ARM TrustZone®
How to use it to make devices secure and safe
Felix Baum
Agenda

Objectives

- Describe the need for hardware enforced security
- Outline the use cases for the ARM TrustZone deployments
- Highlight the pros and cons of various options

Results

- Provide a better understanding of how to secure embedded devices by utilizing the ARM Cortex-A hardware and software capabilities Mentor Embedded products
Recent Security Incidents in the News

- **Blast Furnace** – via plant’s business network
  - Caused “massive” damage to blast furnaces

- **Jeep Grand Cherokee** – via vehicle connectivity
  - Malicious code allowed access to vehicle’s CAN bus

- **Medical Devices** – via medical network
  - Devices accessed to steal patient medical data

- **Power plant in Ukraine** – via spearphishing e-mail attack
  - Power outage to 80,000 people

- More to come...
# Growing threat of attacks

<table>
<thead>
<tr>
<th>Embedded devices</th>
<th>IoT &amp; Industrial IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>15x</strong></td>
<td><strong>70%</strong></td>
</tr>
<tr>
<td>more vulnerable to attack than enterprise endpoints</td>
<td>of IoT devices are vulnerable to attack</td>
</tr>
<tr>
<td>- Columbia University Research Study</td>
<td>- HP Labs research study</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Defense &amp; Aerospace</th>
<th>Medical Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>148</strong></td>
<td><strong>300+</strong></td>
</tr>
<tr>
<td>companies were victims of cyber attacks from Chinese cyber warefare groups.</td>
<td>medical devices with hardcoded passwords</td>
</tr>
<tr>
<td>- Mandiant security report</td>
<td>- ICS-CERT vulnerability report</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Industrial &amp; Critical Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>52%</strong></td>
</tr>
<tr>
<td>Increase in cyber security attacks</td>
</tr>
<tr>
<td>- 2012 US Department of Homeland Security</td>
</tr>
</tbody>
</table>
Consolidation on the SoC level
**Security and Safety via Separation**

**Safety**: Protecting the world from the device

**Security**: Protecting the device from the world

**Mixed criticality**: Protecting of security or safety critical parts of the device from other parts of the device

ISO26262-6 requires “freedom from interference”. If two systems can interfere with each other, they must be certified to the highest ASIL level of the two. Secure separation aims to eliminate such interference.
Starting point: Chain of Trust

Before loading any software, ask:
• Did it come from the OEM?
• Has it been tampered with?

ARM TrustZone can be used for:
• Crypto Key Storage
• Signature Generation and Comparison
• Signature Storage
• Loading OS and Apps
Use Case 1:

Physical Separation aka AMP
Physical Separation aka AMP

- Multicore Device running one Operating System
  - Migrating to multicore device for the next generation or project
  - Need to consolidate applications that require real time and determinism with applications requiring Linux networking or graphics services
  - Addressing performance constrains of existing design
Physical Separation aka AMP

Multicore Device running multiple Operating Systems

- Single user interface for Configure, Edit, Debug, Optimize work
- Framework to configure, boot, execute and communicate across cores and Operating Systems
- Take full advantage of the underlying ‘silicony goodness’ 😊
Remote CPU Lifecycle Management

- Used by master OS to boot remote OSs on remote CPUs
- remoteproc user API for processor lifecycle management
- Conformance to upstream Linux remoteproc implementation
- Stand alone OS agnostic clean-room implementation of remoteproc API
Inter Processor Communications

- For inter-processor communications between OS/software contexts
- rpmsg user API for Inter Processor Communication
- Conformance to upstream Linux rpmsg implementation
- Stand alone OS agnostic clean-room implementation of virtio and rpmsg
- Usable from RTOS and BME contexts
OpenAMP Use Cases

- Separation of Resource constrained for example power management
- Offload work for Computationally intensive operations such as processing, encryption of secure, sensitive data
How this could be accomplished
Use Case 2:

Virtualization Enforced Separation
Multiple boards running various Operating Systems and dedicated applications

- Migrating to multicore device for the next generation or project
- Need to consolidate applications that require real time with Linux
- Must share displays and other resources
Separation via Virtualization

Consolidation to a single Heterogeneous Multicore SoC running multiple Operating Systems and Applications

- Virtualizing GPU to either control multiple displays per application or layer multiple applications on a single display (1:1, 1:N, N:1)
- Framework to configure, boot, execute and communicate across domains in safe and reliable matter
Separation via Virtualization

Infotainment Display

- BusMaster CAN Vehicle Simulator
- USB 2CAN
- CAN BUS
- AUTOSAR & CAN stack on M4
- Nucleus
- IVI Linux

Cluster Display

- FPD-Link Touch Display 10" (1280x800)
- FPD-Link Display 12" (1280x480)
- Hypervisor (2x CortexA) + GPU sharing

AxSB
Use Case 3:

Hardware Enforced Separation
Separation via ARM TrustZone

- ARM TrustZone® can be thought of as a hardware-based solution that can be used to define a subset of the SoC for access by software.

