Maintaining data integrity in Internet of Things applications

By Arlen Baker, Principal Technologist, Wind River

The Internet of Things (IoT) is promising to open up many new opportunities for businesses to offer new and exciting services. However, with the myriad of devices and business assets connected open to the internet, the need for a strict and reliable approach to security is essential. One such security model that has been in use for a long time is that of the CIA triad. The key components of this model (see Figure 1) are based around confidentially, integrity and availability.

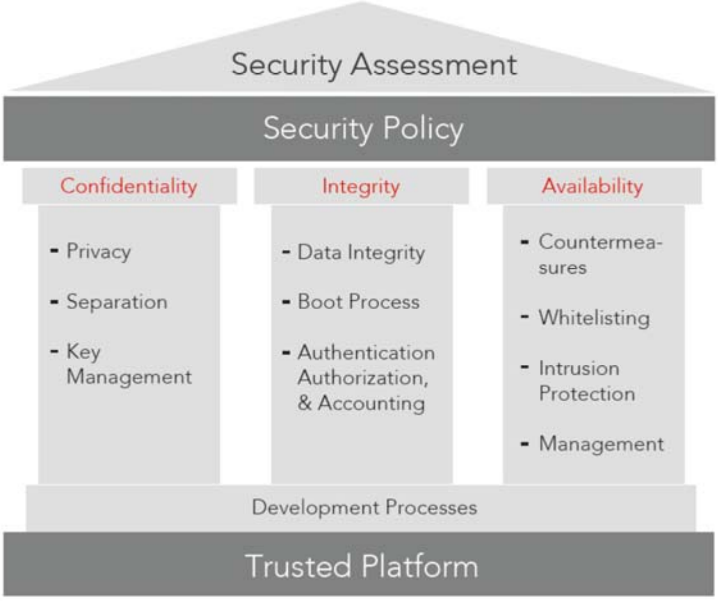

Figure 1: Confidentiality, Integrity and Availability model

In the context of IoT, confidentiality caters for protecting privacy of IoT devices, integrity looks after the data contained within the device while availability covers accessibility of the device. This article will focus on the aspects of maintaining data integrity.

When we think about data of an IoT device we should think about not only the data being generated or used by it but also its own programming data, this including all aspects of program software, configuration parameters and operating system software. To guide the process of integrity it is helpful to consider three different states that data can exist, namely in motion, at rest and in process. Figure 2 illustrates the decomposition of the Integrity principle into sub-principles, and finally implementations that the IoT device can incorporate in protecting the integrity of its data.

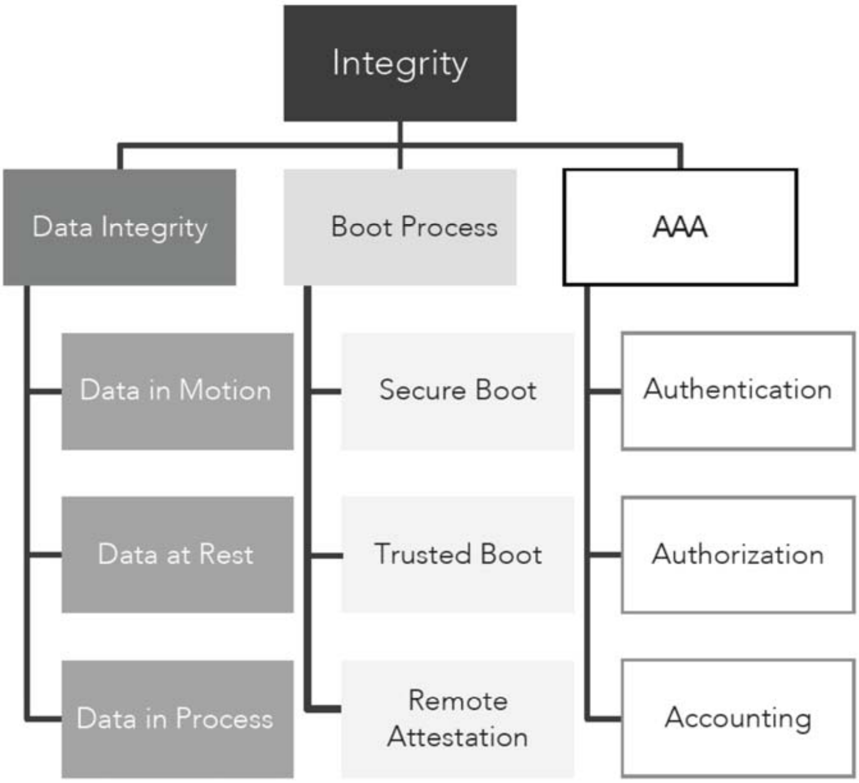

Figure 2: Integrity of data considerations for an IoT device

Any breach of data integrity will mean that an IoT device cannot operate correctly but it also potentially exposes the device to being exploited and become a compromised platform from which other attacks can be launched. The usual method of verifying the integrity of data is by a mathematical algorithm called a hash, of which the secure hash algorithms (SHA) is most popular.

