# Secure boot on embedded Sitara™ processors

Texas Instruments

Amrit Mundra
*Security architect and systems engineer*

Hong Guan
*Security application engineer*

*Texas Instruments*

# Executive summary

The security of an embedded system should not be an afterthought, an after-the-fact add-on or a nice-to-have feature. Either security is designed into the embedded processor so that the device operates as intended from the time power is first supplied, or it's not. And if it's not, there may be problems.

Of course, when an issue as complex as security comes up, usually the first question a designer will ask is where to begin? Without a strong foundation, the security subsystem of an embedded system will be built on sand. Every security feature or capability in the system will be only as strong and as effective as the foundation upon which it is built. Building a powerful security foundation begins when the system boots.

This white paper explains how the secure boot process has been designed into TI Sitara™ processors, as well as their supporting infrastructures, and how these design features provide the root-of-trust upon which designers can begin to build security subsystems to meet their desired security objectives.

## What is secure boot?

The system booting process occurs every time a system is reset or power is applied to an unpowered system. The objective of a boot process is to initialize all of the necessary hardware and firmware to allow the system to operate. Typically, the embedded multicore processor starts booting from on-chip read-only memory (ROM), which does basic device initialization, followed by accessing the boot code/software, which begins the whole process. In typical multicore embedded systems, this boot firmware is stored in nonvolatile memory like a Flash memory that is external to the processor itself.

The first possible point where the security of the system might be compromised is during the boot process when the system is becoming operational. If this process is not secure, then no other subsequent process that executes on the system can be assumed to be secure. That is to say that in the absence of a secure boot process, there is no root-of-trust established in the system. So, to secure the boot process, the boot firmware stored in memory must be certifiably secure and authentic. If hackers have tampered with the boot code, they may have inserted malware that can hijack the system entirely, download the intellectual property (IP), snoop on unsuspecting users or take any number of nefarious actions. One of the fundamental technologies that is used to secure the boot process is cryptography, which can be used to limit access to boot code to only authorized users, to secure code as it is transferred from memory to the processor and to certify the authenticity of boot code as it arrives to be processed.

## Fundamentals of cryptography

Security keys play a major role in many security schemes. The two most common key processing techniques employed in embedded systems to secure the boot process are asymmetric and symmetric cryptography. These techniques often rely on other elements of cryptography, such as random number generators and hashing. Random numbers are used as the basis for many of the encryption algorithms that are currently prevalent in the industry. A hashing engine performs a calculation on a block of data and produces a number, which is associated with the data and referred to as a digest. Surreptitious or accidental changes to the data will change this hash value and thereby alert the security system that someone has tampered with the data. Hashing is particularly useful in certain cryptographic operations such as digital signatures, data integrity, non-repudiation, message authentication and other forms of authentication.

- ***Asymmetric Cryptography***

Asymmetric cryptography involves a matched set of two keys, a Public and Private Key. The Public Key is just that, public. It is available to anyone, but whatever is encrypted, such as code or data, by the Public Key can only be decrypted by the holder of the Private Key. Similarly, the data signed by the Private Key can be verified by Public Key.

The Public Key, since it is public, provides no security, but it can be used to certify the authenticity of digitally signed data. If signed data can be verified by a Public Key, then the source of the data is known since that data could only have come from the holder of the Private Key that signed the data. These techniques can be used by an embedded processor to authenticate the boot code stored in external memory to check that it has not been tampered with or hacked.

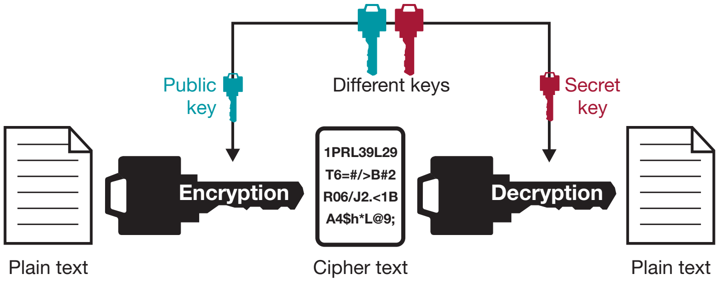

***Figure 1:*** *Asymmetric cryptography*

- ***Symmetric Cryptography***

This cryptographic technique involves only one key for both encryption and decryption. The key size can vary from 56 to 256 bits; the larger the key, the more security it provides. Various security algorithms like AES, DES, 3DES and others require keys of different lengths. Since security depends on just one key, it must be protected to the utmost by the customer.

