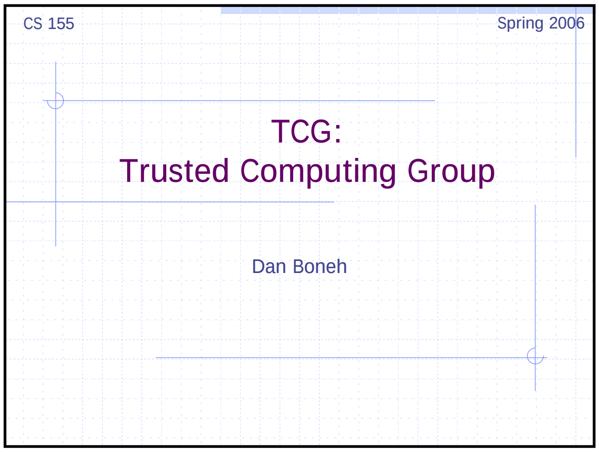
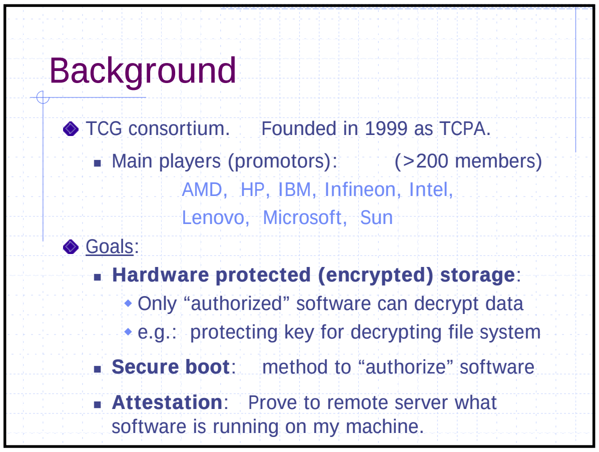


CS 155 Spring 2006

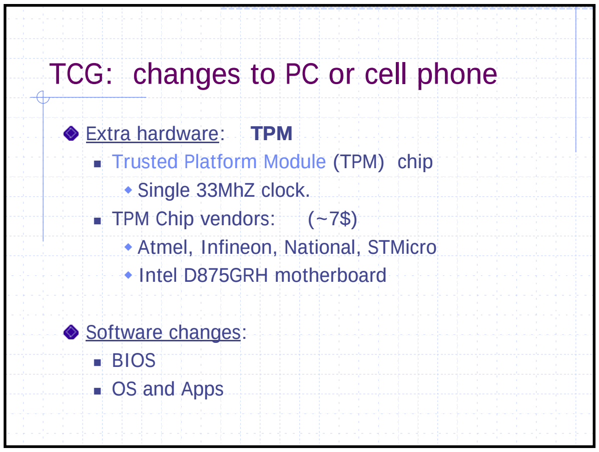
TCG: Trusted Computing Group

Dan Boneh



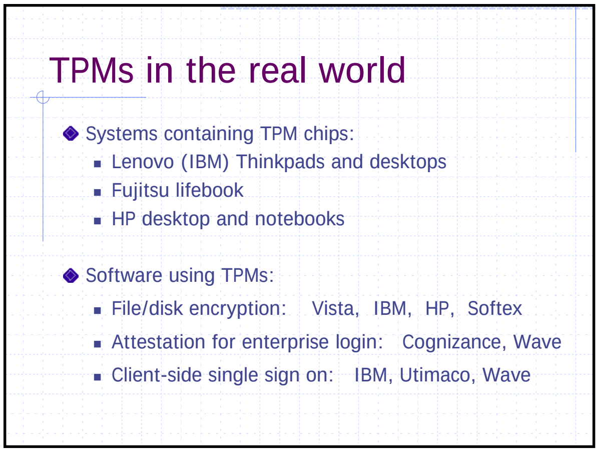
Background

- TCG consortium. Founded in 1999 as TCPA.
  - Main players (promotors): (>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.



