Security issues within embedded software development

Ron Buskey
Fellow of the Technical Staff
Security Architecture Technology Lab

Outline

- Introduction to the Environment
- Skills and Problem Solving
- Boundary Faults
- Why is ‘Why?’ bad
- Are We There Yet?
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- Summary

Outline

- **Introduction to the Environment**
- **Skills and Problem Solving**
- **Boundary Faults**
- **Why is 'Why?' bad**
- **Are We There Yet?**
- **Current Project Example**
  - **JTAG / ICE Interface**
  - **Authenticated Debug project**
- **Summary**

Environment

- Four Processors
- Five Network Interfaces
- Two Operating Systems
- Fixed and removable Memory
- Ten Million Lines of software
- Two Security Models

Outline

- Introduction to the Environment
- **Skills and Problem Solving**
- Boundary Faults
- Why is ‘Why?’ bad
- Are We There Yet?
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- Summary

Skills and Process

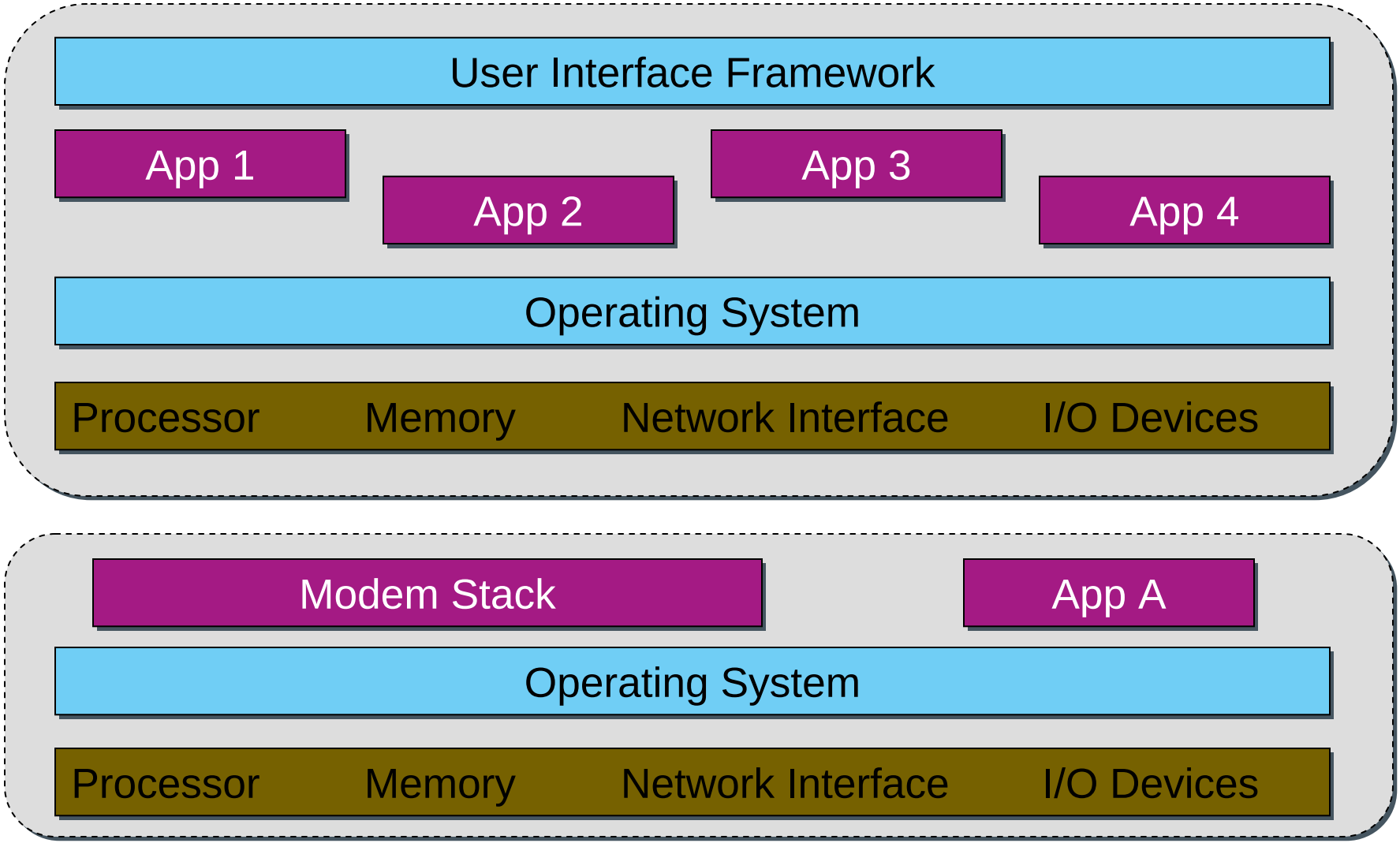
- Define Use Cases
- Requirements Capture
- Test to Meet Requirements
- Develop Interfaces
- Build Architecture
- Write Software
- ‘What can I make it do?’

