INF3510 Information Security
University of Oslo
Spring 2010

Lecture 10

Computer Security and Trusted Systems

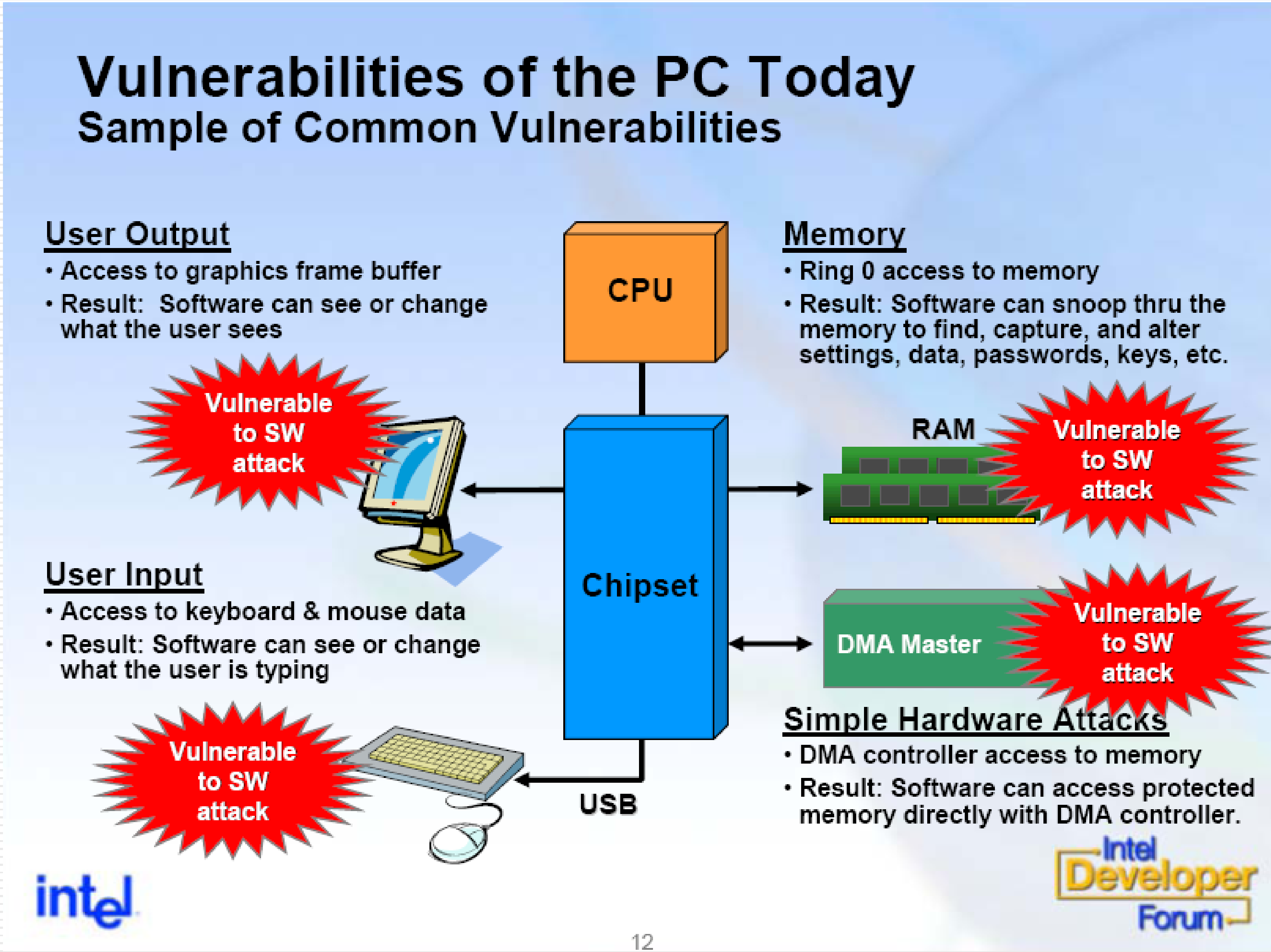
Audun Jøsang

Lecture Overview

- Secure computer architectures
- Trusted computing - background motivation and history
- Trusted Hardware
- Trusted Computing Group (TCG) overview
- Trusted Platform Module (TPM) overview
- Basic trusted platform functions
- TCG issues and challenges

