Energy-Secure Architectures
A Wish List

Prof. Simha Sethumadhavan

Computer Architecture and Security Technologies Lab
Department of Computer Science
Columbia University

Invited Presentation at the 3rd Energy-Secure Architecture Workshop.

To cite this talk please cite: Columbia University Technical Report: cucs-017-13.

Hardware's Role in Security

- Security is a full-system property
  - Both hardware and software need to be secure
  - Traditionally, hardware hasn't received as much attention

- Hardware focus changes notions of what's possible
  - Attacks
    - Hardware is the root of trust
    - Compromised hardware difficult to detect thru software
  - Defenses
    - Hardware mechanisms offer smaller attack surface
    - Energy-efficient

- We've to design systems with security as a first-order design constraint along with other first-order design constraints (e.g., energy).

Outline

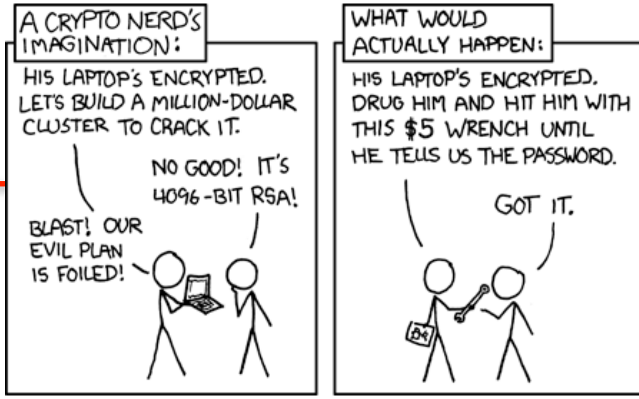
- Introduction to Security for Architects
  - Assumes no familiarity with Security
  - Three cornerstone security properties (CIA)
- Energy-Security wish list
  - Energy optimizations aggressively pursued now
  - Can energy optimizations cause security vulnerabilities?
  - Security added as an afterthought causes more problems than it solves.

Intro: Security is Challenging

Security is about protecting "assets"

- Motives for attack
  - Financial: Banks, petty theft (e.g., music), extortion
  - Information: Cyber offense, espionage (e.g., Flame, Duqu, Stuxnet)
  - Fame (Rare today)
  - Attackers will never go away
- Solutions are difficult: Large attack space
  - Security is a "full system" feature, many attack points
  - Attacker/defender incongruence:
    - Attacker has to find one or few weaknesses
    - Defender has to defend the complete system

A Key Point


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

- **Absolute security is a fool’s errand**
  - **Bribery, beating and burglary workarounds**
  - **Too many holes to plug**

- **Just like other architecture research “walls” there are no perfect or absolute solutions**

- **But**
  - ***You can secure against a threat model***
  - ***The goal is to raise the bar for the attacker***

Threat Model, Attacks and Attackers

- A threat model defines
  - threats that are/are not considered
    - E.g., We consider threats on the confidentiality and integrity of sensitive data, but not Denial of Service threats
  - assumptions of the computing model
  - Assumptions regarding attackers capabilities

- An attack is an instantiation of a threat

- Attacks violate security properties of a system

- Attacks have provenance
  - Vulnerabilities & entities that create them
  - Exploits & entities that carry out exploits

3 Cornerstone Security Properties

- Confidentiality: Protection against unauthorized reads
  - Prevent disclosure of information to an unauthorized entity
    - Example Attack: Eavesdropping

- Integrity: Protection against unauthorized writes
  - Prevent unauthorized modification without detection
    - Example Attack: Corruption attacks (code injection)

- Availability
  - System and services are available when requested by legitimate users
    - Example Attack: Denial of Service

More Security Aspects

- Access control policy
  - Specifies which principals can access which resources
  - Authentication and Authorization are essential aspects
- Accountability/Attribution
  - Specifies if/how actions can be tied to attacks
- Non-repudiation
  - Ability to hold one responsible to messages/events
- Anonymity
  - Ability to carry out actions without authentication
- Privacy
  - Right to determine how one’s personal information is distributed
- Distinction between Confidentiality and Privacy
  - Confidentiality is the obligation to protect secret information
  - Privacy is the right to preserve information
  - Privacy is a property of sentient beings

Sources of Vulnerabilities

Broadly three sources (applies to HW, SW, FW)

