Secure Booting

The Problem
- securely identify (remote) systems
- their hw, os, ...

Sources (ca. 1989)
- DSSA (Digital System Security Architecture)
- BirliX Security Architecture (Michael Groß)

3 Steps

Preparation … os and hw builders

Booting

Authentication

Preparation: CPU, OS (1)

- CPU
  - CPU generates key pair outputs: K^{pub}_{CPU}
  - Manufacturer of CPU (CUM) certifies: $\{CPU\text{-}ID, K^{pub}_{CPU}\}CPU^{priv}_{CUM}$
- OS:
  - Manufacturer assigns name and certifies: $\{OS, ID\text{-}OS\}K^{priv}_{OSM}$

Booting

- CPU verifies signature of OS (using K^{pub}_{OSM})
- and generates key pair for the new unit "loaded os": LOS(CPU/OS)
- CPU certifies: $\{ID\text{-}OS, K^{pub}_{LOS}, k^{pub}_{OSM}\}K^{priv}_{CPU}$

Authentication

Objective: ID-OS, ID-CPU, OSM

- send: nonce
- receive:

  $\{nonce, t\}K^{priv}_{LOS}$

  $\{ID\text{-}OS, K^{pub}_{LOS}, K^{pub}_{OSM}, t\}k^{priv}_{CPU}$

  $\{ID_CPU, K^{pub}_{CPU}, t\}K^{priv}_{CUM}$

Authentication cont.

known: k^{pub}_{OSM} and k^{pub}_{CUM}

trusted: OSM and CUM

k^{pub}_{CUM} verifies: $\{Id_CPU, k^{pub}_{CPU}\}$

k^{pub}_{CPU} verifies: $\{Id_OS, k^{pub}_{LOS}, k^{pub}_{OSM}\}$

k^{pub}_{LOS} verifies the response to the authentication request

System Requirements

- Tamperfree Black-Box:
  - Secure Key Store on CPU
  - no direct manipulation of memory
- OS can protect a key
- Trust in builders of hw, os