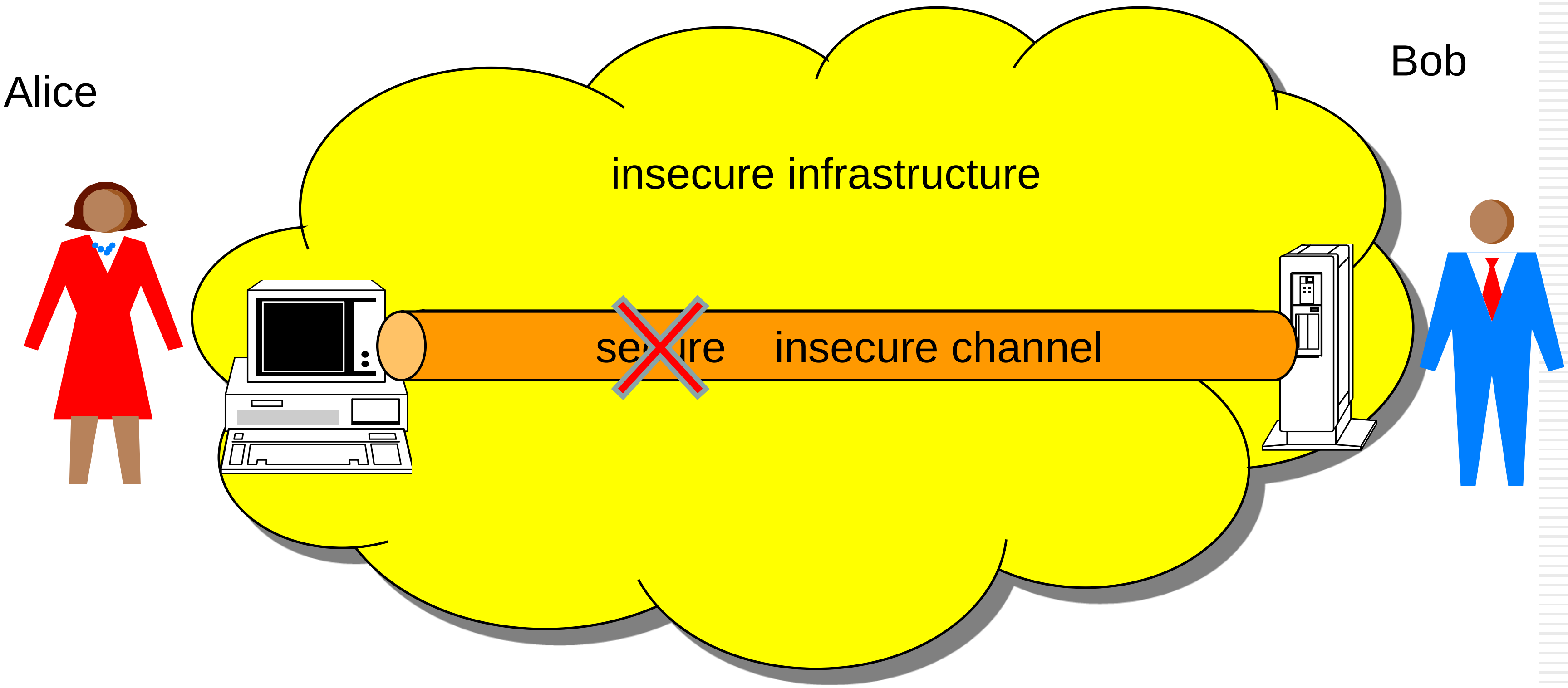
Background

- Increasing reliance on networked computing in commerce and critical infrastructures
- Systems are vulnerable to fraud, vandalism, targeted subversion
- Systems can’t be trusted to operate as expected
- Threats:
  - Subversion via network attacks and mobile code
  - Denial of service
  - ‘Insider’ attacks
  - Application models where the user is motivated to subvert their own device (DRM, software license enforcement, online gaming, electronic cash)



Source: "LaGrande Architecture" presentation by David Grawrock, delivered at Intel Developer Forum, September 2003. http://www.intel.com/idf/us/fall2003/presentations/F03USSCMS18_OS.pdf

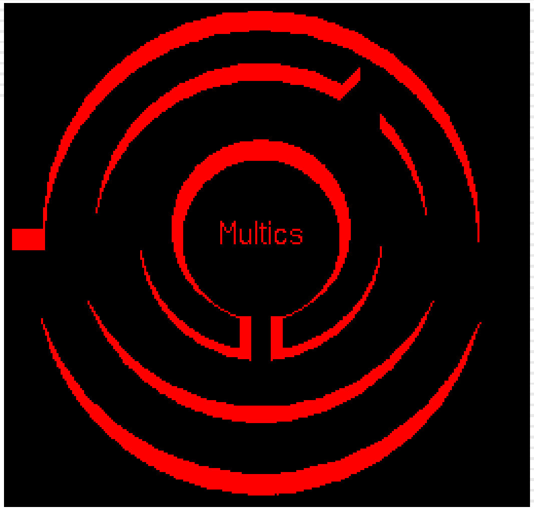
Assumptions and Reality



Approaches to Computer Security

- Harden the commodity platform OS
  - SELinux, Solaris 11
- Add secure hardware to the commodity platform
  - Internal & integrated
- Give up on the commodity platform (?)
- Rely on secure hardware external to the commodity platform
  - Smart cards
  - Hardware tokens
  - Special security hardware in commodity platforms

Some history: Multics



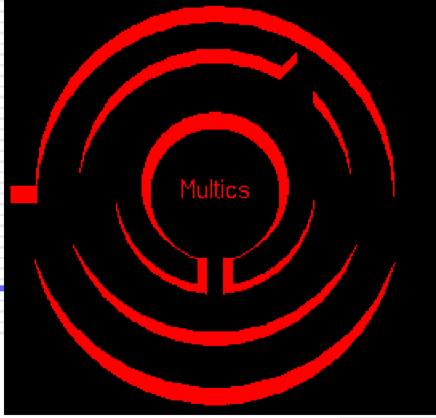
- Operating System
  - Designed 1964-1967
    - MIT Project MAC, Bell Labs, GE
  - Introduced timesharing
  - At peak, ~100 Multics sites
  - Last system, Canadian Department of Defense, Nova Scotia, shut down October, 2000
- Extensive Security Mechanisms
  - Influenced many subsequent systems

http://www.multicians.org/security.html

Multics Innovations

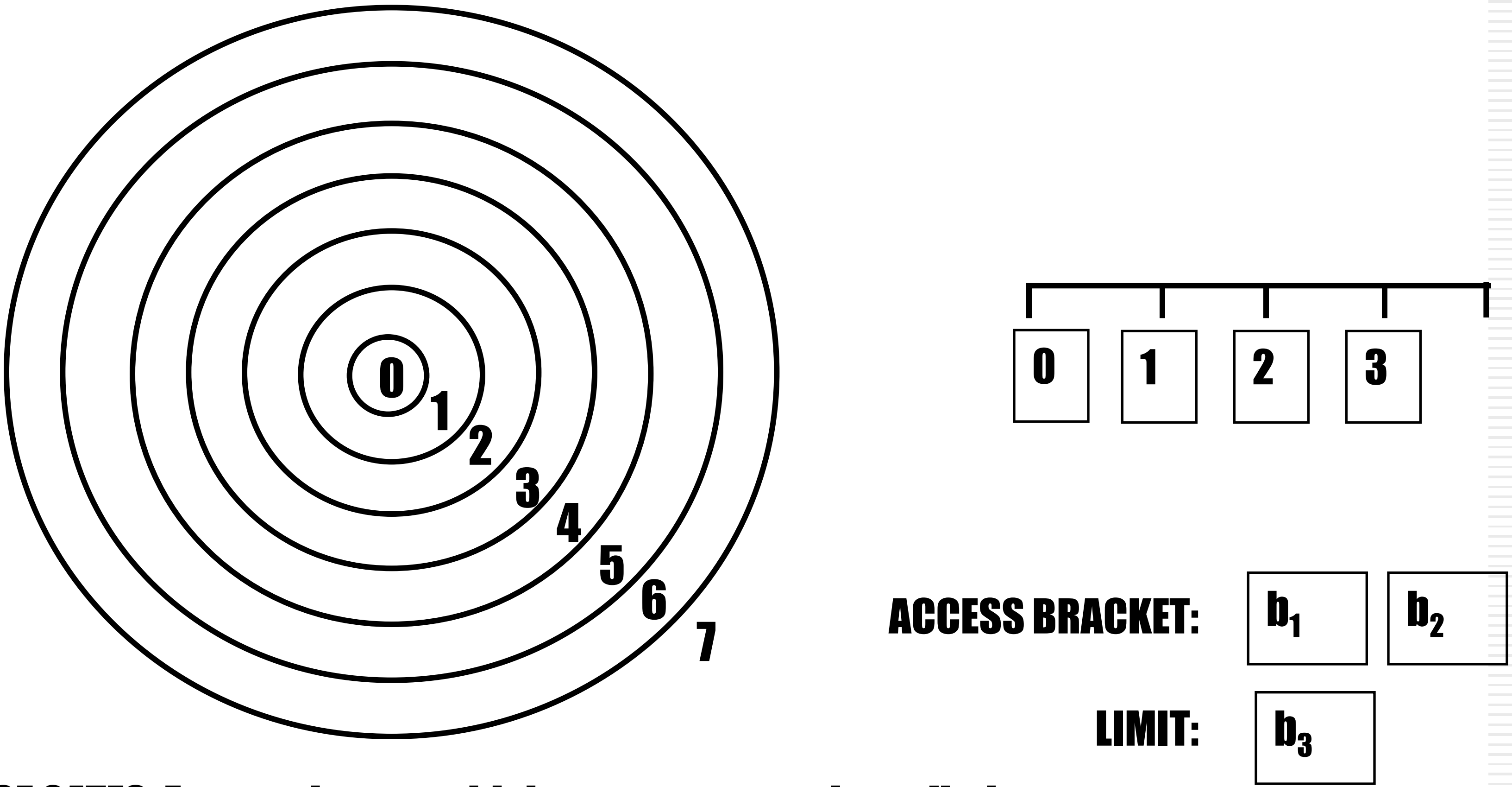
- Segmented, Virtual memory
  - Hardware translates virtual address to real address
- High-level language implementation
  - Written in PL/1, only small part in assembly lang
- Shared memory multiprocessor
  - Multiple CPUs share same physical memory
- Relational database
  - Multics Relational Data Store (MRDS) in 1978
- Security
  - Designed to be secure from the beginning
  - First B2 security rating (1980s), the only one for years

