Security Features in the i.MX31 and i.MX31L Multimedia Applications Processors

by: Asaf Ashkenazi

1 Executive Briefing

Security is an increasingly important feature of wireless mobile devices such as cell phones, ultra-portable computers and integrated media players. Instances of hackers and pirates breaking into portable devices and stealing private information and copyrighted content are becoming more and more common. As such, security is a high priority for i.MX31 and i.MX31L processors. Products built with Freescale applications processors provide a platform that consumers will trust to help protect their personal and confidential data. Private information, such as credit card numbers and access licenses, can be stored in a protected manner and used, allowing e-commerce and advanced content-based subscriber services. Products built using our processors will be able to demonstrate to content providers that they reliably help to protect licensed content.

To address these security challenges and to provide an extensible platform for addressing future security needs, the i.MX31 and i.MX31L processors incorporate

Contents

several hardware blocks and architectural features targeted to help secure the platform:

- Platform Independent Security Architecture (PISA)[1]
  - High-Assurance Boot (HAB)
  - Security Controller (SCC) including chip unique secret key
  - Run Time Integrity Checker (RTIC)
  - Memory Management Unit (MMU)
- Chip-unique identification number
- Random Number Generator Accelerator (RNGA)
- Secure JTAG Controller (SJC)
- Physical tamper detection
- Public Key, symmetric ciphers, and hash cryptography elements implemented in software

2   A Trusted Platform

Secure System's Foundation consists of the hardware platform and the critical code that executes on that platform. This foundation is built with an on-chip ROM-based boot-up process that initiates validation of the platform including the following tasks:

- examining key hardware elements to help ensure that they are functioning properly
- verifying the authenticity and integrity of the critical code that controls the overall operation of the system.

The boot process gains control of the system immediately after reset by executing known boot code that is resident in the on-chip ROM. The boot process, after verifying the authenticity of a start-up script residing in flash, follows that script using established cryptographic techniques to validate the authenticity and integrity of the operating system code and data in external memory. Because the script resides in external memory, it can be tailored to flexibly meet the customer's particular needs.

Further flexibility is achieved using electrically programmable fuses to enable or disable particular system functions. For example, it is possible to configure a production version, security-enabled device so that the JTAG debug port is completely disabled. For early prototype devices though, portions of the security system can be selectively disabled allowing access to otherwise inaccessible areas of the device. Full flexibility between these two extremes is possible.

Software that is security aware is imperative for those products that need security. Sensitive data in plaintext form must not appear on external data buses, and it should be restricted to the minimum number of data paths internal to the chip.

3   Cryptography

Cryptographic techniques are used to obscure sensitive data so it cannot be seen or used by unauthorized users. They are used to authenticate data, ensuring that it came from the expected source. These techniques

---

1. The Platform Independent Security Architecture (PISA) was developed within Freescale to address security concerns in a portable way. PISA is an integral part of many Freescale baseband and applications processors.

are used to decrypt streams of audio or video that have been encrypted to enforce Digital Rights Management (DRM) schemes. provide a brief overview of the main cryptographic elements implemented in the i.MX31 and i.MX31L processors.

3.1 Symmetric Key Cryptography

Symmetric Key Cryptography is the primary process used to obscure sensitive data. It works with a single key that ranges in size from 56 bits to 256 bits, depending on the selected algorithm. This key is used to encrypt and decrypt blocks of data. Data streams at up to 2 Mbps need to be processed in i.MX31 and i.MX31L processors. Crypto algorithms are implemented to meet the needs of various Virtual Private Network (VPN) and Digital Rights Management (DRM) schemes. These include: AES, DES, 3DES, RC4, and C2.

The Security Controller (SCC) hardware module uses a 3DES engine that can only use the chip-unique secret key for all encryption and decryption and is intended for encrypting sensitive data for off-chip non-volatile storage. The SCC crypto engine is restricted to supervisor mode code. It should never be accessible by user applications.

