Hardware Reverse-engineering Tools
new threats – new opportunities
Bio

- 8 years in a security lab
- Technology lover
- Analysis techniques // exploits
- Involved from sample preparation to report writing
  - Optical systems setup
  - Sample preparation
  - Delayering
  - Imagery
  - Software developments
- Semi-invasive attacks
- Invasive attacks – circuit edit
- Micro-probing
- Various experiments
  - Photoemission
  - AFM techniques
  - Electrical glitch
Focus on Hardware reverse engineering

Evolution of the all process
  - Sample preparation
  - Imaging
  - Study

Change in evaluation criterias

Future evolutions
- Attacks summary
- Chip classification
Non invasive attacks - VCC and Clk glitch

- Take advantage of the RTL technology
- Used to skip instructions or to disturb the normal execution

⇒ Finding the glitch pattern is empirical
⇒ The real effect stays hidden
Semi-invasive attacks - Sample preparation techniques

Partial opening - frontside

Context – Attacks summary
Semi-invasive attacks - Sample preparation techniques

Repackaging
Semi-invasive attacks - Sample preparation techniques

In situ:
Semi-invasive attacks – Principle

- 1064 nm laser spot can induce transistor switch
- Silicon is « transparent » @1064 nm
- Metal planes prevent laser fault injection
- Fault is injected at a precise given location

Context – Attacks summary
Semi-invasive attacks – Tests

Fishing:

. Unknown timing
. Vague localization
. Trial and Error

=> Working ;-)
Semi-invasive attacks – Tests

Automated fishing (a first step toward laser scan):

- XY stages for chip positioning
- One position – several laser pulses
- Pass-fail from data returned by the device
- One scan per timing of interest

=> Different effects
Semi-invasive attacks – Tests

Targeted shot:

. Precise localization from laser scan image

. Timing still critical
Invasive attacks

Get access to the circuitry itself and apply modification for

- Shield bypass
- Embedded counter-measures deactivation
- Data extraction

Context – Attacks summary
**Invasive attacks**

*The process: delayering and imaging*

- Delayering requires skills and machinery
- Optical and / or SEM scan
- Pictures stitching is key
- Alignment of layers must be precise
Invasive attacks

The process: optical imaging

Optical scans are fast to perform but:

- Good tilt setup for high resolution scan is a nightmare (narrow depth of field)

- Small features become invisible with technology size reduction

- Oxide layers are light transparent (every deeper layer is visible)

- Pictures lack information such as vias
**Invasive attacks**

*The process: SEM imaging*

SEM scan are slow (hours range) and pictures are distorted but:

- Depth of field is bigger
- Resolution is higher
- Oxide layers are not transparent (one visible layer at a time)
Invasive attacks

The process: “Reverse-engineering”

- Intensive use of pictures
- Generate a test procedure
- Localize points of interests

Context – Attacks summary
Invasive attacks

The process: Fib edit

Context – Attacks summary
Invasive attacks
The process: Micro-probing

Context – Attacks summary
Invasive attacks

Linear Code Extraction

- 2 major types of instructions: sequential / jumps
- Provide only one instruction to the core of sequential type
- Core will execute something useless
- Address will be incremented
- The entire code will be outputted from NVM memory

=> Most successful invasive attack
Invasive attacks

Linear Code Extraction

- Cut and setup an instruction for the core (ex. nop)
- Read data before the cut
Invasive attacks

Linear Code Extraction: Less FIBing – more options

- Use buffer or register / latch signal to prevent read buffer output update
- Read data before the buffer (register / latch)

⇒ Running code extraction is straight forward
⇒ Modification of the code is possible
⇒ Skipping instruction is possible (jumps...)
3 different kind of security levels:

- **Weak**: code can be extracted by old techniques or LCE
- **Adequate**: old techniques do not work // LCE can be done at the costs of Hardware Reverse-engineering
- **Advanced**: Hardware Reverse-engineering is mandatory for a code extraction + hardware functions have to be found and studied
3 different kind of security levels:

<table>
<thead>
<tr>
<th>Chip manufacturer</th>
<th>Pirates</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>Trivial</td>
<td>Dangerous (cheap)</td>
</tr>
<tr>
<td>Adequate</td>
<td>Tricky</td>
<td>Balanced</td>
</tr>
<tr>
<td>Advanced</td>
<td>Headache provider</td>
<td>Overkill (expensive)</td>
</tr>
</tbody>
</table>
Hardware Reverse-engineering Tools: the next step
Analysis techniques evolution:
- Laser fault injection
- ROM code extraction
- LCE
- Other techniques

Sample preparation and imaging evolution:
- Sample preparation
- SEM imaging
- Accurate correlation
- All chip features become visible and usable
Usual tests target registers or memory output
- Where are the working registers?
- Is the memory encrypted?
⇒ Results can be achieved but hardly exploited

Fishing tests are also effective
- Needed equipment price can be quite low
- Effect can not be predicted
- Timing and spot localization have to be found
⇒ Results can be achieved but can’t be fully understood therefore exploits are difficult to build
⇒ Fishing is a real threat

Analysis techniques evolution:
Laser fault injection

HRTs as the next step

Laser fault injection

HRTs as the next step
Analysis techniques evolution:

- Reading extra bytes from RAM while glitching during the ATR routine
- Number of extra bytes depends on glitch location

- Change mode of execution
- Effect is “stored”
- Original mode can be restored
- Instruction skip

⇒ Registers can be found by fishing
⇒ Fault injected inside the core – what happened?