Multics Access Model



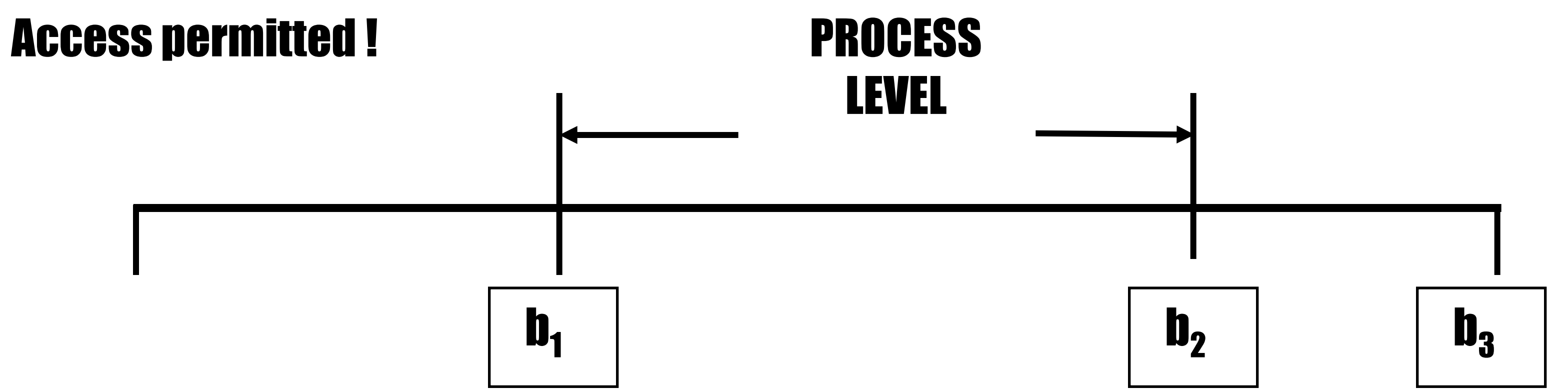
- Ring structure
  - A ring is a domain in which a process executes
  - Numbered 0,1, …7 ; Kernel is in ring 0
  - Graduated privileges
    - Processes at ring i have privileges of every ring $j > i$
- Segments
  - Each data area or procedure is called a segment
  - Segment protection $\langle b1, b2, b3 \rangle$ with $b1 \leq b2 \leq b3$
    - Process/data can be accessed from rings b1 … b2
    - A process from rings b2 … b3 can only call segment at restricted entry points

MULTICS PROTECTION RINGS



LIST OF GATES: Entry points, at which segments may be called

SEGMENT ACCESS BRACKET



- If a process executing in ring i tries to execute a segment with access bracket (b1, b2), then the call is allowed if b1 <= i <= b2, and the current ring number of the process remained i. Otherwise, a trap to the kernel occurrs.

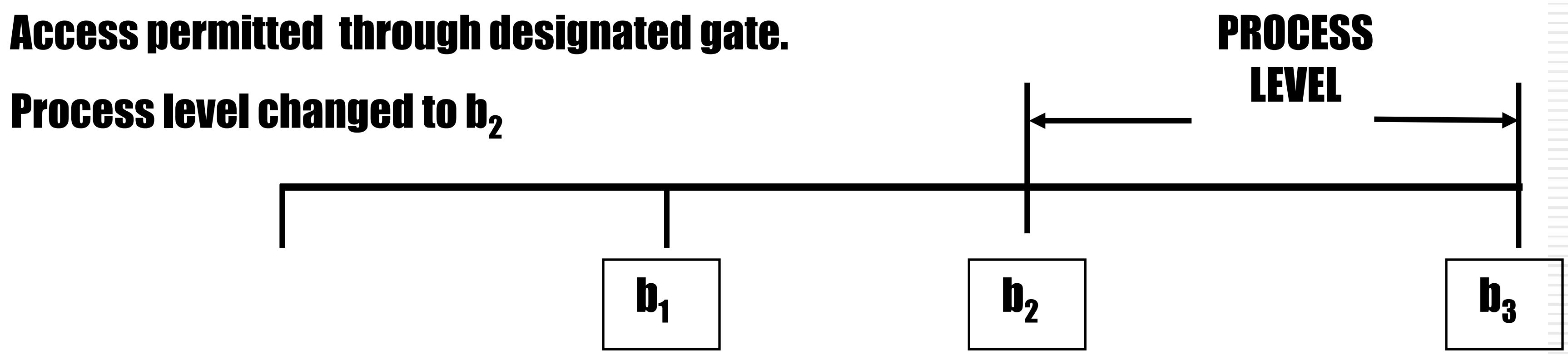
SEGMENT ACCESS BRACKET

- If i <= b1, then the call is allowed to occur and the current-ring-no of the process is changed to b1. Thus the access rights of the process are reduced. If parameters are passed which refer to segments in a ring lower than b1, then these segments were copied into an area accessible in ring b1.

EXECUTE ACCESS

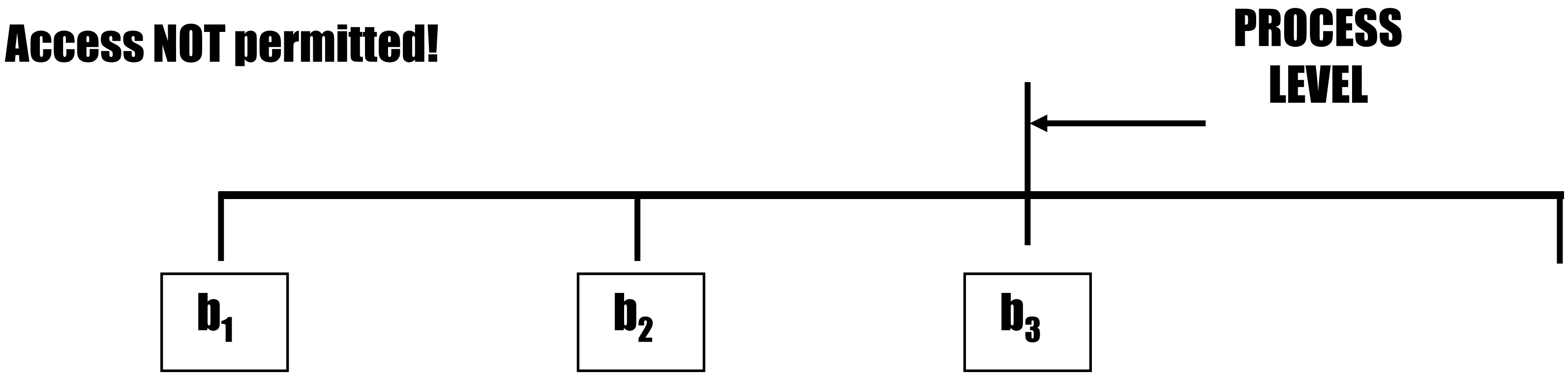
- It would be dangerous if untrusted code were allowed to execute highly privileged processes.
- A list of gates representing entry points at which segments may be called is included with the access bracket.

