3Dsec: Trustworthy System Security through 3D Integrated Hardware [presentation]

Huffmire, Ted

3Dsec: Trustworthy System Security through 3-D Integrated Hardware

Ted Huffmire
14 January 2009
What is Hardware Security?

• Many of the issues of hardware security are similar to traditional computer security
  – Malware, authentication, program analysis, patches, insiders, social engineering, developmental attacks, evaluation, certification and accreditation, flawed implementations, protocols, system-level issues, network security, usability, economic incentives, complexity.

• Anything can be hacked, but the attacker has finite resources.
  – Shades of grey rather than black-and-white (“broken” or not)
  – Make attackers toil, and design systems that “win” decisively
  – Each security technique has its advantages and disadvantages, and we must understand each technique's limitations. Even crypto has limitations. We need to know what specific attacks each technique is capable of preventing.
What is Hardware Security?

• Opportunities of hardware
  – High performance
    • Custom processors for crypto, deep packet inspection, etc.
  – Direct control
  – No intermediate OS layers
  – Physical separation

• Challenges
  – Semantic gap
  – Engineering and fabrication costs
What is Hardware Security?

• Foundry Trust
  – Malicious Hardware (a.k.a. “gate-ware”)
    • Trojan Horse, Rootkit, Kill Switch
  – Design Theft (Protecting Intellectual Property)
  – Start with a secure design before addressing fabrication security
• Operational Attacks
  – Power Analysis, Fault Injection, Heating, Optical
  – Cold Boot, Probing, Math Errors
• Developmental Attacks
  – Malicious Design Tools
  – Malicious IP
• System Assurance
  – Security Architecture, Key Management, PUFs
  – Formal analysis of IP cores (not a panacea)
What is Hardware Security?

• Interfaces [Schaumont 2009]
  – Secure hardware is part of a bigger system.
  – Secure hardware interfaces are tricky:
    • How do you distinguish red wires from black wires?
  – Secure hardware interfaces do not exist yet!
  – Current secure hardware serves software

• Composition [Schaumont 2009]
  – This is not trivial.
  – To resist side channels, you must avoid redundancy.
  – However, fault tolerance requires increasing redundancy.
  – How can you build fault-tolerant, side channel resistant systems?

• Metrics [Schaumont 2009]
  – Security is dimensionless.
  – Metrics are absolutely necessary to do meaningful research.
  – Without metrics, it is impossible to analyze trade-offs.

• Education: Electrical Engineers are trained to make things happen rather than to make bad things NOT happen [Schaumont 2009]
What is 3Dsec?

• Economics of High Assurance
  – High NRE Cost, Low Volume
  – Gap between DoD and Commercial

• Disentangle security from the COTS
  – Use a separate chip for security
  – Use 3-D Integration to combine:
    • 3-D Control Plane
    • Computation Plane
  – Need to add *posts* to the COTS chip design
    • Dual use of computation plane
3DSec: Trustworthy System Security through 3-D Integrated Hardware

**Goal:** Build trustworthy systems using commercial hardware components

**Problem:** Integrating specialized security mechanisms is too costly for hardware vendors

**Idea:** Augment commodity hardware *after fabrication* with a separate layer of security circuitry

**Anticipated Benefits:**
Configurable, protected, low-cost hardware security controls that can override activity in the commodity hardware

**Privacy Applications:**
Detect and intercept the execution of malicious code
Prevent the microprocessor internals from being exploited to leak crypto keys
Tag and Track private information as it flows through a processor
Pro’s and Con’s

• Why not use a co-processor? On-chip?

• Pro’s
  – Bandwidth
  – The ability to override

• Con’s
  – Cooling
  – Manufacturing yield
Cost

• Cost of fabricating systems with 3-D
  – Fabricating and testing the security layer
  – Bonding it to the host layer
  – Fabricating the vias
  – Testing the joined unit
Ramifications

• Device and Packaging Level
• Circuit Level
• Architectural
• Systems Software Level
• Application Level
Problems

• What if the control plane is manufactured in a different process than the computation plane?
• The computation plane must function as normal in the absence of the control plane, but must behave differently when the control plane is present.
• I/O and programming the control plane
• Actively interfering with the computation plane (timing issues)
• What to cut in the computation plane?
Goals

• Cost and performance analysis
  – Thermal/performance/cost model
  – FPGA prototypes
• Circuit-level modifications
  – Timing and power feasibility
  – Optimal placement/cost for commodity cores
  – Automatic insertion with CAD tools
    • Partition, place, optimize
Circuit-Level Modifications

- Passive vs. Active Monitoring
- Tapping
- Re-routing
- Overriding
- Disabling
Goals (Continued)

• Architectural structures
  – Memory (caches, predictors, scratchpad)
  – General logic (interconnect, controllers)

• End-to-end evaluation
  – Validate methods with design examples
  – Cache study, cache monitor
  – System-level security analysis
  – Dynamic management of 3-D control plane
Applications

- Information flow tracking
- Protection mechanism
  - Cache side channels
- Safe context switch
- Secure alternate service
- Instrumentation
- Optimal architecture for a given application
Broader Impact

- Economics of hardware security
- Use commodity processors
- Applicable to a wide variety of platforms
- Education
- Interdisciplinary scope
- Multiple universities
- Attract students to the field
- Student body diversity
Questions?

• faculty.nps.edu/tdhuffmi
Additional Slides

- Additional slides
Low-level architecture

- Routing through the 3-D control plane

![Diagram of 3-D control plane with metal layers, TSVs, contact points, and reference monitor logic. The diagram shows how signals are rerouted and how the bus is disabled by sleep transistors.]
Circuit-Level Modifications

- Tapping, Re-routing, Overriding, Disabling

![Diagram with symbols](attachment:images.png)

- X = Post to the 3-D control plane
- ▲ = Signal flow

(a) Tapping  (b) Re-routing  (c) Overriding  (d) Disabling
Applications

- Cache monitor

Computation Plane

Cache

CPU

Mem

3-D Control Plane

Cache Controller

Address

Tag

Index

Write Enable

Hit?

Data

R/W

Memory Data

PID

Security Bits

Locked?

Lock bit

Address

Grant

PID