Thanks, But No Thanks
Current Cryptographic Standards Are Sufficient for Software

Dan Shumow

MSR Security and Cryptography Group

Microsoft Research

Introduction

- Disclaimer: I am a Software Developer, so this talk is from the software perspective.
- Goals
  - Identify and discuss what “lightweight cryptography” means in the context of the software development community.
  - Explain the conclusion that software does not require specialized lightweight cryptography standards.
- This is not an argument against a lightweight cryptography standard in general.
  - By removing software scenario from consideration, the design requirements for the standard can be most effectively focused on hardware use cases.

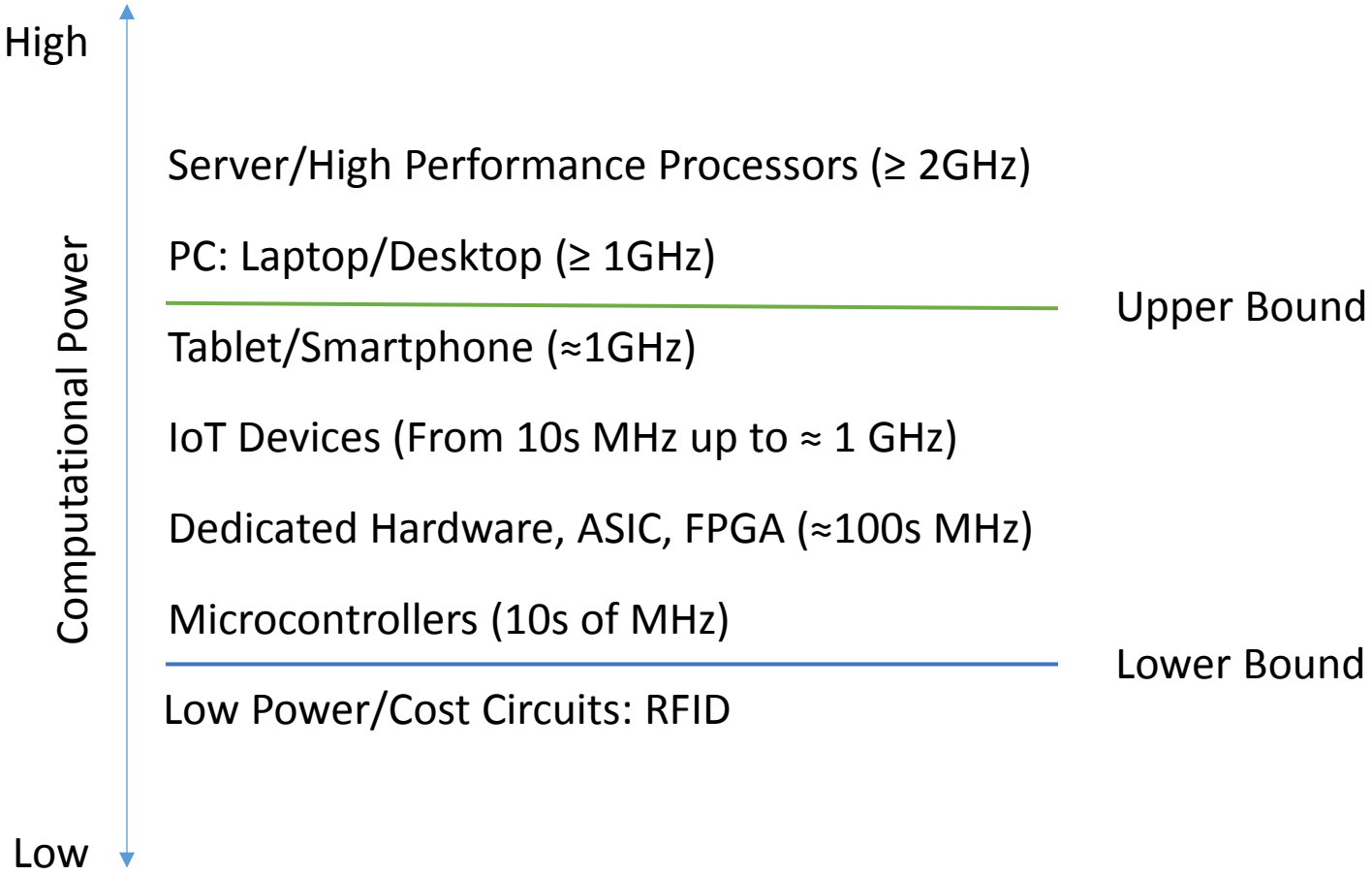
Outline