- Define Assets
- Define Abuse Cases
- Probe Interfaces
- Look for Cracks
- Consider Risk Analysis
- Develop Threat Models
- ‘What can I make go wrong?’

Outline

- Introduction to the Environment
- Skills and Problem Solving
- **Boundary Faults**
- Why is ‘Why?’ bad
- Are We There Yet?
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- Summary

Product Structure



Outline

- Introduction to the Environment
- Skills and Problem Solving
- Boundary Faults
- **Why is ‘Why?’ bad**
- Are We There Yet?
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- Summary

Security Assessments

- Define Assets
- Establish Threat model
- Evaluate the Boundaries
- Define cost for lost/compromised Assets
- Develop cost model for protection of Assets
- Develop cost model for attacks
- Perform Risk Analysis
- Establish Roadmap for Evolution/Revolution
- **Understand Why someone would attack?**

Outline

- Introduction to the Environment
- Skills and Problem Solving
- Boundary Faults
- Why is 'Why?' bad
- **Are We There Yet?**
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- Summary

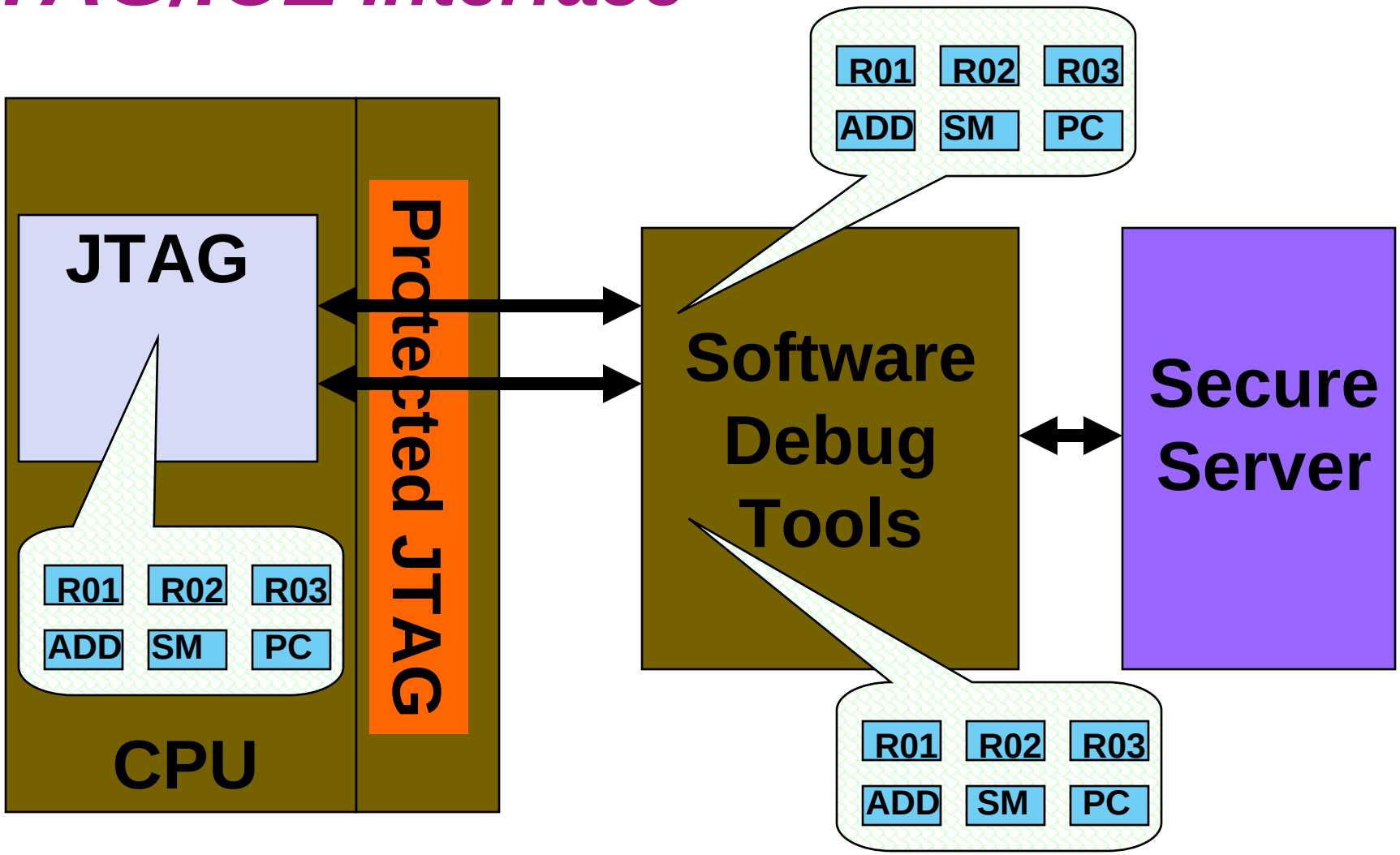
Security under Attack

- **Software and Hardware Features**
  - **Smaller / Faster / Cheaper**
  - **Extra Functions**
  - **Pressure from Competitors**
  - **Consumer Expectations**
- **Security Capabilities**
  - **Under Continuous Attack**
  - **Changing Environment**