Data-in-motion requires that data be protected from modification while on its journey from sensor to cloud application. While a hash technique can be used an attacker could make a change to the message and recalculate the hash. A stronger approach is by using a data integrity check with a shared private key as illustrated in Figure 3. This is called a keyed-hash message authentication code (HMAC), and since it needs a shared private key it must be protected just like any other cryptographic key.

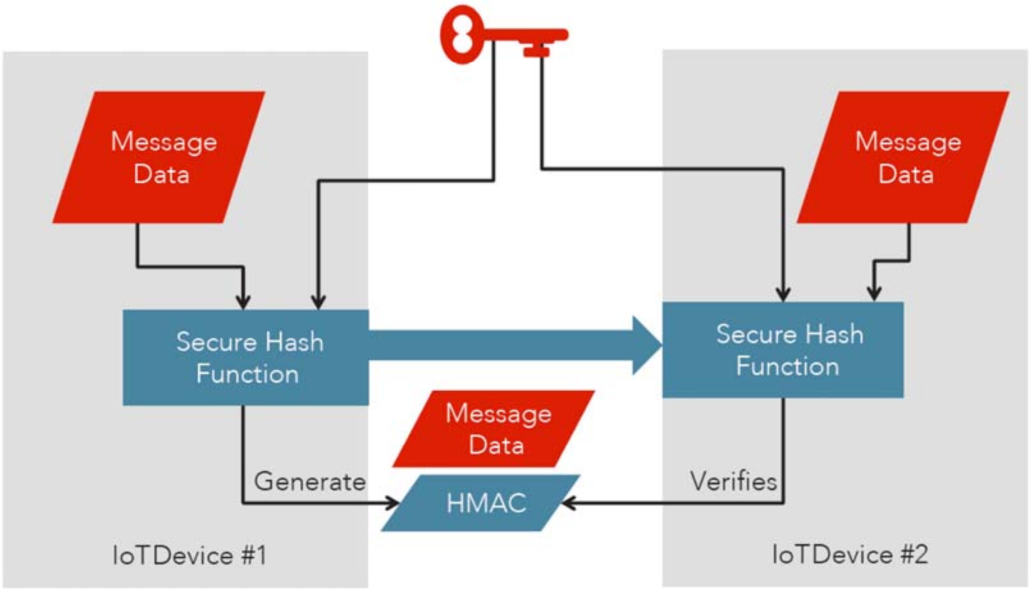

Figure 3: HMAC workflow

When it comes to data-at-rest there are several considerations. Firstly, the stored program data will need to be verified and that will be done at boot time, see secure boot in the next section of this article. Configuration data and any stored device data should always be verified prior to being processed by an IoT device. Periodic integrity checks can be made during operation and always at start-up and shutdown.

Integrity checks also need to be used for data that is being processed to ensure that the data and its flow can be trusted. By using unique enumeration values through the software will ensure APIs are called with parameters unique to only that API.

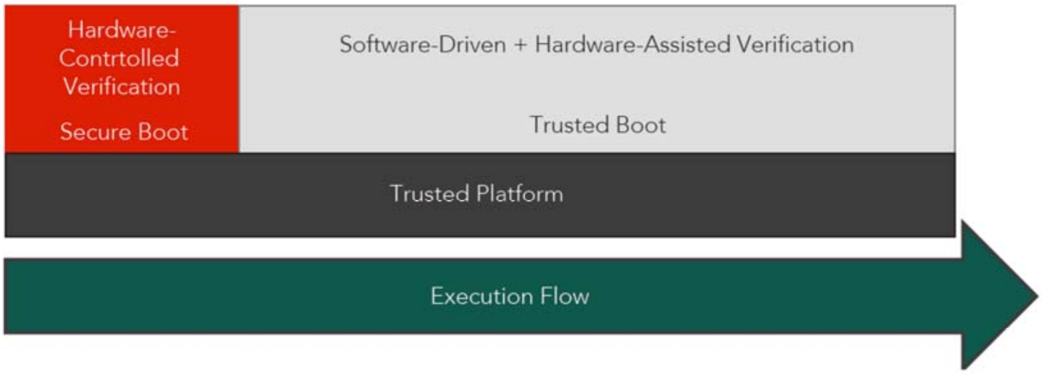

Figure 4: Boot process

The boot process verification can be split into two discrete parts as indicated in Figure 4. Firstly, a secure boot process is initiated by the trusted platform and determines the authenticity of the initial images.

