Side Channel Analysis on Embedded Systems

Job de Haas
*Riscure*

October 8, 2008
Who the hell is…

Job de Haas

• Electro engineer: 1990
• First exploit: 1991
• ITSX: 1998
• Riscure: 2006

Currently at Riscure

• Director Embedded Technology
• Testing security on: Set-top-boxes, mobile phones, smart cards, payment terminals, ADSL routers, VoIP modems, smart meters, airbag controllers, USB tokens, …

HITB 2009
Agenda

Scope

What is SCA on embedded?

How do you test for it in practice?

How to assess the strength of a product?
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
Scope

- **Device**: embedded systems with security functions
- **Focus on**: passive side channels
- **Why?**
  - 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.
What is SCA on embedded?
Attacking Side Channels

- Time
- Power consumption
- Electro-Magnetic radiation
- Light
- Sound
• Signal leakage from busses, registers, ALUs, etc

PIN verification attempts
Statistical data detection

- Where is data processed in presence of noise?
Statistical data detection

- Where is data processed in presence of noise?
- Collect many traces with different data ($n > 1000$)
- Assume data values are:
  - known (e.g. algorithm input or output)
  - uniformly random (typical for crypto)
- We focus on one bit of one variable in the process

Group by known data           Average trace

```
0 0 0 0
```

```
1 1 1 1
```

Subtract

Differential trace
Differential trace

- Input: $n$ traces with known variable (e.g. input or output)
- Output: 1 trace with indication where bit causes trace differences
Not much changes ...

Retrieve secrets
- Key
- PIN
- Unlock code

Reverse engineer
- Program flow
- Crypto protocol
- Algorithm
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

HITB 2009
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

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

<table>
<thead>
<tr>
<th></th>
<th>Smart card</th>
<th>Embedded</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power interface</strong></td>
<td>Standard interface</td>
<td>Implemented on PCB with dedicated power supply</td>
</tr>
<tr>
<td><strong>Triggering of acquisition</strong></td>
<td>Standard interface allows controlled trigger</td>
<td>Trigger may be difficult without control over CPU</td>
</tr>
<tr>
<td><strong>Flexibility of interfacing</strong></td>
<td>Interface restricted</td>
<td>Control over CPU can often be gained through reverse engineering</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>Low power device (few mA)</td>
<td>Low to High power device (0.5A to 4A)</td>
</tr>
<tr>
<td><strong>Clocks</strong></td>
<td>Moderate clocks speeds (&lt;50MHz), limited number</td>
<td>Moderate to high clock speeds, single or multiple clock domains</td>
</tr>
<tr>
<td><strong>Sample preparation</strong></td>
<td>Attacks are often noninvasive</td>
<td>Attacks mostly require invasive action</td>
</tr>
</tbody>
</table>
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
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
XBOX 360 with Infectus board

source: http://beta.ivancov.com
XBOX 360 timing attack

- XBOX 360 has a **secure boot** chain
- First boot loader security implemented with a HMAC-SHA1
- Sequence:
  - Hash secret key + boot loader with SHA1
  - Compare 16 bytes result with stored 16 bytes
- Comparison is per byte → **timing attack**
- Implementation in Infectus board:
  - It can modify stored HMAC-SHA1 value in NAND flash
  - Observes **timing** of diagnostic POST byte on PCB
  - Reset CPU with nTRST
- Brute forcing $16 \times 128 = 2048$ values on average takes **about 2 hrs**

source: [http://www.xboxhacker.net](http://www.xboxhacker.net)
Power analysis

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

➢ All require (more detailed) knowledge on target

http://www.phonewreck.com
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 probe

- Probe is a **coil** for magnetic field
- Generally the **near field** (distance $<< \lambda$) is most suitable
EM scanning DEMO
XY scan examples

Clock pin! (20MHz)

Full spectrum

Around 40MHz

Scans above same chip running at 20MHz
Bonding wire effects

Full spectrum

Full spectrum logarithmic
Hotspot is clock line

Corresponding trace shows no pattern: base clock
Hotspot after resampling

XY plot full spectrum (left); selected higher harmonic (right)

Trace shows pattern
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

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Random Interrupts</td>
<td>• Randomizing flow</td>
</tr>
<tr>
<td>• Data / key masking</td>
<td>• Blinding / masking</td>
</tr>
<tr>
<td>• Shielding</td>
<td>• Algorithm</td>
</tr>
<tr>
<td>• Balancing</td>
<td>• Protocol design</td>
</tr>
</tbody>
</table>

- Patented by Cryptography Research Inc (CRI)
- Licenses required and taken by major vendors (Infineon, NXP, Renesas, Samsung, ...)
- Check with CRI
### Side channel resistance

<table>
<thead>
<tr>
<th>CPU type</th>
<th>Counter-measure</th>
<th>Effort (inc setup)</th>
<th>Skills</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic microcontroller</td>
<td>No</td>
<td>1-2 weeks</td>
<td>SPA/DPA</td>
<td>0</td>
</tr>
<tr>
<td>Basic microcontroller</td>
<td>Basic</td>
<td>2-6 weeks</td>
<td>+ Adv sig proc</td>
<td>1</td>
</tr>
<tr>
<td>Complex processor</td>
<td>No</td>
<td>2-6 weeks</td>
<td>+ Adv sig proc</td>
<td>1</td>
</tr>
<tr>
<td>Complex processor</td>
<td>Basic</td>
<td>1-3 months</td>
<td>+ Adv sig proc</td>
<td>2</td>
</tr>
<tr>
<td>Both</td>
<td>Strong</td>
<td>&gt;3 months</td>
<td>+ High order DPA</td>
<td>3</td>
</tr>
</tbody>
</table>

Note:
A complex processor with a bad RSA can still break in less than a week!
These are only indicators.
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

http://opensca.sf.net
http://www.dpacontest.org

HITB 2009
Side Channel on RSA - DEMO

signal processing to high-light dips

variation of interval between dips

HITB 2009
RSA implementation

- Algorithm for $M = c^d$, with $d_i$ is exponent bits ($0 \leq i \leq t$)
  - $M := 1$
  - For $i$ from $t$ down to 0 do:
    - $M := M \times M$
    - If $d_i = 1$, then $M := M \times C$
Demo Target

- Dev board with LPC2468
- ISP + JTAG can be locked
- Internal Flash for boot and storage
- Internal SRAM

- Can be moderately secured
- Running RSA with internal key
Conclusion
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
Thank you

Job de Haas

Director Embedded Technology

dehaas@riscure.com