Formally verified and upgradable trusted functions

SWITS 2024

Marcus Birgersson marbir@kth.se KTH Royal Institute of Technology
Musard Balliu musard@kth.se KTH Royal Institute of Technology
Cyrille Artho artho@kth.se KTH Royal Institute of Technology

May 27-28, 2024

Abstract

Trusted Execution Environments such as Intel SGX is a hardware assisted isolated environment for confidential computation tasks. It makes it possible to use an untrusted host computer to carry out sensitive tasks while still protecting the confidential data. TEE:s provide an attestation service which makes it possible for an outside party to verify that one communicates with the correct machine, running the correct version of the software. The TEE generates a cryptograpic proof that contains the hash of the current running software. Anyone can hence inspect the software to make sure that it does what it should, compute the hash and compare this with the hash from the attestation process. If the software changes, the hash changes. This is in general a feature, the party that manages the software should not be able to change the code without any user notice this. The problem is that if one needs to update a small part of the software, for for example optimization aspects, all users once again needs to inspect the whole software stack to make sure that the changes was not malicious. In addition, manual inspection of the software is a tedious task that in many cases requires experts. We want to change the inspection of the implementation, to instead inspect pre- and post-condition of the software. By instead inspecting what a certain piece of software will do, rather than how it does it, moves the trust from the developer of the software to the verifier of the software. We fulfil this goal by deploying Dafny, an automatic formal verifier of software, inside the TEE, together with the specifications of the computational functions. This makes it possible for any user to inspect what the functions will do, rather that how they will do it. At the same time, the developer can update the implementation at any time as long as it does not violate the already deployed specifications.