Requirements for IoT and IIoT in the changing connected world

By Oliver Winzenried, WIBU-Systems

This article describes the requirements and solutions to ensure safety and cyber security. It covers aspects including secure boot, firmware updates, licensing and know-how protection as well as new business opportunities for device manufacturers.

Figure 1. CodeMeter technology allows easy integration in products and processes.

Manufacturers of IoT devices need to care about three things: security, security, and security. Consider this: security is required to operate and use IoT devices in the way they are intended, security is required to update and upgrade functionality and features in IoT devices while ensuring that they are not tampered with or hacked, and security is required to monetize features in IoT devices and implement new business models with benefits for both device manufacturers and users. The potential threats are increasing as more and more systems are becoming connected with each other making the integration of security mechanisms a must. A look at the evolution of devices nowadays into IoT devices reveals the benefits.

Table 1 shows the differences. Today, embedded devices are often stand-alone devices with set functionality. Device makers make a one-time sale of the device, and repair or maintenance revenue is optional. In many cases, an individual piece of hardware is used with no operating system or with an easy scheduling OS. This situation is changing: devices are becoming connected, and are being equipped with upgradable features, not unlike the popular app stores for smartphone applications. Using standard hardware and software platforms, development efforts can be reduced and the time-to-market shortened, which promises benefits for the device maker. Furthermore, selling devices at an early stage and with a basic feature set allows additional feature sales at a later point. With these new business models, recurring revenues can be realized. To enable these new benefits for users and makers, the three security challenges have to be solved.

More and more features are software-realized, using the same hardware and software platform. Increasing connectivity then needs a secure identity. The major threats are the following. Counterfeiting by copying software or rebuilding it with standard hardware and OS platforms. Reverse engineering to uncover the actual value-adding algorithms and implementing them without major development efforts. Undermining new business models by enabling all embedded software functions without buying the required licenses. Manipulating devices with faked firmware updates or manipulating existing firmware or configuration parameters, as well as manipulating complete systems with fake identities. The solutions for these challenges rely on the following methods.

Know-how protection: valuable intellectual property in application data and program code is protected via encryption. For performance reasons, symmetric encryption algorithms are used, such as the well-known AES. The required data is decrypted in the memory at runtime, while always staying encrypted on the disk or flash storage. A static analysis is made impossible without access to the plaintext data.

Product protection: creating counterfeit products by copying data or program code to a look-alike system will not work, as the encrypted data and program code cannot be decrypted on counterfeit products. Therefore cryptographic keys need to be stored with protection against cloning.

Flexible licensing: each software-realized feature is assigned a unique license, using different encryption keys. In addition to the encryption key itself, the maker can define how these keys and licenses and the features can be used. The new licensing options include pay-per-use models, time-based licensing, rental and subscription models, and many more. To use these licensing mechanisms and transfer the licenses to a device, license creation, administration, and deployment needs to be integrated in the vendor business process. This can rely on a form of appstore or user license portal with which the user can activate or return licenses himself and is billed accordingly.

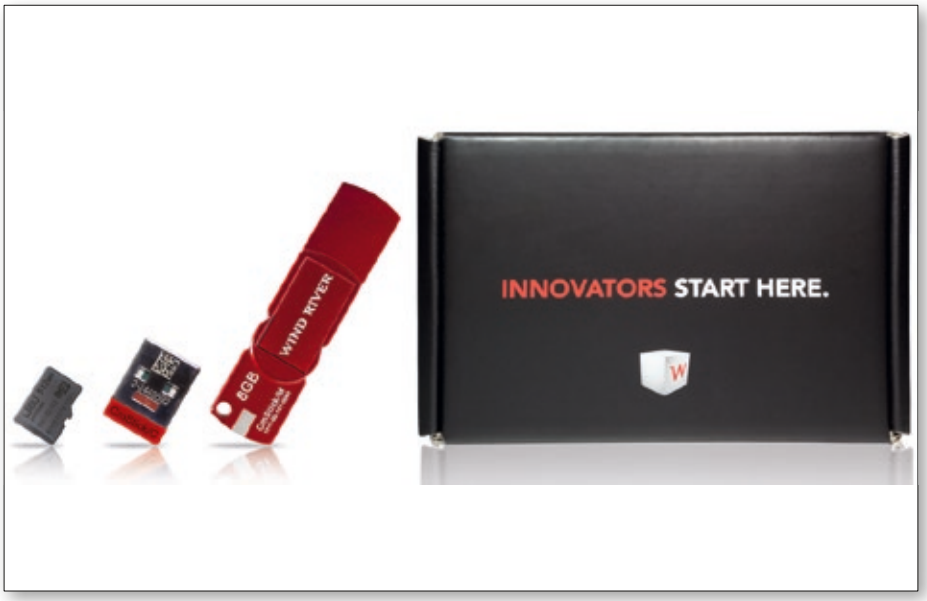

Figure 2. Examples of Wind River Embedded Development Kit including CodeMeter

Today	Tomorrow
Stand-Alone Device	Connected Devices
Fixed Features	Upgradeable Features (App Store)
One-Time Business Products, Upgrades, Service, Spare Parts / Consumables, Replacements	Recurring Revenues Pay-per-Use, Pre Paid, Post Paid
Only Repair / Maintenance	New Business Models Shorter Time-to-Market, Cloud
Individual Hardware	Standard Platforms (HW & SW)

Table 1. Differences between current and future devices.