SEGMENT ACCESS BRACKET



- If i > b2, then the call is allowed to occur only if i <= b3, and the call is directed to one of the designated entry points in the list-of-gates. If successful, the current-ring-nr of the process is changed to b2. This scheme allowed processes with limited access rights to call procedures in lower rings, but only in a carefully controlled manner.

SEGMENT ACCESS BRACKET



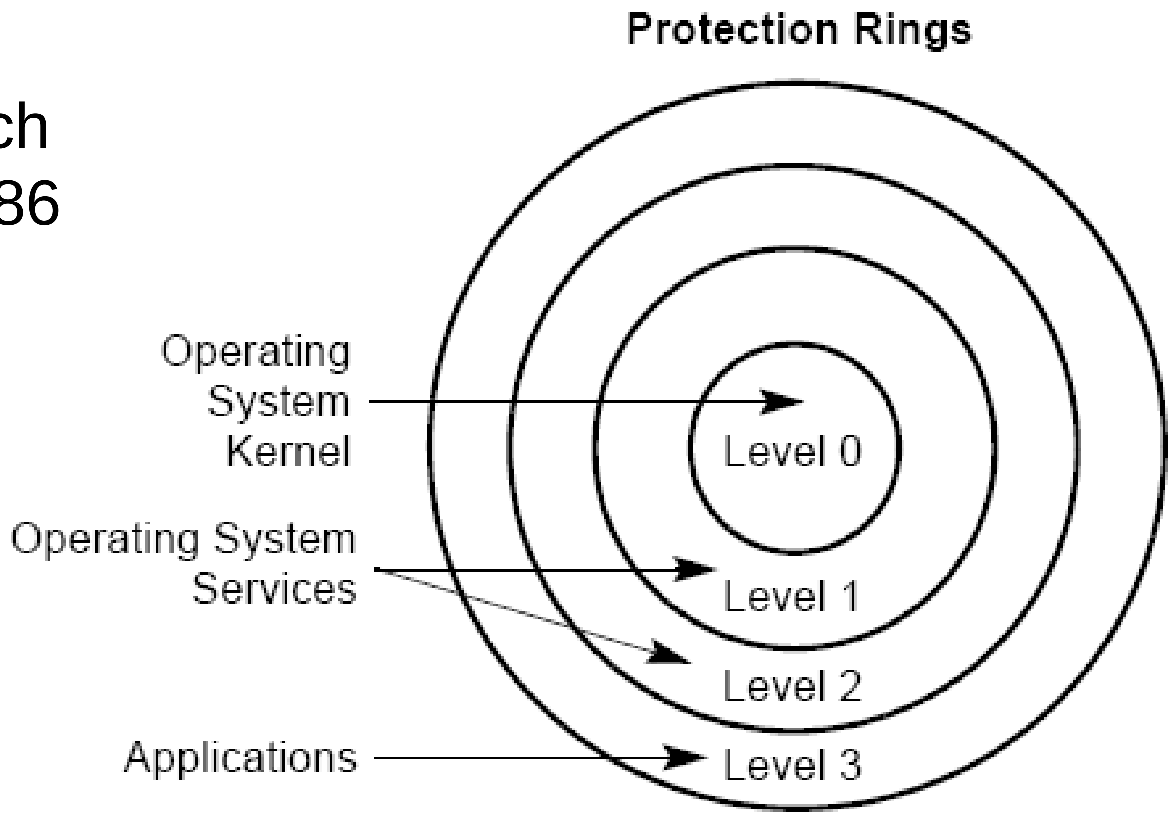
- If i > b3 (the limit) no access is permitted.

MODERN SYSTEMS

- Intel x86 processors have 4 privilege levels with the intended use as follows
  - Ring 0: kernel
  - Ring 1 & 2: device drivers
  - Ring 3: applications
- Documentation:
  - Intel64 and IA-32 Architectures Software Developer's Manual, Volume 3A, Chapter 4
  - http://www.intel.com/design/processor/manuals/253668.pdf

Intel Memory Protection Rings

- Originally in Multics
- In Intel arch since 80386



Privilege Levels

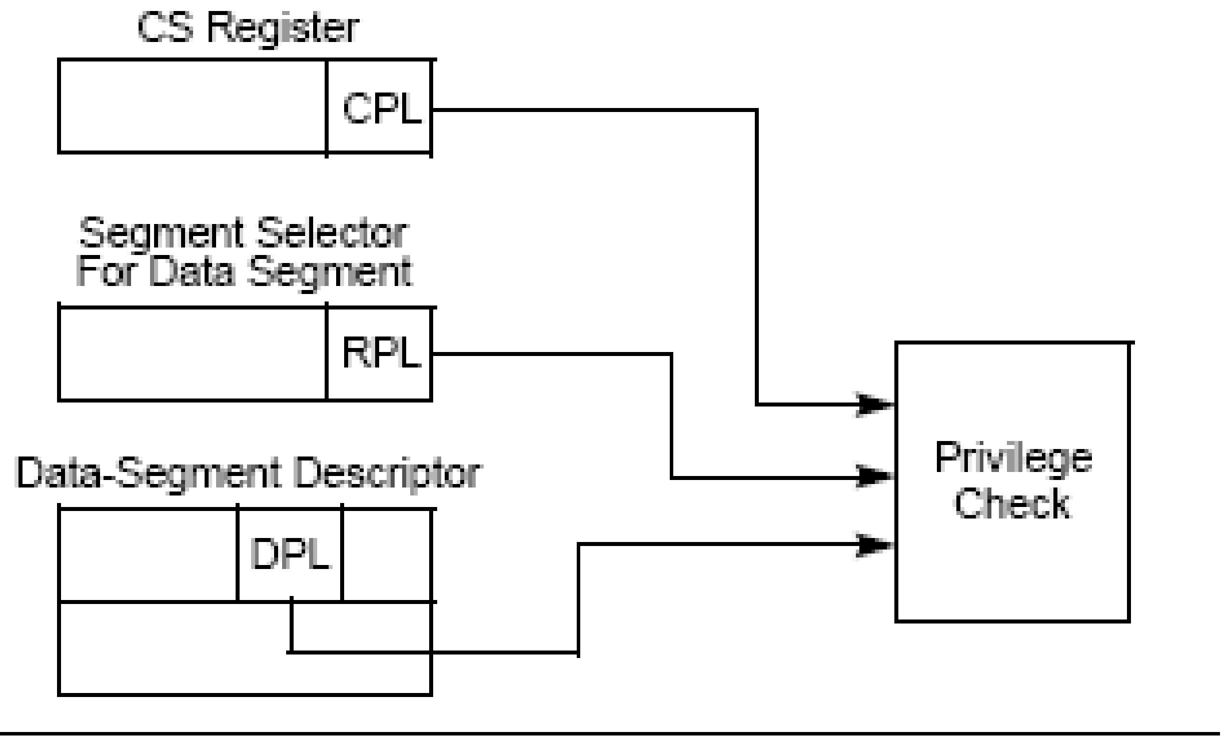
- CPU enforces constraints on memory access and changes of control between different privilege levels
- Similar in spirit to Bell-LaPadula access control restrictions
- Hardware enforcement of division between user mode and kernel mode in operating systems
  - Simple malicious code cannot jump into kernel space

