Securing the Future of IoT

Joseph Yiu
Senior Embedded Technology Manager

China
July 2016
Connectivity in all aspects of modern life

Energy grid  Automotive  Environmental  Home automation  Healthcare  Enterprise  Retail

Smart city  Wearables  Farming  Identity & tracking  VR / AR  Building automation  Connected clothing

Robotics  Sensor  Industrial  IoT  Smart lighting  Smart watch  Space
The Need for Security

Communication Protection
- Cryptography, authentication

Data Protection
- Secret data (keys, personal information)

Firmware Protection
- IP theft, reverse engineering

Operation Protection
- Maintaining service and revenue

Anti-tamper Protection
- Protection against physical attack
TrustZone Concepts

Separation
- Isolate trusted resources from non-trusted
- Isolate non-trusted software
- Reduce attack surface of key components

Trusted Software
- Provision of security services
- Small, well reviewed code

Trusted Hardware
- Hardware assist for cryptography
- Secure access validation built into SoC
Security Domains and Protection

- 1\textsuperscript{st} level: Cryptography and authentication (communication protection)
  - Simple…
  - But a single software component being compromised can lead to the loss of the whole device
Security Domains and Protection

- 1\textsuperscript{st} level: Cryptography and authentication

- 2\textsuperscript{nd} level: Privileged levels and memory protection

- Available today…
- But no separate barrier for on-chip firmware
Security Domains and Protection

- **1st level**: Cryptography and authentication

- **2nd level**: Privilege levels and memory protection

- **3rd level**: ARM TrustZone

---

**Secure hardware**

- TRNG
- Unique ID
- Key storage

**Secure software**

- Crypto lib API
- Secure boot
- GUI lib API
- OS / OS API

---

**Secure**

- Privileged
  - Pre-loaded
  - On-chip firmware
  - Unprivileged

**Non-Secure**

- OS (privileged)
- MPU support
  - Apps software
  - Comms stack
  - Unprivileged

---

**Secure hardware**

- TRNG
- Unique ID
- Key storage
ARM Architecture: For Diverse Embedded Processing Needs

**Cortex - A**

Highest performance

Optimised for rich operating systems

**Cortex - R**

Fast response

Optimised for high performance, hard real-time applications

**Cortex - M**

Smallest/lowest power

Optimised for discrete processing and microcontrollers
Cortex-A: Wide Portfolio for Diverse Embedded Markets

Cortex-A17
Cortex-A15
Cortex-A9
Cortex-A7
Cortex-A5
Cortex-A73
Cortex-A72
Cortex-A57
Cortex-A53
Cortex-A35
Cortex-A32
High Performance
High Efficiency
Ultra High Efficiency

ARMv7-A
ARMv8-A
TrustZone on Cortex-A Processors

- **4 Execution Levels**
  - Level transitions using system calls instructions: SVC, HVC, and SMC

<table>
<thead>
<tr>
<th>Execution Levels</th>
<th>Applications</th>
<th>Applications</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL0</td>
<td>Unprivileged</td>
<td>SVC</td>
<td></td>
</tr>
<tr>
<td>EL1</td>
<td>Privileged</td>
<td>SMC</td>
<td>HVC</td>
</tr>
<tr>
<td>EL2</td>
<td>Hypervisor</td>
<td>SMC</td>
<td>Hypervisor (Virtualization)</td>
</tr>
<tr>
<td>EL3</td>
<td>Secure world</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Normal world** (access to Non-secure memory only)
- **Secure software** (can access to Secure and Non-secure memory)
- **Applications**
  - Secure boot, secure services
  - Secure and Non-secure memory
  - Linux kernel
  - Other guest OS

- **Execution Levels**
  - EL0: Unprivileged
  - EL1: Privileged
  - EL2: Hypervisor
  - EL3: Secure world
Cortex-A32 - ARM’s Most Efficient Cortex-A Processor

- Optimized for 32-bit ARMv8-A
- >25% Higher efficiency than Cortex-A7
- More performance and less power
- Scalability for diverse embedded applications

Performance and power compared to Cortex-A7 for same CPU configuration, same frequency and iso-process (28nm)
Cortex-A32: Extreme Scalability for Diverse Applications

Performance optimized configuration
- >1.0 GHz
- Cortex-A32 Quad Core
- ARMv8-A 32-bit CPU
- NEON SIMD engine
- Floating Point Unit
- 32K I-Cache - Parity
- 32K D-Cache - ECC
- ACP
- SCU
- 1MB L2 Cache with ECC
- 128-bit AMBA AXI4
- 1MB L2 Cache with ECC
- Less than 75mW/core at 1 GHz

Smallest area configuration
- <0.25 mm²
- Cortex-A32 Single Core
- ARMv8-A 32-bit CPU
- 8K I-Cache
- 8K D-Cache
- 128-bit AMBA AXI4
- Less than 4mW at 100 MHz

Total Dhrystone power reported at typical operating conditions, Process – 28HPC
New Architectural Features for Rich Embedded 32-bit Applications

- **ARMv8-A**
  - New 64-bit ISA
  - >100 New 32-bit Instructions
  - Enhanced floating point performance
  - Substantially faster software encryption
  - Enhanced media performance

- Cortex-A35
  - (64/32 bit ARMv8-A)

- Cortex-A32
  - (32-bit ARMv8-A)