- Software that is designated as Secure World software has access to ALL of the SoC, while software that is designated as Normal World can access only those HW elements that are defined as “Non-Secure”. 
Separation via ARM TrustZone

- ARM TrustZone® can be thought of as a hardware-based solution that can be used to define a subset of the SoC for access by software.

- Software that is designated as Secure World software has access to ALL of the SoC, while software that is designated as Normal World can access only those HW elements that are defined as “Non-Secure”.

![Diagram of a house with areas marked as S for Secure and NS for Non-Secure.](mentor.com/embedded)
Separation via ARM TrustZone

- ARM TrustZone® can be thought of as a hardware-based solution that can be used to define a subset of the SoC for access by software.

- Software that is designated as Secure World software has access to ALL of the SoC, while software that is designated as Normal World can access only those HW elements that are defined as “Non-Secure”.

![Diagram of a house with sections labeled S and NS, indicating secure and non-secure zones.](image-url)
Separation via ARM TrustZone

- ARM TrustZone® can be thought of as a hardware-based solution that can be used to define a subset of the SoC for access by software.

- Software that is designated as Secure World software has access to ALL of the SoC, while software that is designated as Normal World can access only those HW elements that are defined as “Non-Secure”.

![Diagram of a house with NS markers indicating Non-Secure areas.](mentor.com/embedded)
GlobalPlatform and TEE

- GlobalPlatform identifies and develops technical specifications which facilitate the secure and interoperable deployment and management of multiple embedded applications on secure chip technology. Its proven technology is regarded as the international industry standard for building a trusted end-to-end solution which serves multiple actors and supports several business models.

- The Trusted Execution Environment (TEE) offers the best route to meeting security objectives. TEE is a separate execution environment that runs alongside the OS and provides security services to that environment. The TEE offers an execution space that provides a higher level of security than an OS; though not as secure as a Secure Element (SE), the security offered by the TEE is sufficient for most applications.
GlobalPlatform identifies and develops technical specifications which facilitate the secure and interoperable deployment and management of multiple embedded applications on secure chip technology. Its proven technology is regarded as the international industry standard for building a trusted end-to-end solution which serves multiple actors and supports several business models.

The Trusted Execution Environment (TEE) offers the best route to meeting security objectives. TEE is a separate execution environment that runs alongside the OS and provides security services to that environment. It offers an execution space that provides a higher level of security than an OS, though not as secure as a Secure Element. However, the security offered by the TEE is sufficient for most applications.
Separation via hardware enforcement

One or more cores running applications of various security or robustness levels

- Migrating to multicore or more powerful device for the next project
- Need to consolidate applications that require secure and non-secure apps
Separation via hardware enforcement

One or more cores running applications of various security or robustness levels

- Migrating to multicore or more powerful device for the next project
- Need to consolidate applications that require secure and non secure apps
Separation via hardware enforcement

Using the ARM TrustZone capabilities of the SoC separating secure or robust applications from the rest of the system

- Control only flows from Secure World to Normal World
- Data could flow either way
ARM TrustZone & MultiCore

- ARM TrustZone® can be thought of as a hardware-based solution that can be used to define a subset of the SoC for access by software.
- Software that is designated as Secure World software has access to ALL of the SoC, while software that is designated as Normal World can access only those HW elements that are defined as “Non-Secure”.

Secure World Apps run on each core
Secure World Apps run on dedicated core
Separation via hardware enforcement

Non-certified

Complex Instrument Logic

Complex Graphics Render

draws content to separate graphics plane managed by safe driver

Certified

Safe Instrument Logic

Safe Graphics Render

Safe Graphics Driver

planes blended in hardware, also managed by safe driver

Single Core

Cluster (unSafe) | Cluster (Safe)

Nucleus
Safe graphics
TrustZone
A9

Dual Core

Cluster (unSafe) | Cluster (Safe)

Nucleus
Safe graphics
TrustZone
A9 A9
Separation via hardware enforcement

1. TZ Enforced Secure Boot and Chain of Trust
2. Integration with IT tools including McAfee and Symantec
3. Device Integrity Monitoring, Audit and Reporting
4. Network Intrusion detection, prevention and reporting via firewall
5. Secure Update
ARM TrustZone limitations

TrustZone includes features that may be helpful to Multi-Core and Multi-OS support, but it alone fails to provide some fundamental capabilities typically required by an embedded system:

- No separation of Normal World resources from Secure World
- It only allows for 2 payloads on one processing core:
  - Normal world content
  - Secure world content
- No Separation of multiple, non Secure Domains

A full safe and secure solution needs a combination of hardware and software elements using virtualization!
Bring it all together

Secure App
Secure RTOS
Secure App
Safety Cert RTOS
Safe App

Graphics/Vision/Web App
Linux

Hypervisor

App
RTOS BME

Cortex M
Cortex A
Cortex A
Cortex A
Cortex A

TrustZone

The World of Embedded Devices

There is no silver bullet or one single button to push to adequately protect an embedded device!

Consider using ARM TrustZone and Mentor Graphics products to meet security and regulatory requirements!
Q & A