Data Access Rules

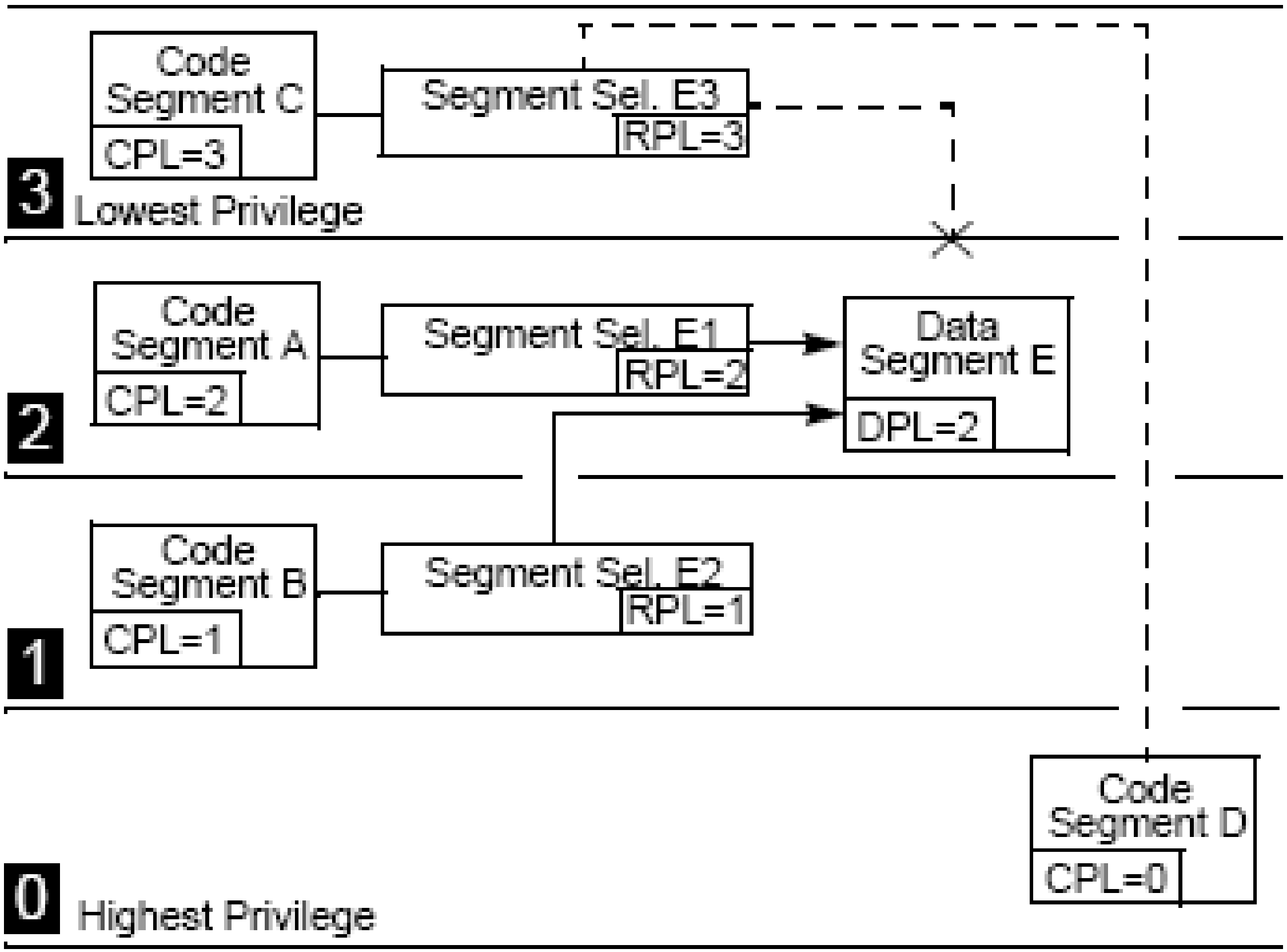
- Three players
  - Code segment has a current privilege level CPL
  - Segment selector has a requested privilege level RPL
  - Data Segment Descriptor for each memory includes a data privilege level DPL
- Segment is loaded if CPL <= DPL and RPL <= DPL
  - i.e. both CPL and RPL are from more privileged rings

Data Access Rules

- Access allowed if
  - CPL <= DPL and RPL <= DPL



Data Access Examples



Comments to example

- Four procedures (located in codes segments A, B, C, and D), each running at different privilege levels and each attempting to access the same data segment.

a) The procedure in code segment A is able to access data segment E using segment selector E1, because the CPL of code segment A and the RPL of segment selector E1 are equal to the DPL of data segment E.

b) The procedure in code segment B is able to access data segment E using segment selector E2, because the CPL of code segment B and the RPL of segment selector E2 are both numerically lower than (more privileged) than the DPL of data segment E. A code segment B procedure can also access data segment E using segment selector E1.

Comments to example

c) The procedure in code segment C is not able to access data segment E using segment selector E3 (dotted line), because the CPL of code segment C and the RPL of segment selector E3 are both numerically greater than (less privileged) than the DPL of data segment E. Even if a code segment C procedure were to use segment selector E1 or E2, such that the RPL would be acceptable, it still could not access data segment E because its CPL is not privileged enough.

d) The procedure in code segment D should be able to access data segment E because code segment D's CPL is smaller than the DPL of data segment E. However, the RPL of segment selector E3 (which the code segment D procedure uses to access data segment E) is greater than the DPL of data segment E, so access is not allowed. If the code segment D procedure were to use segment selector E1 or E2 to access the data segment, access would be allowed.

