ESTABLISHING THE ROOT OF TRUST

Authors:

Vincent Zimmer
Senior Principal Engineer, Intel
Vincent.zimmer@intel.com

Michael Krau
Systems Engineer at Intel
Michael.p.krau@intel.com

August 2016

Establishing the root of trust

The first step in securing a computing device - from a simple embedded device to a supercomputer and everything in between - is to ensure that it can start up under the following conditions:

- It is operating as expected
- All the firmware needed to run the system is intact
- It has not been tampered with in any way

As described in the first white paper in this series, *Understanding the Chain of Trust and Its Vital Role in Keeping Computing Systems Secure*, the UEFI specification includes a mechanism for ensuring the integrity and security of firmware (the all-important link between the hardware and the operating system) as a system starts up. This mechanism is called Secure Boot and uses public key cryptography to validate that each piece of firmware has been digitally signed and is therefore unmodified as the system starts up.

In a chain of trust, each piece of firmware must be digitally signed before it can start up. Once one piece of code has been validated, it can then validate the next section and so on until the system is fully booted and control handed over to the operating system. But how does that chain get started? While difficult, it would be possible for an attacker to inject malicious attack code of some sort prior to start of the chain of trust to gain low-level and nearly undetectable control over the system. To prevent this, the chain of trust requires a strong foundation. In modern systems, this is known as the root of trust.

A root of trust, one that can be counted on to anchor the chain of trust in the face of the most determined attackers, can be established in a number of ways. The most secure approaches use some form of an unchangeable section of hardware to validate the initial keyed signature, but there are a number of effective approaches. Ultimately it comes down to the level of security you're comfortable with and an understanding of the approach used to establish the root of trust. This white paper looks at several common methods for establishing a root of trust as the basis for the UEFI Secure Boot process.

Root of Trust definition

The root of trust is ideally based on a hardware-validated boot process to ensure the system can only be started using code from an immutable source. Since the anchor for the boot process is in hardware it cannot be updated or modified in any way. When this foundation is combined with the cryptographically secured UEFI Secure Boot process, there are no easily accessible gaps for hackers to exploit.

A root of trust can be started by a variety of methods. The simplest mechanism is to run start-up code directly from a non-writable device (and thus protected) in the processor's memory map. Alternatively, to allow updates and more flexibility, the code can be loaded from a protected memory region into a protected memory store of some sort set aside for firmware execution, among a number of other methods. The important aspect for a root of trust is to be sure that the initial code is what the manufacturer intended, before execution. When it starts, the root of trust derives its internal keys from supplied device identity inputs and executes self-tests and code validation for itself. If these tests pass, it can move on to validate the first piece of code in the chain of trust.

A cryptographically based root of trust includes the public key portion of the key pairs used in the cryptographic operations of signing and verifying signatures on digital certificates. The public key is nothing more than a static ASCII text file containing cryptographic keys. If another mechanism is used to establish the root of trust, then this may not be the case. However, even some other root of trust mechanism could benefit by using a cryptographic mechanism to verify that the original firmware is inviolate. A root public key can be a very large number—most of them are currently 2048 bits in size. While this is no problem for PCs or smartphones, it can be a challenge for low-power IoT devices in some cases. The industry is starting to address this problem in a variety of ways.

To be sure, what sounds like a straightforward concept becomes more complex when it comes down to actual implementation. The approaches used by vendors to establish a trusted, unalterable root of trust are different from platform to platform, and even within the same platform there can be variations intended to give OEMs varying degrees of protection vs. cost. A very secure solution may cost more in terms of actual dollars, boot time, and runtime flexibility than a less secure system. What's more, in some cases the section of hardware devoted to initial start-up can be asked to do more than security checks, adding to the level of complexity.

Hardware-based RoT implementations

For any individual or organization concerned about maintaining a secure computing environment, the operating assumption needs to be this: boot securely – or don't boot at all. Major players across the industry have begun to adopt that mantra and implemented mechanisms that provide a hardware-based root of trust. For custom chip designers building Systems on a Chip (SoC), major embedded design suites now include provisions for incorporating a root of trust as the basis for a secure boot process. Here is an overview of notable approaches.

AMD Platform Security Processor

The latest low-power APU designs from AMD incorporate a Platform Security Processor, or PSP alongside the main CPU's x86 core. The PSP is actually a separate 32-bit ARM Cortex-A5 core with its own memory. The PSP is designed to provide a secure processing path, a trusted execution environment (TEE), a Trusted Platform Module (TPM), and a cryptographic co-processor capable of using the full set of ARM's TrustZone hardware-enabled security services.

In normal operation, the PSP's primary role is to protect the x86 core and provide a hardware-based root of trust. This secure core boots first using its own ROM and SRAM and verifies the code that boots the x86 core and launches the UEFI Secure Boot process. The PSP's cryptographic co-processor can also support x86 applications to secure off-chip storage.

