

# TCG: Trusted Computing Architecture

# Background

- ◆ TCG consortium.      Founded in 1999 as TCPA.
  - Main players (promoters):      (>200 members)  
AMD, HP, IBM, Infineon, Intel,  
Lenovo, Microsoft, Sun
- ◆ Goals:
  - **Hardware protected (encrypted) storage:**
    - ◆ Only “authorized” software can decrypt data
    - ◆ e.g.: protecting key for decrypting file system
  - **Secure boot:**      method to “authorize” software
  - **Attestation:**      Prove to remote server what software is running on my machine.

# TCG: changes to PC or cell phone

## ◆ Extra hardware: **TPM**

- Trusted Platform Module (TPM) chip
  - ◆ Single 33MhZ clock.
- TPM Chip vendors: (~.3\$)
  - ◆ Atmel, Infineon, National, STMicro
  - ◆ Intel D875GRH motherboard

## ◆ Software changes:

- BIOS
- OS and Apps

# TPMs in the real world

## ◆ Systems containing TPM chips:

- Lenovo (IBM) Thinkpads and desktops
- Fujitsu lifebook
- HP desktop and notebooks
- Acer, Toshiba, Panasonic, Gateway, Dell, ...

## ◆ Software using TPMs:

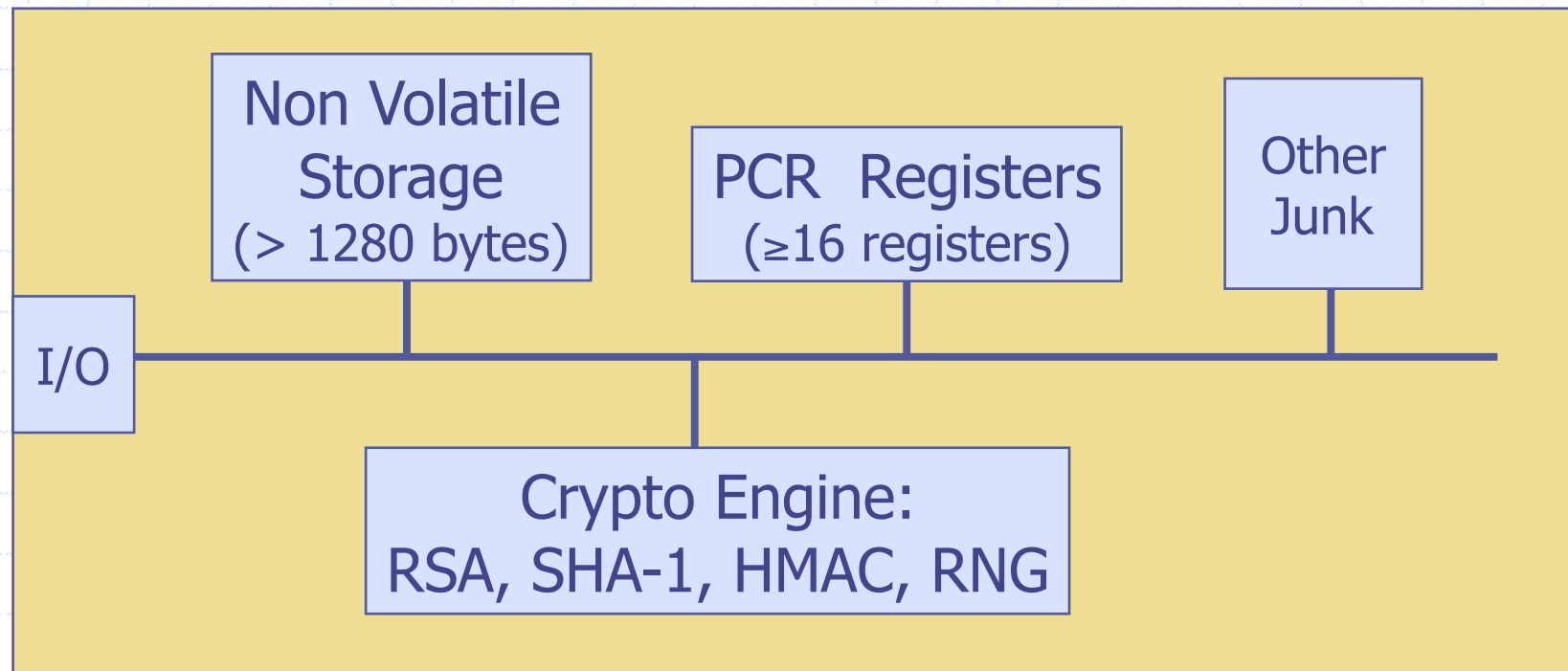
- File/disk encryption: Vista, IBM, HP, Softex
- Attestation for enterprise login: Cognizance, Wave
- Client-side single sign on: IBM, Utimaco, Wave