The verification process is best implemented using digital signatures. Digital signatures combine the use of a hash function's message digest along with the asymmetric private-key encryption of that hash function by the author. The digital signature is verified by recalculating the message digest, decrypting the associated digital signature with the public key, and comparing the message digests. If the message digests match, then the integrity of the software is verified.

The second stage in the boot process is trusted boot where the remaining images and data are verified by previously verified software. It is best if the boot process includes hardware to perform the integrity verification processing, because the immutable properties of hardware (whether system-on-chip (SoC) or field programmable gate array (FPGA)) mitigates the risk of a malicious change causing a breach in verified boot processing. The process of one verified image passing control to another verified image is called a chain of trust. See Figure 5. The chain-of-trust approach ensures that only verified software is loaded into the system.

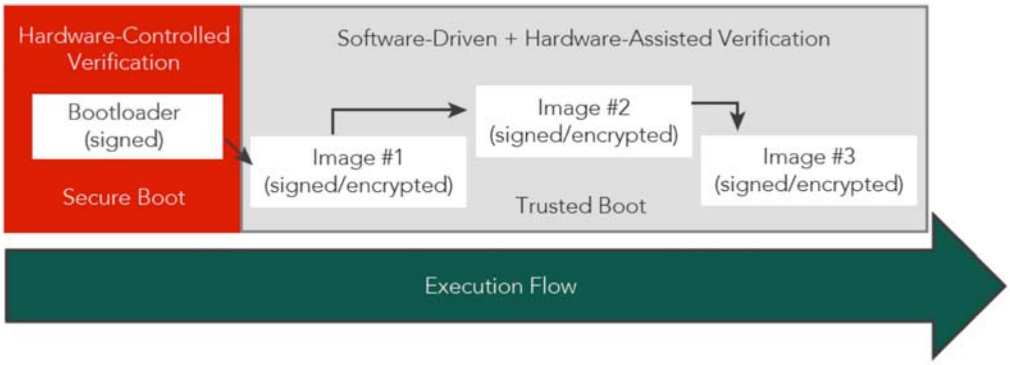
Hardware-Controlled Verification
Bootloader (signed)
Secure Boot
Software-Driven + Hardware-Assisted Verification
Image #1 (signed/encrypted)
Image #2 (signed/encrypted)
Image #3 (signed/encrypted)
Trusted Boot
Execution Flow

Figure 5: Chain of trust

If the IoT device has reliable, always-on internet connectivity, then remote attestation can be used to report and validate key boot parameters during the secure boot and trusted boot phases to a physically separate server. These parameters are typically a hash of the components of the boot process (bootloader, applications, etc.). The server than compares these measurements and determines the trustworthiness of the IoT device. The transmission of the measurements is secured and must include the identity of the IoT device.

Any internet connected device is at risk, and IoT devices are no exception. Providing a further layering of integrity defenses comes under the heading of AAA (pronounced triple A) which stands for authentication, authorization and accounting. Its role is to determine which other devices on the network it can communicate with and what data can be transferred. A trusted party is needed to broker the communication between the IoT device and the distant device or server. A well-established protocol called Kerberos can be used to establish this trust between devices over the network.

Once authenticated, authorization follows. Authorization is the determination of the type of access allowed to a resource within the network. A network-wide security policy is used to list what the resources are, the paths of communication to and from each resource, and to what level the access can occur.

The final component of AAA is that of accounting. This is the generation of a log of events that denote security-related activities on and by the IoT device. Event logs can be stored within the IoT device or, more usually, on an external server. The security policy will denote which events need to be recorded. Reviewing the logs and associated data will help determine what led to an attack, where it came from and what happened during the attack. It is best that these events are recorded as near to real-time as possible to minimize the damage from the attack in order that any initial, front-line response as defined in the security policy can be made.

Because of the large number of IoT devices that typically must be monitored, along with the large number of events that need to be parsed and managed, a specialized server is required. This type of server is called a security information and event management (SIEM) server. A SIEM server is able to correlate received security event messages from IoT devices and use predictive analytics to determine if an IoT device is at risk of an attack.

In conclusion, in order to benefit from the huge potential of connected IoT devices, the need for a strict and reliable approach to security is essential. The CIA triad provides a very simple and convenient model of both discovering and representing the security needs of your IoT device. Maintaining data integrity is certainly one of the key aspects of implementing a set of security policies, and the same considerations and approach needs to be made for that of confidentiality and availability. Wind River provides both a wide and deep range of products and solutions that implement the Integrity principle of the CIA Triad for our customer’s IoT needs.

+++Ends