Security Enablers on Jacinto™ 7 Processors

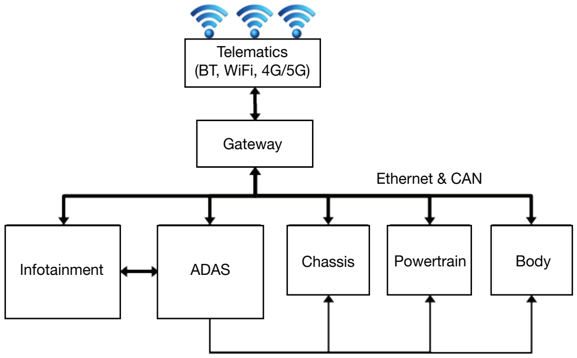
Steve Reis
Systems Applications & Architecture
Jacinto processors

Ever more powerful embedded processors and System-on-Chip (SoC) solutions enable designers to create more capable and powerful systems. Connectivity, both wired and wireless, is now a requirement for most embedded systems in order to interface with remote control and management functions and to enable integration into more complex and capable systems in factories, automobiles or homes.

In addition, the ability to support remote updates to add features and correct errors has become a standard requirement as well. These features, in turn, require greater security to prevent these systems from being co-opted, misused or even from becoming unsafe.

Figure 1 shows an automotive system with chassis, powertrain and body systems, along with an infotainment system and advanced driver assistance system (ADAS), all networked via a network gateway that allows data sharing between the electronic control units. A typical automotive embedded system enables the ADAS to control some vehicle movement, such as self-parking, lane-keeping assist and other automated driving features. A telematics gateway enables vehicle access to the cloud for software updates and other data.

The external interfaces, especially those that are wireless, are vulnerable to remote access. This along with the increasingly networked nature of these systems means that any security breach can have wide-ranging implications and makes it imperative to provide a high degree of protection.



Example interconnect vehicle architecture with wirelelss connectivity

***Figure 1**. Interconnected automotive architecture.*

In this white paper we will discuss the Jacinto 7 processor device family, which includes TDA4x and DRA8x processors, and provide an overview of the security features available in TI's Jacinto 7 family of SoCs that can help system designers achieve your security objectives. We refer to these as **security enablers**. You can learn more about security enablers at TI.com/security.

Security Framework

Starting at the application level, implementing security measures helps protect assets against threats. At the semiconductor level, the main assets in a system that need protection are data, code, device identities and keys. The exposure points (often referred to as the attack surface) in a system can increase the vulnerability of assets at each part of the application and system life cycle and operations.

Based on the assets that need protection and the exposure points, you should consider all of the appropriate security enablers and select security features at the device level to design appropriate protection. **Figure 2** shows an example security framework.

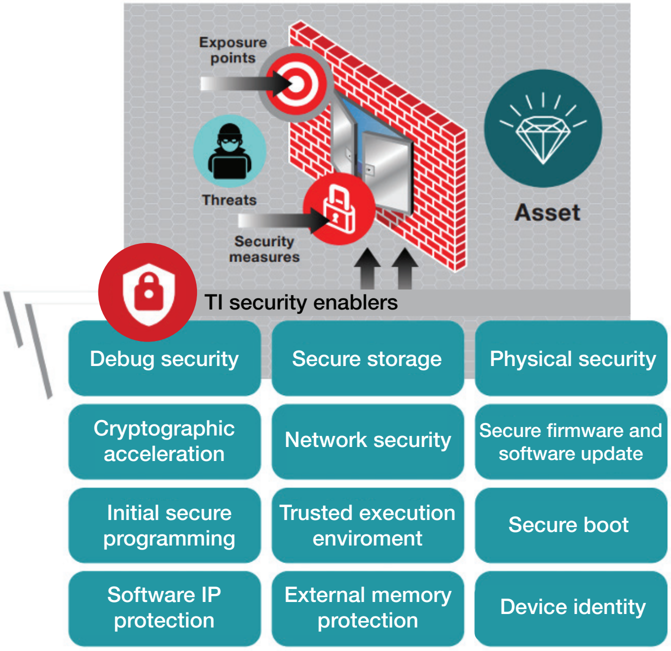


***Figure 2**. Security framework.*

The Jacinto 7 SoC family supports many security enablers to help users implement strong security measures, tailored to their system, to counter potential threats that will limit or prevent access through the system's exposure points, including:

- Device identity (unique ID).
- Secure boot (root-of-trust public key).
- Initial secure programming.
- Cryptographic acceleration.
- External memory protection (firewalls).
- Debugging security (Joint Test Action Group [JTAG] lock with password).
- Software intellectual property (IP) protection (debugging lockout).

TI foundational security processor and firmware

At the core of the Jacinto 7 SoC's security enablers is a dedicated Arm® Cortex®-M processor and secure random access memory hosting the firmware that provides the foundational security functions. These features include secure boot and secure functions, secure eFuse key management, device firewall management, JTAG access authorization and firmware rollback protection. Additional features may also be available depending on device variant.

Device identity, keys and secure boot

An important component of the Jacinto 7 platform's security enablers is the support for secure boot and secure root-of-trust keys. Together, these secure the boot process and prevent the loading and execution of untrusted software.

This security anchor is built around a secure root of trust or set of keys embedded in the Jacinto 7 SoC. These keys consist of an asymmetric public and private key pair, a shared secret key and a device-unique secret key. The public key is fused into one-time-programmable eFuse memory during the hardware manufacturing flow. This public key is used to authenticate the initial software image that starts the system, along with the initial device security configuration components, by validating the digital certificates and signatures embedded in the software. This process can be extended to include establishing trust in additional keys as well as extending the chain of trust by authenticating additional software components – such as additional bootloaders and operating system kernels for the multiple cores on the Jacinto 7 SoC.

