A Survey on Hardware Security Techniques Targeting Low-Power SoC Designs

Alan Ehret\textsuperscript{1}, Dr. Karen Gettings\textsuperscript{2}, Bruce R. Jordan Jr.\textsuperscript{2}, Dr. Michel A. Kinsy\textsuperscript{1}

\textsuperscript{1}Boston University, Boston, MA
\textsuperscript{2}MIT Lincoln Laboratory, Lexington, MA
Introduction

- Embedded systems are integral part of modern life
- More devices are being made “Smart”
- Connected systems provide opportunities for theft, denial of service, etc.
Security Challenges

- IoT deployment environment
  - Unattended
  - Physical access available
- Attackers may have unhindered access to devices for long durations
- Tight power/area budgets limit overhead available for security
Why Not Anti-Tamper?

- Increases size, weight, power consumption
  - Must always check for intrusions
- Adding security directly to IC hardware is more efficient
  - Circuit level security modules may be shut down with device

Point of sale device anti-tamper PCB – hackaday.com
Defense Categories
Defense Categories

- Processing Elements

Diagram: Diagram shows various components including CPUs, Off-Chip RAM, Off-Chip NVM, I/O, NoC Interconnect, and On-Chip Cache/RAM.
Defense Categories

- Processing Elements
- Volatile Memory
Defense Categories

- Processing Elements
- Volatile Memory
- Non-Volatile Memory
Defense Categories

- Processing Elements
- Volatile Memory
- Non-Volatile Memory
- NoC Interconnects
Processing Element Defenses
Secure Enclaves

- Dedicated hardware core
  - Isolate sensitive data
- Commercial examples
  - Apple SEP
  - ARM TrustZone
- Mitigates:
  - User and kernel software vulnerabilities
  - Application Processor side-channels

Execution Obfuscation - Ascend

- Mitigate side-channels in cloud environment
  - Power, I/O, Timing

- Honest but curious cloud provider


Department of Electrical & Computer Engineering
Physical Unclonable Functions

- Use variation in manufacturing to uniquely ID a device
- Prevent impersonation of hardware
  - Silicon Fingerprints
- Useful for hardware based:
  - Key generation/storage
  - Device authentication
Volatile Memory Defenses
Memory Encryption & ORAM

- Prevent main memory leaks
- ORAM has greater overhead than memory encryption

<table>
<thead>
<tr>
<th></th>
<th>Obfuscate Contents</th>
<th>Obfuscate Addresses</th>
<th>Obfuscate Timing*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Encryption</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oblivious RAM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

*ORAM does not necessarily obfuscate timing but some implementations (see Ascend in [9]) add this functionality to obfuscate all aspects of the memory.
Non-Monopolizable Caches

- Reserve cache ways for threads of execution
  - Low hardware overhead
  - Moderate performance overhead
- Mitigate cache timing side-channels
  - Prevent threads from evicting each other's data

Non-Volatile Memory Defenses

Diagram showing the interconnection of Off-Chip RAM, Off-Chip NVM, CPUs, I/O, NoC Interconnect, and On-Chip Cache/RAM.
Full Disk Encryption

- Protects Data-at-Rest
- Encryption key stored in RAM
- Data on disk protected while machine is off
- Key could be leaked by side channel
- Self Encrypting Drives (SED) vulnerable to Hot-Swap attacks
  - Requires physical access

Network-on-Chip Defenses
NoC Interconnect

- Use IP from different sources
- Run multiple applications with different trust levels
  - Must prevent misuse of hardware resources
- NoC performs permission checks on traffic
  - Support virtual isolation of software stacks

Department of Electrical & Computer Engineering
NoC Interconnect

- Permissions stored in RAM
- Permission cache used in Memory Protection Unit
  - Improves performance


Department of Electrical & Computer Engineering
## Defense Comparisons

<table>
<thead>
<tr>
<th></th>
<th>Limited Use Memory Encryption</th>
<th>Memory Encryption</th>
<th>Disk Encryption</th>
<th>Cache Arch.</th>
<th>ORAM</th>
<th>NoC Isolation</th>
<th>PUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Based Cache Side-Channel</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power-Based Cache Side-Channel</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Reverse Engineering</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In-Memory Data Theft</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>On-Chip Data Theft</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Data at Rest Theft</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Counterfeiting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>On-Chip DoS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

- Low-power connected systems vulnerable to a variety of attacks
  - Increased potential for theft, denial of service
- No one-size-fits-all solution
  - Must analyze threat model for each system
  - Consider size, weight, power budgets
References


