Digital signatures: fighting firmware hacking and hardware cloning

By Dirk Akemann, Segger

In the age of the IoT, firmware attacks are becoming an increasing danger. Digital signatures can protect embedded applications against hacking and prevent the cloning of hardware.

The number of reports about successful hacks of different devices increases by the week: from internet-hijacking of cars to remotely stopping anaesthetic machines and opening allegedly super-secure safes through a simple USB stick, from worms attacking Apple computers to voting machines being decommissioned because of serious vulnerabilities. These attacks are potentially dangerous not only for the individual user, but in fact for society.

During the rapid evolvement of embedded and interconnected devices, most developers of hardware or applications have neglected the issue of preparing the firmware for the growing danger of third-party attacks. They still do without signing and authorizing firmware updates, or neglect the necessary diligence for the handling of signatures. Lack of awareness of this threat makes it even easier to infect firmware. Once in the system, intruders remain in the firmware and do not get detected by common scanners, which usually do not operate on this level. Even a reinstallation of the operating system will not help. Such attacks can ruin devices – or turn them into remotely controlled tools in a more malicious undertaking. Some people seem to surrender in view of this threat. “No matter how much time, money and effort we could put into a device or a system to make it as secure as possible, there is always the possibility that someone else would put in the time, money and effort to exploit that system,” the director of a US City Electoral Board said, arguing against decommissioning unsecure voting machines. But that is not a convincing argument, because there are in fact solutions for protecting the firmware, without significant expense or effort.

emSecure, developed by Segger Microcontroller, is the first software package for generation and verification of digital signatures that runs on embedded devices without much effort and, at the same time, is also a complete toolset. It has been developed specifically for embedded applications, is easy to implement, and the process of signing and verifying is so quick that it does not degrade perceived boot time and the user experience. It relies on the concept of digital signatures with a pair of private and public keys. The manufacturer of a device or application couples the public key within the product. Whenever he provides firmware updates or other relevant data for the product, these will be signed by help of the private key. The receiving product then checks, by help of the public key, whether it can validate the firmware by its signature. If so, the update is authentic and will be installed. If not, it will be stopped or erased. This way, non-signed or manipulated firmware cannot invade the product. Usually, checksums or hashes are used to evaluate if data has been corrupted or lost during transfer. However, these instruments do not indicate anything about the sender of the data, i.e. if the software update is from the original manufacturer of a product. They do thus not contribute to higher security – in contrast to digital signatures. Only the latter verify the authenticity of the sender. The digital signature generated by emSecure is based on the asymmetric RSA cryptosystem. Its algorithms have proven their worth for decades. 2,048-bit keys, which are used by default, are currently regarded as absolutely safe and not to be broken by reverse engineering. Governmental institutions like the NSA were not involved in the development of RSA, which means it does not contain a backdoor as is usually demanded by these institutions. However, both DSA and ECDSA signing and verification code is also available from Segger, on request.

emSecure has been designed from scratch for best possible portability and performance together with minimum memory requirements. It can even be used with small single-chip microcontrollers, without the need for additional external memory or hardware. The keys and signatures can best be generated on a stand-alone PC. It has been tailor-made for two areas of application: anti-hacking and anti-cloning.