DICE: Foundational Trust for IoT

Dennis Mattoon, Microsoft

Introduction

- Modern cyber-attacks are often sophisticated and relentless in their continual efforts to seek out vulnerabilities in modern technology-based solutions

- At the same time, market segments like IoT are driving architectures and solutions with challenging power, security, resource, and other constraints. These constraints make an optimal security posture much more difficult to create and maintain

Introduction

- To effectively address these challenges a security architecture must be:
  - Free or very cheap; not just in BOM cost
  - Adaptable with minimal silicon requirements
  - Scalable to millions of endpoints per solution
  - Standards-based
- And most importantly, it takes a combination of hardware support and software techniques

Why Hardware Support?

- There are problems with software-only solutions
- Device Identity
  - If a bug leads to a DeviceID disclosure how do we securely (and remotely) recover a device?
- Device State and Attestation
  - Cannot trust software to report its own health
- Roots of Trust for Storage, data encryption, entropy, etc.
  - How do we securely extend trust chain, store keys, etc.?

Beware of Simplistic HW Solutions

- Why not just store DeviceID key in fuses?
  - If malware can read the fused key, you're no better off than with a software-based key
- TPMs are great but, especially in IoT solutions, systems and components probably won't have TPMs or even similar silicon-based capabilities
- We need something different

DICE and RIoT

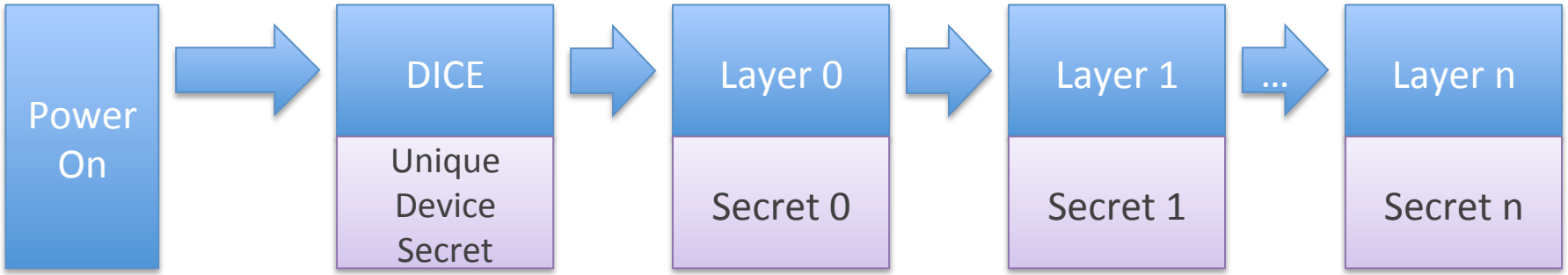
- Device Identifier Composition Engine (DICE, TCG)
- Robust, Resilient, Recoverable IoT (RIoT, MSFT)
- New Root of Trust for Measurement specification from the Trusted Computing Group (TCG)
- Foundational security for IoT at near zero cost
- Simple HW requirements mean DICE is adaptable to most any system or component
- Provides HW-based identity and attestation, as well as sealing, data integrity, device recovery and update

The DICE Model

- Device startup (boot) is layered in a DICE Architecture
- Beginning with a Unique Device Secret (UDS), secrets/keys are created that are unique not only to the device, but each layer/configuration
- This derivation method means that if different code or configuration is booted, secrets are different
- So if a vulnerability exists and a secret is disclosed, patching the code automatically re-keys the device

