Low-Cost Chip Microprobing

Philipp Maier <dexter@srlabs.de>
Karsten Nohl <nohl@srlabs.de>
A few smart cards chips cover numerous security domains

### Security chip applications

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payment cards, electronic ID cards, access badges</td>
</tr>
<tr>
<td>Trusted platform modules (TPM)</td>
</tr>
<tr>
<td>Device and accessory identification</td>
</tr>
<tr>
<td>SIM cards, NFC secure elements</td>
</tr>
<tr>
<td>Content protection</td>
</tr>
</tbody>
</table>

### Chip hacking motivation

- **Recent trend** – Functionality that users traditionally circumvented or exposed in their devices moves into hardware:
  - Usage restrictions
  - DRM, “Dongeling”
  - “Secret” protocols
- Devices are increasingly controlled by their manufacturer, not their owner
- We need more wide-spread hardware analysis knowledge and cheaper tools to combat this trend
Reverse-engineering hardware functions requires specific tools

- Oscilloscope
- Logic analyzer
- Microscope
- Micropositioner

Disassembler, decompiler
This talk introduces the methodology and tools of earlier (and hopefully future) works.

<table>
<thead>
<tr>
<th>Introductory</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deep Silicon Analysis</strong></td>
<td><strong>Mifare, Legic, DECT Reversing</strong></td>
</tr>
<tr>
<td>Starbug, Nohl</td>
<td>Starbug, Plötz, deDECTed*, Nohl</td>
</tr>
<tr>
<td>25C3, HAR 2009</td>
<td>24C3, 26C3</td>
</tr>
<tr>
<td><strong>Reviving Smart Card Analysis</strong></td>
<td><strong>Hacking the Smart Card Chip</strong> et al.</td>
</tr>
<tr>
<td>Tarnovisky, Nohl</td>
<td>Tarnovisky</td>
</tr>
<tr>
<td>Camp 2011, BlackHat 2011</td>
<td>Various BlackHat, ToorCon</td>
</tr>
<tr>
<td><strong>This Talk</strong></td>
<td></td>
</tr>
<tr>
<td>Dexter, Nohl</td>
<td></td>
</tr>
<tr>
<td>29C3</td>
<td></td>
</tr>
</tbody>
</table>

*deDECTed team: Karsten Nohl, Andreas Schuler, Erik Tews, Ralf-Philipp Weinmann, Mathias Wenzel
Microprobing background

- Probing with simple tools
- Advanced probing techniques
Basic CPU principle: Current instruction decides which instruction gets loaded next

Program memory: (EEP)ROM, Flash

CPU Core

- Instruction latch
- For most instructions:
  - Program counter ++
- For flow-altering instructions (branches, jumps, returns):
  - Register / memory
  - Program counter

Data

Address
Probing and glitching with only two needles allows full memory read-out

Attacker goal: Make the CPU go to all places in memory independent of security checks and other flow characteristics. This attack is called **Linear Code Extraction.**

**Attack vector 1:** Freeze the latch clock so that one instruction is executed indefinitely (second needle)

**Attack vector 2:** Overwrite one bit that is set for all flow-altering instruction (works on ARM and a few other instruction sets)

Attacker probes one bus bit at a time (first needle)
- Microprobing background

**Probing with simple tools**

- Advanced probing techniques
Basic probing setup

Micropositioner
Buffer
Needle
Chip target
Moveable stage
Step 1: Decapsulate chip

Option A – **Mechanically pry open package**

Option B – **Chemically etch into package**
Step 2: Expose wires (1/2)

Option A – **Scratching**

- A micropositioner in combination with hard needles can break the silicon oxide (aka glass) above metal tracks
- Works best for highest ("top") metal layer on non-planarized chips
Step 2: Expose wires (2/2)

Option B – Lasering

- A laser shot through the microscope destroys chip structures to:
  - Expose wires for probing -or-
  - Cut wires to alter chip logic; i.e. permanently reset lock bit
- Optimal for planarized chips and for working on top metal
- Takes practice to not destroy chips through shorting wires or inducing overvoltage
- Higher cost alternative when compared to scratching but more reliable after practice
Step 3: Find exploitable wires

- Most interesting chip structures:
  - Security bits (fuses)
  - Data buses
- Discover them:
  - Reverse-engineer (HAR 09, Camp 11)
  - Trial and error

Source: Chris Tarnovsky, Flylogic (fuse image)
Step 4: Connect to wires with probing needle
Some level of probing can even be done with simple optical microscopes and few extra components.

**Old optical microscope**
- € 2,000

**Micro-positioner**
- € 300

**Stage**
- parts: € 70

**Drawback:**
- Probe distance limited to 2mm

**Compare to 2cm w/ special probing microscope**

**Measurement amplifier** (design to be released under CC soon)
- parts: € 50

SECURITY RESEARCH LABS
- Microprobing background
- Probing with simple tools

**Advanced probing techniques**
Focused Ion Beams prepare small feature-size chips for probing

Focus Ion Beam (FIB) operation mode 1: Cut into chip to expose structures

FIB operation mode 2: Deposit material to allow probing connectivity to hidden/small wires

Source: Chris Tarnovsky, Flylogic
Protection meshes create additional complexity for FIB probing

- Tight metal trace on top layer hides critical structures
- Mesh breach triggers “alarm”
Meshes slow down attack, but do not prevent them

**Mesh circumvention**

- Either remove parts of mesh, then bridge/fix the mesh with FIB edits -or-
- Circumvent mesh alarm:
  - If defense is bulk-erase of non-volatile memory: Cut off the programming pumps
  - If chip it is logic reset: Exploit in the small time window before the reset triggers

Source: Chris Tarnovsky, Flylogic
Arms race around front-side FIB attacks makes back-side attacks more attractive

Invasive attacks and defenses have thus far concentrated on touching metal tracks on the chip.

The same critical data can also be intercepted or manipulated at transistors, which are reachable from the other side of the chip.
Take aways

- Device functionality is increasingly hidden in hardware and needs to be freed
- Software can be extracted from chips using fuse overwrites or linear code extraction
- Simple controllers can be attacked with cheap tools; smart cards require focused ion beams

Questions?

Philipp Maier <dexter@srlabs.de>
Karsten Nohl <nohl@srlabs.de>