- Incomplete specifications
  - e.g., System was not designed for a threat
- Incorrect implementations
  - Implementation bugs weaken security
  - e.g., bad crypto implementation
- Improper policy
  - Admins and users don’t use the system correctly

- Each source can be further classified into:
  - Malicious vs. Unintentional
  - Malicious is typically harder than unintentional

Today’s Top 25 Software Errors

Category	Examples (from Top 25 CWEs)
Input Validation	CWE-79 (Cross-site scripting), CWE-89 (SQL inj Path traversal), CWE-98 (PHP file inclusion), CWE-434 (unre ith dangerous type),CWE-78 (OS Command injection), C ect), CWE-190 (Integer overflow)
Buffer Errors	CWE-120 (Classic buffer overflo CWE-805 (Buffer access with CWE-129 (Improper valid CWE-131 (Incorrect c
Permissions, Privileges and Access Control	CWE-285 (impr CWE-732 (in ment for critical resource) CWE-80 puts in a security decision) CWE gery)
Cryptographic Issues	r risky crypto algorithm) cryption of sensitive data) ad of code without integrity check)
Authentication Issues/ Credentials Manag	ssing authentication for critical function) (use of hard-coded credentials)
Resource Managem	WE-770 (allocation of resources without limits or throttling)
Race Condition	CWE-362 (race condition)
Information Leak/Disclosure	CWE-209 (information exposure through an error message)
Logic Error	CWE-754 (improper check for unusual or exceptional conditions)

Create a similar list of hardware vulnerabilities?

Exploits

- The set of vulnerabilities that have been used to successfully attack a system
- Requires great familiarity with low-level operational details of the system
- Are often reused
  - http://www.metasploit.com/

An Important Distinction

- Distinction between Security and Reliability

- System does not behave as expected because of:
  - Faults which have characteristic behavior (in Reliability)
  - Malicious attacker (in Security)

- Attackers are more intelligent than faults
  - In security, attackers will attack the protection mechanism
  - Attacks on AV system are quite common

Quiz

1. Based on the information presented in the tutorial so far, which of the following statements is not true:

- ❑ Each year Russian online mafia makes more money than Columbian drug cartels
- ❑ The goal of any security research is to guarantee absolute security
- ❑ Security is a full system property
- ❑ Today, security is a first-order design constraint for computer architects

2. What are the three fundamental security properties?

3. Fill in the blanks with three sources of vulnerabilities in hardware/software/firmware.

Incorrect __________
Incomplete _________
Improper ___________

4. True or false:

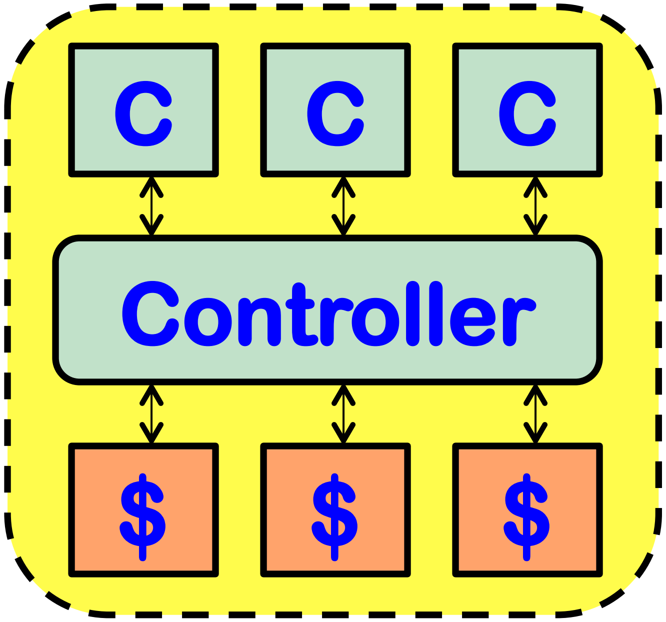
All security exploits leverage vulnerabilities, but not all vulnerabilities have been exploited.

