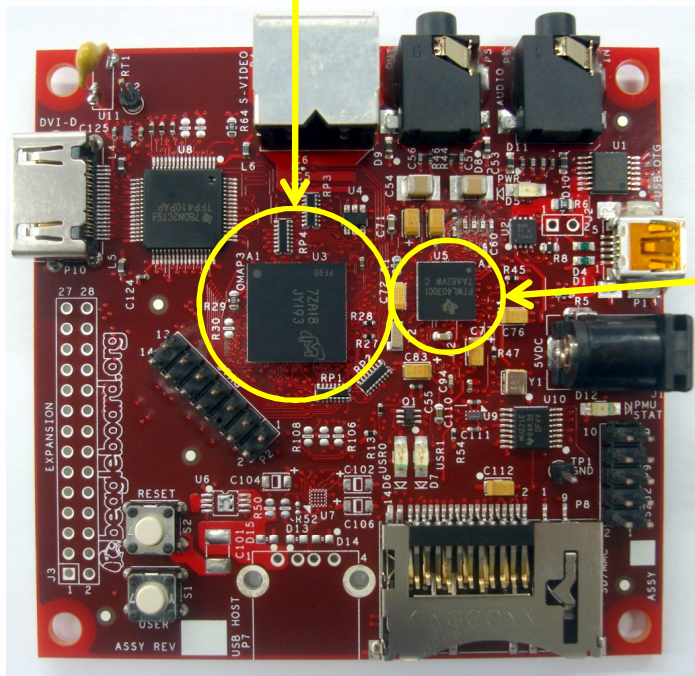
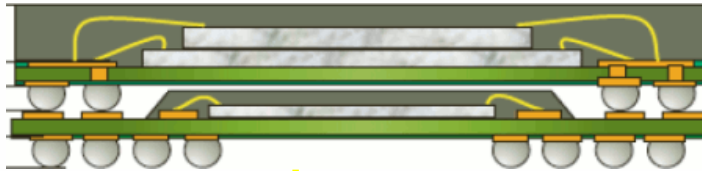


Practical encounters (1)

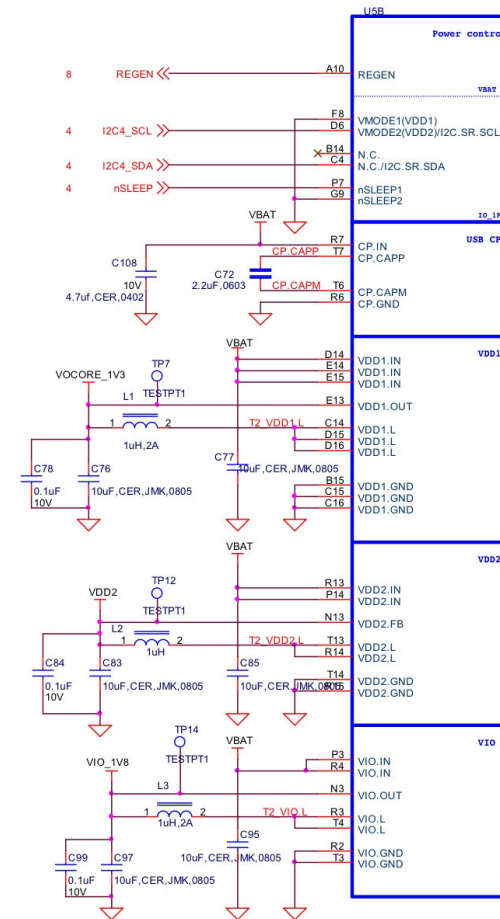


Practical encounters (2)

Package-on-package



Main power rails



How to assess the security strength of a product?



Threat and impact

SCA can break security functions, because:

- Few countermeasures
- Significant leakage
- Fast acquisition
- Examples in the field: Keeloq, ...

However...

Effort can be considerable

Required level of control

Attacks needed to achieve control

High noise level, increased acquisition times

- Even without countermeasures,
but countermeasures do improve this!

