Trust Architecture: Freescale's Security Solution for Industrial Control Systems (ICS)

Matt Short, Geoff Waters

Why Security for ICS?

Complexity in factory automation and process control applications has been increasing. Historically ICS were controlled by mechanical devices, and later the control gradually shifted to electrical devices communicating with more than 50 fieldbus standards. In recent times the control has further shifted to networking equipment communicating with more than five standards. Hence, ICS are frequently equipped with Ethernet-based fieldbus or wireless fieldbus for communication.

To manage the functionality of these complex networks, ICS require increased CPU power. This has led malicious third parties to gain access to ICS networks and make them vulnerable to potentially devastating malware attacks and cyber security threats which can impact the entire system infrastructure. The more networked the ICS is, the more vulnerable it becomes to external spyware attacks. The 2010 "Stuxnet" malware attack that particularly targeted Siemens Supervisory Control and Data Acquisition System (SCADA) control systems and caused nationwide infrastructure damage in Iran is just one example.[1] So critical is ICS security that even the Department of Homeland Security runs a program integrating efforts to protect the United States' critical infrastructure and key resources.[2] Due to increased vulnerabilities of ICS to the dangerous consequences from viral infections such as Stuxnet, numerous cyber security standards and certifications have been developed to evaluate and certify the security of industrial automation products.

Freescale's Solution for ICS Security: Trust Architecture

Freescale understands its responsibility for delivering the processors that secure the traffic passing through ICS against any external attack. Freescale's trust architecture helps OEMs and users to build a trusted system. A trusted system specifically checks for any suspicious activities or external attacks on the system and counters them through built-in functionalities (discussed later in this article).

The starting point for a trusted system is assurance that it boots and executes only authentic code. Consequently, secure boot is a cornerstone of the QorIQ platform's trust architecture, which also includes secure runtime, secure debug, tamper detection and device-specific secret key usage features. QorIQ P1010 processors, along with other QorIQ processors such as P2040, P2041, P3041, P4080, P4040, P5020 and P5010 support trust architecture, providing system developers with the hardware anchor points needed to develop a trusted system.

Objectives of Secure Boot

Secure boot is a process through which the QorIQ processor determines whether the system's image is trusted. System developers digitally sign their code to allow the P1010 processor to distinguish authentic trusted code from untrusted non-authentic code. The ability to distinguish between trusted and untrusted code enables the following capabilities:

- Prevent CPU from running untrusted code rather than authentic OEM signed code
- Detect and reject modified security configuration values and device secrets
- Allow trusted code to use a device-specific, one-time programmable master key (OTPMK) when the trust architecture says the P1010 processor is in a secure state
- Prevent extraction of sensitive values from the device by any means, short of de-processing

Key Advantages of Freescale's Trust Architecture Solution for ICS Security

The security mechanisms within the trust architecture have many systems that help prevent threats to ICS security. Below are some of the key functionalities of trust architecture:

- Gives OEMs the tools they need to create trusted systems without Freescale as part of the chain of trust
- Provides a unique untamperable silicon identifier, and creates an untamperable binding between hardware and public key
- Validates system image prior to allowing it to execute and allows validated images to use a device-specific secret key
- Detects and responds to hardware and software security violations that could lead to exposure of secret key or use of secret key by untrusted software
- Supports strong partitioning of system resources
- Secures debug interfaces
- Protects arbitrary numbers of session keys without impacting security acceleration performance

[1] Falliere, Nicolas. "Stuxnet Introduces the First Known Rootkit for ICS | Symantec Connect Community." Symantec - AntiVirus, Anti-Spyware, Endpoint Security, Backup, Storage Solutions. Symantec, 19 Aug. 2010. Web. 13 Oct. 2011. **symantec.com/connect/blogs/stuxnet-introduces-first-known-rootkit-scada-devices**

[2] Chertoff, Michael. "DHS | National Infrastructure Protection Plan." Department of Homeland Security | Preserving Our Freedoms, Protecting America. DHS, 2009. Web. 02 Nov. 2011. **dhs.gov/files/programs/editorial_0827.shtm**

Trust architecture features are not enabled by default, so only those ICS manufacturers who want to leverage its benefits can enable it. Customers can even enable only the relevant features to strike a balance between security and manufacturing constraints.

Components of Trust Architecture

Figure 1 demonstrates Freescale's trust architecture implementation on the single-core QorIQ P1010 processor. Blocks shown in red have a role in trust architecture. The following sections describe how each block functions to enhance ICS security.