- Identify the Internet of Things (IoT) as the primary “lightweight cryptography” scenario in software.
- Brief overview of the current state of IoT security protocols.
- Discuss some of the problems with adding new cryptographic primitives.
- Conclusion for software: IoT does not require new cryptographic primitives.

Goal: Identify Lightweight Cryptography Scenarios

- From the description of this workshop:
  “NIST is currently investigating whether there is a need for NIST to standardize lightweight cryptography.”
- What does “lightweight” mean?
  - means different things in software and hardware.
- Potentially refers to:
  - Mobile platforms
  - Low powered RFID Cards
  - Microcontrollers
  - Dedicated hardware
  - Internet of Things (IoT)
  - …

Computational Platforms by Power



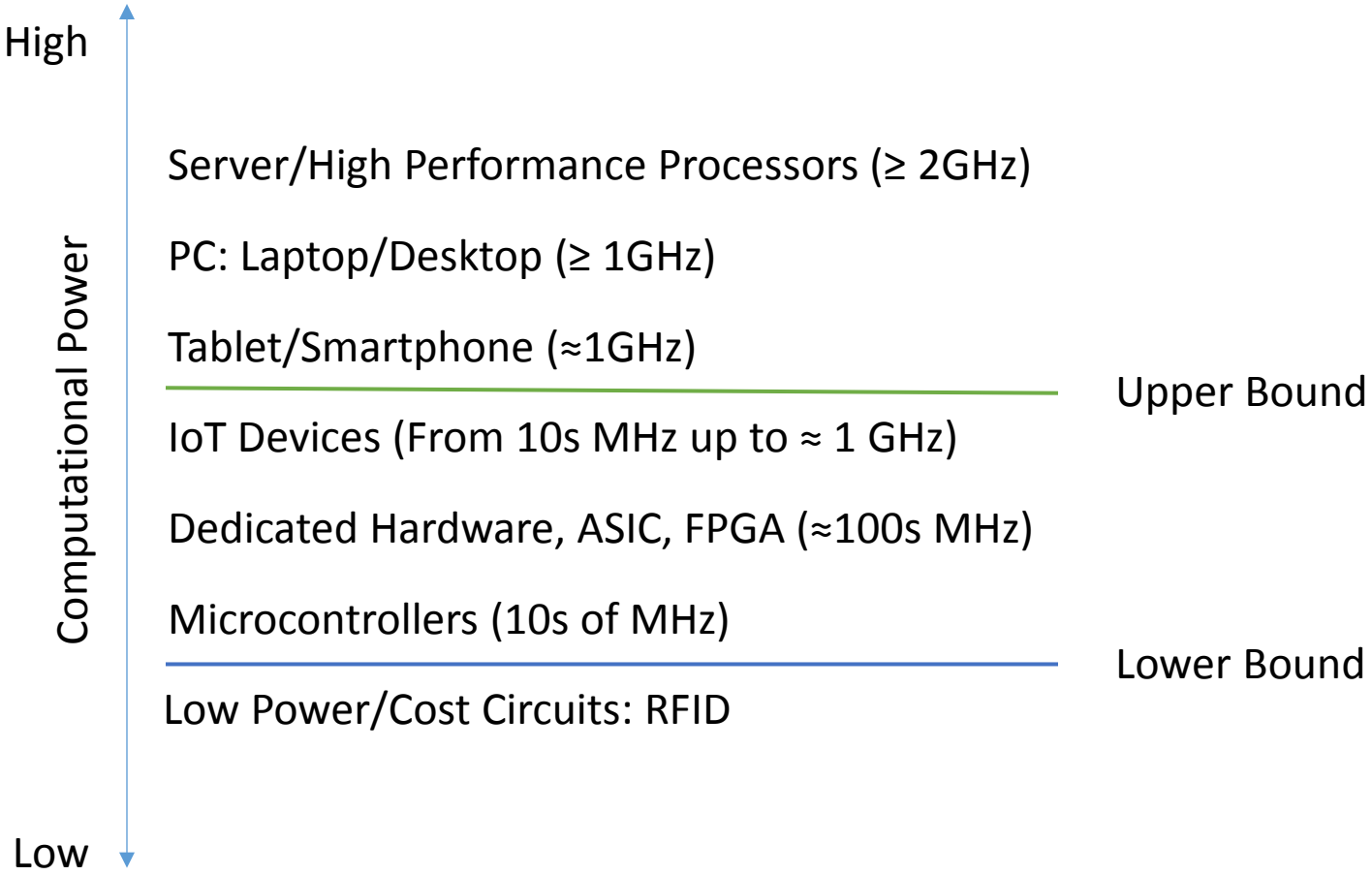
What counts as lightweight?