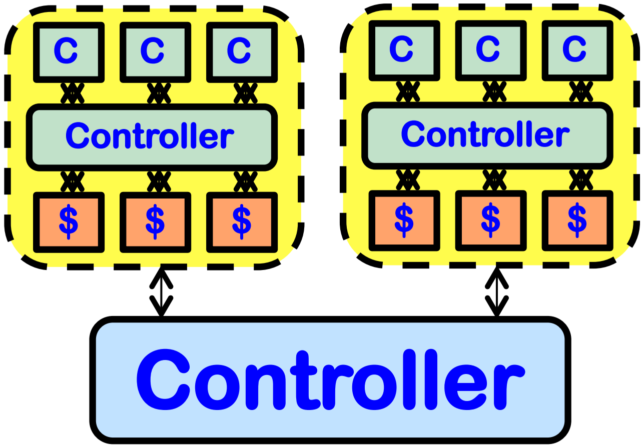
Energy-Secure Wish List

Let us answer the following questions:

- What security properties can be violated by energy optimizations?
    - Approach by analyzing 3 cornerstone security properties.
- What are the root causes for vulnerabilities in energy optimization algorithms? Are they exploitable?
- Can we quantify "risk" from vulnerabilities? Can we build tools to automatically discover vulnerabilities?
- Can we prove security properties in realistic systems?

Abstract View of Energy Mgmt.





For Each Epoch:

- Receive Activity Factors for Previous Epoch
- Estimate Activity Factor for Next Epoch
- Actuate Voltage, Frequency etc. based on Policy

Some Desirable Features:

- Allow VMM, OS, Programmer to control policy
- Co-ordination Control *(across HW/SW domains)*

Safeguard:

- Never actuate to an unhealthy V/F/T setting

A Confidentiality Vulnerability?

- Co-ordination is desirable to maximize efficiency
  - Spatial: across units on a chip
  - Temporal: across scheduling quanta
  - But co-ordination can be insecure!

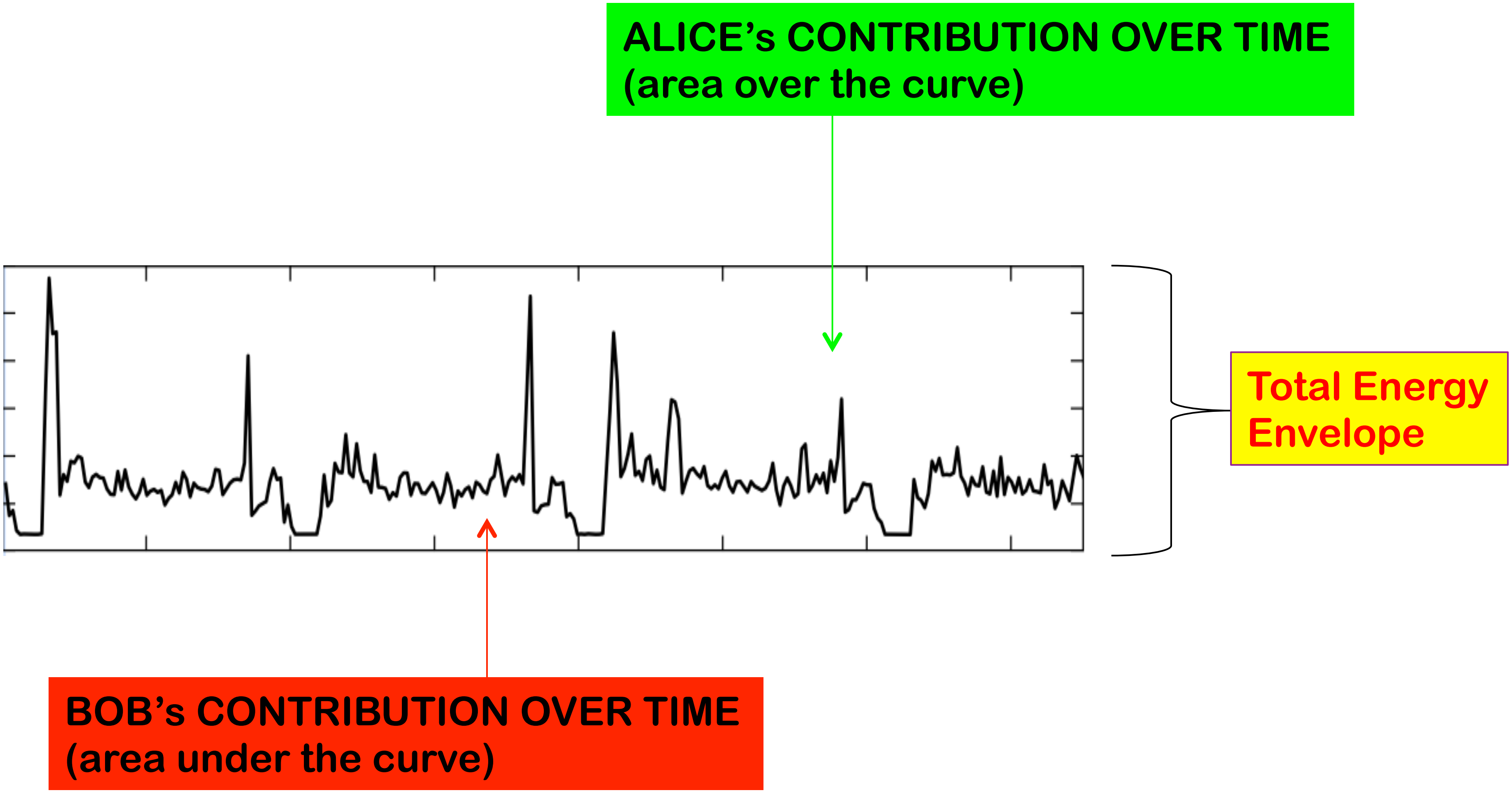
A hypothetical attack:

