Problem

The Trusted Platform Module (TPM) currently supports secure (internal) execution of only a small set of specific-purpose commands. In order to base the security of applications solely on the TPM (and not on a trusted boot into some trusted OS), we need to develop new techniques for existing and future TPMs.

Approach: We are developing new techniques for existing TPMs, as well as a new device we call the Trusted Execution Module (TEM), which can replace the TPM in the future. The TEM accepts packets of partially encrypted virtual machine code, which can be generated even by untrusted parties. Mechanisms for secure, fresh, and shareable virtual memory enable many applications, including: Trusted Storage on Untrusted Servers, Offline Count-Limited Decryption and Delegation, Secure Mobile Agents, Count-Limited Programs, Secure Computation, etc.

Approach and Impact

New approach

Research Impact

• Virtual Monotonic Counters

• Evolution and Design Future TPM

• Count-Limited Objects/Instructions

• Better Scalability/User-Freedom

• Trusted Execution Module

• Transactions among ‘Offline Users’

Results: As a first application, we used existing TPMs to address the problem of using an untrusted server to provide trusted storage for a large number of clients, where each client may own and use several different devices that may be offline at different times and may not be able to communicate with each other except through the untrusted server (over an untrusted network). We implemented tamper-evident storage where clients are guaranteed to immediately detect illegitimate modifications to their data (including replay and forking attacks) at the time of critical operations. We implemented a virtual counter manager maintaining a large number of virtual monotonic counters using untrusted storage and a TPM. We tested an actual implementation using PlanetLab and a PC with a TPM 1.2 chip.