Industry 4.0 and the IoT: paths to secure data communication

By Konrad Zöpf, TQ-Group

Against the background of Industry 4.0 and the IoT, the need for secure communications and the associated networking of systems and components is increasing. New requirements on hardware and software performance are the result, as described in this article.

(image source: y123rf.com/Prasit Rodphan)

■ IoT (the Internet of Things) is the hot topic today and requires a secure system. However, this should not be confused with functional reliability, which is used to describe the reliability and resistance to a system outage. Rather it refers to vulnerability to the outside world. Usually, external attacks are intended to cause damage or gain an economic advantage by tampering.

Until now, secure systems could only be developed with a lot of individualization effort. For more than three years now, research and the business world have been occupied with the topics of Industry 4.0 and IoT. Basically, these topics depend on the necessity of having the increasing volumes of data available everywhere at all times. The whole subject describes a very complex topic, starting with small controllers that accept data from actuators and sensors and transfer these data to the cloud. One clear trend can be seen in different market segments. More and more clients are asking for solution modules that cover the stricter requirements in the security and safety area.

Until now, embedded systems were usually stand-alone solutions and possessed only limited connection options, if they had any at all. In this way, it was relatively easy to ensure the security of a system because access was possible only on a very limited basis. With newer systems, easily accessible communication options are required. Networked production systems must have suitable countermeasures to protect against external access (from the Internet, say) and against internal attacks, too. It must be ensured that the sender is the person or device that it purports to be (authentication). It must be verifiable later that only this sender transferred the data or message (non-repudiation). The data were not modified over the transmission path (validation). The data are encrypted and cannot be read or interpreted by others (secrecy). The hardware of the sender or receiver cannot be manipulated (protection).

All these countermeasures require high-performance hardware and secure software. Another aspect critical for security is that only a safe (reliable) system secured against outages (safety) is able to resist external attacks. In addition, hardware accelerators integrated into the CPU support the software and reduce the load on the CPU. This has a direct effect on the performance of the system and on the power consumption.

Versatile countermeasures are needed to implement a secure system. The most important of these are described in the following. These include High Assurance Boot (also called a Secure Boot), trusted execution, hardware accelerator for cryptography, secure debug controller, protected memory access (encryption) and hardware security measures (tamper detection, runtime integrity checker – RTIC).

A secure system boot must ensure that only authenticated program code may be run on the CPU. Usually, this software is the boot loader. Secure Boot, also known as a High Assurance Boot, is an important element in security measures and prevents the CPU from executing untrusted or unauthorized

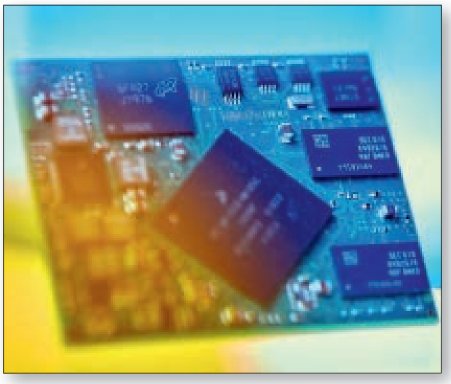

Figure 1. The TQMa7x mini module, based on the i.MX7 processor from NXP, combines the ARM Dual Cortex-A7 core technology with a variety of interfaces.

code. It must also be ensured that unauthorized modifications to the CPU configuration by unauthorized program modifications are detected and prevented. This makes it impossible to read sensitive data from the chip without authorization. The operating process for a secure boot usually contains the following steps: signing of the image, encryption of the image and use of a secure boot sequence.

The term TrustZone from ARM is used for a concept that separates a non-secure or untrusted environment in the hardware from a secure, trusted environment by blocking non-secure software from accessing secure resources. Within this process, the software is in either a secure or a non-secure environment. Software switches between these two environments, known as the secure monitor (Cortex-A) or as the core logic (Cortex-M).

