Paul Curtis, Katrin Berkenkopf
Embedded Experts Team, SEGGER Microcontroller

Fighting product clones through digital signatures

Product piracy and forgery are growing problems that not only decrease turnover and profits of Original Equipment Manufacturers (OEMs), but also pose a threat to their image and brand awareness, and might result in legal action when counterfeit parts cause damage to equipment or, worse, threaten human life.

In this paper, the functionality and application of digital signatures is described as a means to tackle the problem. SEGGER's emSecure software package, which applies an electronic signature to any fragment of electronic data or *digital asset*, is presented as a means to assist OEMs in this endeavour.

European customs officials seized almost 36 million Intellectual Property infringing goods at the EU borders in 2013 alone, with an estimated value of 760 million Euro. According to a 2014 survey by the German Engineering Federation VDMA, the financial damage caused by product piracy has been just below 8 billion Euro annually for the past years — this is just for the engineering industry.

The survey suggests that 71% of engineering companies have already been affected by product forgery, however 41% of the participants have not taken any measures in response. Using digital signatures is a precautionary measure which is easy to employ and brings a high level of security.

What is a digital signature?

Broadly speaking, a digital signature is rather like a written signature: it's a distinctive mark used as a form of identification in authorizing a document. The problem with a handwritten signature on a signed document is that its authenticity can be challenged, which is usually why such signings are witnessed.

As many documents are now prepared on a computer, it would seem natural to find some way of signing an electronic copy of a document using a computer, leaving only ceremonial signings on public occasions for heads of state. Digital signatures remove *all ambiguity* from the question of whether a signature is genuine.

A digital signature is a small quantity of data, in the order of 128 to 256 bytes, that verifies two things:

- The content of the document that was signed.
- The identity of the person who signed the document.

An unsigned document combined with a digital signature forms a digitally signed document. Digitally signed documents are now commonplace with companies springing up to offer online electronic signing services. Whilst digital signatures authenticate the signatory and the document, they can be deployed more widely to sign things other than electronic documents, as we shall see. In the following sections, instead of documents we are thus talking about any *digital asset*, be it firmware, a license key, or manufacturing data. The ability to sign any digital asset to show authenticity is where the true potential of a digital signatures lies.

The advantages of digital signatures

Digital signatures offer many advantages over written signatures:

- Digital signatures cannot be forged so there is no doubting the authenticity of a digital signature.
- With appropriate information, anybody can verify the authenticity of a digital signature.
- A digitally signed asset cannot be changed without making the signature invalid — if the signature or asset is tampered with, it is immediately apparent.

Digital signatures also offer many advantages over familiar schemes using message authentication codes (MACs) such as CRCs, checksums, message digests, or high-reliability error detection and correction codes. These schemes offer no assurance that the asset that they are applied to has not been tampered with as the MACs can be regenerated as needed by the tamperer. Digital signatures affixed to the asset cannot be regenerated as needed because the tamperer has no access to a crucial part of digital signature technology, the private key, which we discuss next.

What makes a digital signature trustworthy?

emSecure uses digital signatures based on the RSA cryptosystem which has proven robust against decades of attacks on the algorithms. RSA algorithms are the cornerstone of many protocols in use today by financial, defense, and communication industries. In fact RSA is now commonplace and pervasive: when you open up a web page on many Internet sites and see a padlock icon, your computer will more than likely be using multiple RSA digital signatures to ensure the site that you visit is authentic and trustworthy.

RSA uses a special type of cryptography that is hard for many to understand—it uses a system of asymmetric keys where the key that signs a message is not the same as the key that verifies the message. This dual-key cryptographic asymmetry underpins many of the encryption, authentication, and signature schemes in use today.

RSA's security is based on the premise that you cannot compute the private key from the public key without significant computing resources. For 2048-bit key sizes, it is considered well beyond the capability of governments, with all their computing power and using the very latest number-theoretic methods, to recover a properly-generated RSA private key before 2030, and most probably well beyond that.

We should say that RSA signatures are not the only form of digital signature—there are others that are based on different mathematical problems, but RSA signatures are the widest deployed form of digital signature. RSA digital signatures verify quickly and much faster than other digital signature schemes. Because verification has low computational and memory overhead, an RSA digital signature is a perfect candidate when targeting constrained embedded systems.

The signing process

Preparing to sign an asset

Before signing an asset, you generate an RSA key pair that is known only to you. The key pair is composed of a private key that you keep safe and a public key that is genuinely public — you publish or distribute the public key and declare that it is your public key. The two keys serve different purposes: the private key is used when creating a digital signature for an asset and the public key is used when authenticating the digital signature of an asset.

emSecure uses a system of proven-prime generation that strictly implements the appropriate National Institute of Standards and Technology (NIST) specifications, using the Shawe-Taylor proven prime algorithm, in full. This ensures that the RSA key pair generated is cryptographically strong and, given an appropriate seed, the primes used in the generated keys are certified.

Signing the asset

To digitally sign an asset requires two cryptographic operations. The first is to compute a message digest over the content of the asset to be signed that “fingerprints” the asset. In today’s world, the message digest of choice is one of the Secure Hash Algorithms (SHA) developed by NIST the USA. Given only an asset’s fingerprint, you cannot reconstruct the original asset from it, nor can you easily construct an asset whose contents would generate the same fingerprint.

Once you have the fingerprint, the next step is to encrypt the fingerprint using your private signing key of the RSA key pair. The output of this part is the digital signature that you attach to the asset.