1. e500v2 core

The Power Architecture® e500mc CPUs play important roles in both secure boot and secure runtime operations. If the QorIQ processor is configured to perform secure boot, CPU0 bit is released to begin execution from the internal boot ROM at power-on reset. The instructions executed from internal boot ROM allow CPU0 to determine whether code outside the former is safe to execute.

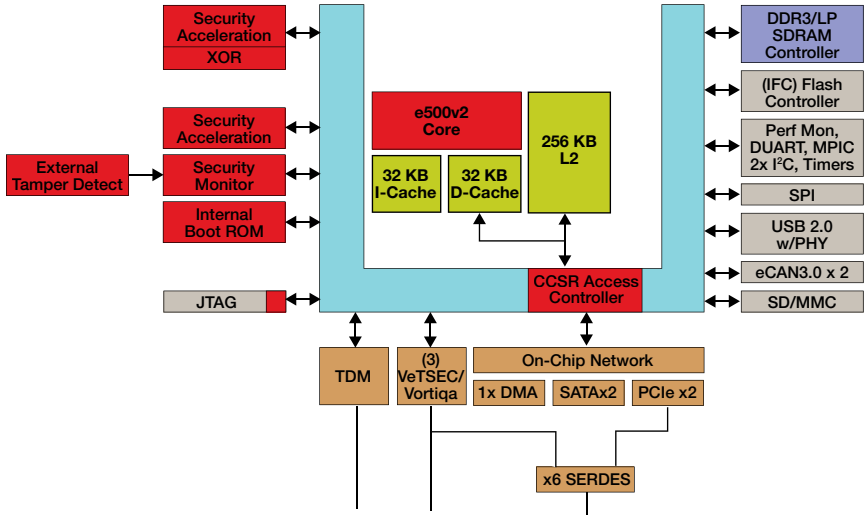
1.1 No execute bits (UX, SX Bits)

The UX and SX bits in the translation lookaside buffer (TLB) control whether the contents of the page can be executed as instructions by user- or supervisor-level processes. The ability to define pages as non-executable provides a barrier against attacks that overflow data buffers into code memory space.

2. CCSR access control unit (ACU)

CCSR Access Control Unit provides a mechanism through which reads or writes from unauthorized masters to the CCSR space can be blocked. This block uses transaction source ID to restrict the access to different regions in CCSR space. Boot firmware should be used to program the ACU for appropriate access controls by non-CPU bus masters. In case an unauthorized master tries to access restricted CCSR space, this block sends a violation to the security monitor. Some QorIQ devices perform this function in an I/O MMU also known as PAMU, discussed later.

Figure 1: P1010 with the QorIQ Platform's Trust Architecture



3. Internal boot ROM

The unmodifiable internal boot ROM contains code known as the internal secure boot code (ISBC). The function of ISBC is to validate a signature over next code to execute, referred to as external secure boot code (ESBC).

4. Security fuse processor

The security fuse processor (SFP) supports the following three trust architecture functions:

- Physically burns fuses during device provisioning
- Enforces basic security policy in the pre-boot phase
- Securely passes provisioned keys and other secret values to other hardware blocks if the QorIQ processor reaches a trusted/secure state

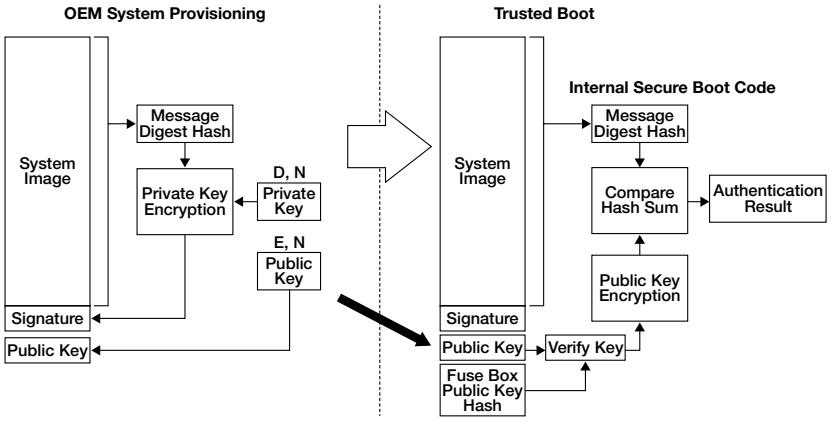
5. Secure monitor

Security monitor (Sec_Mon) senses and controls the security state of the QorIQ processor. Once in the trusted/secure state, the Sec_Mon provides ongoing monitoring of the QorIQ processor's security, with security violations causing configurable actions ranging from master key lock-out or zeroization to full SoC reset.