WHAT ABOUT DRIVERS

- *Can somebody please tell me why a fault in my sound card driver has to crash my system?*
- Answer: MS Windows only uses ring 0 and 3.
  - Windows 98 had device drivers in ring 3. Execution became very slow.
  - From Windows 2000, they are in ring 0, for performance reasons

Limiting Memory Access Type

- The Pentium architecture supports making pages read/only versus read/write
- A recent development is the Execute Disable Bit
  - Added in 2001 but only available in systems recently
  - Supported by Windows XP SP2 an later MSWindows
- Similar functionality in AMD Altheon 64
  - Called Enhanced Virus Protection

Trusted Computing Motivation

- Computer Security
  - Well established since 1960s
- Trusted Computing Base (TCB)
  - The totality of protection mechanisms within a computer system, including hardware, firmware and software
  - Concept developed during 1980s
- Physical access to computers open up for attacks that can circumvent traditional TCBs, e.g. secure operating systems
- Complexity of contemporary systems makes it impossible to remove all software vulnerabilities

Basic idea of Trusted Computing

- Addition of security hardware functionality to a computer system
- Enables external entities to have increased level of trust that the system will perform as expected/specified

Related Concept: Trusted Platform

- Trusted platform = a computing platform with a secure hardware component that forms a security foundation for software processes
- Trusted Computing = computing on a Trusted Platform

Motivation for Trusted Hardware

- Trusted Computing is not a new idea!
- Tygar and Yee: A System for Using Physically Secure Coprocessors 1991
  - Cryptography assumes the secrecy of keys
  - Secrecy requires physical security
  - All security algorithms and protocols rely on physical security

(J. D. Tygar and B. Yee. A System for Using Physically Secure Coprocessors, *Technical Report CMU-CS-91-140R*, Carnegie Mellon University, May 1991)

Motivation for Trusted Hardware 2

- Computing platforms are deployed in hostile environments , in contrast to 1960’s 1970’s protected computing centres
  - There is a gap between the reality of physically unprotected, network connected systems and the assumption of confidentiality and integrity
  - The gap must be closed if systems are to be trustworthy

What is “trust” in the sense of TC?

- To have faith or confidence that something desired is, or will be, the case
- Trust engenders confident expectations
- Trust allows us to believe assertions

“*A trusted component, operation, or process is one whose behaviour is predictable under almost any operating condition and which is highly resistant to subversion by application software, viruses, and a given level of physical interference*”

- A ‘trusted’ component can violate the security policy if it breaks
- A ‘trustworthy’ component can be relied on to enforce the security policy, because it doesn’t break
- A ‘trusted system’ can be <u>verified</u> to enforce a given security policy
- The big question: “Trusted by whom to do what?”

ISO/IEC. Evaluation criteria for information technology, 1999. Standard ISO/IEC 15408.

Trusted by whom to do what?

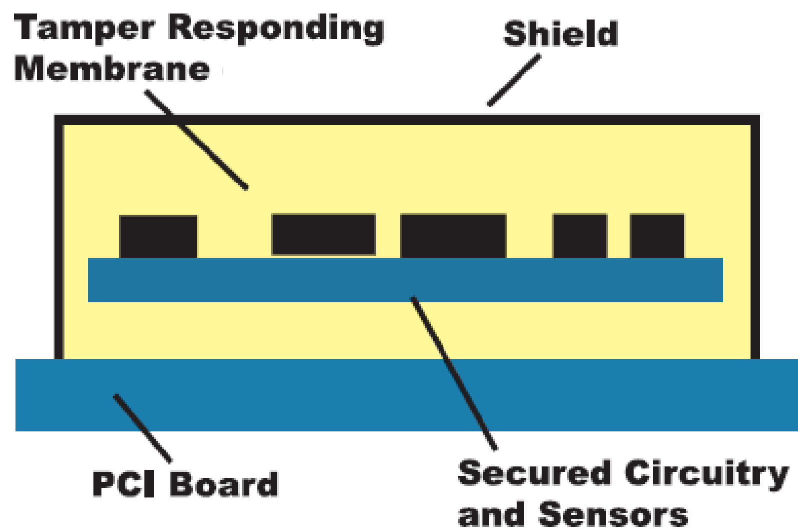
- Has the OS been subverted?
  - Virus/Trojan/Spyware/Rootkit
  - Keystroke/screen/mouse logger
  - Smart card reader, biometric reader access
- How would the user know?
- How would a program on another computer know?

Characteristics of Trusted Hardware

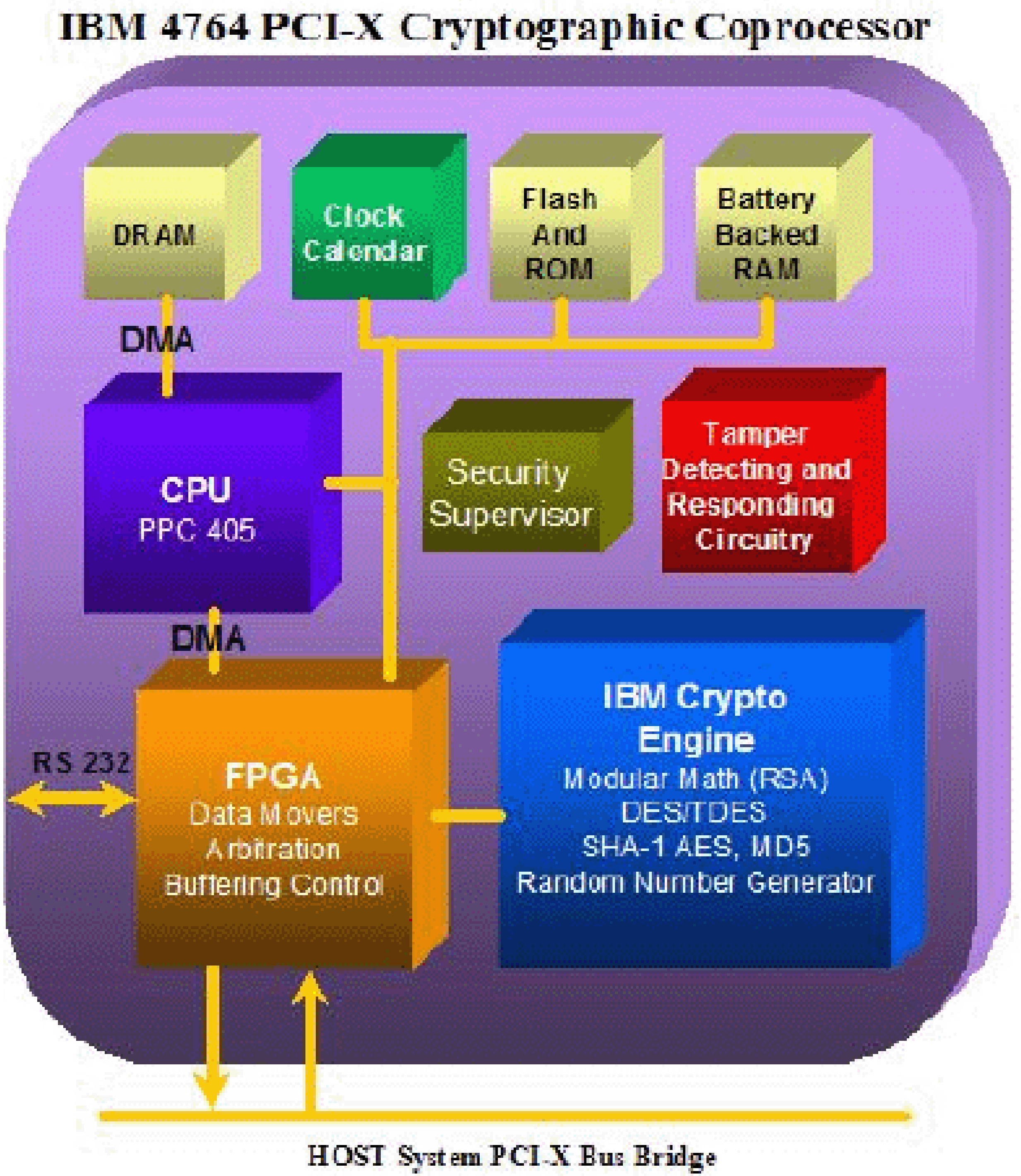
- Physically secure module
- Environmental monitoring (temperature, power supply, structural integrity)
- Tamper responsive
- CPU
- ROM for OS and application code
- NVRAM (Flash), EEPROM, BBRAM for secrets and data (zeroisation)
- Optimized hardware support for cryptography
- I/O interface