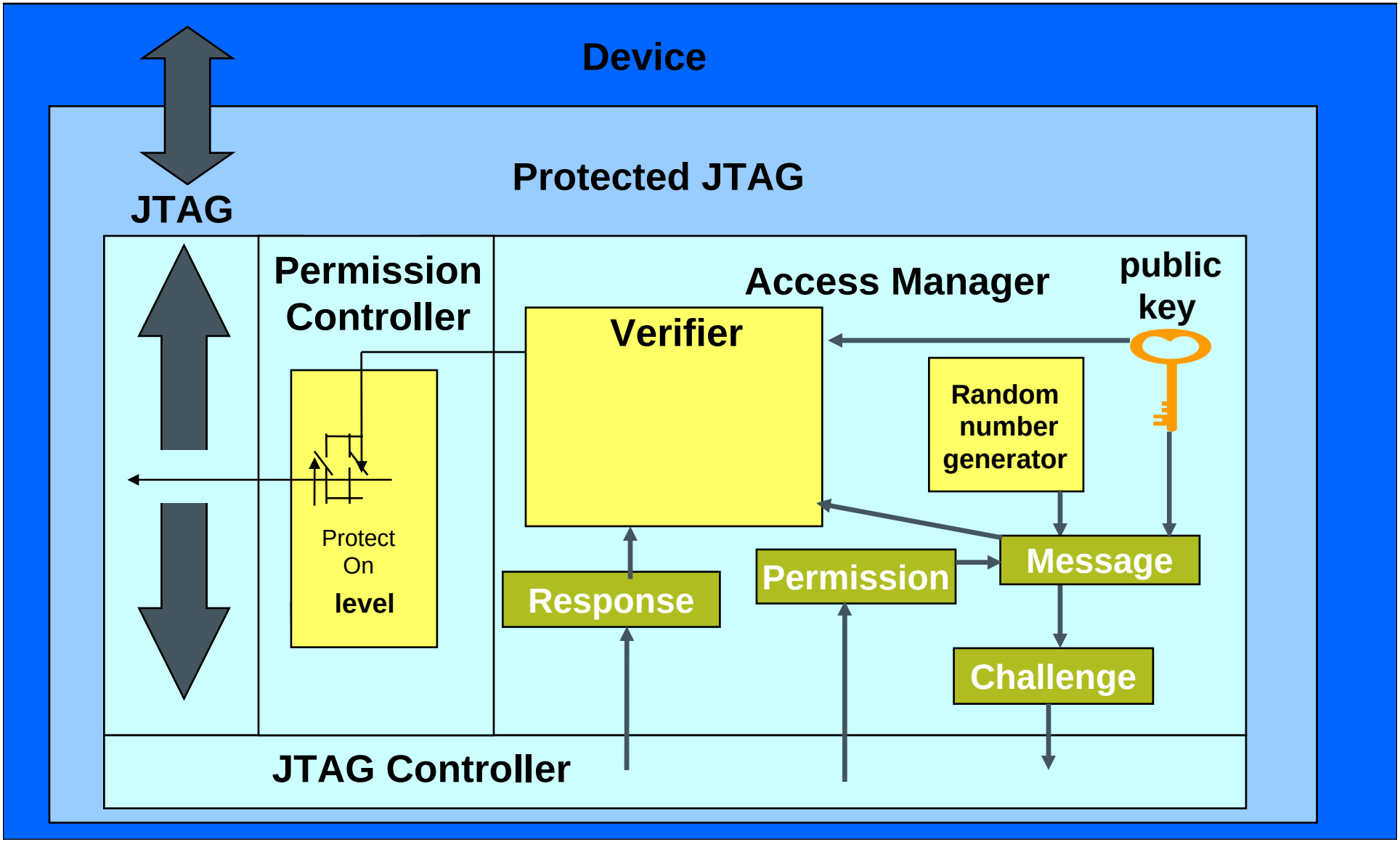
Outline

- Introduction to the Environment
- Skills and Problem Solving
- Boundary Faults
- Why is ‘Why?’ bad
- Are We There Yet?
- **Current Project Example**
  - **JTAG / ICE Interface**
  - **Authenticated Debug project**
- Summary

JTAG/ICE Interface



Protected JTAG

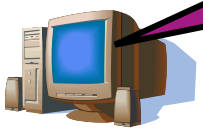


ECC Delegated Authentication

- **Problem**
  - **Protocol that fulfills Protected JTAG objectives**
    - Different permission levels of access to a device
    - Robust against “Man in the Middle” attack (and others)
  - **Hardware only solution required**
    - Minimize gate count implementation on embedded device
    - Processor may be component under study (i.e. no software)
- **Solution**
  - **Restrict operations to a single function (point multiply)**
  - **Removes finite field inversion operation from limited device (only finite field add and multiply required)**
  - **Offers multiple permission levels of operation**
    - Can be expanded to multiple interfaces

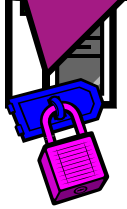
ECQ Authentication Protocol

P - point on curve
Q - public key
Q' – partial public



U – User Credentials

a - private key
Q = aP



Generate Random k

Generate Random r

Open (level - l)

Public Key (Q)

m = 0x01 || lP->x || r ||

c = kP, m(kQ->x),kQ->z

c

Full Public Key
if using Partial
Storage

12 bits

l ~ 20 bits

l must = l'
U must be valid for level l
l' and lP->x must match
m' must start with 0x01

l, U

/ kQ->z / akP->x

l' and r' from m'

U, l and l'

V= m'akP

V

Verify mkQ = V

If valid, Open level l

Outline

- Introduction to the Environment
- Skills and Problem Solving
- Boundary Faults
- Why is ‘Why?’ bad
- Are We There Yet?
- Current Project Example
  - JTAG / ICE Interface
  - Authenticated Debug project
- **Summary**

Summary

- Embedded Systems are complex with a lot of opportunities for applied security work
- Understand the tension between product development and security engineering
- All system boundaries need to be evaluated for possible cracks
- Concentrate on the How, Where and If
  - NOT the Why
- Embedded System Security is all about the journey, not the destination

Contact Info

Ron Buskey
Fellow of the Technical Staff
Security Architecture Technology Lab
Motorola Labs

Email: Ron.Buskey@Motorola.com

Current work: http://www.motorola.com/techpubs