**Overview**

- Executive Summary
- Hardware Attack Basics
- Paper Summary
- Strengths
- Weaknesses
- Questions and Discussion
Executive Summary

• **Motivation:** Modern processors are often manufactured by third party foundries, which creates the risk of **Hardware Attacks** on the chips by some malicious party

• **Observation:** Known, digital **Hardware Attacks** are either big & power hungry and therefore easy to detect, or they have limited capabilities

• **Goal:** Show that **small & stealthy** Hardware Attacks are possible by using **analog components**

• **A2:** Novel **analog trigger** for hardware attacks **using capacitors** as its main component

• **Evaluation:**
  - Simulation and implementation in real hardware
    - The attacks can be triggered by user level programs
    - No accidental triggering from normal usage
  - Testing effectiveness of existing detection mechanisms
    - A2 is good at avoiding detection
Overview

• Executive Summary

• Hardware Attacks:
  • Basics
  • Attack vectors
  • Detecting attacks

• Paper Summary

• Strengths

• Weaknesses

• Questions and Discussion
Hardware Attack

- **Payload**: predefined action in the chip
  - Causing a crash
  - Activating supervisor mode
  - Leaking data

- **Trigger**: activates *payload* if chip in specific state
  - Specific wire activation count
  - Rare event happens
  - Always on
Counter-Based Hardware Attack

on_every(RBACE) do
    if(count == 12345) then
        do_attack()
    else
        count = count + 1
    end
end

RBACE := Rare But Attacker Controllable Event
How Can Hardware Attacks Happen?

• Several Attack Vectors
• Later steps leave fewer options for attacker
  • Paper: Focus on attack during manufacturing
Detecting Hardware Attacks

• Detectable at any point after insertion
  • Design Time Attacks:
    • Verification
    • Simulation
    • Post-fabrication testing
  • Fabrication Time Attacks:
    • Post-fabrication testing
Post-Fabrication Testing

• Run **tests**, check functionality

• Use **side-channel information** for more subtle changes
  • Timing changes
  • Power draw

• **Delayer** a chip, then search for unexpected gates with an electron scanning microscope
  • Expensive & time consuming
  • Can be focused critical parts

 Delayered Intel Itanium CPU (2001)
 Source: Youtube, Der8auer
Overview

- Executive Summary
- Hardware Attack Basics

**Paper Summary:**
- Goal & Key ideas
- Design & Implementation
- Testing the attack
- Defenses
- A2 vs x86

- Strengths
- Weaknesses
- Questions and Discussion
Goal of the Paper

• Show that small & stealthy hardware attacks are possible

Key Ideas

• Design a simple counter-based trigger with capacitors
• Enable complex triggers by logically combining simple triggers
Overview

• Executive Summary

• Hardware Attack Basics

• Paper Summary:
  • Goal & Key ideas
  • Design & Implementation
  • Testing the attack
  • Defenses
  • A2 vs x86

• Strengths

• Weaknesses

• Questions and Discussion
Counter-Based Hardware Attack

\begin{align*}
on\_\text{every}(\text{RBACE}) & \text{ do} \\
\text{if}(\text{count} == 12345) & \text{ then} \\
\text{do\_attack}() & \\
\text{else} & \\
\text{count} = \text{count} + 1 & \\
\text{done} &
\end{align*}

\text{RBACE} \equiv \text{Rare But Attacker Controllable Event}

Needs a lot of digital circuitry
Single Stage Trigger (1 capacitor)

Problems:

- Very small capacitance (~$10^{-14}$ Farad) due to manufacturing and size constraints
- Needs currents in the nano-Ampere range
- High variability across manufacturing process, temperature and voltage variations
- E.g. Leakage currents change with conditions

Solution: Use a second, smaller capacitor to charge main capacitor
Single Stage Trigger (2 capacitors)

\[ \Delta V = \frac{Cap_{unit}}{Cap_{unit} + Cap_{main}} \times (VDD - V_0) \]
Multi-Stage Trigger Circuit

• Use logical combinations of single-stage triggers
  • \((A \land B) \lor C\) ⇒ *Activate payload*

• Can make detection by accidental activation arbitrarily hard
Designing the Malicious Chip [1/3]

• Choose a target processor
  • Authors used 32-bit OpenRISC 1200 core
    • Performance comparable to an ARM10 processor (2000 – 2003)
    • Complexity much lower than modern x86 processor
Designing the Malicious Chip [2/3]

• Find **Victim Wires**:
  • Simulate chip to find low activity wires
  • $\approx 3\%$ of wires in OR1200 had nearly zero toggle rate
Designing the Malicious Chip [3/3]

• Find **empty space** to place trigger
  • With typical placement and routing, only $\approx 60\% - 70\%$ of chip area used
  • A2 is tiny
Implementation in Real Hardware

- 65 nm CMOS technology
Overview

- Executive Summary
- Hardware Attack Basics

**Paper Summary:**
- Goal & Key ideas
- Design & Implementation
- Testing the attack
- Defenses
- A2 vs x86

- Strengths
- Weaknesses
- Questions and Discussion
Activating the Attack (Pseudo Code)

- User level program
- Perform specific pattern to trigger attack
- Detect success by reading special register

```
{r0 is a non-zero register but reads as zero in user mode}
Initialize SR[0]=0 {initialize to user mode}

while Attack_Success==0 do
    i ← 0
    while i < 500 do
        z ← a/b {signed division}
        z ← c/d {unsigned division}
        i ← i + 1
    end while
    if read(special register r0) ≠ 0 then
        Attack_Success ← 1
    end if
end while
```

Actual code (and more) can be found here:
https://github.com/impedimentToProgress/A2
Testing Results

