digicert®

Hash signing with DigiCert Software Trust Manager

Quickly and easily secure code with hash signing

In our connected world, businesses and users rely on file sharing for everything from software and app releases to patches and updates. But with many security solutions, signing code means giving up control over the very files you're trying to protect.

With DigiCert Software Trust Manager, files are never uploaded, so your valuable property stays at all times inside your environment. More, hash signing helps you avoid common local signing management issues like key protection and user access. It's the best of both worlds—the speed of local signing with the security assurance of the signing service.

How does hash signing work?

1. Client-side libraries, including KSP for Windows, Apple CryptoTokenKit for Apple and PKCS11 for all others, facilitate a working partnership between Software Trust Manager APIs and the local signing tool
2. Client-side libraries enable the secure signing of large files by generating the hash of the requested application, then sending that hash to Software Trust Manager, where it's signed with a protected private key
3. When service-based signing is complete, the client-side libraries give the signed hash back to the signing tool, where the signature integrates with the original application. Now, the original application is signed, having never left your environment

Key features and benefits

1. **Protect**
   Unsigned software never leaves your organization's environment
2. **Control**
   Client-side libraries are called in conjunction with local signing tools via command prompt
3. **Support**
   Full integration into CI/CD for signing automation using common tools like Azure DevOps, Jenkins, Apache Ant, Gradle, and Apache Maven
4. **Manage**
   Fully integrated with Software Trust Manager APIs to support user access management, private key protection and auditing of all code signing events

Service signing with files	Local signing with files	Service signing with hashes
Slow but secure	Fast but not secure	Fast and secure
Whole files must be uploaded and downloaded to the signing service, which slows processing time, expecially with large files.	There is no upload or download, so processing is quick. However, keys may be in multiple places, and there are no signing permission controls to track who is using the keys. If the keys are not in a FIPs-compliant device they can be copied and replicated.	Only the hash is uploaded and downloaded, so transfer speeds are quick, and the full process is protected by encryption. Keys are kept in Hardware Secure Modules (HSMs), and files are kept locally, so the entire system is secure. For compliance to code signing standards, keypair user access permission is checked and signature records are logged.

DigiCert Software Trust Manager leverages client-side libraries to support hash signing in the following ways:

KSP for Windows

- Supports the signing of Authenticode files with Windows SignTool, Mage, Nuget, Clickonce, HLK, HCK
  - Authenticode file extensions *.EXE, *.DLL, CAB, *.MSI, *.JS, *.VBS, *.PS1, *.OCX, *.SYS, *.WSF., *CAT, *.MSP, *.CPL, *.EFI, *.ARX, *.DBX, *.CRX, *.XSN, *.DEPLOY, *.XAP, and more
- Supports Extended Validation (EV) and Organization Validation (OV) public code signing, as well as private code signing

CryptoTokenKit (CTK) for Apple

- Supports the signing of Apple application bundles and frameworks. File types supported: *.dmg, *.ipa, *.app
- Supports Apple productsign used with signing installer packages and archives. File types supported: *.pkg,*.mpkg

PKCS11 for Java, Android, Linux, Docker, OpenSSL, GPG, XML, and others

- Supports signing Java file formats (*.JAR, *.WAR, *.SAR, *.EAR) and Android *.APK with JarSigner
- Supports Docker Notary, APKSigner for Android, OpenSSL, GPG, Debian, XML, JSign, osslsigncode, and more
- Supports EV and OV public code signing certificates, and private certificates with 25-year duration in order to meet Android requirements for signing

Additional features

With hash signing, you retain the benefit of key protection, user management, and reporting, which Software Trust Manager provides. In addition, keys can be in offline and online mode. Offline keys require an approval for user to sign during a pre-approved release window, adding an extra layer of security to key protection and usage. Software Trust Manager also supports the incorporation of timestamping into your signature as part of your request. This is common practice for Authenticode, EV code signing, and Java signing.

For more information on SoftwareTrust Manager, contact one of our PKI experts at pki_info@digicert.com or visit: www.digicert.com/software-trust-manager