Application Signing

See also: Certificates and their uses

Why are applications signed?

Application signing means that an application is signed with a private key. For each private key, there is a corresponding public key and a **certificate**, for example with TrustCenter Publisher ID certificate, developer certificate or a self-created certificate (usually called 'self-signed' certificate).

When installing a signed application package on the device, the application installer checks the signature and then checks the certificate of the public key against a root certificate that has been provided with the device. If the certificate (of whoever signed the application) can be traced back to a valid root certificate, the application package is deemed as **trusted** to the extent that the root certificate was trusted. If the application was signed with a key that only has a self-signed certificate, then the certificate cannot be traced back to a valid root certificate, and the application package is then **untrusted**.

For trusted packages, there will be less warning messages, and the software has access to more powerful capabilities. For Java ME, the signed MIDlets will cause fewer confirmation questions from the user, when trying to access restricted Java APIs.

Terms and definitions

Don't confuse signing with certification, or certification with certificates, or certificate issuance with application certification. These terms are sometimes used interchangeably, but really shouldn't, as they are completely different things.

- **Signing** is the technical act of creating a digital signature on a software package (SISX file on S60, signed JAR on Java) using a private key. The key pair (consisting of public and private keys) can have an authority-issued certificate or a self-signed certificate - this has no effect on the signature itself, only the trust that the signature conveys.
- **Certificate** is a proof of ownership for the public and private keys. A certificate issued by a certification authority (a.k.a. the 'issuer', Nokia or Verisign, for example) indicates that the public/private key pair belongs to the entity ('subject') described in the certificate. If the certificate is self-created (self-signed), the certificate does not convey any trust whatsoever - in this case, the certificate is just created because of a technical necessity.
- **Certification** means that an application is tested against a commonly accepted test criteria and then certified. The application is signed with a key that has a related certificate that can be traced back to the root certificate that is available in devices. Then, when a user installs such an application to a device (that has correct root certificate), there is trust between the application and the device. Application installer checks the signature and verifies that the application comes from a trusted source. If application is not certified (there is no corresponding root certificate in the device), user will get a note when installing the application. The note says that the application comes from an untrusted source.
- **Certificate issuance** is the act of a certification authority issuing a certificate tied to a certain public/private key pair. This is not the same as certifying applications, and certification authority is not the same thing as a testing house that certifies applications.
- A **trusted** software package is usually something that has been signed with a key whose certificate can be traced back to a trusted root certificate. Packages signed with self-signed certificates, or not signed at all, are **untrusted**.

S60 considerations

An important difference to understand when talking about S60 3rd Edition is that **signing of software packages is mandatory, but certification is not**. The software package needs to be technically signed, but the associated certificate does not need to be issued by a trusted party. However, certain capabilities may require also the package to be certified (signed by a key that has a trusted certificate).

Signing in testing programs

When an application is tested via an industry-wide and commonly accepted testing program (Symbian Signed or Java Verified), the test house (which, to reiterate, is not the same as certification authority) takes care of the certification (i.e. signing the software package with a key that has a trusted certificate) after tests have been passed.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.