Secure Boot with the EmSPARK™ Security Suite

Secure Boot is a process in which software on a device is identified, authenticated, and started up when the device is powered on. Secure Boot also includes the protection of the firmware images stored in the non-volatile memory whether or not the device is powered on. This process requires several stages of authentication, protection, and encryption/decryption, in order to ensure that the device is secure. This solutions brief covers the process step-by-step and explains how these steps are implemented using the EmSPARK™ Security Suite.

INITIATION

Secure Boot is initiated in **READ-ONLY MEMORY (ROM)**. **ROM** is memory on the device that has instructions built into the device's hardware (e.g. the processor), which tell the device what to do when it is powered up. The two hardware components that can be used to create configuration for the ROM are:

+ **Pins**: these are connections from the System-on-Chip (SoC) to external components on the device
+ **Fuses**: these can be "burned" so they store actual permanent data directly in the SoC; for example, private keys or machine-access-code (MAC) addresses for networking

SOFTWARE AUTHENTICATION

The following steps are covered to begin the Secure Boot process:

1. **ROM** initiates the process by following instructions set by pins and fuses. This process is called the **Boot Loader**.
2. **ROM** loads the first software (Secondary Program Loader, or SPL) from **Non-Volatile** (i.e. **Flash**) **Memory (NVM)**—memory that contains data in the absence of a power supply—into **RAM (Random Access Memory)**, which temporarily stores the data, serving as the device's "working" memory.
3. The software is verified by comparing the SPL to information burned into the fuses. This is typically done by reading a signature, which was created using a cryptographic key and read using a corresponding key burned into the fuses.
4. After verification, the SPL is loaded and the process of "unpacking" (i.e., decrypting and locating) the remaining software begins.

"Secure Boot" is often described as steps 1-4. To complete the process, the steps below should be followed.

MEMORY ISOLATION, TRUSTED EXECUTION ENVIRONMENT (TEE) ESTABLISHMENT, AND APPLICATION AUTHENTICATION

5. The Secondary Program Loader (SPL) separates the RAM into two partitions: Secure and Non-secure. The Secure area will manage the Trusted Applications (TAs), keys, certificates, and sensitive software, like an Artificial Intelligence (AI) algorithm or other form of intellectual property (IP). The Non-Secure area will house software that is well known to the outside world, such as Linux components.

6. The SPL then loads the encrypted object located in the NVM, called a Binary Large Object (BLOB), into the secure area of the **RAM**. The BLOB is authenticated (i.e., signature is verified), un-encrpyted using keys, and Sequitur's secure Operating System (OS), called the CoreTEE™, is then loaded. The CoreTEE™ is Sequitur's Trusted Execution Environment (TEE), or secure OS, enabled by ARM's TrustZone™ architecture.

7. The SPL then passes control to the CoreTEE™, which then sets up the secure operating environment, which will house keys/certificates, Trusted Applications (TAs), etc.

8. CoreTEE™ passes control back to the SPL, which then sets up the non-secure operating environment. The SPL then executes the loading of the OS and applications in the Non-secure Environment. A Linux example is below:

    a. The SPL loads and runs Linux uBoot (verifies and decrypts). UBoot is a Bootloader, which is a program that loads an Operating System.

    b. The SPL loads the Linux kernel (verifies and decrypts)

        i. Also loads .dtb files (device tree binary or blob—this file passes hardware information about the board to the Linux kernel)

        ii. Also loads a ramfs, which is a file system that allows the creation of a RAM-based storage area for Linux files

After the Operating System (OS) and its associated files are loaded/booted, the **Secure Boot** process is complete.

Learn More

EVAL KIT Download the free evaluation kits and user's guide for more details and code examples

See a real-word implementation of Secure boot in the Johnson Controls Case study

SEQUITUR LABS

PO Box 1127
Issaquah, WA 98027

Sequitur Labs is developing seminal technologies to improve trust in a connected world, reducing the cost and complexity to build secure embedded and IoT devices. Sequitur's products span a range of disciplines required for trusted computing, from boot through the full device lifecycle. Sequitur's security solutions provide real business value to device makers, such as reducing BoM costs, protecting revenue by thwarting IP theft, improving product reliability and reducing liability, and improving device lifecycle management processes. To learn more about Sequitur's security platform, visit us at www.sequiturlabs.com or follow us at @SequiturLabs.

MCSEm-0001-Rev A. Printed in the U.S.A.