3.2 Public Key Cryptography

Public Key Cryptography (PKC) is a compute-intensive process that uses very large keys (1024 or 2048 bits). PKC uses two paired keys, a public key and a private key. Encryption uses the public key and decryption uses the private key. The public key is published and can be used by anyone. However, only the holder of the private key can decode the messages. In the reverse direction, information encrypted with the private key can be decrypted only with the public key. Since the public key is freely available, this doesn't provide any security. However, the source of the information is absolutely known. Information successfully decrypted with a public key can have originated from only the holder of the private key. This is the basis of authentication and digital signatures.

Applications that use PKC, such as DRM, use it at only the beginning of a session, typically to verify authenticity and exchange keys. As such, i.MX31 and i.MX31L processors do not include a hardware accelerator for PKC. Well known, efficient, software implementations of the ECC and RSA algorithms are available and execute with adequate speed to meet our customers' needs.

3.3 Hashing

Hashing is another fundamental cryptographic element. A hash is a calculation, similar to a CRC, over a block of data that results in a number, known as a Message Digest (MD). The SHA-1 hash engine is provided in i.MX31 and i.MX31L processors as part of the RTIC module and is being used in the boot authentication process.

3.4 Random Numbers

Most cryptography systems rely on strong random numbers for their robustness. Software-driven, pseudo-random number generators (PRNGs) appear random, but, in fact, are predictable, and thus, vulnerable to prediction attacks. To strengthen the security of crypto systems relying on strong random numbers, i.MX31 and i.MX31L processors each include a hardware Random Number Generator (RNGA). The random numbers produced by the RNGA are used as seeds to a PRNG, providing entropy and enhancing unpredictably.

4 High Assurance Boot (HAB)

The HAB ensures two basic attributes of the whole system: authenticity and integrity. This section describes details of the i.MX31 HAB implementation.

4.1 HAB Overview

The HAB is the software process that helps to ensure trusted execution of the operating system and application code. This is done by validating that the code image, stored in external memory, originated from a trusted authority (authenticity), and by verifying that the code is in its original form (integrity). The HAB can also be set to control the version of the loaded operating system (code revocation). The HAB uses digital signatures to validate the external code images, and thereby establishes the security level of the system.

The boot process is flexible because it is controlled by authenticated scripts that reside in external memory. Depending on the instructions in the control scripts, various levels of security can be easily established.

4.2 Signing and Verifying

As previously stated, the HAB uses digital signatures to verify the authenticity of the code and data resident in off-chip memory. The signatures are created using hashing and public key cryptography.