HRTs as the next step
Analysis techniques evolution:

- Principle does not change
- Memory encryption
- Multiplexers mixed with the core

HRTs as the next step
Analysis techniques evolution:

* LCE evolution: hidden mux

HRTs as the next step

8 bits processor
32 bits FLASH output going to the core
Analysis techniques evolution:

LCE evolution: hidden mux

HRTs as the next step

Lines have to be traced inside the core to find the 8 bits data bus.
Analysis techniques evolution:
LCE evolution: hidden mux

3 paths can be followed:
2 of them can not be exploited
Analysis techniques evolution:

LCE evolution: hidden mux

- Finding the correct spot took some time
- Multiplexers were hidden
- Data was not encrypted

HRTs as the next step
Analysis techniques evolution:

LCE evolution: state of the art

- Multiplexers are hidden
- NVM content is scrambled
- NVM content is encrypted
- Hardware custom functions are implemented as part of the core
- Several thousands gates have to be reversed

HRTs as the next step
Analysis techniques evolution:
ROM reading: ROM “optical reading”

HRTs as the next step
Analysis techniques evolution:

ROM reading: principle

- Define 4 corners for alignment
- Affine transformation to compensate “tilt deformation”
- Define horizontal bit spacing
- Define vertical bit spacing
- Choose criteria for bit value
- Extract defined zone

HRTs as the next step
Analysis techniques evolution:

ROM reading: correlation issue

As ROMs are getting bigger, correlation errors have to be considered

HRTs as the next step

4700 pictures have to be stitched
Analysis techniques evolution:

ROM reading: correlation issue

Smarter procedure:
- Do not try correlating pictures (especially SEM pics) of a large scan
- Do not try to tell your script where the bits are
- Find bits corresponding to a noticeable value
- Extract a grid from their position
- From the grid, recover the missing bits
- Correlate bits from an image with those of the adjacent one and so on
Sample preparation and imaging evolution:  

**Deprocessing:**

By using plasma etching as the only technique for deprocessing, picture quality is poor.
Sample preparation and imaging evolution: HRTs as the next step

Deprocessing:

Using combination of techniques such as Plasma etching, Chemical Mechanical Polishing and wet chemical etching leads to “perfect” deprocessing, suitable for SEM scan.
Sample preparation and imaging evolution: HRTs as the next step

Deprocessing:

- One layer visible at a time
- Vias also visible
- Custom process to distinguish vias and lines has been defined
Sample preparation and imaging evolution: HRTs as the next step

SEM imaging:

Major issue was found and solved: SEM picture distortion
- Tilt adjustment table has been machined
- Fast scan
- Distortion is calculated for a given scan and reversed
Sample preparation and imaging evolution:

Features on grid:

HRTs as the next step
Sample preparation and imaging evolution:  
Find vias:

HRTs as the next step
Sample preparation and imaging evolution:  
*HRTs as the next step*

Find lines:
Sample preparation and imaging evolution: **HRTs as the next step**

**Accurate correlation**

- Correlation is performed on feature coordinates “grid pattern”
- At worst, lines are “jittering” around the calculated grid position
Sample preparation and imaging evolution: HRTs as the next step
Accurate correlation
Sample preparation and imaging evolution:

2 layers example:

HRTs as the next step
Sample preparation and imaging evolution:

All information available

- Vias and lines are extracted on a grid
- Gates are detected from the same mechanics
- No correlation error
- Layers are aligned “perfectly” without further picture transformation
- No more pixels – polygons only 😊

HRTs as the next step
Hardware Reverse-engineering Tools outcomes
New possibilities:

Some possible studies

- semi-invasive preparation
- LCE preparation
- Shield global bypass
- Other embedded counter-measures bypass

HRTs’ outcomes
New possibilities:
Other techniques:

- Photoemission
- EMA
- Dynamic voltage contrast

HRTs as the next step
New possibilities:

New tools

FIB navigation files can be generated
- Planarised chip
- Backside edit

HRTs’ outcomes
New threats:
Laser fault injection:

Particular gates can be highlighted without any further study
- Fire at the located registers and see the effects

Tracing signals is easy as a click
- Fire first, with for example a pass-fail scan
- Look at what you hit at “fail” location
- Understand the effect

=> From laser glitching to laser fault injection.
New threats:
Laser fault injection become cheaper
New threats:
Scan chains analysis:

- Path chain are very easy to spot
- Used to debug / program the device

HRTs’ outcomes
Timing consideration:

Real world example:

- ROM chip
- ROM is scrambled
- Multiplexers are hidden inside the logic
- ROM is encrypted
- Data encryption based on address and hard-wired key