Verifying the asset’s signature

When presented with a digitally signed asset, you would like to verify its authenticity. For this you require the public verification key of the signatory's RSA key pair. If you are exchanging signed electronic documents between humans, the public key will either be published or exchanged and agreed between recipient and signer in advance. There is a means of exchanging public keys to establish identity, called a Public Key Infrastructure (PKI), that is beyond the scope of this document to describe.

With possession of the public key, verifying the signature is a relatively straightforward matter. You take the signed asset, detach the digital signature, and apply the public verification key to the signature to recover the signed asset’s fingerprint. The signature is verified if the unsigned asset's fingerprint matches the (recovered) signed asset's fingerprint exactly.

If the fingerprints do not exactly match, the forgery is revealed. The forgery is detected in all cases, tampering with the asset, tampering with the signature, or fraudulent signing:

- Tampering with the asset’s content leaving the signature unchanged leads to a different asset fingerprint, different to the signed fingerprint.
- Tampering with the asset’s signature without changing the asset’s content leads to a different signed fingerprint, different to the asset’s fingerprint.

- Signing the asset with a fraudulent RSA key generates a different signature and, when recovered by the verifier, the recovered signature differs from the asset's signature.

Applying digital signature technology

In this section we show you how you might apply digital signatures. These examples are by no means exhaustive, but should help you to realize how versatile digital signatures can be.

Clone prevention

Many microcontrollers have a unique identifier (UID) laser-etched into them during production. Developers can use this UID, should they want, to generate equipment serial numbers, device addresses, or record it as part of their manufacturing traceability data. However, with digital signatures, it's now possible to use this UID to stop blind copying of hardware and software in order to clone a device. As part of the manufacturing process, the original equipment manufacturer (OEM) can sign the firmware, together with the UID, and write the signature into the equipment's internal memory. Each signature will be different because each device has a different UID. As part of the power-up sequence, the equipment verifies the computed signature of the firmware combined with the UID against the signature programmed into permanent memory. If the signature does not verify, we can draw one of these conclusions:

- The device's firmware has been tampered with.
- The device's signature has been tampered with.
- The device's firmware has been cloned into a new device that has a different UID.

In all cases, the course of action is up to the equipment manufacturer, and the manufacturer may decide that it's best to shut the device down safely to prevent injury, or to report some error to the operator.

Because the cloner cannot generate a signature over the firmware and UID, as he does not have access to the private signing key, there is no possibility of copy-and-paste cloning of your equipment.

Authenticated firmware upgrades

In much the same way as clone prevention, it is relatively simple to sign new firmware upgrades to ensure that replacement firmware only originates from the manufacturer.

The OEM signs the new firmware image and transmits it, together with the digital signature, to the equipment. The equipment reads the new firmware image, verifies the signature, and only if the signature is valid does it proceed to replace the operating firmware.

As the private key is held securely by the OEM, there is no means for an attacker to create a correctly-signed firmware image and, therefore, the equipment remains safe from exploitation as long as the digital signature is checked as part of the firmware replacement process.

Software license keys

Because signing a digital asset authenticates the signatory and the asset's content, a digital signature makes a perfect means to distribute license keys for applications.

Many license schemes use a home-grown encryption protocol applied to a license key in order to hide the sensitive data and make it confidential during transfer to the application. The application decrypts the delivered packet and uses some other form of verification to ensure that the delivered license key is valid. It's natural, as a software engineer or manager, to consider the license key data as sensitive and somehow hide its content such that it cannot be uncovered, cloned, or tampered with.

With a digital signature, it is not necessary to hide or encrypt the license key and, in fact, it proves to be a positive benefit. If the license key format is understandable by a human, for instance a sequence of text fields indicating granted capabilities, expiration dates, and hardware IDs, all observers can agree on what the license capabilities are by simple inspection. As the proposed license key format is nothing more than a text string, additional features can be added over time without invalidating any previously distributed license key.

It may seem counterintuitive to have the license key in the open and readable, but application of digital signatures make it secure. If the software author affixes a digital signature to the license key, we now have a signed license key that can be verified by anybody, including the licensed application. As nobody but the software author is able to apply a valid signature to the license key, and tampering with the key or the signature is immediately detected, using digital signature technology is a perfect way to distribute secure license keys.

The utility of emSecure

emSecure makes the application of digital signatures in the fight against cloning and hacking straightforward and takes little effort to deploy. It is the first software package which has been tailor-made for embedded systems, but works with other applications, too.

Compared to hardware-based solutions, emSecure is advantageous in terms of costs and space, and can be applied to existing equipment that is in manufacturing as well as new, planned equipment.

Summary

We conclude by summarising the salient facts relating to digital signatures:

- Digital signatures are one solution to the growing problem of product cloning and counterfeiting.
- Digital signatures cannot be forged.
- Anybody can verify the authenticity of a digital signature.
- Once verified, there is no doubting the authenticity of a digital signature.
- Using RSA asymmetric keys, one private and one public, provides the basis of security.

- The RSA cryptosystem remains unbroken despite decades of research and analysis.
- Security scales with key sizes — the larger the key, the more secure the signature.
- Key sizes of 2,048 bits suffice for signatures to 2030. emSecure supports key sizes up to 16,384 bits.
- RSA signatures verify quickly, much faster than other forms of digital signature.
- Verification requires low computational power and resources, ideal for even the most constrained embedded system.
- Tampered digitally-signed assets are immediately revealed upon signature verification.
- Applied digital signatures can detect modified firmware and, with a UID, detect cloning.
- Authenticated firmware upgrade is simplified with a digital signature.
- CRCs, hash algorithms, or error detection and correction codes do not offer the ability to detect tampering; digital signatures do.