Building Trustworthy Embedded Systems
Agenda

• The Move to IoT
  – Threats & Defenses
  – Managing Risks

• Trusted Computing
  – Embedded Applications
  – Tips and Tricks

• Q&A
The Move to IoT
**IoT Trend Affects All Markets**

<table>
<thead>
<tr>
<th>Smart vehicles</th>
<th>Smart city &amp; energy</th>
<th>Smart industry &amp; business</th>
<th>Smart home &amp; consumer devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart cars</td>
<td>Energy &amp; infrastructure</td>
<td>Factory automation</td>
<td>Smart home</td>
</tr>
<tr>
<td>› ADAS / autonom. driving</td>
<td>› Generation (renewables)</td>
<td>› Industrial automation</td>
<td>› Home automation incl.</td>
</tr>
<tr>
<td>› Connected car</td>
<td>› Advanced transmission &amp; distribution / storage</td>
<td>› Motor &amp; motion controller</td>
<td>› home appliances</td>
</tr>
<tr>
<td>› Car security</td>
<td>› Utilities (water), traffic (elev. toll collection), outdoors, government</td>
<td>› Power quality</td>
<td>› Home energy management</td>
</tr>
<tr>
<td>› (H)EV</td>
<td>› Environmental sensors</td>
<td>› Power tools</td>
<td>› Home security &amp; safety</td>
</tr>
<tr>
<td>Commercial, agriculture &amp; construction vehicles incl. Trucks &amp; Buses</td>
<td>Building automation</td>
<td>› Industrial robotics</td>
<td>› Lighting</td>
</tr>
<tr>
<td>› ADAS / autonom. driving</td>
<td>› Automation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>› Connected car</td>
<td>› Access control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>› Car security</td>
<td>› Air conditioning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>› (H)EV</td>
<td>› Elevators/escalators</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other forms of transport</td>
<td>Professional lighting</td>
<td>Other businesses</td>
<td>Consumer Electronics &amp; wearables</td>
</tr>
<tr>
<td>› Commercial aircraft</td>
<td>› - Building lighting</td>
<td>› e. g. Banking &amp; securities, education, mining, retail and wholesale, transportation and logistics</td>
<td>› Media players, smart glasses, smart watches</td>
</tr>
<tr>
<td>› Connected trains</td>
<td>› - Street lighting</td>
<td></td>
<td>› Well-being (health &amp; fitness, assisted living)</td>
</tr>
<tr>
<td>› Ships (ferry &amp; container)</td>
<td>› - etc.</td>
<td></td>
<td>› Gaming</td>
</tr>
<tr>
<td>› Light electric vehicles</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ICT**

<table>
<thead>
<tr>
<th>Communication Networks</th>
<th>Data Center / Server Farms</th>
</tr>
</thead>
</table>

Source: Infineon Technologies | graphics are courtesy of Infineon
Why IoT?

1. New capabilities and services
2. Greater efficiency
3. Increased flexibility and customization

Smart Home
Automotive
Industrial
ICT

Source: Infineon Technologies | graphics are courtesy of Infineon
IoT Architecture

1. **Gather data**
2. **Analyze**
3. **Send commands**

- **Device**
- **Network**
- **Server**

- **Reliably convey data and commands**
- **Send and receive data and commands**

Source: Infineon Technologies | graphics are courtesy of Infineon
IoT Attacks Growing
Each Layer can be Attacked

An Eavesdropper listening in on data or commands can reveal confidential information about the operation of the infrastructure.

A Bad Server sending incorrect commands can be used to trigger unplanned events, to send some physical resource (water, oil, electricity, etc.) to an unplanned destination, and so forth.

A Bad Device injecting fake measurements can disrupt the control processes and cause them to react inappropriately or dangerously, or can be used to mask physical attacks.