Tamper protection: in order to avoid manipulation or cyber-attacks, the application data and program code is digitally signed. An anchor of trust is the root public key, which must be securely stored on the device. Digital signatures are created using asymmetric cryptographic means. First, the data or program code is hashed (a hash function is a function that maps data of any length to a character string of fixed length. It is also collision-resistant, or a one-way function or both. One example is SHA-256.) Second, this hash value is asymmetrically encrypted using the private key of the code signer. The public key of this code signer key is digitally signed by the root private key. The result is called a certificate, which is attached to the signature of the hash. By using these means, a public key infrastructure can be set up to allow different authorized parties to sign code and data. RSA or ECC (Elliptic curve cryptography) are established as forms of asymmetric encryption. The IoT device itself can then validate the signed data and program code. This makes sure that the data and program code have not been tampered with and that they have been created by an authorized party.

Device identity: connected devices need to be able to authenticate themselves. This can be realized by using standard cryptographic authentication schemes, which requires the storage of a device private key in a way that cannot be compromised. For authentication and trusted communication, an open standard like OPC UA is an excellent solution, as it enables devices of different makes to be used in one system with full interoperability. For implementing the already-mentioned mechanisms, specific care is required to achieve a complete protection against the identified threats.

To achieve highest security for IP and tamper protection, a high level of security is required. Unlike the situation with software licensing in the desktop PC area, where attackers are mostly users trying to avoid license fees, the attackers in this new field are often found in organized crime, terrorism, secret services, or industrial competition. What all of these attackers have in common is that they have considerable financial resources for highly professional attacks. Therefore, to inspire trust and confidence in the user and safeguard the family jewels of intellectual property of the makers, cryptographic keys and calculations need to be produced in so-called secure elements. These can be a Trusted Platform Module, TPM, any other solutions like TEE or Trustzone, or an industrial-grade dongle like CodeMeter that contains a smart card chip with the highest security level, certified according to Common Criteria. Using any of the described secure elements makes it possible to develop a truly secure IoT device. The secure element will become an essential small and isolated part of the device.

Flexible storage for multiple licenses: like the apps on a smartphone, different data and applications need to be licensed on an IoT device. Either the secure element needs to be capable of storing many cryptographic keys and license options, like CodeMeter with its large secure key storage of more than 300kB, or external license files with the keys need to be handled and loaded into the secure element, as with TPM 2.0. To realize new business models, this is definitely a must for device makers, and it is not enough to store single cryptographic keys.

Communication security: connected devices need to be able to authenticate themselves and guarantee privacy by encrypted communication and authenticity by digitally signed communication. OPC UA is an excellent solution, but the secure storage of private keys and the execution of crypto algorithms in a secure and efficient way are again key.

The industry-proven CodeMeter technology made by Wibu-Systems is employed in several widely used systems. It allows developers an easy integration into their products and processes and comes with the following components.

Secure key storage is the heart of the technology, either with storage in an encrypted and digitally signed file bound to an existing secure element in an IoT device, or as hardware with a smart card chip. The libraries allow decryption and encryption (AES, ECC, RSA) as well as the storing of flexible license models.

Software integration: powerful and easy-to-use tools allow automatic code protection and offer a protection API. This is available for many operating systems, either without OS modifications using self-extracting protected code, AxProtector technology, or with operating system integration, ExProtector technology. Integration in OPC UA stacks makes implementation easy.

Back Office integration: key and certificate deployment, license deployment, and license administration is made easy with CodeMeter License Central. Customizable connectors using web services are available for use with ERP systems like SAP, CRM systems like Salesforce, or ecommerce applications.

The IEC61131 development tools Codesys, B&R Automation Studio or Rockwell Automation Studio 5000 allow the user to protect his project source code as well as the target code created for the runtime system. The developer can apply the protection by setting certain properties in the application build options. Further features, such as license management, are available when using the extended API to check for licenses.

Secure boot and a check of the signature of software components in the OS loader are integrated in Wind Rivers VxWorks, and similar mechanisms can be used with Linux OS as well as with Uboot, Grip, and UEFI bios.

OPC Unified Architecture (UA) is getting more and more established in many applications. It offers interoperable security functions in accordance with IEC 62541. The distribution of certificates to many networked OPC UA servers and clients and their secure storage is a major challenge for medium-sized applications. Wibu-System CodeMeter solution will integrate OPC UA in its components to make the interoperable use of OPC UA security mechanisms easier in practice.

The DAVE microcontroller developer tools from Infineon offer an integrated solution to sign and encrypt application code and store licenses in an XMC4000 microcontroller, bound to its chip ID with CodeMeter technology.

Kontron, one of the leading CPU board and module manufacturers in the world, has decided to embed a CodeMeter smart card chip on all new designs to make them IoT ready.

The integrity of IoT devices can be ensured by using cryptographic methods in a clearly defined process and a secure hardware device for key storage and state machine. A secure implementation of symmetric and asymmetric encryption methods as well as hash functions and functions for signature validation allow the implementation of the proposed mechanisms. New business models can be realized on this basis. ■