# TPM 101

- What the TPM does
- How to use it

# Components on TPM chip



RSA: 1024, 2048 bit modulus

SHA-1: Outputs 20 byte digest

# Non-volatile storage

1. **Endorsement Key (EK)** (2048-bit RSA)
  - Created at manufacturing time. Cannot be changed.
  - Used for "attestation" (described later)
2. **Storage Root Key (SRK)** (2048-bit RSA)
  - Used for implementing encrypted storage
  - Created after running  
`TPM_TakeOwnership( OwnerPassword, ... )`
  - Can be cleared later with `TPM_ForceClear` from BIOS
3. **OwnerPassword** (160 bits) and persistent **flags**

Private **EK**, **SRK**, and **OwnerPwd** never leave the TPM

# PCR: the heart of the matter

## ◆ PCR: Platform Configuration Registers

- Lots of PCR registers on chip (at least 16)
- Register contents: 20-byte SHA-1 digest (+junk)

## ◆ Updating PCR #n :

- TPM\_Extend(n,D):  $\text{PCR}[n] \leftarrow \text{SHA-1}(\text{PCR}[n] \parallel D)$
- TPM\_PcrRead(n): returns value(PCR(n))

## ◆ PCRs initialized to default value (e.g. 0) at boot time

- TPM can be told to restore PCR values via  
TPM\_SaveState and TPM\_Startup(ST\_STATE)