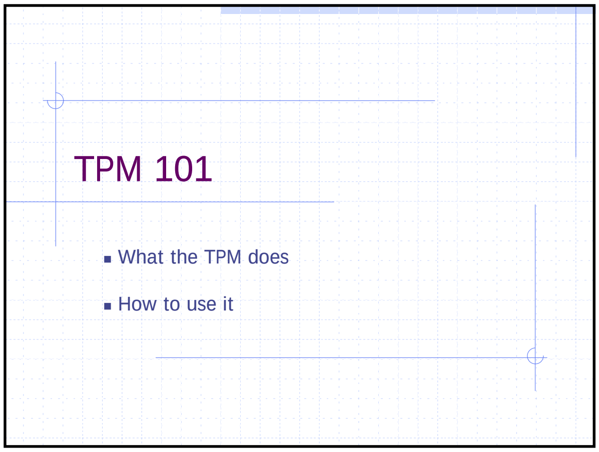
TCG: changes to PC or cell phone

- Extra hardware: **TPM**
  - Trusted Platform Module (TPM) chip
    - Single 33MhZ clock.
  - TPM Chip vendors: (~7$)
    - Atmel, Infineon, National, STMicro
    - Intel D875GRH motherboard
- Software changes:
  - BIOS
  - OS and Apps



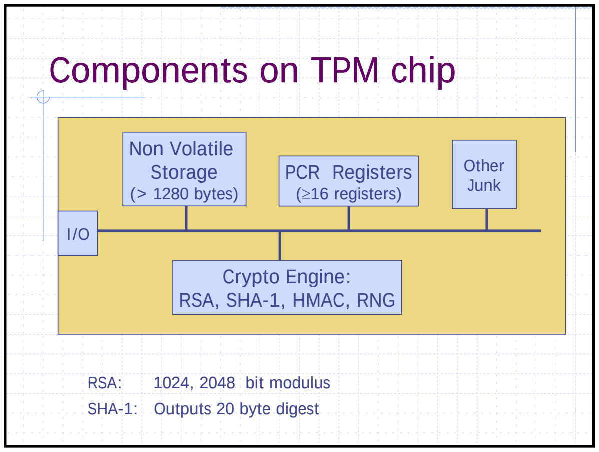
TPMs in the real world

- Systems containing TPM chips:
  - Lenovo (IBM) Thinkpads and desktops
  - Fujitsu lifebook
  - HP desktop and notebooks
- 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

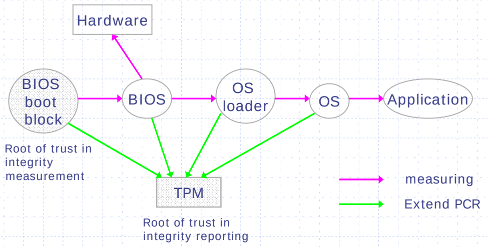
PCR: the heart of the matter

- *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): PCR[n] ← SHA-1 (PCR[n] || 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

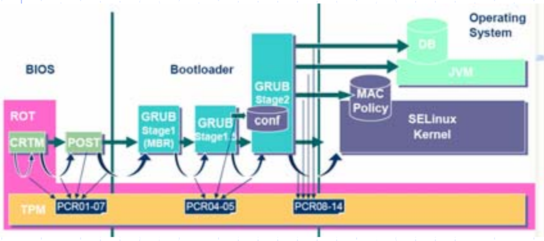
- At power-up PCR[n] initialized to 0
- BIOS boot block executes
  - Calls PCR_Extend(n, <BIOS code>)
  - Then loads and runs BIOS post boot code
- BIOS executes:
  - Calls PCR_Extend(n, <MBR code>)
  - Then runs MBR (master boot record), e.g. GRUB.
- MBR executes:
  - Calls PCR_Extend(n, <OS loader code, config>)
  - Then runs OS loader ... and so on

In a diagram


Hardware
BIOS boot block
BIOS
OS loader
OS
Application
TPM
Root of trust in integrity measurement
Root of trust in integrity reporting
measuring
Extend PCR

- 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:
    - Ownership Enabled flag ← False
  - Done once by IT department or laptop owner.
- (optional) Step 2: TPM_CreateWrapKey
  - Create more RSA keys on TPM certified 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 othrwise

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.
- Why can't attacker disable TPM until after boot, then extend PCRs with whatever he wants?
  - Root of trust: BIOS boot block.
- Gaping hole: role-back attack on encrypted blobs
  - e.g. undo security patches without being noticed.
  - Can be mitigated using Data Integrity Regs (DIR)

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 reverse-engineering
- Web server: seal server's SSL private key
  - Goal: only unmodified Apache can access SSL key
  - Problem: updates to Apache, config, or content
- General problem with software patches:
  - When updating MBR, must re-seal blobs
  - Not a simple process ...

TPM Counters

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

Non-volatile TPM memory

- Stores:
  - Storage Root Key (SRK)
  - Owner Password

    } Generated when user takes ownership

  - Endorsement Key (EK)
    - Created once for the life of the TPM
    - Certificate for EK issued by TPM vendor
    - Basis of attestation
  - Persistent flags (e.g. ownership flag)

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.