Digital signatures are simply SHA-1 hash values that are encrypted with a RSA private key crypto systems. Figure 1 shows the signing and verifying procedure. A hash of the original code block is calculated and encrypted with the private key creating a signature. The signature is then bundled with the code and prepared for storage in flash. This procedure is used for each protected block of code or data.
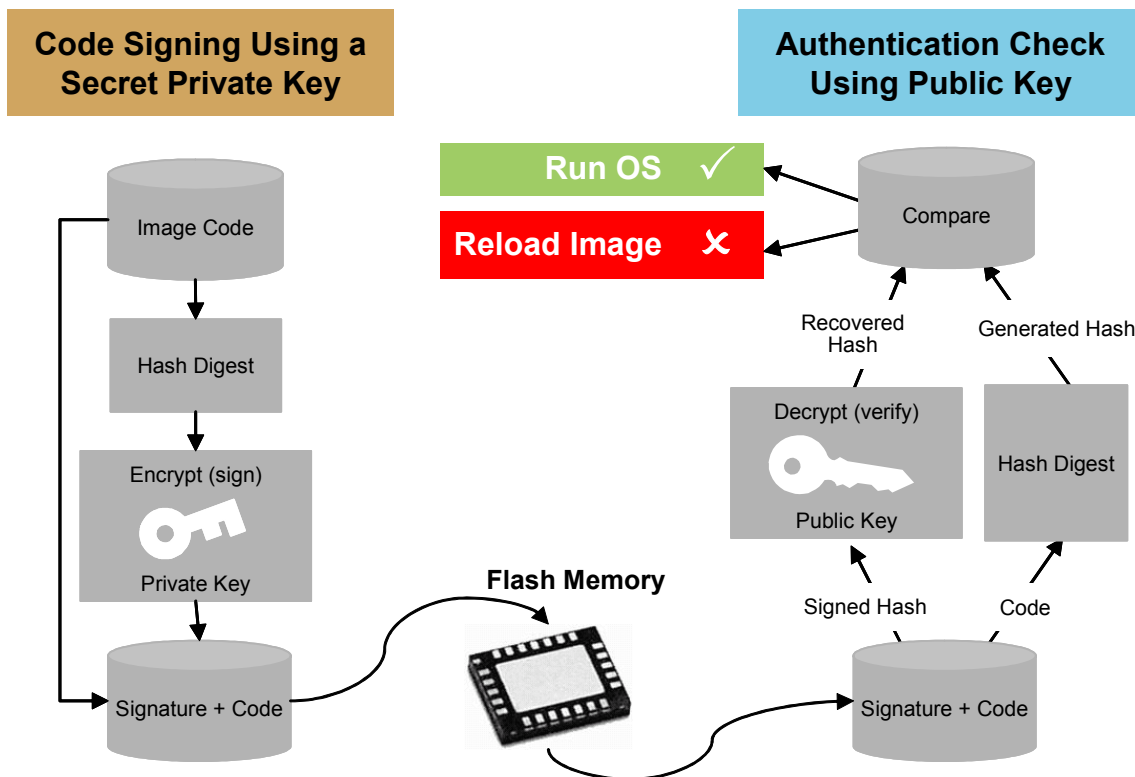
Figure 1. Signing and Verifying

Verification by the HAB uses the public key, also known as the Super Root Key (SRK), which is stored in on-chip non-volatile memory. To enhance the robustness of the HAB security, multiple Super Root keys (RSA public keys) are stored in internal ROM, and the applicable one for a given manufacturer is selected using electrically blowable fuses. Once the fuses are blown, they are protected from further modification. The i.MX31 and i.MX31L processors each support both 1024 bit SRK and 2048 bit SRK for enhanced security products.

The signature, extracted from the bundled code block and the SRK, is used to decrypt it into the original hash value. A new hash of the code block is re-calculated, then compared with the decrypted hash value. If the two hash values are equal, the code is verified to be correct, unaltered, and authentic. If the two hash values don't match, it is certain that the code has been modified, either intentionally or unintentionally, and the HAB process takes appropriate measures.

5 Hardware Security Elements

This section describes the security platform's hardware elements. Figure 2 shows the i.MX31 and i.MX31L processors' security subsystem high level block diagram.