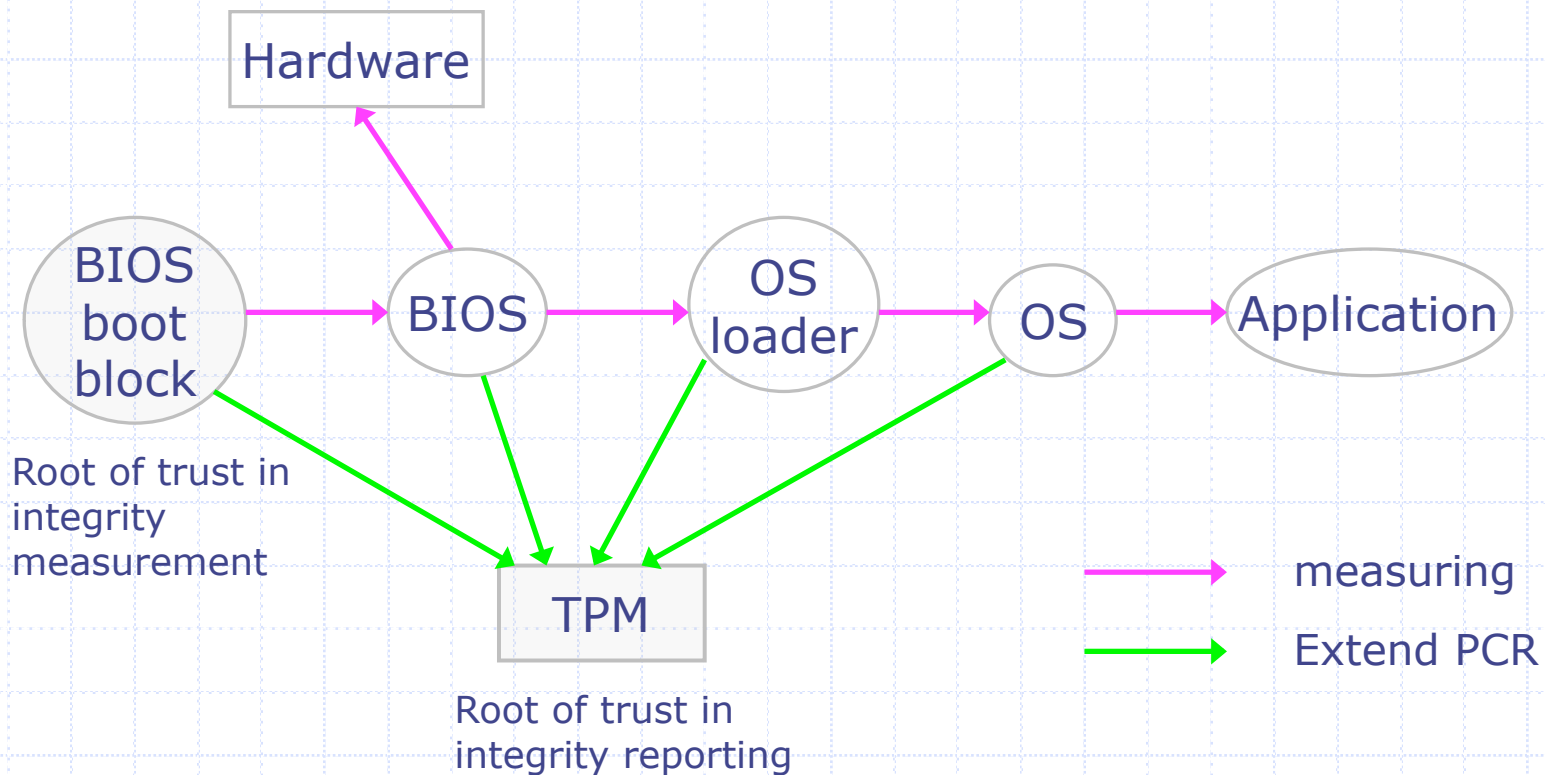


# Using PCRs: the TCG boot process

- ◆ BIOS **boot block** executes
  - Calls **TPM\_Startup (ST\_CLEAR)** to initialize PCRs to 0
  - Calls **PCR\_Extend( n, <BIOS code> )**
  - Then loads and runs BIOS post boot code
- ◆ BIOS executes:
  - Calls **PCR\_Extend( n, <MBR code> )**
  - Then runs MBR (master boot record), e.g. GRUB.
- ◆ MBR executes:
  - Calls **PCR\_Extend( n, <OS loader code, config> )**
  - Then runs OS loader