IoT Platforms

Development Board	Processor	Memory
Beaglebone Black	1 GHz ARM Cortex-A8	512 MB
Raspberry Pi 2	900 MHz quad-core ARM Cortex-A7	1 GB
Raspberry Pi	700 MHz ARM	256 MB
Intel Galileo	400 MHz Intel Quark	256 MB
Arduino Due	84 MHz Cortex-M3	96 KB
Arduino Uno	16 MHz AVR (8 bit)	2 KB

This is a list of sample platforms and not meant to be exhaustive.

Computational Platforms by Power



What counts as lightweight?

IoT Protocols

- AllJoyn
  - Open Source project run by AllSeen Alliance.
  - Industry stakeholders include Qualcomm, Microsoft and AT&T.
- Iotivity
  - Open source project that has recently announced an association with the AllSeen Alliance.
  - Backed by the Open Interconnect Consortium
  - Industry stakeholders include Intel, Samsung and Cisco.
- Thread
  - Open protocol run by the Thread Group.
  - Industry stakeholders include ARM, Samsung and Qualcomm.

IoT Security Protocol

- AllJoyn
  - Protocol derived from TLS hand shake and message protocol.
  - Only asymmetric algorithms are negotiable, not authenticated encrypt.
- Iotivity and Thread both rely on DTLS for security.

Cryptographic Algorithms Used:

Protocol	Asymmetric	Bulk Encryption	Authentication
AllJoyn	RSA, ECDSA/ECDHE P256	AES-CCM	AES-CCM
Iotivity	RSA, DSA/DHE, ECDSA/ECDHE (NIST Curves)	AES, AES-CCM, AES-GCM, 3DES	HMAC-SHA1, HMAC-SHA2
Thread	RSA, DSA/DHE, ECDSA/ECDHE (NIST Curves)	AES, AES-CCM, AES-GCM, 3DES	HMAC-SHA1, HMAC-SHA2

