How to Build Hardware Trojans

TrustEd 2014 – Trustworthy Embedded Devices
Scottsdale, November 3, 2014

Christof Paar
Ruhr Universität Bochum & University of Massachusetts Amherst
Acknowledgement

- Georg Becker
- Pawel Swierczynski
- Marc Fyrbiak
- Philipp Koppe
Agenda

- Introduction to Hardware Trojans
- Sub-Transistor ASIC Trojans
- FPGA Trojan
- Auxiliary Stuff
Agenda

- Introduction to Hardware Trojans
  - Sub-Transistor ASIC Trojans
  - FPGA Trojan
- Auxiliary Stuff
Hardware Trojans

Malicious change or addition to an IC that adds or remove functionality, or reduces reliability

Many rather unpleasant “applications”
Defense Science Board
Task Force
On
HIGH PERFORMANCE MICROCHIP SUPPLY

February 2005
Hardware Trojans & the Scientific Community

Publications w/ „Hardware Trojans“ or „malicious Hardware“ (Google Scholar, Aug 2013)

- only title
- in paper

2007 2008 2009 2010 2011 2012

0 15 15 17 18 32 34

133 167 199
Trojan Injection & Adversaries Scenarios

DoD scenario 2005

- **Manufacturing**
  - Malicious factory, esp. off-shore (foreign Government)

- **Design Manipulation**
  - 3rd party IP-cores
  - malicious employee

Not-so-unlikely 2013

- **During shipment**
  - cf. NSA’s interdiction

- **Built-in**
  - backdoors etc.
Related to Trojans: Product Piracy and Counterfeits

Products sold under false brand name.

Allegedly up to 1/3 of all SanDisk MircoSD cards are fakes.

Lenovo Y1011 Tablet: Lenovo does not produce such a table, but are available on the Internet.

Micron Elpida DRAM Memory Chips: Identically-looking fake chips (but serial numbers differ)

https://thecounterfeitreport.com/
Where are we with “real” HW Trojans?

- No true hardware Trojan observed in the wild

- All examples from academia

- Vast majority of publications focus on detection
Proposed Trojan Detection Methods

- Formal verification
- Functional testing
- Optical inspection
- Side-channels
- Trojan detection circuitry
Agenda

- Introduction to Hardware Trojans
- Sub-Transistor ASIC Trojans
- FPGA Trojan
- Auxiliary Stuff
Our Thoughts ca. 2012

1. *Designing* Trojan could be fun too
2. Especially those that go *undetected*
Dopant-level Hardware Trojans

Main idea
Change the design at sub-transistor level

Why?
• Poorly studied in the literature
• Malicious factories and OEM are realistic threat model (perhaps “motivated” by local government)
• Can yield extremely stealthy Trojans with zero overhead

⇒ Many detection methods (optical, side-channel) will fail
Simple Example: Inverter Trojan

Let’s modify an inverter so that it always outputs “1” (VDD) without visible changes.

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
PMOS Transistor Trojan

Unmodified PMOS transistor

Trojan trans. w/ constant VDD output
Inverter with Trojan PMOS transistor

short circuit if A=1
... avoid GND connection by NMOS transistor
NMOS Transistor Trojan

Unmodified NMOS transistor

Trojan transistor w/ floating output
“Always One” Trojan Inverter

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1. PMOS transistor (top) acts as closed switch
2. NMOS transistor (bottom) acts as open switch

Q: Can the Trojan be detected?
Layout view of Trojan inverter

Which has the Trojan?

Original Inverter

“Always One” Trojan

Unchanged:

- All metal layers
- Polysilicon layer
- Active area
- Wells

⇒ Dopant changes (very ?)
difficult to detect using optical inspection!
“Small remaining question”

• Unfortunately, circuits will not function correctly with this simple stuck-at fault ...

• ... functional testing (after manufacturing) will detect fault right away

Can we build a **meaningful** Trojan using dopant modifications that passes functional testing?

• Are their circuits parts that are not fully tested?
Intel’s Ivy Bridge RNG design

Entropy Source

Online Health Test (OHT)

Conditioner (Based on AES)

Build-In Self Test (BIST)

256 bit state

Rate Matcher (Based on AES)

Get Status

RnRand

Dopant Trojan
Rate Matcher, simplified  
(digital post-processing)

- Rater Matcher uses AES in counter mode
- Stage registers k and c contain true random bits
- Stage registers k and c are updated after each output
**Trojan Rate Matcher**

- Trojan: registers \( k \) only holds constant bits
- Trojan: \((128-n)\) bits of register \( c \) are also fixed
- output of RNG depends **only on** \( n \) random bits
- for \( n=32 \), RNG still passes NIST random number tests

32 bits entropy → very insecure keys are generated
How to avoid detection by built-In self test

For security reasons only the BIST is used for functional testing.