6. Secure debug controller

By means of control fuses, OEMs can control the level of debug access available to external debuggers. The secure debug controller supports four different levels of access.

Figure 2: Code Signing and Signature Validation



Note: Program and Signature may also be encrypted for IP protection.
Private Key has to be carefully managed and protected.

7. SEC 4.4 with run-time integrity checker

Freescale assumes that most QorIQ-based systems will use more session keys than can be reasonably stored in on-chip memory. Consequently, SEC 4.4 supports the option of encrypting session keys as they are created, and storing them as ciphertext in external memory. SEC offers the following two additional trust-related features: Key encryption keys (KEKs) and the run-time integrity checker (RTIC). When in a trusted state, the SEC can randomly initialize KEKs, which are stored within the SEC. The KEKs can be used to encrypt and decrypt hundreds to thousands of session keys negotiated for protocols such as IP security (IPsec) and secured sockets layer (SSL). Figure 2 demonstrates the code signing and signature validation process.

8. External tamper detection

The trust architecture on QorIQ processors provides an input for OEM-defined tamper detection circuitry. Magnetic switches, light sensors, out-of-specification voltage sensors and out-of-specification temperature sensors are examples of these circuits. Some high-end QorIQ devices support a few additional trust architecture components, including those on the following pages.

Zeroizable Secret Key (ZSK)

Use of a ZSK significantly raises the consequences of a security violation. Rather than being locked out until the next successful secure boot cycle, the ZSK is zeroized. Anything encrypted with the ZSK is unrecoverable, and if this happens to include the majority of the system's software, subsequent secure boot attempts will fail, effectively "bricking" the system.

Platform Memory Management Units (MMUs)

To prevent system masters other than the e500mc cores from reading or writing sensitive memory regions, QorIQ processors implement a number of I/O MMUs (also known as platform MMUs or PAMUs). These PAMUs prevent internal and external DMAs (non-CPU masters) from accessing memory for which they have not been granted explicit access permission.

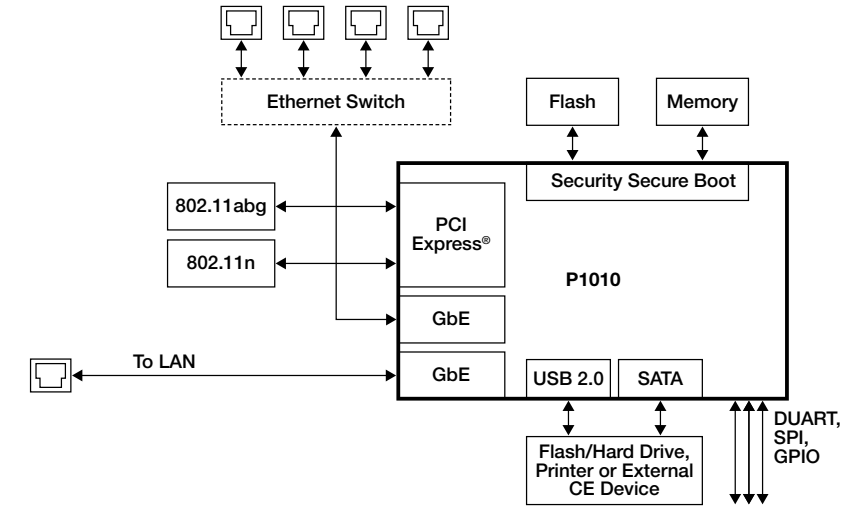
Hypervisor

The e500mc embedded hypervisor architecture introduces a third privilege level called guest state (GS bit in the e500mc machine state register) allowing hypervisor software to run at the most privileged level, which can further help virtualize the e500mc core (e500mc-vcpu) and block any guest OS attempts to modify critical security configurations, such as modifications of page table entries.

Trust Architecture in Operation

This section describes some of the practical scenarios where trust architecture combats against security attacks.