A symmetric key is used in secure boot flow to provide code/data confidentiality to meet the goal of IP protection, where the code and data while sitting in external Flash is encrypted with the symmetric key.

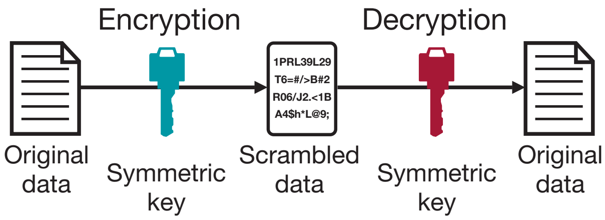

***Figure 2:*** *Symmetric cryptography*

## Secure boot key management

Key management is an important aspect to allow the security of keys involved in the secure boot. TI's Sitara™ processors are designed to keep general Public Key infrastructure concepts in view to promote easy role-based key definition and associated management. Typically customers use a hardware security module to protect keys and use Public Key infrastructure for key management.

Sitara AM335x, AM43x and AM57x processors provide the ability for a customer to specify the root security key (Public Key) that acts as a root-of-trust and is fused into the device. Once fused this root security key cannot be reversed or muted.

This root security key is typically entrusted to the manufacturer's chief technical officer for security. The Chief Technology Officer (CTO) is able to further bind multiple Public Keys for the use of their organization's development teams and cryptographically associate these keys with Root Public Key. These keys are analogous to a keyring containing the Public Keys assigned to the various software development teams within the development organization that are working on a system's boot code or the various product lines of the company. At the discretion of the CTO, any of these Public Keys can be revoked at any time should this be required.

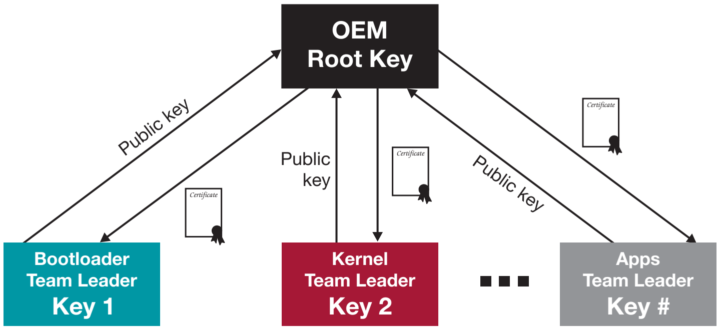

***Figure 3:*** *Device keys delegation*

## Provisioning keys in hardware

The first step in enabling secure boot on the device is to create cryptographic keys and provision providing the generated keys in the hardware. Using the hardware security module is a common industry practice to generate and protect booting keys.

TI provides evaluation modules (EVMs), software development kits (SDKs), tools and documentation to showcase the generation of keys and procedure to provision booting keys into the device. These collaterals are designed for early evaluation of key provisioning and support customers to transition from development to volume production.