Countermeasures

Hardware

- Random Interrupts
- Data / key masking
- Shielding
- Balancing

Software

- Randomizing flow
- Blinding / masking
- Algorithm
- Protocol design

➤ Expert design at Cryptography Inc (CRI)

Side channel resistance

CPU type	Countermeasure	Effort (inc setup)	Skills	Strength
Basic microcontroller	No	1-2 weeks	SPA/DPA	0
Basic microcontroller	Basic	2-6 weeks	+ Adv sig proc	1
Complex processor	No	2-6 weeks	+ Adv sig proc	1
Complex processor	Basic	1-3 months	+ Adv sig proc	2
Both	Strong	>3 months	+ High order DPA	3

Developments

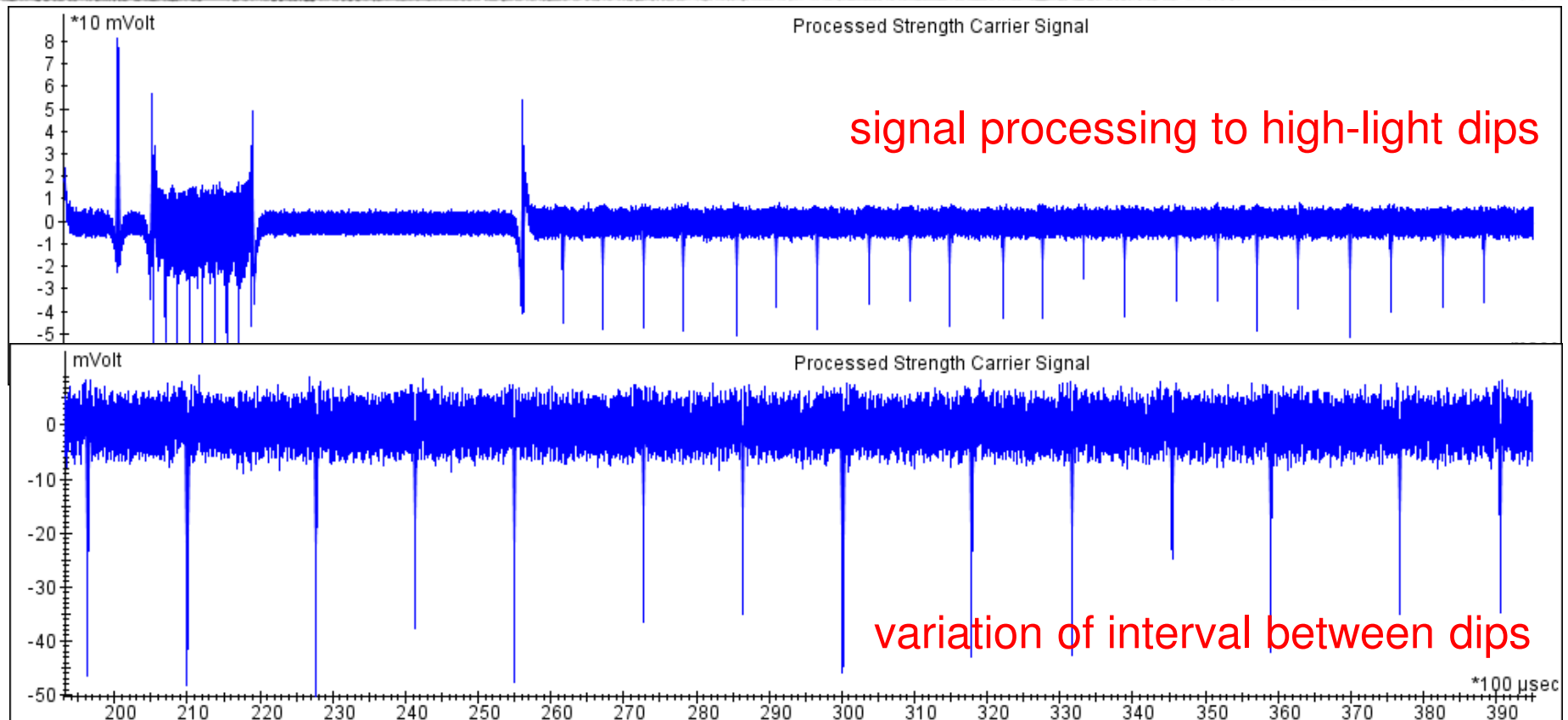
Side channel analysis related

- Increasingly high speed acquisition
- Combined analysis of EM and power
- SCA becomes more mainstream
 - Tools
 - Techniques