The system manufacturer maintains the root keys in a secure computing environment to ensure system integrity and limit access to authorized users, who can only indirectly access them to sign and encrypt software for the Jacinto

7 processor cores for their systems. The software is authenticated through the standard X.509 certificate format, which requires no custom certificate generation or signing tools and can thus be created with common tools. This in turn enables a straightforward implementation in the user's secure computing environment and maintains the security of their private keys.

The Jacinto 7 SoC's secure boot feature ensures that the software running on the device is always authenticated and thus prevents unauthorized software from being loaded at the critical initial boot phase. The initial secure boot process is implemented with a secure boot read-only memory that enforces the authentication of the software components against the device root of trust. Boot authentication options can support either Rivest-Shamir-Adleman (RSA) (up to 4,096 bit keys) or Elliptic Curve Digital Signature Authentication (ECDSA) elliptic curve cryptography (ECC) up to 521 bit keys, combined with strong SHA2-512 hashes for the software and certificate signature. There is also support for optional AES-256 encryption of the bootloader.

Initial secure programming

Device key provisioning is a process that must occur securely when programming secret keys. Device key provisioning processes are fully controlled by the system manufacturer in their own factory for security, simplicity and full flexibility in key programming, using TI-provided secure provisioning tools. Encryption protection prevents the symmetric secret key from being exposed during the provisioning process and allows key provisioning and manufacturing in an untrusted factory environment.

Cryptographic acceleration

Cryptographic functions can be computed on general-purpose computing cores or specialized hardware accelerators, depending on flexibility and throughput requirements. Jacinto 7 SoCs include a set of cores that accelerate common cryptographic functions and include support for:

- Asymmetric cryptography: RSA and ECC functions.
- Hashing: Message Digest Algorithm (MD5), SHA1 and SHA2-224/256/384/512.
- Symmetric cryptography functions: AES-128/192/256.
- Hardware TRNG module with deterministic random bit generator (DRBG) post-processing.

In addition, Arm Cortex-A CPUs support ARMv8 cryptography extensions, which add new instructions that accelerate the execution of AES, SHA1 and SHA2 algorithms.

Software IP protection (firewalls)

The Jacinto 7 SoC contains a set of heterogeneous processor cores optimized for various tasks, including 64-bit Arm cores and 32-bit Arm microcontroller cores, along with TI digital signal processors (DSPs) and specialized DSP accelerators on some devices. Some of these components may be allocated to tasks that work with secure assets and thus require protection and isolation from other general-purpose functions. Jacinto 7 includes a comprehensive set of system firewalls for runtime security protection as well as isolation for safety. Firewalls allow users to define the hardware elements and memory ranges that each processor core or system initiator is allowed to access. This firewall infrastructure is a key enabler to prevent exposure of secrets, maintain freedom from interference and limit the impact of any possible intrusion.

Debugging security

The ubiquitous JTAG debugging port found on most programmable devices provides many accessibility features such as easy access to device registers and memory, easy initial flashing methods and program tracing. This accessibility also means that it may be the most vulnerable point of exposure in a system. As a result, the Jacinto 7 SoC's JTAG debugging port is disabled by default for secure devices, so it cannot be used to gain access to the SoC operation. At the same time, the Jacinto 7 device JTAG can be enabled – in a secure manner – for debugging and analysis if necessary. Enabling JTAG access requires authorization or authentication through a certificate mechanism tied to the root of trust. Furthermore, each debugging certificate is tied to one device and can only enable debugging for the selected device ID contained in the certificate. Finally, JTAG access can also be permanently disabled through one-time programmable eFuse programming, if system security protocols require. These features provide layers of protection and access and enable users, with flexibility for access during development and security when in production.

Trusted execution environment

The Jacinto 7 SoC's Arm Cortex-A72 TrustZone® feature provides isolation for the execution of secure software components and can protect important assets such as keys, data and specialized algorithms. To simplify use of this secure environment, a trusted execution environment (TEE) provides a secure software environment for isolated security applications. The Jacinto 7 device's Linux® software development kit enables integration of the Linaro OP-TEE secure stack, which in turn enables security applications using standard GlobalPlatform application programming interfaces for developing secure applications for the Arm platform. Another benefit of the TEE is that secure applications are isolated from each other as well as the rest of the Linux stack. Thus, multiple clients can safely use the TEE without exposing assets between them.

Secure firmware and software updates

The need for secure firmware update capability, especially OTA updates, is proliferating in embedded systems to enable in-field updates of new and enhanced features, bug fixes and security patches quickly without the time and expense of service technicians or factory service. However, this update process can also be a vulnerability point if others can impersonate, roll back to a previous version, or use the update mechanism to install compromised software images.

Update images should always be hashed and signed in order to verify both their integrity and authenticity. Authenticity checks verify that the update is from a known and trusted source, and integrity checks verify that the images have not been altered or tampered with during transmission and loading. The same Jacinto 7 SoC features used for secure boot authentication can also authenticate software and data updates

Conclusion

The security enablers available in the Jacinto 7 device family comprise a comprehensive set of embedded security features that can enable designers and architects to achieve the security goals needed for the system. These are typically assessed as part of a security implementation cycle: the identification of the specific security goals, risks and measures for each project and the security enablers that can help achieve those security goals. For more information, see ti.com/security.

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (https:www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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