... and so on

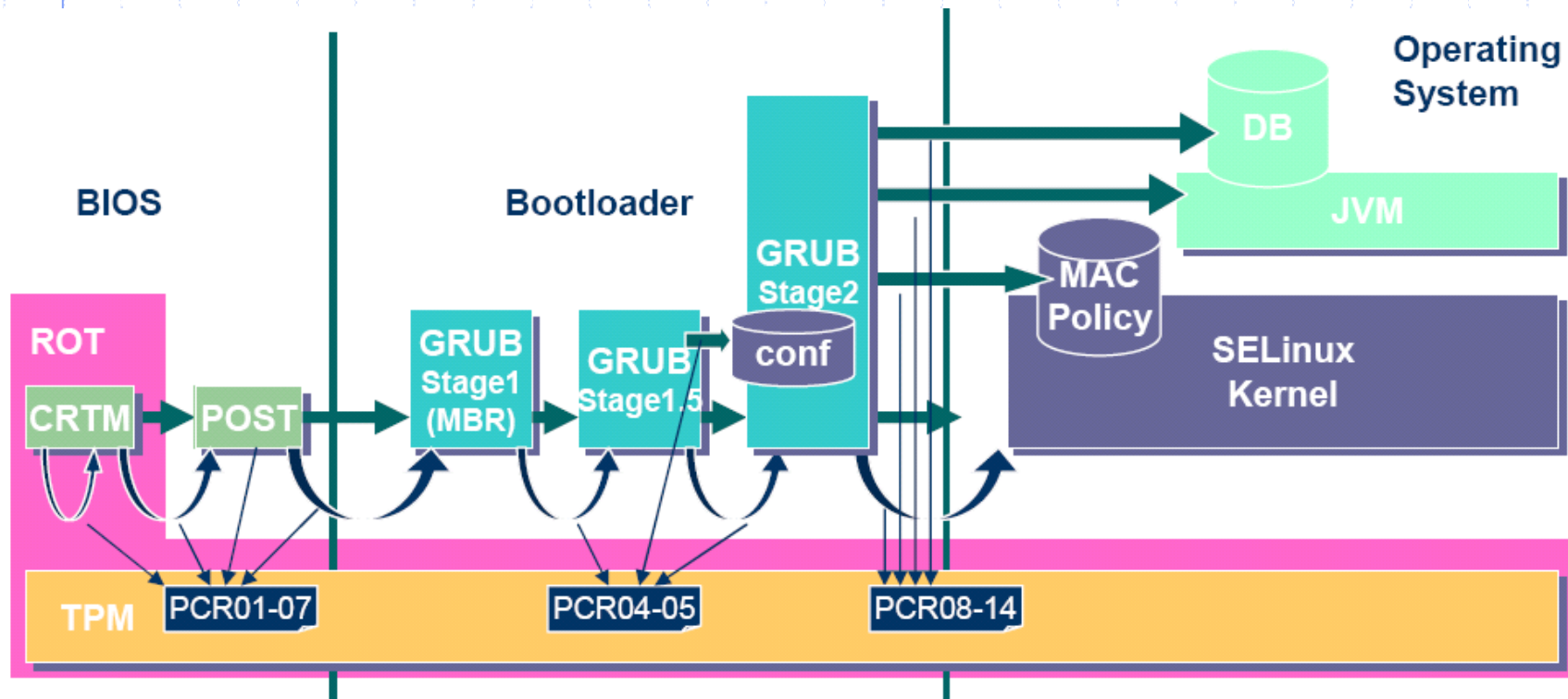
# In a diagram



- After boot, PCRs contain hash chain of booted software
- Collision resistance of SHA1 (?) ensures commitment

# Example: Trusted GRUB

(IBM'05)



What PCR # to use and what to measure specified in GRUB config file

# Using PCR values after boot

- ◆ Application 1: encrypted (a.k.a sealed) storage.
- ◆ Step 1: `TPM_TakeOwnership( OwnerPassword, ... )`
  - Creates 2048-bit RSA Storage Root Key (SRK) on TPM
  - Cannot run `TPM_TakeOwnership` again without `OwnerPwd`:
    - ◆ Ownership Enabled Flag ← False
  - Done once by IT department or laptop owner.
- ◆ (optional) Step 2: `TPM_CreateWrapKey / TPM_LoadKey`
  - Create more RSA keys on TPM protected by SRK
  - Each key identified by 32-bit keyhandle

# Protected Storage

◆ Main Step: Encrypt data using RSA key on TPM

■ **TPM\_Seal** (some) Arguments:

- ◆ keyhandle: which TPM key to encrypt with
- ◆ KeyAuth: Password for using key `keyhandle`
- ◆ PcrValues: PCRs to embed in encrypted blob
- ◆ data block: at most 256 bytes (2048 bits)
  - Used to encrypt symmetric key (e.g. AES)

■ Returns encrypted blob.

◆ **Main point:** blob can only be decrypted with **TPM\_Unseal** when PCR-reg-vals = PCR-vals in blob.

■ TPM\_Unseal will fail otherwise

# Protected Storage

- ◆ Embedding PCR values in blob ensures that only certain apps can decrypt data.
  - e.g.: Messing with MBR or OS kernel will change PCR values.