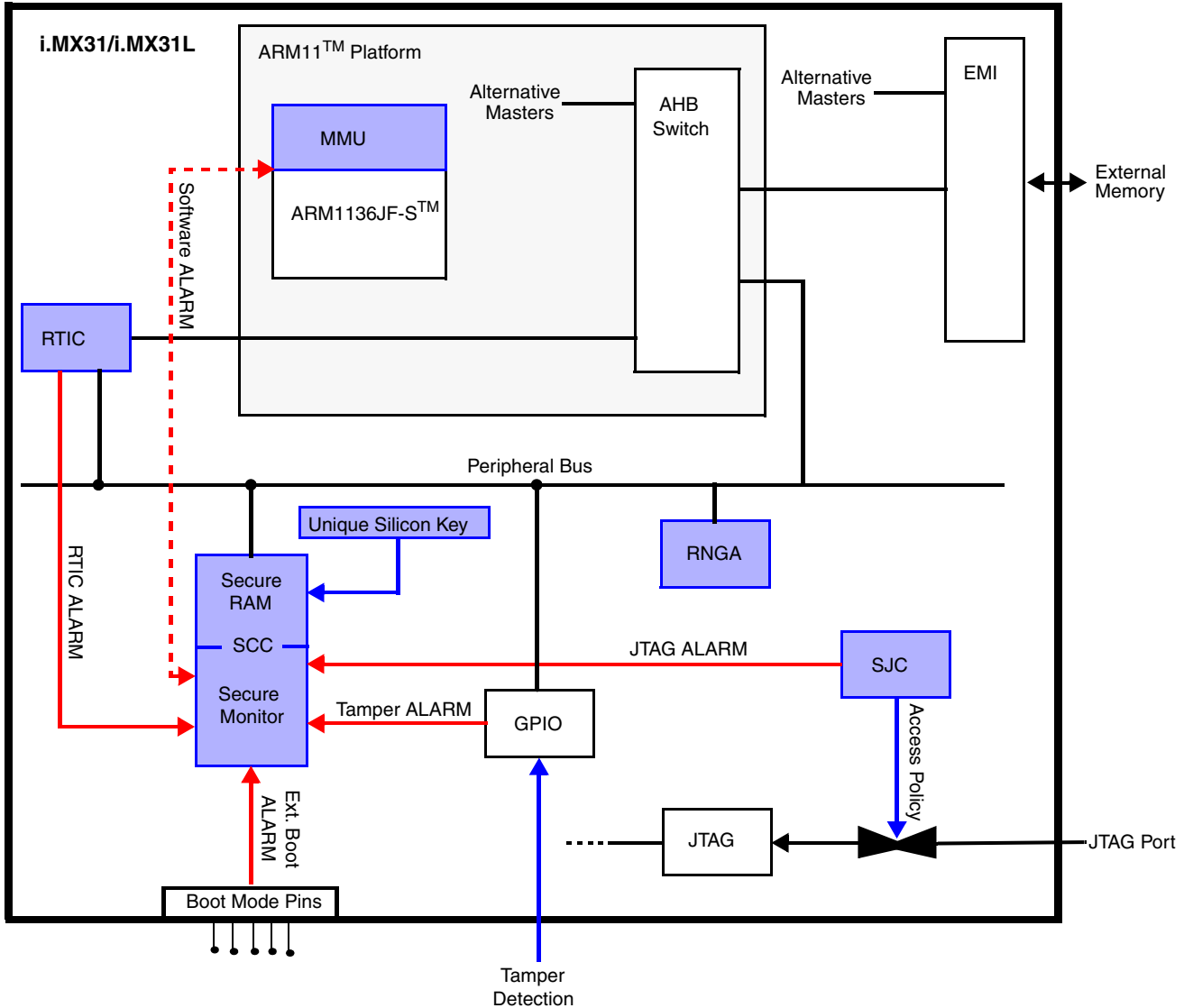**Figure 2. Security Subsystem High-level Block Diagram**

5.1   Security Controller (SCC)

The Security Controller (SCC) helps to provide a secure memory system, along with supervisory hardware to allow access to secure memory only under certain conditions. This allows sensitive information to be encrypted and stored in internal RAM or external flash for future use.The SCC uses a 168-bit chip-unique secret key for its 3DES crypto engine. Because the key is secret, data that is encrypted with one chip cannot be decrypted by any other chip.

The SCC is comprised of the Secure RAM and the Security Monitor.

5.1.1   Secure RAM

The SCC's internal Red RAM has two functions—a secure storage area for actively working keys or other sensitive data, and a marshalling area for encrypting sensitive data. Access to the Red RAM is possible only in supervisor mode. If a security fault, such as a JTAG debug session or tamper detection, is detected by the Security Monitor, the Red RAM is zeroed to protect its contents from unauthorized viewing. Backup copies of the Red RAM's contents are stored in encrypted form in external memory.

The SCC contains hardware that performs several checks on the secret key to ensure its strength.

5.1.2   Security Monitor

The Security Monitor ensures that the system is running in such a manner as to provide protection for the sensitive data that resides in the system. Specifically, it determines when and how the Secure RAM resources can be used, as well as providing mechanisms for verifying software algorithm integrity.