Trusted Hardware – Example

- IBM 4764 Secure Coprocessor



IBM 4764 Architecture



IBM 4764 Security Functionality

- Performs symmetric and public-key cryptography in a highly secure environment
- Supports loading of software for highly sensitive processing, even when under the physical control of a motivated adversary.
- Secure envelope around the electronics to detect penetration attempts
- Will zeroize critical secret memory area when tampering is detected.

IBM 4764 Security Algorithms

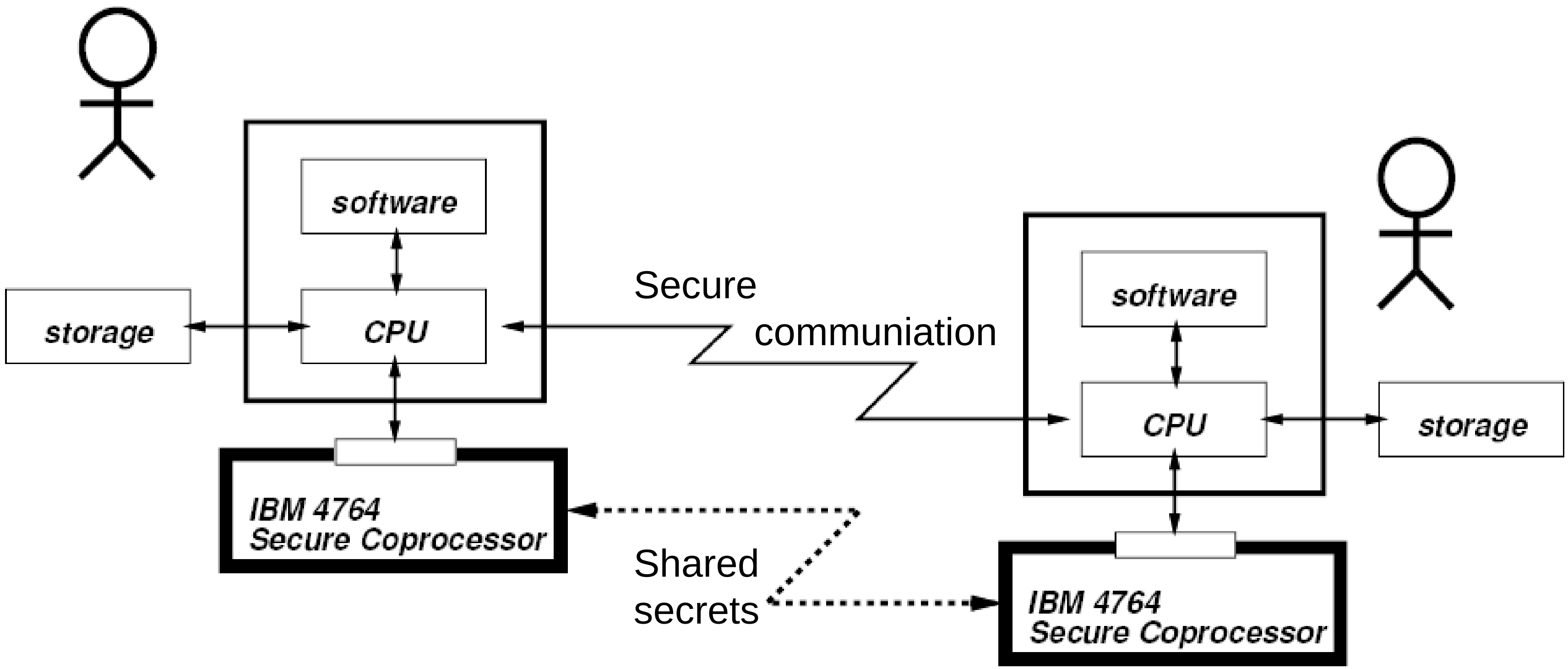
Includes these NIST-approved algorithms:

- AES (Advanced Encryption Standard)
- TDES (Triple DES)
- DES (Simple DES, for compatibility with legacy applications)
- DSS (Digital Signature Standard)
- SHA-1 (Secure Hash Algorithm)
- Software DRNG (Deterministic Random Number Generator)
- Any combination of encryption/decryption and ECB/CBC

Includes these *non-approved* algorithms:

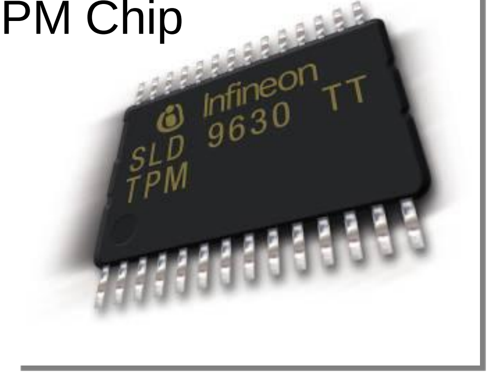
- MD5 (Message Digest 5 Hash Algorithm, which is broken)
- RSA (for signing/signature verification)
- ISO9796 padding for public-key signatures
- Hardware random number generation

IBM 4764 Application Example



Trusted Hardware Examples

TPM Chip



iButton

IBM 4764

Fortezza PC Card

Smart Card

OS Boot Integrity Protection

- System trust relies on boot integrity
  - Which OS has been loaded?
  - Has the OS been modified?
  - Integrity of system layer *n* dependent on system layers < *n*
- System components are hashed (fingerprinted) by trusted hardware before they are loaded or executed
- Reference values of trusted component hashes are stored in NV protected memory
- Runtime measurements are compared with stored reference measurements
- System boots only if runtime measurements match reference measurements

J D Tygar and B Yee, A System for Using Physically Secure Coprocessors, Technical Report CMU-CS-91-140R, Carnegie Mellon University, May 1991.