Source: Infineon Technologies | graphics are courtesy of Infineon
IoT Defenses

- Audit
- Crypto Key Establishment and Management
- Crypto Offloads
- Lifecycle Management
- Platform Integrity Verification
- Stored Data Protection
- Authentication
- Boot Process Protection
- Secure SW/FW Update
- Secure Communications

Source: Infineon Technologies | graphics are courtesy of Infineon
Managing Risks
Balancing Risk and Security

![Graph showing the relationship between Cost of Security and Cost of Risk]
Risk Management

• Risk Analysis
  – Identify and assess risks
    • Analyze threats, vulnerabilities, risks (w/ impacts)

• Risk Handling
  – Avoid (e.g. skip activity)
  – Mitigate (e.g. reduce)
  – Transfer (to insurer, customer, ...)
  – Accept

• Presentation to management in business terms

• Implementation

• Review risk analysis periodically or as needed
Benefits of Risk Management

- Makes business case for IoT security
- Management understands rationale and impact
- Avoids taking on too much risk
- Avoids excessive or inefficient spending on IoT security
Risk Assessment in Smart Home

- Threat
- Vulnerability
- Risk

Cloud-Based Smart home server

WLAN Internet Router

Low energy gateway

Oven

Low energy gateway

Smart Meter

Smoke alarm

Game&DVD

Door
Risk Management in Smart Home

Cloud-Based Smart home server

WLAN Internet Router

Low energy gateway

Oven

Low energy gateway

Smoke alarm

WLAN Internet Router

Smart Meter

Game&DVD

Door

©2019 Trusted Computing Group
Trusted Computing
What is a Trusted System?

A trusted system is...
- predictable, even under stress
- trusted based on experience and/or evidence
- based on fundamental properties (identity, integrity)
How to Build a Trusted System

• Start with a Root of Trust (RoT)
  – Minimized, strongly protected security function

• RoT used for highly security-sensitive functions
  – Generate random numbers
  – Store and use long-term keys
  – Verify system integrity

• Benefits
  – Reduce risks
    • Compromise of long-term keys
    • Undetected system compromise
Trusted Platform Module (TPM)

- **Standard Hardware Root of Trust**
  - TPM = ISO/IEC 11889

- **Benefits**
  - Foundation for secure software
  - Resistant to attacks/hacks
  - Built-in virtual smart card

- **Features**
  - Authentication
  - Encryption
  - Attestation
  - Identity
  - Integrity
Levels of Assurance in TPM

- Tamper Resistant
- OS Bug Resistant
- App Bug Resistant
- Passive Network Attack Resistant

Assurance

- Software TPM
- Firmware TPM
- Integrated TPM
- Discrete TPM
Why Hardware?
Why Hardware?

Software Security is Not Enough

Graph used with permission of Capers Jones.
Hardware Storage Encryption

- **Hardware Security**
  - Self-Encrypting Drive (SED)

- **Benefits**
  - Always on encryption
  - No performance impact
  - Protection against Physical Attacks, loss and theft
  - Cryptographic instant erase / Wipe

- **Features**
  - Encryption
Embedded Applications
Typical Applications

• Cars
• Financial Services
  – ATMs
  – Cash registers
  – Slot machines
• Industrial Control Systems
• Network Equipment
• IoT Gateways
• Hard Copy Devices
Trusted Computing for IoT

• Use cases
• Tips and Tricks
• Implementation guidance
• Coming Soon
• IoT Security Resources
TCG and IoT Defenses

- Device Identity
- Secure Communication
- Random Number Generation
- Secure Software Update
- Attestation & Measured Boot
- Stored Data Protection
- Tamper Resistance
- Resale and Decommissioning
- Zero Touch Provisioning
- Audit Logging
- Remote Management
- Licensing
- Pay-Per-Use
- Etc.
Device Identity

• Many embedded devices are ‘remote’
  – Difficult to protect or identify reliably

• Provision them with a TPM-based cryptographic identity
  – Typically, a private key and certificates stored in TPM

• Use that identity for many purposes
  – Secure communications
  – Access control
  – Remote management
  – Anti-counterfeiting
  – Licensing
Secure Communications

• Remote communications must be protected
  – Confidentiality
  – Authentication & Authorization
  – Integrity Protection
  – Replay Protection
  – Perfect Forward Secrecy