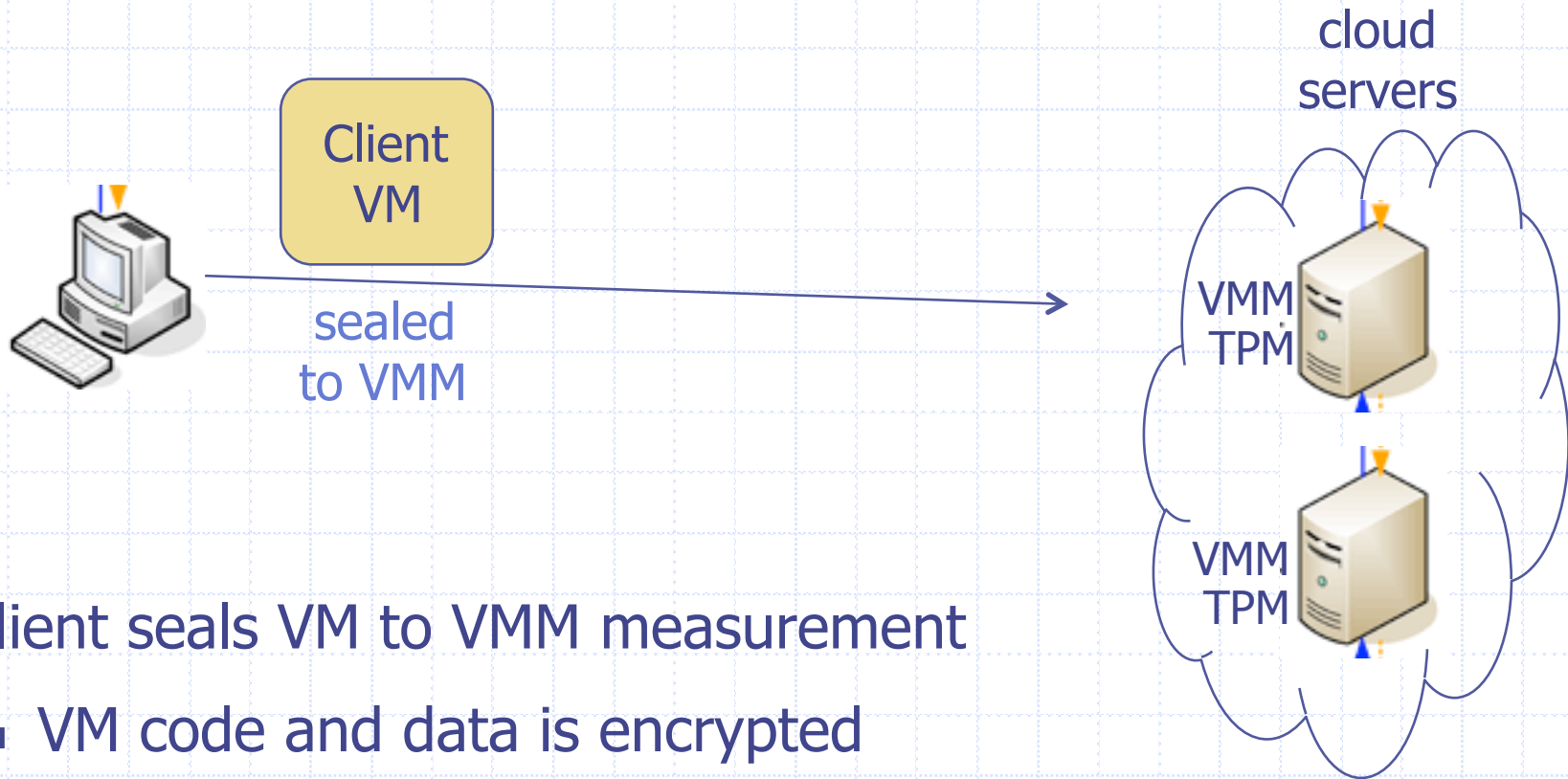
# Sealed storage: applications

- ◆ Lock software on machine:
  - OS and apps sealed with MBR's PCR.
  - Any changes to MBR (to load other OS) will prevent locked software from loading.
  - Prevents tampering and reverse engineering
    - ◆ e.g. software integrity on voting terminals

---

- ◆ Web server: seal server's SSL private key
  - Goal: only unmodified Apache can access SSL key
  - Problem: updates to Apache or Apache config
- ◆ General problem with software patches:
  - Patch process must re-seal all blobs with new PCRs

# A cloud application [JPBM'10]



- ◆ Client seals VM to VMM measurement
  - VM code and data is encrypted
  - Can only be decrypted on valid cloud server
  - Cloud operator cannot easily access data



# Security?

- ◆ Can attacker disable TPM until after boot, then extend PCRs with whatever he wants?
  - Root of trust: BIOS boot block
    - ◆ Defeated with one byte change to boot block [K'07]
- ◆ Resetting TPM after boot (by sending **TPM\_Init** on LPC bus) allows arbitrary values to be loaded onto PCR.
- ◆ Other problems: role-back attack on encrypted blobs
  - e.g. undo security patches without being noticed.
  - Can be mitigated using Data Integrity Regs (DIR)
    - ◆ Need OwnerPassword to write DIR

# Better root of trust

- ◆ DRTM – Dynamic Root of Trust Measurement
  - AMD: **skinit**      Intel: **senter**
  - Atomically does:
    - ◆ Reset CPU.    Reset PCR 17 to 0.
    - ◆ Load given Secure Loader (SL) code into I-cache
    - ◆ Extend PCR 17 with SL
    - ◆ Jump to SL
  
- ◆ BIOS boot loader is no longer root of trust
- ◆ Avoids **TPM\_Init** attack:    TPM\_Init sets PCR 17 to -1