Due to clever choosing of the Trojan constants

Known, fixed input

256 bit state Rate Matcher (Based on AES)

512 bits → CRC Checksum

32 bits → Reference Checksum

Known, fixed input

TROJAN Rate Matcher (Based on AES)

512 bits → CRC Checksum

32 bits → Reference Checksum

= ≠
Meaningful hardware Trojans are possible without extra logic
Many detection techniques don’t guarantee a Trojan free design!
Built-in self tests can be dangerous
More details:
Becker, Regazzoni, P, Burleson, *Stealthy Dopant-Level Hardware Trojans*. CHES 2013

... but the scientific community functions as it is supposed to do:

Trojan detection is possible w/ scanning electron microscope
Sugawara et al., *Reversing Stealthy Dopant-Level Circuits*. CHES 2014
Agenda

- Introduction to Hardware Trojans
- Sub-Transistor ASIC Trojans
- FPGA Trojan
- Auxiliary Stuff
Principle Ways of Realizing Digital Systems

System (e.g., AES)

Software

+ fast development, flexible
  - energy, performance

ASIC

+ performance, low energy
  - high NRE, inflexible

FPGA

+ performance, development + low NRE
  - device costs
  - programming complex
FPGAs are widely used

appr. 5b devices sold in 2013
FPGAs = Reconfigurable Hardware

Configuration file “bitstream”
Can an we build *hardware* Trojans by manipulating the bitstream?
Algorithm Substitution Attack

Idea: Replace regular (strong) crypto alg. with a weak one

AES

00101
11101
10110

AES*

00101
00111
10110

weak version of AES, allowing key extraction etc.
The Mechanics of FPGAs

Configuration bitstream is
a) large and complex
b) proprietary

How can we alter the bitstream?
... but detecting S-boxes in bitstreams is possible

- S-boxes are realized as 6x1 look-up tables (LUTs)
- LUTs can be found in bitstream
- S-box contents is very specific (luckily)
AES manipulation in practice

Works for many different implementations of AES...

<table>
<thead>
<tr>
<th>Impl.</th>
<th>Architecture</th>
<th>AES</th>
<th>LUTs with S-box logic</th>
<th>S-boxes in memory</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Round-based</td>
<td>128</td>
<td>(16+4)·32 = 640</td>
<td>no</td>
<td>100 %</td>
</tr>
<tr>
<td>#2</td>
<td>Round</td>
<td>128</td>
<td>0</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#3</td>
<td>Round</td>
<td>192</td>
<td>0</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#4</td>
<td>Round</td>
<td>256</td>
<td>0</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#5</td>
<td>Round-based</td>
<td>128</td>
<td>(0+4)·32 = 128</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#6</td>
<td>Round-based</td>
<td>128</td>
<td>0</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#7</td>
<td>Round-based</td>
<td>128</td>
<td>0</td>
<td>yes</td>
<td>100 %</td>
</tr>
<tr>
<td>#8</td>
<td>Round-based</td>
<td>128</td>
<td>(16+4)·32 = 640</td>
<td>no</td>
<td>100 %</td>
</tr>
</tbody>
</table>

**TABLE IV: Overview of evaluated AES implementations**
Algorithm substitution attack and its implications

1. Replace S-boxes in flash with identity

AES*

2. weak AES is configured

CT = AES*(k, PT)

“Useful” attacks are still possible!

1. **Storage encryption** (Cloud, harddisk etc.)
   - Attacker can recover plaintext without access to k

2. **Recovery of secret k from CT**, by solving system of lin. equations
   2a) easy if attacker has complete control over the device
   2b) also possible for network encryption etc. if attacker can swap S-Boxes back-and-forth quickly, e.g., for 1 block PT
Conclusion

- New attack vector against FPGAs!
- Reconfigurability allows “hardware” Trojans designed in the lab
- Bitstream protection is crucial! (but not easy, cf. our work at CCS 2011 & FPGA 2013)
- Details at:
Agenda

- Introduction to Hardware Trojans
- Sub-Transistor ASIC Trojans
- FPGA Trojan
- Auxiliary Stuff
... yet another textbook on cryptography (but targeting non-mathematicians)

www.crypto-textbook.com

• includes 25 videos of 2-semester course (in English)

• complete set of slides

• many further resources

• ... flyers with discount
Thank you very much for your attention!

Christof Paar

Ruhr Universität Bochum & University of Massachusetts Amherst