- Two VMs “Alice” and “Bob” share a processor.

- VM Bob may be able to determine what is running in VM Alice by monitoring his power, throughput, chip temperature, fault rate etc. across time.

- With finer-grained energy management Bob may even be able to steal Alice’s data (e.g., keys) across VMs!

Illustration



ALICE's CONTRIBUTION OVER TIME
(area over the curve)

Total Energy Envelope

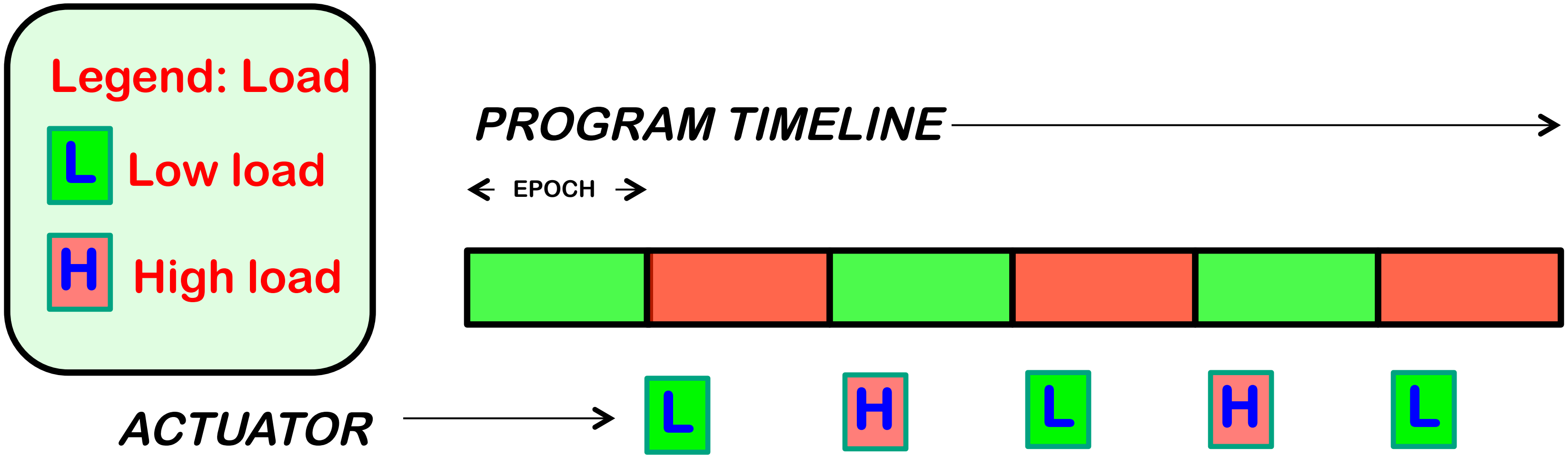
BOB's CONTRIBUTION OVER TIME
(area under the curve)

Integrity Vulnerability?