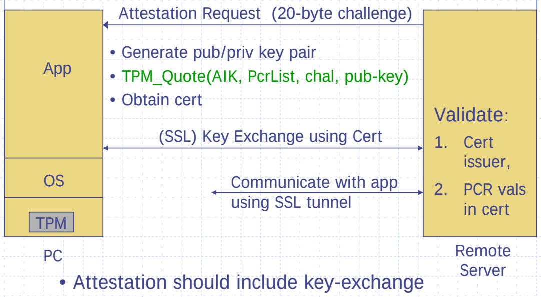
Attestation: how it works

- 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.
  - 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.

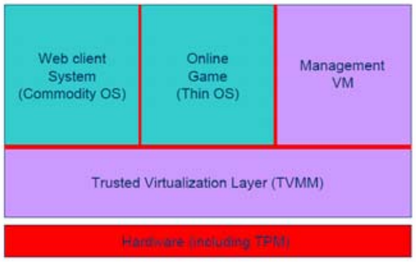
Attestation: how it (should) work


Attestation Request (20-byte challenge)
App
• Generate pub/priv key pair
• TPM_Quote(AIK, PcrList, chal, pub-key)
• Obtain cert
(SSL) Key Exchange using Cert
OS
Communicate with app using SSL tunnel
TPM
PC
Validate:
1. Cert issuer,
2. PCR vals in cert
Remote Server

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

Using Attestation

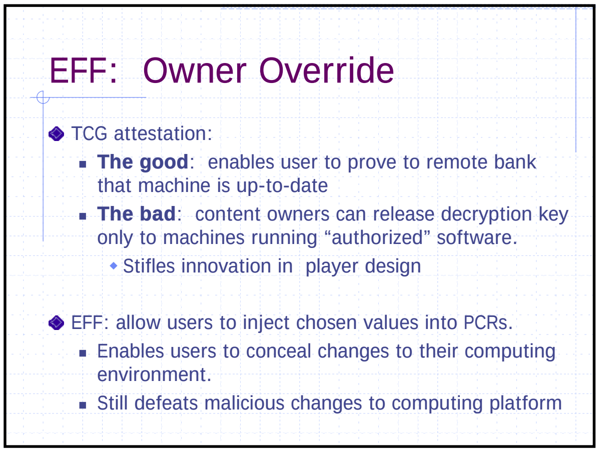
Attesting to VMs: Terra [SOSP'03]


Web client System (Commodity OS)
Online Game (Thin OS)
Management VM
Trusted Virtualization Layer (TVMM)
Hardware (including TPM)

TVMM Provides isolation between attested applications

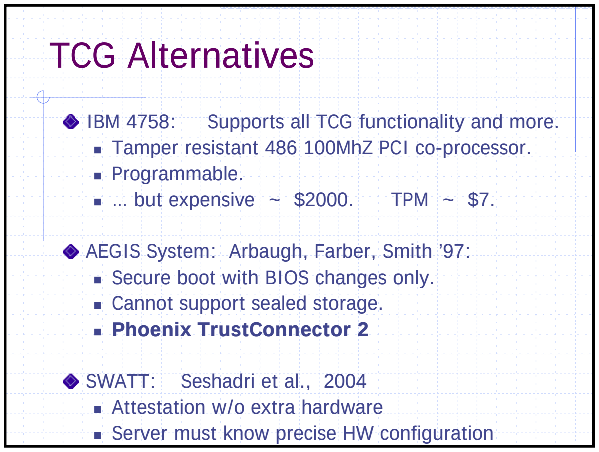
Nexus OS (Sirer et al. '06)

- 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

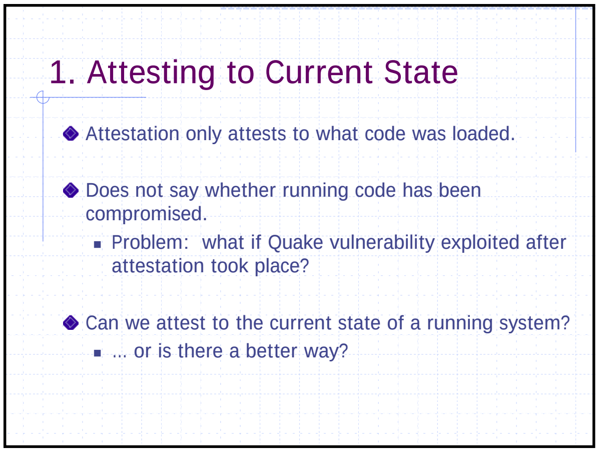
- TCG attestation:
  - **The good**: enables user to prove to remote bank that machine is up-to-date
  - **The bad**: content owners can release decryption key only to machines running “authorized” software.
    - Stifles innovation in player design
- EFF: allow users to inject chosen values into PCRs.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



