TEEs

Trust Measurement

- trust is non-measurable because it's a subjective property
- is either static or dynamic
- **Static Trust**
    - base on a evaluation against a specific set of security requirements
    - trustworthiness is measured only once and before its deployment
- **CC (Common Criteria)** are an internation standard that provides assurance measures for the security evalution
    - 7 evaluation assurance levels (EAL1-EAL7) where high numbers include all requirements of the preceding levels
- **Dynamic Trust**
    - based on the state of the running system
    - trustworthiness is constatly measured throughout its lifecycle
    - in this context, trust can be defined as an expectation that the system state is as it is considered to be:secure
- **Root of Trust (RoT)** - a trusted entity to provide trustworthy evidences
    - **Role:**
        1. **trusted measurement**
            - Trustworthiness of the system, namely the generated code, depends on the reliability of the trust measurement
            - If a malicious entity can influence the trust measurement, then the generated code is of no value

2. f**unction that computes the trust score**
    - **trust score** → *boolean* that indicates the integrity state of the code
    - f(TEE, protection profile, RoT, measurements) → returns the trust level of a given TEE depending: certificating protection profile, reliability of RoT and integrity measurements

- **RoT** is a tamper-resistant hardware module, sometimes called **trust anchor**, depending from the hardware platform that is used to guarentee the isolation properties
- **TrustZone-based** systems rely on **secureROM** or **eFuse** technology as trust anchor
- **PUF** (Physically Unclonable Function) is a promising RoT technology for **TEE**

Trust in TEE is a Hybrid Trust → both static and semi-dynamic

- before deployment, a TEE must be certified by thoroughly verifying its security level in accordance of a *Protection Profile* (predefined set of security requirements)
- GlobalPlatform defines a protection profiel conforms to EAL2
- RoT protects the integrity of the TEE code →during each boot, RoT assures that the loaded TEE is the one certified by the platform provides
- **Semi-dynamic** → TEE is not supposed to change its trust lvel while running because it is protected by the separation kernel

Building Blocks

- **Secure Boot**
    - assures that only code of a certain property can be loaded
    - if a modification is detected, the bootstrap process is interrupted

- consists of variuos stages to a chain of trust to be established
- Chain representation
  - $I_0 = True; I_i + _1 = I_i \hat{} V_i(L_i + 1)$
  - I_i → integrity of layer i
  - I_0 → integrity of the initial boot code
  - V_i → verification function that performs cryptographic hash of the $i^t.^h$ layer and compares the result to the reference value
  - Without the integrity of I_0 integrity becomes pointless
  - Initial boot code is protected by a tamper-evident hardware module

- **Secure Scheduling**
  - assures coordination between TEE and the rest of the system
  - assures that tasks running in the TEE don't affect the responsiveness of the main OS.
  - should take real-time constraints into consideration
- **Inter-Environment Communication**
  - interface allowing TEE to communicate with the rest of the system
  - introduce new threats:
    - message overload attacks
    - user and control data corruption atacks
    - memory faults caused by shared pages being removed
    - unbound waits caused by the non-cooperation of the untrusted part of system
  - each mecanism should satisfy:
    - reliability (memory/time isolation)
    - minimum overhead (unnecessary data copies and context switches)
    - protection of communication structures

- **Secure Storage**
  - storage where confidentiality, integrity and freshness are guaranteed and where only authorized entities can access the data
  - freshness → protect against replay attacks and to enforce state continuity
  - **Sealed Storage** → based on:
    - integrity-protected secret key that can be accessed only by the TEE
    - cryptographic mechanisms - authenticated encryption algorithms
    - data rollback protection mechanism - replay-protected memory blocks RPMB
- **Trusted I/O Path**
  - protects authenticity and confidentiality of communication between TEE and peripherals
  - input and output data are protect from being sniffed or tampered with by malicious applications
  - protects against four classes of attacks:
    - screen-capture attack
    - key logging attack
    - overlaying attack
    - phishing attack
  - allows a human user to directly interact with applications running inside TEE

Formal Methods

Set of formal specifications with a formal language

Design of TEE consists of two aspects:

- requirements specification
- implementation

Two goals:

- specification
- verification

Formal Specifications → describes the requirements of the system

→necessary condition to perform proof-based verification

Formal Verification → used to analyze the formal model for the desired properties

→ two approachess:

- model checking
    - technique in which systems are modeled as finite state systems
- theorem proving
    - proves that a system satisfies the specifications by deductive reasoning
    - proofs can be constructed by hand but most cases machine-assisted proofs are used
    - is used more than model checking because it can efficiently deal with complex properties