- On-chip firmware, VMM, VM, and OS all participate in energy management.
- Bugs in mgmt. software can be exploited to override default policies and mechanisms
  - Mgmt. SW/HW are hundreds of thousands of lines of code. Is likely to have many bugs.
    - Software bug rate = 5 bugs/1000 lines of code.
    - Hardware bug rate substantially lower, but still possible.
- What can you do if you had control of energy management software? (say exploiting firmware, OS or other bugs in energy mgmt software)
  - Can cause silent data corruptions to intentionally weaken victim's security say by causing interfererence during cryptographic dynamic or static "root of trust" operations?

Availability Vulnerability?

- Sense-Actuate delays in energy mgmt may open up vulnerabilities



Legend: Load

L Low load

H High load

PROGRAM TIMELINE →

← EPOCH →

ACTUATOR → L H L H L

Example: A cyber missile (or grindstone) for destroying data centers?

- An attacker can run extremely high load for small part of the epoch that can damage the chip (slowly over time) but provide overall ‘safe’ averaged behavior within an epoch. Consider voltage overshoot, temp effect on aging etc.

- An attacker can run turn off safeguards in firmware to break the chip?

Exploits

Real exploits offer existence proofs of the dangers of vulnerabilities. Currently no known realistic exploits on energy management systems. This is largely because:

- Vendors have not revealed their energy mgmt. architecture
  - Classic "Security through Obscurity" approach – traditionally has failed to provide security in the long run.
- Also, typical hacker today is not familiar with energy mgmt., or for that matter, even low-level microarchitectural details.

Demonstrate real-world exploits.

- Value/Impact: Real-world exploits >> Toy exploits in simulated settings >> Toy exploits in simulators.
- Real-world exploits clearly show what capabilities attackers need to exploit current systems, how much effort is required on part of the attacker, and if these attacks/exploits are more valuable than existing exploits.

Conclusions

- Energy & Security are important issues that if unmitigated can impede continued computing advances.

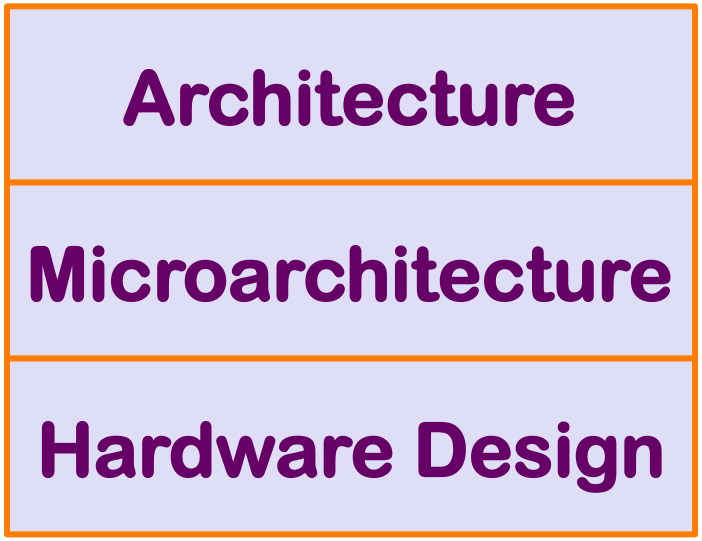
- Presented potential vulnerabilities on current energy management systems (limited by my deviousness). Looks like they can be used as potent cyber weapons.

- Much needs to be done.
  - Descriptions of energy management systems
  - Believable real-world exploit demonstrations
  - Formal analysis, modeling, vulnerability estimation
  - Risk mitigation etc.

- Special thanks to Pradip Bose at IBM!
- Plug: come to our ISCA talk on hardware malware detection!

Hardware Security at Columbia

- Security as a first-order design requirement
- Doctrine: Security must be designed from hardware up for practical, usable secure systems
- Contributions so far:



Architecture

Instruction Set Randomization
Information flow tracking in SoCs (CODES 13)

Microarchitecture

Support for Migration, repl. execution
Hardware Malware Detection (ISCA 13)
Side-channel Mitigation: Timewarp (ISCA12)
Side-channel Measurement: SVF (ISCA 12)

Hardware Design

Hardware Backdoors (Oakland 10,11, DT 13)

Funded by the DARPA CRASH, DARPA MRC, and NSF CAREER programs.