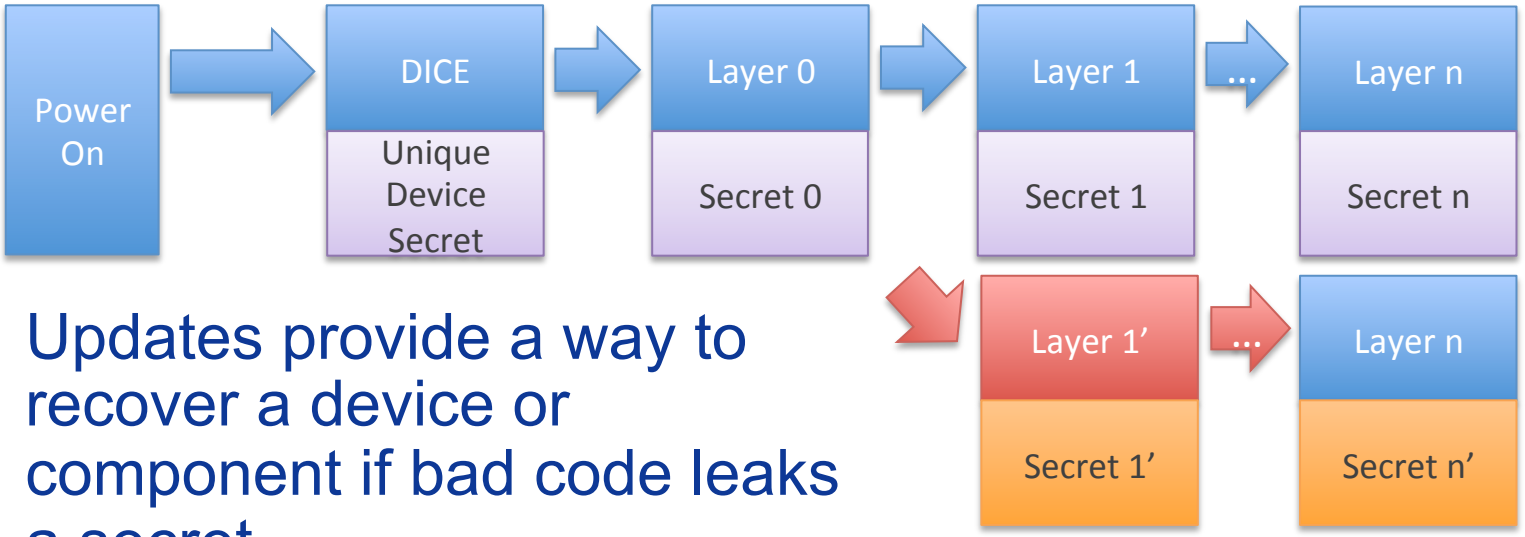
The DICE Model

- Power-on unconditionally start the DICE
- DICE has exclusive access to the UDS
- Each layer computes a secret for next layer (OWF)
- Each layer must protect the secret it receives



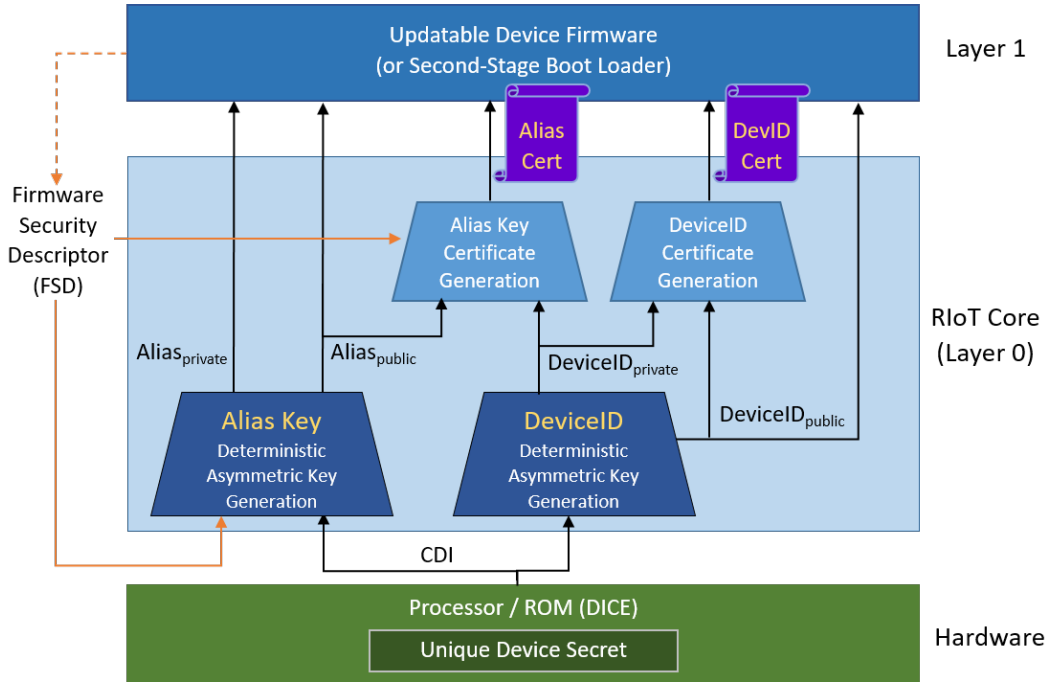
When Something Changes

- The branch illustrates the result of a code/config change



- Updates provide a way to recover a device or component if bad code leaks a secret.

A DICE Architecture (RIoT)



- Underlying architecture for HW-based device identity and attestation (Azure)
- DeviceID – Stable and well protected long term identifier for a device or component
- Alias Key – Derived from combination of unique device identity (HW) and identity of Device Firmware (SW)
- Integrates DICE-enabled HW with existing infrastructure

RIoT is Just One Example

- Build on DICE to enable other high-value scenarios
- Secure remote device recovery (Cyber Resiliency)
  - Recover unresponsive (e.g., p0wned, hung, etc.) devices
  - Greatly reduced cost: no need for physical device interaction
- Supply chain management
  - Several recent damaging cyber-attacks were the result of malware introduced in the supply chain
  - DICE attestation lets end-customers trust far less of the supply-chain, e.g., just the storage-subsystem or flash vendor
- Component identity, authenticity, licensing

DICE Takeaways

- Flexible security framework, not one size fits all
- Minimal Si requirements, low barrier to entry
- Foundation for strong cryptographic HW-based device identity and attestation, data at rest protection (sealing), and secure device update and recovery
- Public announcements from flash memory, SoC, and MCU vendors so far and many more on the way
- Represents the ongoing work of the DICE Architectures Workgroup (DiceArch) in the TCG. Come join us!

References

- Device Identifier Composition Engine (DICE) spec:
  - http://www.trustedcomputinggroup.org/wp-content/uploads/Device-Identifier-Composition-Engine-Rev69_Public-Review.pdf
- RIoT – A Foundation for Trust in the Internet of Things:
  - https://aka.ms/RIoT
- Azure IoT
  - https://azure.microsoft.com/en-us/blog/azure-iot-supports-new-security-hardware-to-strengthen-iot-security/
  - https://azure.microsoft.com/en-us/blog/announcing-new-functionality-to-automatically-provision-devices-to-azure-iot-hub/