ARM TrustZone

According to ARM, TrustZone is hardware based security built into ARM SoCs by semiconductor chip designers who want to provide secure end points and roots of trust. The family of TrustZone technologies can be integrated into any ARM based system, from the smallest microcontrollers to high performance applications processors.

TrustZone technology within ARM processors is commonly used to run trusted boot and a trusted OS to create a Trusted Execution Environment (TEE). Typical use cases include the protection of authentication mechanisms, cryptography, key material and DRM. Applications that run in this secured area are called Trusted Apps.

The hardware-based root of trust is provided by what ARM calls CryptoCell, a range of security sub-systems and hardware components that provide platform level security as well as hardware support for security acceleration and offloading. CryptoCell includes efficient hardware cryptographic engines for root of trust/key management, secure boot, secure debug and lifecycle management. It is comprised of hardware, firmware and SoC-external tools

Intel Boot Guard

Introduced with Intel's 4th generation core processor platforms, Intel Boot Guard is a hardware-based technology designed to prevent malware and other unauthorized software from replacing or tampering with the low-level UEFI firmware. Boot Guard has three separate modes: verified boot or measured boot, or a combination of both. Verified boot mode cryptographically verifies an initial boot block while measured boot uses a measuring process. The verified boot approach is the one used by the majority of OEMs.

Boot Guard configurations vary somewhat across OEMs. In general, the OEM is responsible for configuring a public key for the verified boot and establishing boot policies. The security of the verified boot is rooted to the OEM's key pair. The OEM generates a 2048-bit key that is only used for verifying the initial boot block, the private portion of which must be kept securely. The public portion of the key is then programmed into field programmable fuses during the manufacturing process. These fuses cannot be updated once written, thus providing a solid starting point for UEFI Secure Boot. The OEM is also responsible for setting policies around what steps to take if boot integrity checks fail, such as shut down the system or enter some sort of restricted repair mode.

Internet of Things

Moving forward, IoT devices will be asked to control thousands of critical systems and will be exposed to network-based threats. There is a strong need for a hardware-based root of trust mechanism to thwart large-scale attack coupled with cryptographically signed firmware, BIOS and operating systems to verify that the code has not been tampered with.

One example of how this can be implemented for IoT devices is Synopsis's DesignWare tRoot Secure Hardware Root of Trust. According to Synopsis, tRoot is a "highly-secure hardware root of trust that is designed to easily integrate into SoC ASICs and provide a scalable platform to offer diverse security functions and applications." Similar to AMD's PSP, tRoot uses a secured companion processor to the host processor to host the public key for the root of trust. It uses a key management module to store keys in non-volatile memory in the SoC and provides secure key ports to derive and load keys to other security subsystems on the chip. Once the root of trust has been established, UEFI Secure Boot or a proprietary chain of trust implementation can be used to start up the system.

In addition to options for custom chip designers, secure microcontrollers are available that incorporate a root of trust with a pre-loaded, immutable root key. The root of trust in these devices is located in internal read-only memory or internal flash that is locked at the factory. In such cases, the IoT designer can submit their public key to the chip manufacturer for greater flexibility and control.

Security is a major concern for the IoT and could be a critical factor in the ultimate growth and prevalence of these devices in society. Major security breaches could easily derail their momentum. As such, it’s critical that IoT developers embrace suppliers who offer a hardware-based root of trust, and take advantage of specifications such as UEFI’s Secure Boot to minimize potential attack vectors.

Building trust

Security protection for computing systems large and small must start at the very beginning and cover the entire boot process. The most rigorous security schemes can be easily undermined if the starting point is not to the task. To address this, hardware manufacturers have added root of trust technologies to their products. Regardless of the implementation, the intention is the same: to provide a hardware -based unalterable, cryptographically secure basis for starting up the system.

There’s little doubt that a hardware-based root of trust combined with the chain of trust process used in UEFI Secure Boot (or a similar approach) is the best way to ensure system security during the start-up process while still ensuring flexibility and allowing for secure updates and revisions.

How to Access the Latest Technology

Anyone can access UEFI forum Secure Boot specifications and easily understand the technology, member of the forum or not. However, to participate in discussions and effect future UEFI forum standards, you must join as a contributing member.

About UEFI Forum

The UEFI forum is a world-class non-profit industry standards body that works in partnership to enable the evolution of platform technologies. The UEF forum champions firmware innovation through industry collaboration and the advocacy of a standardized interface that simplifies and secures platform initialization and firmware bootstrap operations. Both developed and supported by representatives from more than 300 industry-leading technology companies, UEFI forum specifications promote business and technological efficiency, improve performance and security, facilitate interoperability between devices, platforms and systems, and comply with next-generation technologies.

The Forum’s spheres of input and influence are large: Membership represents major voices from all players in the industry—open source to proprietary technology, hardware to software, mobile to stationary devices, etc. The Forum liaises with other fundamental standards groups that are essential to computing. For more information about the UEFI Forum and current specifications go to www.uefi.org.

# # #