For the Secure RAM, it ensures that the secret on-chip key can be used only in secure state, and that the secure state can be reached only by booting internally with the high assurance ROM code. Any of the following conditions is considered a security violation, and will put the Security Controller module in non-secure state or failure state:

- External boot
- Any JTAG access
- Activation of any test bus mode, Embedded Trace Macrocell^{TM} (ETM^{TM}) interface, "show cycles," watchpoint, or activation of any other debug interface
- Scan mode
- Tamper detection
- RTIC error
- Software alarm

5.2   Run-Time Integrity Checker (RTIC)

The Run-Time Integrity Checker (RTIC) is part of the PISA family of platform security components. Its purpose is to ensure the integrity of the peripheral memory contents and assist with boot authentication. The RTIC has two basic operation tasks:

- Verify the memory contents during system boot
- Check memory integrity during application run-time execution. If the memory contents at run-time fail to match the original hash signature, an error in the security monitor is triggered.

The RTIC offers the following features:

- SHA-1 message authentication
- Input DMA (AMBA^{TM}-AHB Lite bus master) interface
- Segmented data gathering to support non-contiguous data blocks in memory (up to 2 segments per block)
- Works with High Assurance Boot process

- Secure scan DFT security
- Support for up to 4 independent memory blocks
- Programmable DMA bus duty cycle timer and watchdog timer
- Power saving clock gating logic
- Full word memory reads (word aligned addresses, multiple of 32-bit lengths)

5.3 Secure JTAG Controller (SJC)

JTAG manipulation is one of the known hackers' ways of executing unauthorized program code, getting control over the OS and run code in privileged modes. The platform's JTAG port provides a debug access to several hardware blocks including the ARM® core, and the system bus. This allows program control and manipulation as well as visibility into system peripherals and memory. The ETM interfaces allow bus transactions to be traced. Together, these tools provide the hacker all the access needed to completely compromise the system.

The JTAG debug port must be accessible during platform initial laboratory bring-up, manufacturing, test, and troubleshooting, as well as for software debugging by authorized partners. However, all other access to the JTAG port should be strictly limited to properly secure the system. The i.MX31 and i.MX31L processors each incorporate a flexible mechanism to secure the JTAG port while still allowing full use of the JTAG-based debugging features by authorized parties.

The SJC allows four different JTAG security modes (which represents four security levels):
- Mode 1: No Debug—Maximum Security. All security sensitive JTAG features are permanently blocked.
- Mode 2: Secure JTAG—High security. JTAG use is regulated by secret key-based authentication mechanism.
- Mode 3: JTAG Enabled—Low security. JTAG always enabled.
- Mode 4: SCC JTAG—No Security. The SCC is forced to remain in its secure state during JTAG operation. For secure operation debug, operating in this mode should be done only under tight control.

The JTAG security modes are configured using electrical fuses, which are part of the IC Identification Module (IIM). The fuse burning is an irreversible process. Once the fuse is burned, it is impossible to change the fuse back to its original state. Burning of the IIM JTAG security mode fuses can only increase the level of JTAG security.

5.4 Unique Identification (UID)

As future devices enter into new territory involving such uses as audio playback, e-book, e-cash, and others, it becomes increasingly important to provide a means for customers and media vendors to be assured that their IP is protected. The UID provides the first step in enabling this protection. The UID contains 64 hardware unique ID bits that are implemented using electrical fuses with the sense circuit to allow unique identification code for every IC. The purpose of the UID is to support Digital Rights Management (DRM). The unique ID number cannot be altered without the destruction of the IC in order

to prevent malicious users from mimicking the unique ID and gaining access to materials protected by the DRM.

5.5 Random Number Generator Accelerator

The Random Number Generator Accelerator (RNGA) module is a digital integrated circuit capable of generating 32-bit random numbers. It is designed to comply with FIPS[1]-140 standards for randomness and non-determinism. The random bits are generated by clocking shift registers with clocks derived from ring oscillators. The configuration of the shift registers ensures statistically good data, meaning data that looks random. The oscillators with their unknown frequencies provide the required entropy needed to create random data.