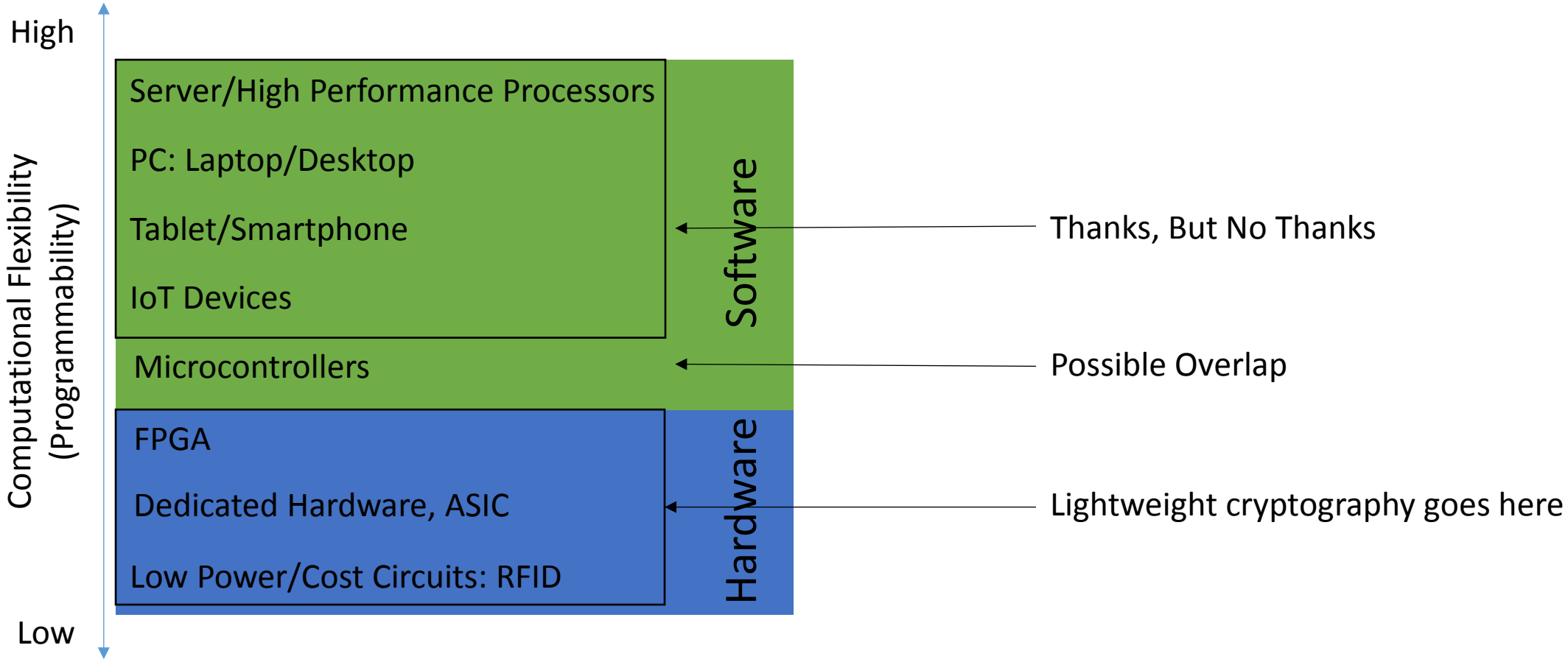
IoT Does Not Need Its Own Crypto Standards

- Any NIST standardization of cryptography for IoT is late to the party.
  - Current IoT protocols (AllJoyn, Thread, IoTivity) already use current cryptographic standards.
  - The protocols have already been deployed with existing algorithms. For backwards compatibility and interoperability these will need to remain in deployment.
- Current cryptographic standards work for IoT
  - Current standards are not a limit on IoT performance.
  - Perspective: Common IoT platforms are approximately as powerful as PCs from 15 years ago when AES was standardized.

Lightweight Crypto Standards: Why Not?

- We already have a set of cryptographic primitives.
  - Some may say ***too many*** cryptographic primitives.
  - Paradox of choice: more options is not necessarily better.
- Adding new standards can be problematic:
  - New standards, especially with lower key sizes could be used in scenarios where they aren't intended.
    Example: Standardizing ECC over 160bit prime for an RFID card and it ends up being used for https; block cipher with 80bit key space ends up being used to encrypt hard drives.
  - Giving developers more choices can lead to security vulnerabilities.
    Example: MAC-then-Encrypt vs Encrypt-then-MAC

Computational Platforms by Flexibility



What does this have to do with lightweight cryptography?

Conclusion For Lightweight Cryptography Standard

- The usage scenarios for lightweight cryptography are limited to non-existent for software platforms.
  - Developers already have too many choices for cryptography.
  - Current standards are good enough.
- This is not to say “No lightweight cryptography.”
  - “No lightweight cryptography in software.” (With a possible exception for Microcontrollers.)
- Any such standard should apply to hardware only.
  - Focus on hardware implementation for performance, side channel security etc.
  - Limit standards to apply only to low powered hardware platforms (RFID etc.)

Thank You