TECHNOLOGY WHITEPAPER

Hardware Features for Maintaining Security During Operation

Trusted Computing: The COTS Perspective Series

CURTISS-WRIGHT

DEFENSE SOLUTIONS

Read About

- NXP® QorIQ® Trust Architecture
- Intel® SGX and OS Guard
- Arm® TrustZone®

Introduction

In our previous Trusted Computing white paper, we discussed the importance of secure boot for trusted computing. After secure boot is implemented, our focus turns to hardware features built-in to the most popular defense and aerospace processor architectures to ensure the continued security of a trusted system during operation. Understanding these features, what they protect against, and how to effectively use them will enable embedded systems to operate securely even in the face of attacks. In most cases, some software has to be modified as well in order to take advantage of these hardware features. In a future column we will discuss the software aspect of maintaining system security during operation.

Because different processor architectures support different security features, this column will consider some examples of those features. It's important though to review your own specific platform's architecture to determine which features are available and germane to your system's unique requirements. Generally, the system integrator will want to use all of the security features that are available. Variables, such as cost, complexity, and the system's security requirements and threat assessment, can influence the decision about which features will actually be implemented. Each individual program will have to review their program requirements and make the appropriate tradeoff decision regarding security and cost, schedule, complexity. Discussions with commercial off-the-shelf (COTS) hardware vendors at the earliest stages of system development can be of great help in making the right choices.

NXP QorIQ Features

NXP's QorIQ Trust Architecture, supported in devices with either Power Architecture® and Arm processors, provides many hardware security features that can be used as the foundation to build a secure processing embedded environment in order to maintain the trusted environment during runtime. Because Trust Architecture features vary depending on the specific processor, it's important to look at your particular platform to determine which features are available and which are not.

IO MMU

To protect sensitive memory regions from being maliciously read or written to, QorIQ Trust Architecture implements IO MMUs that prevent memory from being accessed without explicit permission. These IO MMUs ensure that particular IO devices are prohibited from accessing regions of memory that are meant to be protected. It helps protect the integrity of data by ensuring that input cannot overwrite restricted memory regions. It also helps ensure the integrity of applications and data by protecting against unauthorized modification.

Security Monitor

The QorIQ Trust Architecture provides a Security Monitor (SM) to sense and control the security state of the QorIQ. The SM monitors and responds to potential physical changes to the underlying security features in the hardware. After the QorIQ passes secure boot, and if no hardware security violations are detected, SM enters a Trusted/Secure state. In this state, SM provides on-going monitoring of the QorIQ's security. If a security violation is detected, the SM can be configured to respond with a range of appropriate actions, ranging from Master Key lock-out to full SoC reset. It provides the ability to lock out access to root keys in the hardware based on failures. Monitoring includes checking hardware fuses to verify that error correction is still in line and no bits have been flipped unexpectedly. Failures can be detected by hardware self-checks (looking at error correction for fuse information), via external sources (analyze pre-defined pins that can indicate if something has failed system-wide), or via software sources (an application or the OS has started, and a software identifies a problem and decides to fail at that point).

Run Time Integrity Checker

The QorIQ Trust Architecture Run Time Integrity Checker (RTIC) can actively monitor a system for unexpected changes in operation. The RTIC is used to monitor the integrity of designated sections of system memory. It can periodically examine these specified memory regions to verify that contents have not changed from what is expected. RTIC compares cryptographic hashes with the hash of the memory created when the RTIC was first configured. If the hash values don't match, system memory is considered to have changed outside of program control, and a security violation is reported to the Security Monitor. This sort of protection ensures that even if application contents are modified via an attack, the modifications will be noticed, and the system can respond to the attack. For example, if someone has broken into the system and implemented a buffer overflow in order to change the instructions in order to initiate an attack. The type of security provided by RTIC is an example of a layered approach to system security, providing a second layer of defense able to notice and respond to attacks.

Secure Debug

In the system development process there is always a tension between securing the system (i.e., locking it down so that it can't be attacked) and opening up the system to enable it to be debugged. Secure debug is a capability that allows the hardware to manage the ability to debug. It is used to progressively shutdown the ability to debug through the system development cycle, from the lab to deployment. Using QorIQ's Secure Debug Controller enables a range of accessibility, from a completely open system, to a system that requires authentication to debug, to a system that completely disables the ability to debug. This hardware capability ensures that system security can respond to the changing requirements during system development lifecycle.

The Secure Debug Controller supported by the QorIQ architecture is implemented through control fuse settings through which OEMs can control the level of access available to external debuggers. Secure Debug supports four levels of access:

+ 1 - Open
+ 2 - Conditionally Closed without Notification
+ 3 - Conditionally Closed with Notification
+ 4 - Closed

When set up the system is completely open. To restrict access, the user burns specific fuses via software instructions, after which the debug feature is available only after authentication. To completely lock down access to the debug feature requires additional fuses to be burned.

Intel Security Features

Intel builds security technology into their products, including their embedded IoT lines, to provide a hardware foundation for building a trusted system. Apart from the secure boot capabilities, Intel processors support capabilities to ensure that system security is maintained during operation. Again, because features may vary depending on the specific processor, it's important to look at your particular platform to determine which are available and which are not. To help ensure secure operation following secure boot, Intel provides the following hardware features.

Figure 1: Curtiss-Wright's recently introduced VPX3-1260 SBC, powered by the new Intel Xeon® E-2176M (former codename "Coffee Lake") 8th Gen processor, is designed to take full advantage of hardware security features such as SGX and OS Guard.