Hardware Cryptographic Accelerators (CAAM)	■ Symmetric key authentication: ■ AES-128/256, with DPA protection ■ DES, 3DES ■ ARC4 Asymmetric Authentication (public key) ■ RSA (up to 4096) ■ Elliptical curve „ECDSA“ (up to P-521)
Cryptograhic hash function	MD5, SHA-1, SHA-224, SHA-256
Tamper detection & protection	■ DryICE (On-chip voltage, temperature, frequency monitoring) ■ Dedicated tamper pins, 10 pins total (can be configured to be 10 passive or up to 5 active pairs) Tamper logging
DRAM encryption	On-the-fly DRAM data encryption/decryption with AES-128
Hardened readback disable	Yes, lock bit can disable the access of the key
DPA resistant	DPA protection for AES
Obfuscated key storage protection	On-chip zeroable 8x4 kB secure RAM Off-chip key/data blobs AES-256 master key (CAAM/SNVS)
Permanent JTAG disable	Yes – Secure JTAG controller (with electrical fuses)
Internal key clearance	On-chip zeroable 8x4 kB secure RAM (32 kB)
Unique ID (Device DNA)	Yes, as the OTPMK is secret and this is unique per part. The OTPMK cannot be directly read but can be used to encrypt a constant to create a unique number for use as a unique ID. Each chip has a 64-bit unique ID in the OTP fuse.
Unique ID (user eFuse)	Yes – General purpose OTP fuse for customer use
Secure storage	■ Zeroable master key (256 bit) ■ General purpose 32-bit register ■ Secure High Assurance Boot ■ Up to 2 kbit eFuse
Permanent decryptor disable	Yes (export disable fuse – disable all crypto except hash engine and RNG)
Secure RAM with battery backup	Yes – 256-bit master key storage with secure RTC (real-time clock) power (SNVS)
Additional security features	■ Run-time integrity checker and security controller ■ Random number generator (NIST SP 800-90) ■ ARM TrustZone ■ 2x EMV-compatible SIM V2 & EMVSIM module

Table 1. Safety functions supported in the i.MX6UL CPU from NXP as an example (source: NXP)

The two CPU architectures offer similar security concepts but with completely different implementations. The concept of the secure and non-secure environment extends beyond the processor to include the memory, software, interfaces, bus systems, interrupts and peripherals within an embedded module.

In the basic configuration, the TrustZone technology offers system-wide security and the potential of creating a trusted platform. The system can be designed such that every part is also a part of the secure environment including the debugger, peripherals, interrupts and memory. Parts to be protected against software attacks and prevalent hardware attacks can be defended by using a security system such as TrustZone from ARM.

NXP also supports the following range of functions with internal IP on i.MX product. The concept includes the isolated execution of code in a secure CPU mode. There are also hardware firewalls between the CPU or the master DMA controllers and the peripherals and memory. The concept is comparable to two virtual processor cores where one is considered non-secure and may only access non-secure resources while the other, secure core may access all system resources.

For secure data communications, hardware-supported security acceleration may also be used together with a purely software-based solution. The CPU can, of course, perform security countermeasures and run the algorithms for security operations by itself, including the encryption algorithms and checksum calculations. This is supported by the fact that the programmer alone can design the algorithm. Programmers by themselves can decide how to design the algorithms and can incorporate the newest security research results into the software on a near-real-time basis, a capability that provides the greatest possible flexibility. To design secure data communications on a CPU-based software solution, a lot of computing power is needed.

This is countered by the facts that considerable effort is needed to create the programs, the programs are prone to errors and the CPU is burdened with tasks that are perhaps needed for the actual application. In addition, a general-purpose CPU is not optimized for algorithm calculations. In terms of program execution time, power consumption and throughput per watt, a general-purpose CPU is not really efficient. For this reason, more and more CPU fabricators are implementing hardware accelerators in many microcontrollers and processors. These accelerators are modules that perform the security tasks, or some of them, in hardware, reducing the load on the CPU.