# Vista – BitLocker drive encryption

- ◆ tpm.msc: utility to manage TPM (e.g TakeOwnership)
  - Auto generates 160-bit OwnerPassword
  - Stored on TPM and in file computer\_name.tpm
  
- ◆ Volume Master Key (VMK) encrypts disk volume key
  - VMK is sealed (encrypted) under TPM SRK using
    - ◆ Master Boot Record (MBR) Code (PCR 4),
    - ◆ NTFS Boot Sector (PCR 8),
    - ◆ NTFS Boot Block (PCR 9),
    - ◆ NTFS Boot Manager (PCR 10), and
    - ◆ Volume Key and Critical Components (PCR 11)
  - Note: VMK does not depend on BIOS PCRs

# Vista – BitLocker

- ◆ Many options for VMK recovery
  - Disk, USB, paper (all encrypted with password)
  - Recovery needed after legitimate system change:
    - ◆ Moving disk to a new computer
    - ◆ Replacing system board containing TPM
    - ◆ Clearing TPM
  
- ◆ At system boot (before OS boot)
  - Optional: BIOS requests PIN or USB key from user
  - TPM unseals VMK, if PCR and PIN are correct
    - ◆ TPM defends against dictionary attack on PIN

# TPM Counters

- ◆ TPM must support at least four hardware counters
  - Increment rate: every 5 seconds for 7 years.
- ◆ Applications:
  - Provide time stamps on blobs.
  - Supports “music will pay for 30 days” policy.



# Attestation

# Attestation: what it does

- ◆ **Goal:** prove to remote party what software is running on my machine.
  
- ◆ **Good applications:**
  - Bank allows money transfer only if customer's machine runs "up-to-date" OS patches.
  - Enterprise allows laptop to connect to its network only if laptop runs "authorized" software
  - Quake players can join a Quake network only if their Quake client is unmodified.
  
- ◆ **DRM:**
  - MusicStore sells content for authorized players only.

# Attestation: how it works

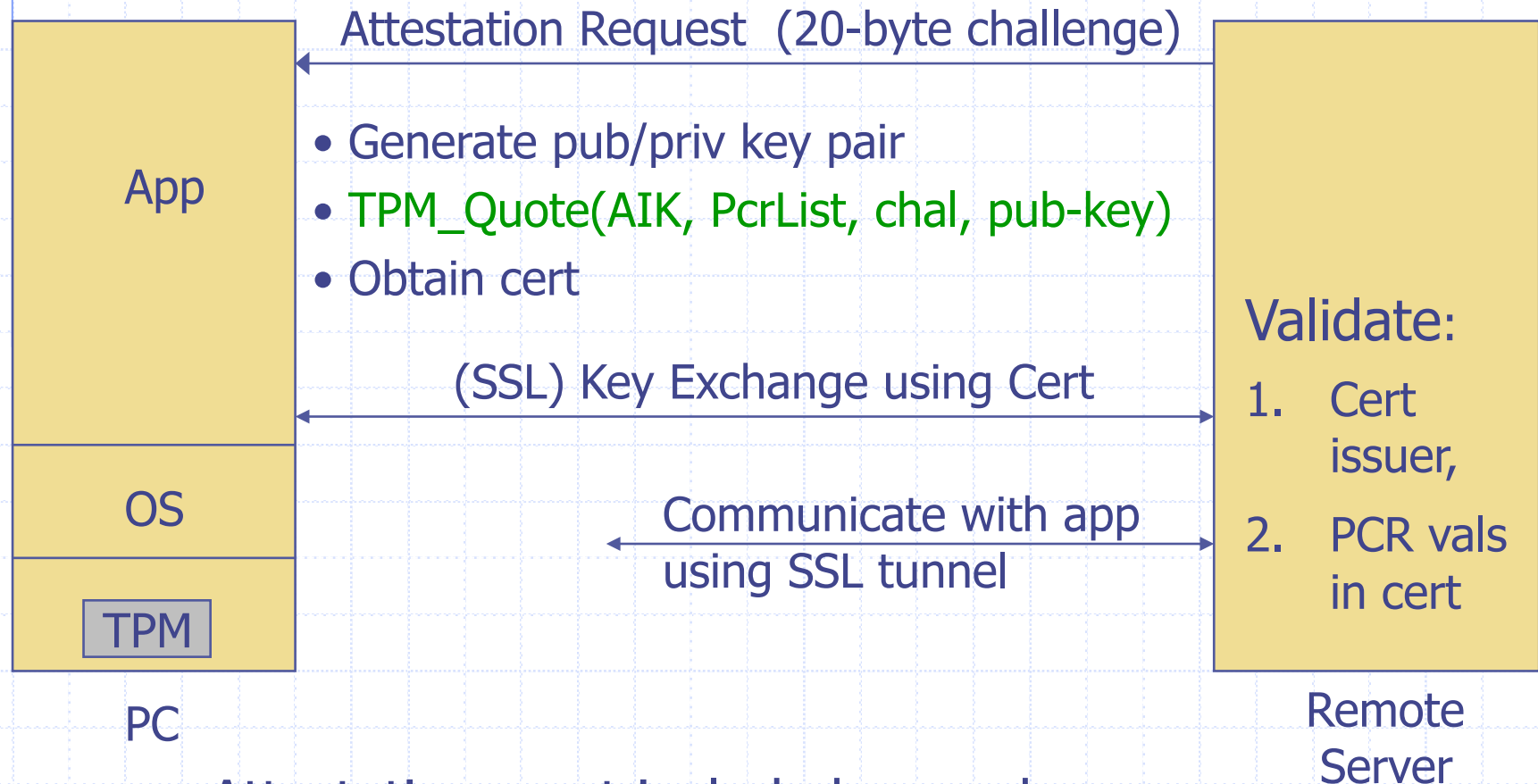
- ◆ Recall: EK private key on TPM.
  - Cert for EK public-key issued by TPM vendor.
- ◆ Step 1: Create Attestation Identity Key (AIK)
  - Details not important here
  - AIK Private key known only to TPM
  - AIK public cert issued only if EK cert is valid