Software Guard Extensions

Available for "Skylake" and newer platforms, Intel's Software Guard Extensions (SGX) technology enables software to define "enclaves" that act as a firewall to protect portions of memory from unauthorized access, even if the access comes from privilege mode. An enclave is Intel terminology that refers to a defined portion of memory protected with a "security fence" within which the software code can interact, but to which external access is only available via defined instruction sets that the processor provides. The SGX instructions enable the protected memory regions to be defined via software. It then uses hardware to enforce access to the software defined regions. This capability provides an additional layer of defense, on top of the distinction between supervisor or user modes. If an attacker is able to inject code to attempt unauthorized access, even from a privileged access mode, SGX ensures that the code cannot necessarily gain access to another enclave's data and instructions. To effectively use SGX requires more than just modifications to the OS. It also requires modifications to application code.

OS Guard

Typically a system is running in either the supervisor mode or it is running application code. When the OS is done with its kernel operations, and is ready to switch over to running the application code again, it needs to un-set a bit in a jump register that allows it to transition from privileged mode to un-privileged mode. This means that even if there is a buffer overflow attack possible within the kernel which allows an attacker to try to jump to unprivileged user code, if that jump register bit isn't set a fault will occur. Intel's OS Guard provides a mitigation mechanism in hardware that protects against a user trying to execute non-privileged application code when the system is executing in privileged mode. This capability allows the hardware to protect a secure system from attacks that try to escalate privilege.

Arm and TrustZone

In addition to the secure operation hardware features provided by NXP's QorIQ Trust Architecture, Arm IP-based processors also support TrustZone. TrustZone adds another layer of defense on top of user mode and privilege mode. Using TrustZone, a bit can be set to place user or privilege modes into a trusted mode of operation in order to limit access for such operations as accessing specific memory regions, or to prevent code from being modified so that it can be made run certain processes. TrustZone positions security at the heart of the hardware and enables much finer grained management of partitioning hardware resources. It also ensures that non-trusted code cannot gain unauthorized access to resources used for trusted operations.

Virtualization Technology

Power Architecture, Arm, and Intel processors also support virtualization, which enables multiple separate OSs to run on a single piece of hardware concurrently. Virtualization provides a mechanism for enforcing security at a level above the OS. Its use can ensure that unauthorized access cannot leak across to other guest OSs even if the OS has a bug that allows an attacker to gain access to privileged areas of operation. However, this level of security can introduce a performance penalty because of the overhead of virtualizing system resources.

Virtualization can be realized in two ways: Type-1 hypervisors operate directly on the hardware, while Type-2 hypervisors operate on top of an OS, and virtualize the use of the host OS's resources so they can be used by multiple guest OSs. Virtualization works in a manner similar to TrustZone in that it provides an additional level of execution privilege that operates distinctly from supervisor mode. This allows the hypervisor to manage resource requests from the multiple guest OSs. Normally the hypervisor will trap requests for resources (eg., memory, CPUs, I/O, etc.) that come from the guest OS and manage their allocation and consistency across all running guest OSs.

When More Security Capabilities are Required

While the secure operation hardware features discussed above provide examples of fundamental building blocks for trusted computing, for many military and aerospace systems these features may not be sufficient to meet all security requirements. In those cases, additional security capabilities will need to be integrated into the system in order to increase the security level as needed. To enable secure operation of the system, these additional security capabilities typically require a COTS vendor with security experience in order to design in the appropriate capabilities at the component and module level during product development. In addition, firmware and software will likely need to be developed in order to take full advantage of these additional components and develop the required system level security solution.

It's important to think of system security as more than just a collection of hardware features. Security results from understanding the system's fundamental building blocks, how the system operates, and the potential attack vectors. This knowledge is then applied to design a system that allows the system developer to make appropriate selections based on their program's security, performance, and cost profile.

For information on Curtiss-Wright's Trusted COTS™ (TCOTS) Program for protecting critical technologies and data in deployed embedded computing systems visit our website at www.curtisswrightds.com/technologies/trusted-computing/.

Author(s)

Steve Edwards
Director, Secure Embedded Solutions & Technical Fellow
Curtiss-Wright Defense Solutions

Learn More

Curtiss-Wright Products

- Data at Rest Protection
- Security Enabled Curtiss-Wright SBC and DSP Products

Curtiss-Wright Case Study

- Building a Truly Trusted Computing Solution with COTS Hardware and Intel Security Capabilities

Curtiss-Wright White Papers

- The Many Faces of Trusted Computing: What You Need to Know to Protect Critical Platforms and Data
- Getting Secure, Intel-Based Solutions to Market Faster - Why the Hardware Vendor's Boot Security Implementation Is So Important
- Beyond Trusted Computing: Extending Protection Capabilities to Deliver TrustedCOTS Solutions
- Trusted Computing: The COTS Perspective Series
  1. Introduction to COTS-based Trusted Computing
  2. Trusted Boot
  3. Hardware Features for Maintaining Security During Operation
  4. Considering the Role of Hardware in Securing OS and Hypervisor Operation
  5. Application Development, Testing, and Analysis for Optimal Security
  6. Developing a Secure COTS-Based Trusted Computing System
  7. The Impact of Protecting I/O Interfaces on System Performance
  8. Decomposing System Security Requirements
  9. Establishing a Trusted Supply Chain
  10. Certification Authorities for Trusted Computing in Military and Avionics Products