- Cortex-A7
  - Cortex-A5
  - ARMv7-A
  - Cortex-A7 Extensions
  - ARMv7-A

Cortex-A35 (64/32 bit ARMv8-A)
Cortex-A32 (32-bit ARMv8-A)
Cortex-A7
Cortex-A5
ARMv7-A
Cortex-A7 Extensions
ARMv7-A
Cortex-A32:Boosts 32-bit Performance and Efficiency

- Higher performance than Cortex-A5 and Cortex-A7
- Same performance as Cortex-A35

Efficiency defined as integer performance/power (Performance/mW)
Iso-process (28HPC), Iso-frequency, comparisons, identical processor configurations (32KB L1 caches, NEON, 1MB L2)

>25% vs. Cortex-A7
>30% vs. Cortex-A5
>10% vs. Cortex-A35
ARMv8-M: Taking Embedded to the Next Level

- **Security**: Taking TrustZone security to the smallest devices
- **Productivity**: Making scalable software development even easier

Bringing security within reach of all developers.
ARMv8-M Sub-profiles

Scalable architecture

- **ARMv8-M Baseline**: Lowest cost, smallest, ARMv8-M implementations.
- **ARMv8-M Mainline**: For general purpose microcontroller products. Highly scalable. Optional DSP and floating-point extensions.

**Similar to ARMv6-M / ARMv7-M**
- 32-bit architecture, architectural memory map
- Nested Vectored Interrupt Controller (NVIC)
- Architecturally defined sleep modes
ARMv8-M Additional States

Existing handler and thread modes mirrored with secure and non-secure states

- Secure and Non-Secure code run on a single CPU
  - For efficient embedded implementation.

- Secure state for trusted code
  - New Secure stack pointers for robust operation
  - Addition of stack-limit checking.

- Dedicated resources for isolation between domains
  - Separate memory protection units for Secure and Non-secure
  - Private SysTick timer for each state.

- Secure side can configure target domain of interrupts.
High Performance Cross-Domain Calls

Efficient microcontroller focussed implementation

- Security inferred from instruction address
  - Secure memory considered to hold Secure code.

- Direct function calls across boundary
  - High performance and high security
  - Multiple entry points
  - No need to go via “monitor” for transitions.

- Uses Secure Gateway instruction “SG”
  - Only permitted in special Secure memory with Non-secure-callable attribute (NSC).
Secure transitions handled by the processor to maintain embedded class latency
Virtualization (ARMv8-R)

Virtual Machine #1
Virtual Machine #2

App/Libs
App/Libs

OS
OS

Hypervisor
Processor

Enables consolidation of multiple systems into one system.

TrustZone for ARMv8-M

NON-SECURE STATES

Non-Secure App / Libs
Non-Secure OS

OS API / Secure OS

SECURE STATES

Secure App/Libs

Enables security protection with very low overhead:
Execution cycles, memory footprint
Communication Protection

- TrustZone CryptoCell
  - Platform level security solutions

- mbedTLS library
  - Enables cryptographic and SSL/TLS capabilities in the connected embedded products
  - Available as standalone version, and also as part of mbedOS
SecurCore - Security Against Physical Attacks

- **SC300**
  - ARMv7-M architecture
  - Similar feature set as Cortex-M3 processor
  - Configurable Anti-tampering features

- **SC000**
  - ARMv6-M architecture
  - Similar feature set as Cortex-M0 processor
    - Support optional privileged/unprivileged levels
    - Support optional MPU
  - Configurable Anti-tampering features
ARM TrustZone Covers Multiple Security Requirements

Data Protection
- Secure data is protected from Non-Secure access

Firmware Protection
- Protects firmware while allowing interactions

Operation Protection
- Secure interrupts can have higher priority

Additional protection technology:
- Communication from ARM TrustZone CryptoCell
- Anti-tampering from ARM SecurCore processors
ARM Security Solutions for Embedded and IoT

Industrial
Retail
Smart Lighting
Automotive
Agriculture
Wearables
Building Automation
Medical
New Wave of Innovation

Driving a new wave of custom SoC development

Sensor and mixed signal companies: integrated IoT solutions

Start-ups: innovative solutions

OEMs: reduce cost, reduce power, differentiate
ARM is Accelerating Innovation

Free access to an ARM Cortex-M0 processor system for design and simulation

Low-cost, FastTrack $40k license to design, manufacture, and commercialize Cortex-M0 based products

A global network of ARM enabled design houses for best in class SoC development
Easy Access to ARM IP

- ARM Cortex-M0 processor is now available for download from ARM’s DesignStart portal (designstart.arm.com)

- Includes system IP and tools to simplify system design:
  - Cortex-M0 processor*
  - Cortex-M0 System Design Kit (key components of SDK including system IP, peripherals, test bench and software)
  - ARM Keil MDK development tool for software development – 3 month license
  - Prototyping option on ARM supplied FPGA board ($995)

* Supplied as obfuscated, fixed configuration, fully synthesizable RTL
ARM DesignStart: Fast Path to Silicon

- Free design & simulation of Cortex-M0
- $40k fast track, simplified commercials

- Free evaluation
- Simplified and quick access to EDA tooling

- Recommended by ARM
- Design partners you can trust
Thank You!

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2016 ARM Limited

© ARM 2016