• Attacks can be triggered by malicious program
• No accidental triggering by 5 different benchmarks from MiBench
• Cycles-to-trigger were accurately simulated

<table>
<thead>
<tr>
<th>Toggle Rate (MHz)</th>
<th>Measured cycles-to-trigger (10 chip average)</th>
<th>Simulated cycles-to-trigger</th>
</tr>
</thead>
<tbody>
<tr>
<td>120.00</td>
<td>7.4</td>
<td>7</td>
</tr>
<tr>
<td>34.29</td>
<td>8.4</td>
<td>8</td>
</tr>
<tr>
<td>10.91</td>
<td>11.6</td>
<td>10</td>
</tr>
</tbody>
</table>
Trigger Consistency

- Trigger behavior changes depending on several factors:
  - Temperature
  - Voltage
  - Frequency
Overview

• Executive Summary

• Hardware Attack Basics

• Paper Summary:
  • Goal & Key ideas
  • Design & Implementation
  • Testing the attack
  • Defenses
  • A2 vs x86

• Strengths

• Weaknesses

• Questions and Discussion
Existing Defenses VS A2

• **Detection by Testing unlikely** if trigger sequence is complex enough

• Using **power draw side-channel** information:
  • Additional power draw only when victim wire is toggled
  • Low activity in standard use means low additional power draw
  • For A2, extra power draw less than normal power fluctuations

• **Delayering and imaging** the chip can detect A2, but:
  • A2 trigger is way smaller than previous digital attacks, making discovery difficult

<table>
<thead>
<tr>
<th>Program</th>
<th>Power [mW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Math</td>
<td>23.703</td>
</tr>
<tr>
<td>Unsigned Division</td>
<td>23.206</td>
</tr>
<tr>
<td>Two-stage Attack</td>
<td><strong>22.575</strong></td>
</tr>
<tr>
<td>Search</td>
<td>21.960</td>
</tr>
<tr>
<td>Single-stage Attack</td>
<td><strong>19.505</strong></td>
</tr>
<tr>
<td>FFT</td>
<td>18.120</td>
</tr>
<tr>
<td>SHA</td>
<td>18.032</td>
</tr>
<tr>
<td>Dijkstra</td>
<td>16.550</td>
</tr>
<tr>
<td>Standby</td>
<td>6.210</td>
</tr>
</tbody>
</table>
Other Possible Defenses

• **Split Manufacturing**
  • Only have part of the chip (FEOL) manufactured externally
  • Wires (BEOL) & Final Assembly are done by **trusted foundry**
  • As shown in paper “Is Split Manufacturing Secure?”, attacker can still reconstruct \( \approx 96\% \) of all wires
  • A2 attacks a Flip-Flop, which is still possible

• **Eliminating empty space** to prevent insertion of attack logic
  • A2 is small, need to fill **ALL** empty space with **non-redundant** circuitry, otherwise attacker can replace some parts of it
  • Less empty space \( \Rightarrow \) routing wires gets harder and performance may decrease
Overview

• Executive Summary

• Hardware Attack Basics

• Paper Summary:
  • Goal & Key ideas
  • Design & Implementation
  • Testing the attack
  • Defenses
  • A2 vs x86

• Strengths

• Weaknesses

• Questions and Discussion
Danger to x86 Processors

• A2-like attack on x86 processor might be easier to implement and harder to detect than on OR1200
  • **Complexity of trigger** not dependent on processor complexity
  • More viable **victim wires** expected (highly-variable and controllable activity)
  • Redundant functional units might make attack **probabilistic**
Overview

- Executive Summary
- Hardware Attack Basics
- Paper Summary

- Strengths

- Weaknesses

- Questions and Discussion
Strengths

• First paper to show how analog triggers can create powerful hardware attacks
  • Smaller & more stealthy than previous attacks
  • Does not interfere with design phase
• Implementation and testing in actual hardware
• Proposals for possible defenses for future research
Overview

- Executive Summary
- Hardware Attack Basics
- Paper Summary
- Strengths

- Weaknesses

- Questions and Discussion
Weaknesses

- Trigger needs to be redesigned for each process
  - Requires a lot of design work
  - Smaller nodes will decrease capacitance

- Power consumption side channel poorly explained
  - Only compared within malicious chip

- Unverified information:
  - Size of digital version of A2 mentioned, but no design shown
  - Mention of A2 trigger having 0.08% of chip’s area, but no calculation

<table>
<thead>
<tr>
<th>Program</th>
<th>Power [mW] (malicious chip)</th>
<th>Power [mW] (normal chip)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Math</td>
<td>23.703</td>
<td>??</td>
</tr>
<tr>
<td>Unsigned Division</td>
<td>23.206</td>
<td>??</td>
</tr>
<tr>
<td>Two-stage Attack</td>
<td>22.575</td>
<td>??</td>
</tr>
<tr>
<td>Search</td>
<td>21.960</td>
<td>??</td>
</tr>
<tr>
<td>Single-stage Attack</td>
<td>19.505</td>
<td>??</td>
</tr>
<tr>
<td>FFT</td>
<td>18.120</td>
<td>??</td>
</tr>
<tr>
<td>SHA</td>
<td>18.032</td>
<td>??</td>
</tr>
<tr>
<td>Dijkstra</td>
<td>16.550</td>
<td>??</td>
</tr>
<tr>
<td>Standby</td>
<td>6.210</td>
<td>??</td>
</tr>
</tbody>
</table>
Overview

- Executive Summary
- Hardware Attack Basics
- Paper Summary
- Strengths
- Weaknesses

• Questions and Discussion
Questions?
Discussion Starters

• Is additional trusted hardware needed for protection of chips?

• How would you rate the threat to modern processors?
  • e.g., x86, ARM

• What can be done in software?
Thanks for your attention!