The microcontrollers of the Cortex-M class mostly use the mm-CAU (memory-mapped cryptographic acceleration unit). This is a coprocessor that, with the help of specialized operations, improves the throughput of software-based encryption algorithms and cryptologic hashing functions. Software libraries usually come from the controller manufacturers. Depending on the controller manufacturer, flash memory protection is also offered. By activating various security functions, access to the memory contents by way of a JTAG interface (debug controller) is prevented by means of a stored key.

In terms of CPUs, very different security functions in the form of a hardware module are used as described depending on the CPU fabricator and the time at which the CPU was put on the market. In this case, the hardware manufacturers are providing innovative functions to meet market requirements. Security concepts for the memory are usually permanently integrated into the TrustZone with the CPUs. The RTIC (runtime integrity checker) allows verifying the memory contents during the system boot and during runtime. If a deviation or tampering is found, an interrupt is sent to the security hardware monitor.

Another function is DRAM encryption. This prevents manipulation of or attacks on the system memory. Tamper detection can be used to make manipulation attempts and physical attacks on the device or even on the pin of the CPU more difficult, to limit or prevent them. Table 1 shows the functions supported in the i.MX6UL CPU from NXP as an example. These functions can be applied by the user to implement a secure system.

Besides the hardware, the operating system to be used is a very important factor that must be considered when evaluating a CPU and ultimately an embedded module intended for use as the foundation of an IoT/Industry 4.0 system. Particular attention must be paid to make sure that, for the selected operating system, there is already support for the hardware accelerators, if necessary. Selecting the correct software for the upcoming project should not just consider the device requirements. An overall concept describes how all needed functions can be brought into line with the necessary requirements. Considering the tasks to be performed, how and in what areas a system must be made secure, the appropriate OS offers different drivers suitable for various market segments such as medicine, industry, railroad, automotive, up to and including military applications.

When comparing earlier applications to those of today, the CPUs provide more and more interfaces. Starting with UART, USB, I²C, SPI, field buses and Ethernet, more and more wireless interfaces such as Wi-Fi, NFC, Bluetooth or the cellular telephone network are being used. Wired interfaces, with the exception of Ethernet, are used only within a limited environment. It is easier to attack radio-based solutions and Ethernet from outside if they are part of a public network. Consequently, when using radio-based solutions, particular attention must be paid to ensure that the vendor can provide an appropriate security concept including an update capability.

An operating system should consider as many attack vectors as possible to satisfy the security requirements of IoT applications. The term attack vector indicates various vulnerabilities such as access rights of data and users and encryption mechanisms that attackers may undermine. Commercial operating systems use their own security models here, usually specific extensions and functions that can be considered secure. Another advantage to this is that a true microkernel is used. This offers fewer vulnerabilities compared to monolithic operating systems.

For example, if new security gaps or bugs are discovered in Linux, everyone is responsible on their own to maintain their system. This can quickly result in considerable expense. For system development using Linux, it is recommended that the user revert to the security features integrated into the CPU to be able to withstand the different types of attack scenarios (side-channel attacks, differential power analysis, crypto analysis, physical attacks).

Clients, even those dependent on comprehensive security concepts, will be supported in the future with all embedded modules from TQ-Systems on the basis of the security functions available in the CPUs. Chiefly those clients looking to use cost-effective solution modules consisting of the module and security-related software benefit from the many years of partnership with various CPU manufacturers. Even in the case of upcoming certifications, TQ-Systems is available as a competent development partner and supports the client.

Solution modules consisting of embedded modules with various CORE architectures and specially matched security concepts extending from accepting data to transferring data to the cloud allow starting a development effort faster and faster without complications for systems that ensure networking takes secure data transmission into consideration. Comprehensive hardware and software support also promises both a quick entry and the efficient and cost-effective implementation of security-related project requirements. ■