# Attestation: how it works

- ◆ Step 2: sign PCR values (after boot)
  - Call **TPM\_Quote** (some) Arguments:
    - ◆ keyhandle: which AIK key to sign with
    - ◆ KeyAuth: Password for using key `keyhandle`
    - ◆ PCR List: Which PCRs to sign.
    - ◆ Challenge: 20-byte challenge from remote server
      - Prevents replay of old signatures.
    - ◆ Userdata: additional data to include in sig.
  - Returns signed data and signature.

# Attestation: how it (should) work



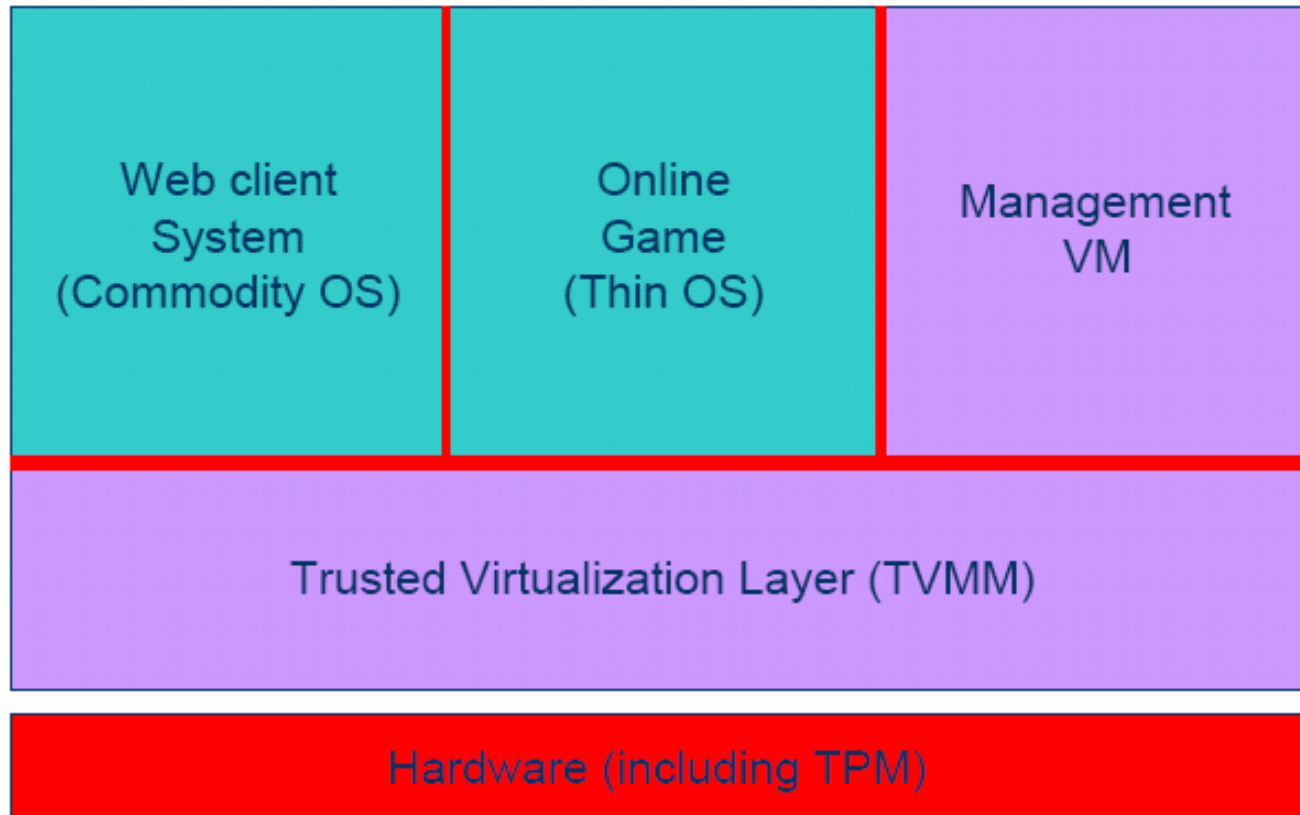
- Attestation must include key-exchange
- App must be isolated from rest of system