Processor related

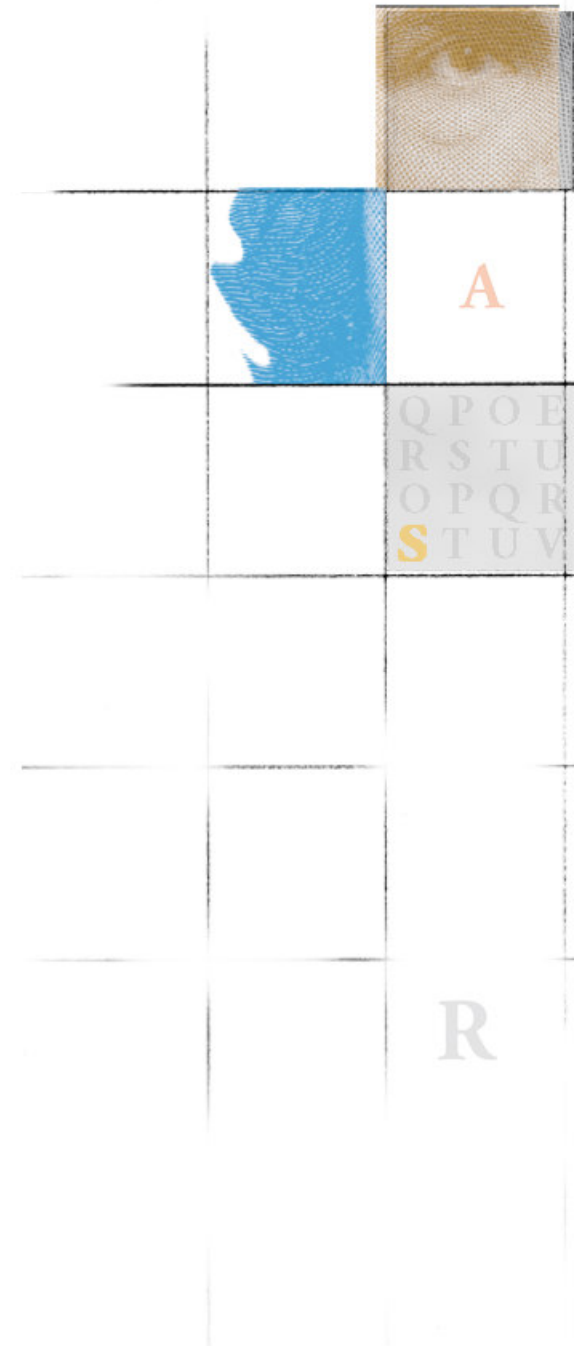
- More security features everywhere
- Basic countermeasure introduced

Side Channel on RSA - DEMO



1 0 1 0 1 0 0 1 0

Conclusion



Findings

Embedded systems provide a different environment for SCA

- New obstacles for attackers: interfacing, noise, triggering
- Potential exposure due to: limited/no countermeasures, speed of acquisition, software implementations

Side channel is primarily a threat to

- Devices with basic microcontrollers
- High security devices that protect something very valuable

Recommendations

To achieve strong protection against SCA, strong countermeasures must be added

Demand countermeasures from manufacturers if you need the security level

Do not rely solely on the hardware for protection

Verify SCA protection if you need that security level

Apply

- Determine if assets are vulnerable to SCA
 - Algorithm type and implementation, Countermeasures used, Type of CPU
- Are other attacks than SCA more feasible?
 - Improve resistance to such attacks first
- If getting asset through SCA is feasible:
 - Verify how much it leaks
 - Select countermeasures in software / hardware
 - Can the design / protocol be improved?