5.6 Tamper Detection

The tamper detection mechanism's purpose is to provide evidence of any physical attempt to remove the device cover. The i.MX31 and i.MX31L processors each give the OEM vendor the option of implementing a tamper detection system. This system is composed of a hardware tamper detector (i.MX31 and i.MX31L external hardware) that can be connected to an i.MX31 or i.MX31L GPIO pin. The GPIO pin is able to wake the ARM[TM] processor and generate an interrupt. The interrupt executes a software routine that can give the user an indication that the device was tampered with. Tamper detection can also generate a Secure Monitor alarm that causes the SCC Red Memory to be zeroed. Attempts to shutdown the device will not disable the tamper detection system, as the tamper IRQ line can wake up the ARM processor. The tamper detection GPIO pin is also connected to the SCC security monitor. When a tamper attempt has been detected, the GPIO tamper detection pin signals the SCC, which zeros the Secure RAM and transits into failure mode.

The tamper control is designed in such way that once software has enabled the tamper-detect GPIO pin, it stays enabled until the reset, meaning the software cannot turn off the tamper-detect function, once it is enabled)

5.7 IC Identification Module (IIM)

This module contains several one-time programmable (OTP) elements that are used to enable or disable various chip features. The IIM also is a repository for the 168-bit secret key and the 64-bit unique chip ID value.

5.8 On-Chip RAM

An on-chip RAM of 16 Kbytes for buffer and working memory that can be used by the various crypto algorithms is implemented.

---

1. FIPS - Federal Information Processing Standard

5.9 Task Separation

Task separation is critical in a secure system; this prevents untrusted tasks from interfering with any other task. The ARM1136JF-S^{TM} provides a Memory Management Unit (MMU), which enforces such restrictions.

6   Cryptographic Algorithms (Software)

Support in the i.MX31 and i.MX31L processors for the various encryption and hashing algorithms is largely provided in software. Software performances, available through the ARM1136^{TM} processor, alleviates the need for dedicated hardware accelerators in most cases. Table 1 shows typical cryptographic algorithms and i.MX31 and i.MX31L implementation performance.

NOTE

Numbers shown are preliminary, and subject to change as more reliable data is generated

Table 1. Security Use Cases

Use Case	Algorithm	System Performance
Digest generation for digital signatures, (Hashing of external flash as part of the secure boot process—done by the RTIC hardware acceleration)	SHA-1	RTIC 115.5 cycles/byte - ARM1136
General purpose symmetric block cipher, used in many protocols.	3DES	430 cycles/byte - ARM1136 (16 MCPS for 152Kbps + 152 Kbps encrypted video conference)
General purpose stream cipher, used in Microsoft Windows Media DRM.	RC4	34 cycles/byte (8.7 MCPS for decrypting 2 Mbps streaming video over RC4-encrypted based DRM)
General purpose symmetric block cipher, specified as the replacement to DES (and 3DES in many cases).	AES	117 cycles/byte (30 MCPS for 2 Mbps streaming video over AES-encrypted based DRM)

Table 2 shows a matrix of use case protocols and the crypto algorithms that are used by the protocols.

Table 2. Crypto Use Cases

Functionality:	VPN	Transaction Security			DRM		
Algorithm:	IPSec	SSL	TLS	WTLS	OMA 2.0	Windows Media	CPRM
DES	X	X	X	X		X	
3DES	X	X	X	X			
AES					X		
RC4						X	

Table 2. Crypto Use Cases (continued)

Functionality:	VPN	Transaction Security			DRM		
Algorithm:	IPSec	SSL	TLS	WTLS	OMA 2.0	Windows Media	CPRM
C2							X
RSA	X	X	X	X	X		
ECC				X	X	X	
SHA-1		X	X	X	X	X	
MD5	X	X	X	X			
HMAC			X	X	X		

Document Number: IMX31SECURITYWP
Rev. 0.0
06/2005