• Use TPM-based identity with TLS, DTLS, Cloud, P2P, etc.
  – Device can prove its identity
    • Using private key and certificate stored inside the TPM
  – Device can authenticate other parties
    • Using trusted CA certificate stored in TPM
Random Number Generation

• Protocols like TLS need high-quality random numbers
  – For cryptographic keys, nonces, initialization vectors, etc.
  – Typically generated within the embedded device itself
  – These random numbers are like passwords:
    • *If you can guess the numbers, you can break the protocol.*

• Random numbers are hard to get in embedded systems
  – Software algorithms can only generate pseudo-random sequences
  – Hardware is needed for truly random numbers

• Most TPMs contain a physical source of randomness (aka entropy) that can be used to generate good keys
Secure Software Update

• Software needs regular updates
  – To add features, fix bugs, and repair security vulnerabilities
  – An unpatched system is highly vulnerable
    • Once a patch is released, attackers quickly develop and deploy exploits

• But software updates can cause problems
  – Reliability issues, failed updates, or incompatibilities
  – Compromise if the update mechanism is weak

• So every system needs a secure update mechanism
Phases of Secure Update

- Secure Development
- Secure Update Signing
- Robust Distribution
- Secure Update Installation
- Post-Update Verification and Attestation

Trusted Computing can help with all of these!

To learn how, read our forthcoming document “TCG Guidance for Secure Update of Software and Firmware on Embedded Systems”
Attestation & Measured Boot

• **Purpose:** verify software on device
  – Cannot trust software alone to report its status

• **Measured Boot**
  – Securely record integrity measurement of each loaded code module

• **Attestation**
  – Verify loaded firmware and software
  – Can verify locally or remotely
Attestation & Measured Boot

1: Challenge (nonce)
2: Signed measurements
3: Validate signature and check measurements
Measured Boot vs. Secure Boot

• Secure Boot works well for *deterministic* early stages of boot
  – Multi-core processors tend to be less predictable once the OS layer starts.
  – Doesn’t provide evidence: is it XYZv3 or XYZv4 that was loaded?

• We want to prove exactly which firmware & software was loaded
  – Before we trust it to do something else we need to trust

• Who checks code and when?
  – With Measured Boot, verifier checks code hash during attestation
  – With Secure Boot, loader checks code signature during boot
Stored Data Protection

• The TPM can be used to protect secrets like:
  – Passwords
  – Authentication keys
  – Disk Encryption keys or other decryption keys
  – Non-cryptographic data (formulas, configuration, code, etc.)

• Configure the TPM so it will only decrypt the secrets for use when the platform is in a specified state:
  – Known, unmodified firmware or OS
  – Specific platform configuration
  – User password entered
  – Etc.
Stored Data Protection

• Use an Opal drive
  – Create multiple sections with different access controls
    • Read-only
    • Read-only unless a certain key is presented
    • Inaccessible unless a certain key is presented
    • Etc.

• Consider protecting Opal keys with a TPM
  – Restrict access to those keys using TPM policies
    • Require an authorization value
    • Require certain PCR values
    • Or many other combinations
Tips & Tricks

• TPM Software Support Options
  – TCG Software Stack (TSS)
  – Mocana TrustPoint
  – WolfSSL

• Dev Boards and Other Tools
Why Host Software

- Host system communicates with the TPM via software
  - At various abstraction levels
TCG TSS working group defines the following working packages:

- **System API (SAPI)** Lowest level interface to build command byte streams and decompose response byte streams
TSS Definitions

TCG TSS working group defines the following working packages:

- **Enhanced System API (ESAPI)** sits directly above the System API to reduce the complexity of individual “system level” calls to the TPM, but allows crypto operations on the data.
TCG TSS working group defines the following working packages:

- **Feature/environment API (FAPI)**
  Provides a higher level software abstraction to application developers
Implementation Help

• TCG IoT Architect’s Guide

• TCG Guidance for Securing the IoT