TCG Alternatives

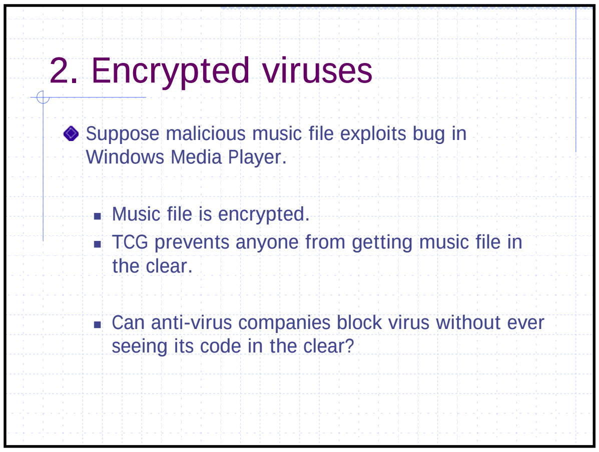
- IBM 4758: Supports all TCG functionality and more.
  - Tamper resistant 486 100MhZ PCI co-processor.
  - Programmable.
  - ... but expensive ~ $2000. TPM ~ $7.
- AEGIS System: Arbaugh, Farber, Smith ’97:
  - Secure boot with BIOS changes only.
  - Cannot support sealed storage.
  - **Phoenix TrustConnector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

Attestation: challenges



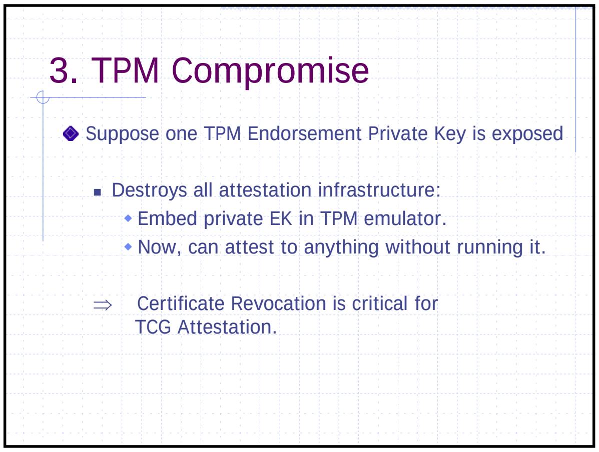
1. Attesting to Current State

- 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

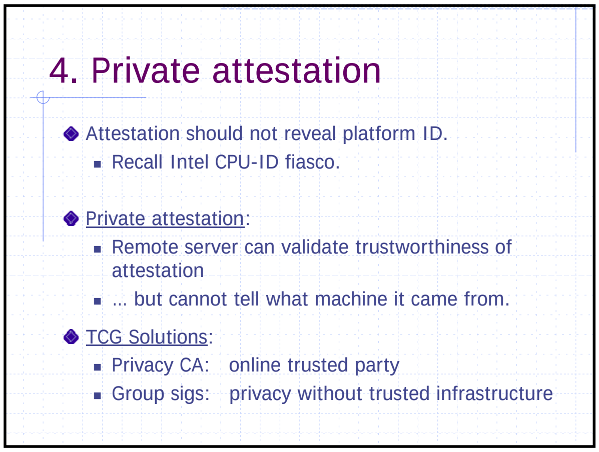
- Suppose malicious music file exploits bug in Windows Media Player.
  - Music file is encrypted.
  - TCG prevents anyone from getting music file in the clear.
  - Can anti-virus companies block virus without ever seeing its code in the clear?



3. TPM Compromise

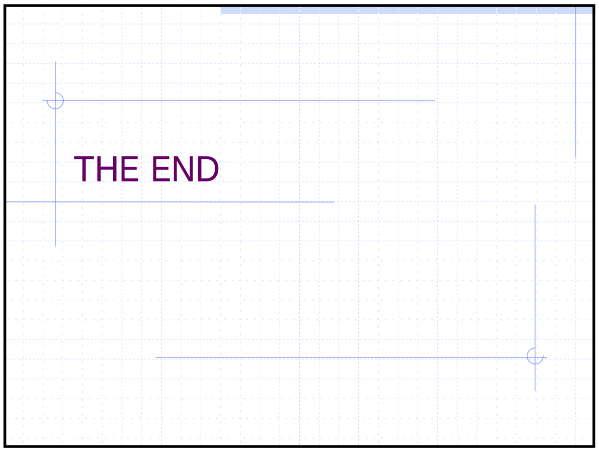
- Suppose one TPM Endorsement Private Key is exposed
  - Destroys all attestation infrastructure:
    - Embed private EK in TPM emulator.
    - Now, can attest to anything without running it.

⇒ 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