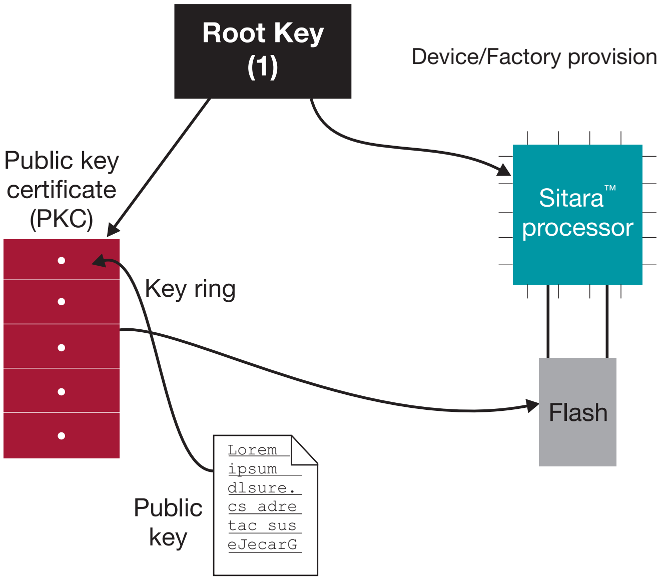

***Figure 4:*** *Device key management*

## Code encryption and signing

The other aspect of secure boot is to cryptographically associate the code/software with the keys provisioned in the device ***such that the device only loads and executes trusted code***. This trusted code is then downloaded to system non-volatile memory on a production line.

The code/software which makes up bootloaders, kernel, file system, etc. is signed and encrypted with associated keys that are provisioned in the device. In a typical scenario, this step involves pulling code into the secure system like hardware security module and cryptographically signing the code. If the encryption option is used, then the code/software is also encrypted with associated booting keys.

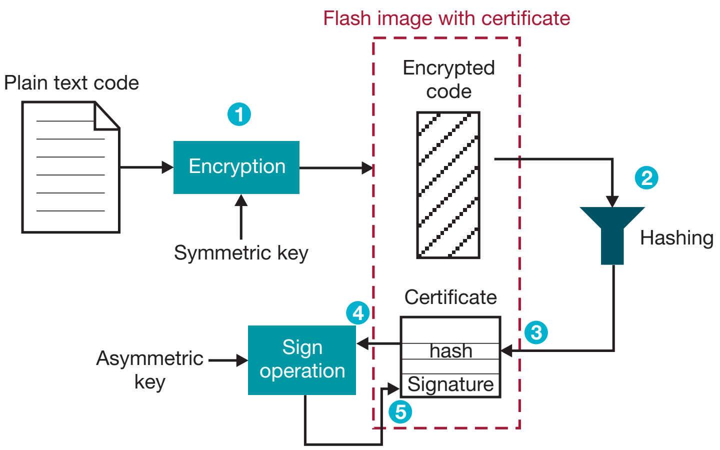

***Figure 5:*** *Code signing and code encryption*

## Secure booting

### *Takeover protection (Authentication):*

The prime function of secure boot is to provide takeover protection, that when properly configured can assist customers in designing their systems such that the device only executes authentic code and rejects code that is not signed by authorized keys. Sitara processors use cryptography along with non-mutable secure boot architecture to ensure the device always forces the check for takeover protection during boot up.

Secure booting starts at reset, as part of secure boot the device initializes itself in preparation to receive signed and encrypted software/code from boot media. This step involves creating secure partition of on-chip SRAM, initializing hardware cryptography engines, switching to secure CPU mode and configuring various controls within devices to create a secure environment.

Device on-chip ROM then fetches first code along with secure boot certificate from external media and loads into secure SRAM. On-chip ROM then extracts the root Public Key from the certificate and compares against the value programmed in the device, if the Root Public Key match then the certificate is deemed trusted. On-chip ROM then uses Root Public Key to cryptographically authenticate key ring data structure that contains more Public Keys, if authentication passes for key ring then all keys in key ring are deemed trusted and can be used to authenticate subsequent software components.

Furthermore, device on-chip ROM fetches bootloaders and other software components and authenticates using either the Root Public Key or one of the keys from the key ring. The device also offers an application programming interface which can be invoked by trusted software to authenticate the next software and pass control to next software.

### *IP Protection (Confidentiality)*

IP Protection is intended to offer system manufacturers a way to have their IP/code encrypted while sitting in external boot media. IP Protection is also sometimes required for certain security certification.

Sitara processors support IP Protection where the code/software in external boot media is encrypted. As part of secure boot, the Sitara processor also decrypts the code/software using keys provisioned in the device.

## Conclusion

Secure boot, when properly configured, is the foundation for providing a root-of-trust and a requisite for system security. Sitara processors are enabled with the hardware, powerful cryptographic algorithms and support infrastructure needed to support secure booting that will enable designers to build security subsystems that meet their desired security objectives.

## Additional resources

Please visit TI's Sitara processors website or to obtain more information about secure boot on AM437x processors, please submit your request here.

# IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265