Frequently Asked Questions (FAQs) on code signing and Software Trust Manager

What is code signing?

digital binaries have not been altered. This method leverages the Public Key Infrastructure (PKI) framework to attests to the integrity of the code or binaries. Code signing acts like a digital shrink wrap: the process of code signing creates a “package” with the signed

can be trusted. And vice-versa: if the “packaging” was

cannot be trusted.

Why is code signing important?

Code signing minimizes the risks of code tampering.

when the integrity check fails during download. This helps recipients to avoid downloading tampered code which may contain malware.

You can sign all types of digital binaries including drivers, firmware, containers, applications, mobile apps as well as source code artifacts.

What is the traditional approach to code signing?

is issued to the requester to use on behalf of their

one location and provides trust based on the website domain(s) and sub-domain(s) that it secures. Code

developers and or engineering teams will need access at different times.

The traditional approach to code signing is where

the organization has to take responsibility for protecting the private key locally while still permitting those with signing responsibilities to have access

organizations to undertake and can be both expensive and time consuming to implement correctly. In a

on-travels. Others may be assuming temporary roles in signing or need access to the keys at certain times. The appropriate keys need to be available at the right time to the right developer. All the keys will need to be protected always.

What could happen if my code signing private keys were to fall into the wrong hands?

Code signing private keys are critical assets in every

application came from them and if the signature is

malware or vulnerabilities to infect or exploit user

[illegible] exploit u[illegible]

DigiCert® Software Trust Manager

enables organizations to adhere to code signing best