# Using Attestation

# Attesting to VMs: Terra

[SOSP'03]



TVMM Provides isolation between attested applications

- application: secure login into a corporate network

# Nexus OS

(Sierer et al. '06)

- ◆ Problem: attesting to hashed application/kernel code
  - Too many possible software configurations
  
- ◆ Better approach: attesting to properties
  - Example: “application never writes to disk”
  
- ◆ Supported in Nexus OS (Sierer et al. '06)
  - General attestation statements:
    - ◆ “TPM says that it booted Nexus,  
Nexus says that it ran checker with hash X,  
checker says that IPD A has property P”

# EFF: Owner Override

- ◆ TCG attestation:
  - **The good:** enables user to prove to remote bank that machine is up-to-date
  - **The bad:** content owners can release decryption key only to machines running “authorized” software.
    - ◆ Stifles innovation in player design
- ◆ EFF: allow users to inject chosen values into PCRs.
  - Enables users to conceal changes to their computing environment
  - Defeats malicious changes to computing platform

# TCG Alternatives

- ◆ IBM 4758: Supports all TCG functionality and more.
  - Tamper resistant 486 100MhZ PCI co-processor.
  - Programmable.
  - ... but expensive ~ \$2000. TPM ~ \$7.
- ◆ AEGIS System: Arbaugh, Farber, Smith '97:
  - Secure boot with BIOS changes only.
  - Cannot support sealed storage.
  - **Phoenix TrustConnector 2**
- ◆ SWATT: Seshadri et al., 2004
  - Attestation w/o extra hardware
  - Server must know precise HW configuration



# Attestation: challenges



# 1. Attesting to Current State

- ◆ Attestation only attests to what code was loaded.
- ◆ Does not say whether running code has been compromised.
  - Problem: what if Quake vulnerability exploited after attestation took place?
- ◆ Can we attest to the current state of a running system?
  - ... or is there a better way?

## 2. Encrypted viruses

- ◆ Suppose malicious music file exploits bug in Windows Media Player.
  - Music file is encrypted.
  - TCG prevents anyone from getting music file in the clear.
  - Can anti-virus companies block virus without ever seeing its code in the clear?

# 3. TPM Compromise

- ◆ Suppose one TPM Endorsement Private Key is exposed
  - Destroys all attestation infrastructure:
    - ◆ Embed private EK in TPM emulator.
    - ◆ Now, can attest to anything without running it.
  
- ⇒ Certificate Revocation is critical for TCG Attestation.

# 4. Private attestation

- ◆ Attestation should not reveal platform ID.
  - Recall Intel CPU-ID fiasco.
- ◆ Private attestation:
  - Remote server can validate trustworthiness of attestation
  - ... but cannot tell what machine it came from.
- ◆ TCG Solutions:
  - Privacy CA: online trusted party
  - Group sigs: privacy without trusted infrastructure



**THE END**