Figure 3: Trust Architecture in ICS Application



Defense against theft of functionality for ICS

- **Defense against system modification**
  ICS OEMs and service providers can prevent their systems and end-customers from the following system modification threats using trust architecture:
  - **Threat:** Tricking ICS into booting an alternate image. For example, replacing flash and changing the boot location
    Protection: As long as attackers do not have the OEM image signing key, the QorIQ processor's secure boot will detect a fraudulent image and refuse to execute
  - **Threat:** Modifying ICS code after boot (for example, use of buffer overflows, debug interfaces and "mod chips")
    Protection: The following act against these attacks: Power Architecture CPU enforcement of non-executable memory regions, run-time integrity checking and secure debug
  - **Threat:** Exploiting a remote management interface or firmware update facility in ICS
    Protection:
    - Perform a two-way authentication of the remote management server, using OTPMK protected credentials
    - Use IPsec, SSL or SSH for privacy and integrity of firmware updates over the network
    - Verify digital signature of the new firmware before allowing it to execute

- **Denial of service (DoS) as theft of functionality**
  Trust architecture is designed to fail safe, which means that attackers can deliberately activate the trust architecture defenses as a way of denying service and functionality to legitimate users. While trust architecture may block all DoS attacks, ICS developers need to exercise control in deciding how and when their systems should respond to the detected attacks, because ICS unavailability may hinder or even stop critical infrastructure operations.

  Classic networking DoS and distributed denial of service (DDoS) attacks attempt to crash ICS by flooding them with connection setup requests, malformed packets and other types of traffic that are intended to overwhelm the processing capability of the system. While P1010 processors and other low-end QorIQ devices have enough CPU horsepower to run filtering protocols to defend against DoS, some high-end QorIQ devices support Data Path Acceleration Architecture (DPAA). DPAA supports fast flow classification and policing, which in turn can be used to rate-limit floods of connection setup request packets and similar protocol exploits.

Defense Against Theft of Uniqueness for ICS

- **Counterfeit equipment**
  Counterfeit equipment has the potential to be an exact copy of an ICS. This counterfeit equipment may be a repaired system sold as a new system, or an extra system built by an unscrupulous contract manufacturer using grey market components. Cloners are assumed to have limited reverse-engineering capabilities. OEMs who use the trust architecture to validate and decrypt their code achieve significant resistance to this type of attacker because the driver modification causes secure boot to fail. By including the OEM's unique ID in the code signature, only cloners who purchase their own QorIQ processors are able to create duplicate systems. However, by including the Freescale unique ID in the code signature, even cloners with unprovisioned QorIQ processors cannot create systems capable of booting the OEM's code.
- **Functional clones**
  A functional clone is a reverse-engineered system intended to compete with the original ICS. For this example, we will call the original manufacturer OEM 1 and the cloner OEM 2. The cloner is typically another OEM, assumed to have the ability to build a system from scratch, and capable of working around most obstacles OEM 1 may put in his path.

  By leveraging the ability of the trust architecture to protect long-term secrets such as the proprietary code of OEM 1, OEM 1 can force OEM 2 to acquire (and destroy) multiple legitimate systems before being able to conduct comprehensive reverse engineering. If it is acceptable for a system to leave the factory with minimal functionality, remote provisioning can be used to raise the cost of attack. A remotely provisioned system must connect to a provisioning server before receiving the code needed to support a full suite of features. If a system has been tampered with, the credentials required to authenticate to the provisioning server will be locked out by the trust architecture.

  The goal of OEM 2 may not be to build a perfect clone but to offer some of the same advanced features as OEM 1. Although business or regulatory considerations may compel OEM 1 to support standards-based interactions with other vendors' systems, OEM 1 can include proprietary functions as well. For example, if OEM 1 supports proprietary protocols for advanced networking functions between two of OEM 1's systems, these systems can require mutual identification and authentication prior to using the proprietary protocol. This allows OEM 1's systems to refuse certain types of connections to cloned systems. The credentials for mutual authentication can be strongly protected by previously described trust architecture mechanisms.

Conclusion

ICS and some of the critical infrastructure which use them are multibillion-dollar industries, and the economic impact of the unavailability of these embedded systems (or of being practically unavailable due to lack of trust in their operation) is significant. Professional criminals have replaced thrill-seeking hackers, causing even bigger threats to security. Increasingly, the trend will be toward multicore processors, which run independent operating systems and are open to licensed applications or possibly even end-customer developed code. The QorIQ platform's trust architecture provides OEMs with the hardware anchor points they need to develop a trusted system and helps enable complete security in ICS.

How to Reach Us:

Home Page:
freescale.com

Power Architecture Portfolio Information:
freescale.com/power

e-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274
480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447
303-675-2140
Fax: 303-675 2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright license granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™

For more information, visit freescale.com/power

Freescale, the Freescale logo and QorIQ are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. All other product or service names are the property of their respective owners.
© 2012 Freescale Semiconductor, Inc.

Document Number: PWRARBYNDBITSTA REV 0

BUILT ON Power™