HRTs’ outcomes

- Clear data bus location?
- Custom encryption reverse?
Timing consideration:

Image preparation: “manual process”

HRTs’ outcomes

Correlation is based on pixel value:
- From 10 minutes to several hours
- Errors are inevitable

Image stitching is not reliable
- One picture = one photoshop layer
- Local adjustments are performed when needed

Alignment of 2 layers almost unfeasible but fast
- Local adjustments are performed when needed
Timing consideration:
Finding data bus: “manual process”

- Tons of layers are used and moved for local adjustment: Errors

- A schematic must be drawn to avoid being lost: Errors + you will get lost anyway
Timing consideration:

Finding data bus: “manual process”

- Each found gate must be analyzed even if already studied: Errors + stay patient

- Equations have to be written in “mathematical form”: Many errors

\[
\begin{align*}
\text{cell}_3_{\text{12}} &= (\text{cell}_9_{\text{24}} \oplus \text{cell}_9_{\text{18}}) \oplus \text{cell}_1_{\text{18}} \\
\text{cell}_3_{\text{13}} &= (\text{cell}_9_{\text{20}} \oplus \text{cell}_9_{\text{19}}) \oplus \text{cell}_1_{\text{23}} \\
\text{cell}_3_{\text{13bis}} &= (\text{cell}_3_{\text{14}} \oplus \text{cell}_9_{\text{25}}) \oplus \text{cell}_5_{\text{2}} \\
\text{cell}_3_{\text{14}} &= (\text{cell}_8_{\text{4}} \oplus \text{cell}_g_{\text{1}}) \oplus \text{cell}_6_{\text{7}} \\
\text{cell}_3_{\text{15}} &= (\text{cell}_{\text{alpha}_1} \oplus \text{cell}_9_{\text{40}}) \oplus \text{cell}_1_{\text{36}} \\
\text{cell}_3_{\text{16}} &= (\text{cell}_{\text{a}_1} \oplus \text{cell}_1_{\text{19}}) \oplus \text{cell}_1_{\text{36}} \\
\text{cell}_3_{\text{17}} &= (\text{cell}_9_{\text{21}} \oplus \text{cell}_9_{\text{22}}) \oplus \text{cell}_1_{\text{24}}
\end{align*}
\]
Timing consideration:

Finding data bus: “manual process”

-> Finally, with help of vhdl software (for example), schematic can be re-arranged to understand the functions.

- Localization of the clear data bus is possible
- LCE is working

-> My FIB is down but I have reverse-engineered every single gate, I can read the ROM...

- VHDL simulations will show.... that there are errors 😞
- Localizing the errors can partially be made from simulations
- Where are the last errors?
Timing consideration: Deprocessing

- Deprocessing for hardware reverse-engineering takes extra steps
- This process is not suited for optical imaging
- Complete deprocess can be achieved in about a week

HRTs’ outcomes
Timing consideration:
Image preparation: with HRT

- Select area of interest and selection of rejected features (10 minutes per layer)
- Find vias (> 100 images per minutes)
- Find grid and lines (< 1 hour per layer)
- Extract gates (1 hour)
- Correct one layer: (1 day)
- Correlate and transform pictures + generates layer netlist (< 1 hour)
- Align 2 layers together (2 minutes)
**Timing consideration:**

Reverse-engineering custom logic: with HRT

- One layer per layer
  - No stitching problem
  - No local layer adjustment
- One click to follow net(s)
- Equations are generated automatically as well as schematic
  - No re-writing errors, software is highlighting what is missing

**HRTs’ outcomes**
Timing consideration:

Timing:

- After 2-3 weeks, every features are detected with good HRT
- 2 weeks later (average), LCE can be started
- Same work with « manual process » would take months
Timing consideration:

Timing difference: reversing custom logic

- 6 months after start of the study, results are still not exploitable

- With first generation of HRT, same study was performed in 2 months

- With next generation of the tool, time will be reduced to 1 month
  * With classical method, you would not have found the correct spots for LCE at this stage
New threats: Possible “achievements”:

- 100% success rate for hackers (excluding customized chip)
- 6 to 12 microcontrollers a year (first extraction)
- XX customized chip a year

⇒ The advanced security level becomes at best adequate
  Custom hardware functions become a new kind of ROM that could be extracted from pictures only

⇒ Piracy

⇒ Counterfeiting

⇒ Patent violation
New opportunities:

- Better security level: in depth security evaluation with new techniques
- Design and routing new strategies to make invasive work more complicated
- Anti-piracy by changing the nature of the hardware custom functions
- Affordable patent violation studies
- Counterfeiting characterization
- ICs’ obsolescence

HRTs’ outcomes
Future developments
Future developments

- Schematic creation and interaction
- From gates to functions: automatic gates grouping to reduce number of blocs to study
- Fast detection of « non aligned » features: from core to chip
- Simulator, specific analysis
- ...

Texplained
Hardware Reverse-engineering Tools
new threats – new opportunities

Q&A...