Tamper Evident Microprocessors

Adam Waksman
Simha Sethumadhavan
Computer Architecture & Security Technologies Lab (CASTL)
Department of Computer Science
Columbia University

Modern Hardware is Complex

- Modern systems built on layers of hardware
- Complexity increases risk of backdoors
  - More hands
  - Easier to hide
- A significant vulnerability
  - Hardware is the root of trust
  - All hardware and software controlled by microprocessors

Prior Work and Scope

- Microprocessor design stages
  - Front End: Specification, High-Level Design, Logic Synthesis, Place and Route, Layout, Detailed Implementation, Detailed Layout, Placement, and Routing
  - Back End: Physical Design, Tapeout, Fabrication
- Prior work focuses on back end
  - More immediate threat
    - Example: IC fingerprinting [Agrawal et al., 2007]
- Front end is the extreme root
  - Common assumption: golden model from front end
  - Focus of this work

Key Idea: Use Inherent Division of Work

- Bob
  - Nice Guy
  - Donates $100
- Eric
  - Evil Accountant
  - Steals $10
- Alice
  - Charity President
  - Receives $90

Microprocessor Pipeline Stages Analogue

<table>
<thead>
<tr>
<th>Front End</th>
<th>Back End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>Physical Design</td>
</tr>
<tr>
<td>High-Level Design</td>
<td>Tapeout</td>
</tr>
<tr>
<td>Logic Synthesis</td>
<td>Fabrication</td>
</tr>
<tr>
<td>Place and Route</td>
<td></td>
</tr>
<tr>
<td>Layout</td>
<td></td>
</tr>
<tr>
<td>Detailed Implementation</td>
<td></td>
</tr>
<tr>
<td>Detailed Layout</td>
<td></td>
</tr>
<tr>
<td>Placement and Routing</td>
<td></td>
</tr>
</tbody>
</table>

Outline

- Taxonomy
  - Ticking Timebombs, Cheat Codes, Emitters, Corrupters
- Solutions
  - TrustNet and DataWatch
- Results
  - Correctness, Coverage and Costs
- Future Work

Taxonomy of Attacks

- Backdoor = Trigger + Payload
  - Trigger: Turns on an attack
  - Payload: Malicious, illegal action
Taxonomy of Attacks: Triggers

- Data
- Time

Taxonomy of Attacks: Payloads

- Emitter Attacks
  - Extra malicious events
  - Separate from normal events
- Corrupter Attacks
  - No extra malicious events
  - Normal instructions altered

Taxonomy of Attacks: Summary

- Emitter Timebomb
- Corrupter Timebomb
- Emitter Cheatcode
- Corrupter Cheatcode

Assumptions

- Large design team
  - Each designer works on one unit or part of one
  - Security add-ons cannot be done by one member
- Full knowledge
  - Attacker has complete access to all design specifications
  - Attacker also knows about additional security mechanism
- Equal distrust
  - Any one designer/unit may be evil
  - Security add-ons may contain backdoors

Outline

- Taxonomy
  - Ticking Timebombs, Cheat Codes, Emitters, Corrupters
- Solutions
  - TrustNet and DataWatch
- Results
  - Correctness, Coverage and Costs
- Future Work

Sample Emitter Backdoor

- Consider a malicious instruction decoder
- Decoder emits instructions not in the original program
- Execution unit faithfully executes them
TrustNet

- Predictor and Reactor monitor the Target
- Division of work prevents one bad guy from breaking two units
- Scaling to larger number increases design complexity

Corrupter Backdoors

- Bob
  - Still nice
  - Donates $100

- Eric
  - Evil (and smarter)
  - Converts to Canadian $

- Alice
  - Still president
  - Fooled by Eric’s C$100

DataWatch

- Scaled up version of TrustNet
- Multiple bit messages
- Confirms types of messages (instead of just yes/no)

Outline

- Taxonomy
  - Ticking Timebombs, Cheat Codes, Emitters, Corrupters

- Solutions
  - TrustNet and DataWatch

- Results
  - Correctness, Coverage and Costs

- Future Work

Experimental Context, Correctness, Costs

- Context
  - Simplified OpenSPARC T2

- Correctness
  - Designed attacks
  - No false positives or negatives

- Costs
  - Low area overhead (2 KB per core)
  - No performance impact

- How to measure coverage?

Coverage: Vulnerability Space

Paper has plots for other units at a chip level
Summary and Future Work

- **Strengthen root of trust: microprocessors**
  - Hardware-only solution. No perf impact, low area overhead
  - Security add-on highly resilient to corruption
- **Applicability of TrustNet & DataWatch**
  - Covered: pipelines, caches and content associative memory
  - Not covered: ALU, microcode, power mgm., side-channels
- **Moving Forward**
  - Expand coverage
  - Out-of-order processors
  - Motherboard components
  - Design automation tools
  - Reaction to errors
  - Applying techniques for reliable execution
  - **First steps toward a secure trusted hardware w/ untrusted units**

Thank You! and Questions?