Software Protection and DRM

- **Context**: an entity owns a right in a work
  - The work is represented as a string of bits e.g.
    - Software
    - MP3, MP4, AVI, PDF, MPEG,WAV
- **Aim**: provide assurance that a usage policy defined by the rights owner will be followed when the *string of bits* is on a computing platform not under rights owner's control
- Requires trust in the hardware/software environment
  - Will it reliably enforce the usage policy?

Software Protection and DRM 2

- Cryptography can protect digital content when it is stored or transmitted
- Digital content bits must be *in the clear* to be rendered in a perceptible manner
- Robust DRM assumes that these plaintext bits can be protected from access by the rendering platform owner/administrator
  - An access control problem
  - Difficult for open computing platforms (PC) where untrusted owner/administrator has control

Software Protection and DRM 3

- DRM applications require complete trust in environment that manipulates plaintext or keys
  - Rights owner must be able to trust remote OS
  - Very hard problem on an open platform that can run arbitrary software: requires robust domain separation
  - Kernel debugger or malicious device driver can access memory hence plaintext
  - Direct Memory Access (DMA) also a big problem

Software Protection and DRM 4

- Current solutions based on closed devices
- DRM on open devices requires a reliable way of reporting the current software config
- If rights owner trusts this config, keys to decrypt bits can be released by trusted hardware
- System must revoke “trusted” status if software environment changes
- Requires trusted OS
- TCG provides some building blocks for this but does not address how revocation of trusted status should be achieved

Trusted Computing to DRM

- Content owner needs assurance of mandatory access policy enforcement on devices they do not own or control
  - Requires isolation of mutually distrustful applications/processes on the same platform
- Trusted systems theory is concerned with Mandatory access policy enforcement

Trusted Computing Group (TCG)

TCG Promoters

intel

AMD

Microsoft

hp HEWLETT PACKARD

Sun microsystems

IBM

SONY

TCG History & Evolution

- October 1999: TCPA formed
  - Trusted Computing Platform Alliance
  - Founders: IBM, HP, Compaq, Intel and Microsoft
- 2001: 1st TPM specification released
  - Trusted Platform Module
- 2002: TCPA becomes TCG
  - Trusted Computing Group
  - Incorporated not-for-profit industry standards organization
- 2003: TCPA TPM specification adopted by TCG
  - Currently TPM specification 1.2

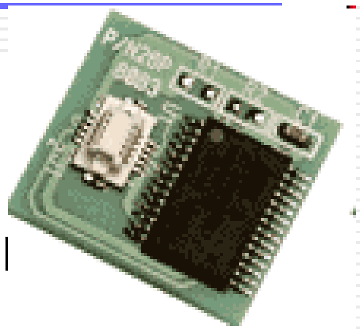
TCG Technical Working Groups 1 - 6

1. TPM WG
   - Specifies how the architecture can be implemented
2. TCG Software Stack (TSS) WG
   - Specifies APIs to be used by application vendors
3. Mobile Phone WG
   - Adapting TCG concepts to mobile devices.
4. Trusted Network Connect WG
   - enables network operators to enforce policies regarding endpoint integrity at or after network connection
5. Server Specific WG
   - Specifies how TCG technology can be implemented in servers
6. Storage System WG
   - Specifies security standards for dedicated storage systems

TCG Technical Working Groups 7 - 11

7. Virtualized Platforms WG
   - Specifies how to secure virtualiplatforms
8. PC Client WG
   - Specifies functionality requirements for including TPMs in PCs
9. Infrastructure WG
   - Specifies how to integrate TCG technology in Internet applications
10. Authentication WG
    - Defines the role of authentication in Trusted Computing
11. Hard Copy WG
    - Defines functionality for hardcopy components

TPM Specification



- Current available spec Version 1.2
  - Revision 103
- TCG aims to be OS and platform independe
- Based on a crypto coprocessor
- Currently available from:
  - Atmel, Infineon, ST, National Semiconductor , Broadcom, Sinosun, STMicroelectronics, Winbond
- TPM 1.2 equipped desktops, laptops available from
  - HP/Compaq, Dell, Gateway, IBM etc.

Trusted Platform Module (TPM)

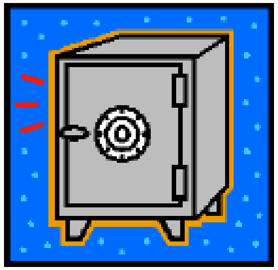
- Core that implements TCG functionality
- Separate crypto capable microprocessor
  - random number generator
  - hashing (SHA-1, HMAC)
  - asymmetric crypto (2048 bit RSA)
  - Asymmetric key pair generation
- The TPM does not expose general purpose symmetric encryption
- Fixed part of the device – can’t be easily transferred to another platform
- TPM Protection Profile specifies EAL3 (augmented)
  - Tamper resistance not required – just tamper evident
  - Side channel analysis attack resistance not required
  - Not aimed at protecting against an attacker with physical access
- Protected memory (key storage, platform configuration metrics)

3 Main TCG Services

1. Protected Storage
   – HW storage for keys
   – Based on Root of Trust for Storage (RTS)
2. Platform Integrity Measurement / Sealed Storage
   – Reliably discover which software is loaded
   – Based on Root of Trust for Measurement (RTM)
   – Sealed storage based on integrity measurement
3. Remote Attestation
   – Reliably report software environment to a remote third party challenger
   – Based on Root of Trust for Reporting (RTR)

TCG Services- Protected Storage

- Portal to a platform's own storage resources
- Storage Root Key (SRK) (asymmetric key) stored inside TPM
- All other keys protected by SRK.
- Protected keys are called *objects*
- Objects are arranged in a hierarchy
  - Parent nodes (asymmetrically) encrypt child nodes



- TPM can generate and protect new signing only key pairs:
  - Private key not releasable, TPM does signing
  - This offers a significant security improvement for the PC

Protected Storage 2

- TPM can stop migration of stored objects to another platform (i.e. only works on this TPM) – ‘non-migratable blobs’
- Access to stored objects can be restricted by:
  - Authentication (prove knowledge of a shared secret using HMAC)
  - Platform Configuration Register (PCR) values (called sealed storage): binds key release to a defined SW config.

TCG Integrity Measurements

- Measurement values reflect software version
- Platform can be configured to only allow predefined software to be loaded:
  - PCR (**P**latform **C**onfiguration **R**egisters) contain predefined software values
  - PCR values same as when the object was wrapped or;
  - Required PCR values defined by someone else (e.g. information owner)
- Platform can also be configured to simply report integrity values

Sealed Storage

- Places data in encrypted blob:
- Availability of data depends on predefined PCR values
  - TPM delivers data only if measurement values mach PCR values
  - otherwise data remains encrypted
- Usage Scenarios:
  - Cryptographic keys for accessing networks
  - Documents, Media files, etc.
- Key question: What happens to sealed data when patching the TCB?

Integrity Protected Booting

- TCG security services, particularly remote attestation, and sealed storage build on an integrity protected boot sequence
- Integrity protected booting is fundamental to TCG design
- Based on chain of trust – critical components (BIOS, ROMS, OS Loader, OS etc.) are measured *before* control is passed to them
  - Same idea as Tygar and Yee, Lampson, Arbaugh

Integrity Protected Booting 2

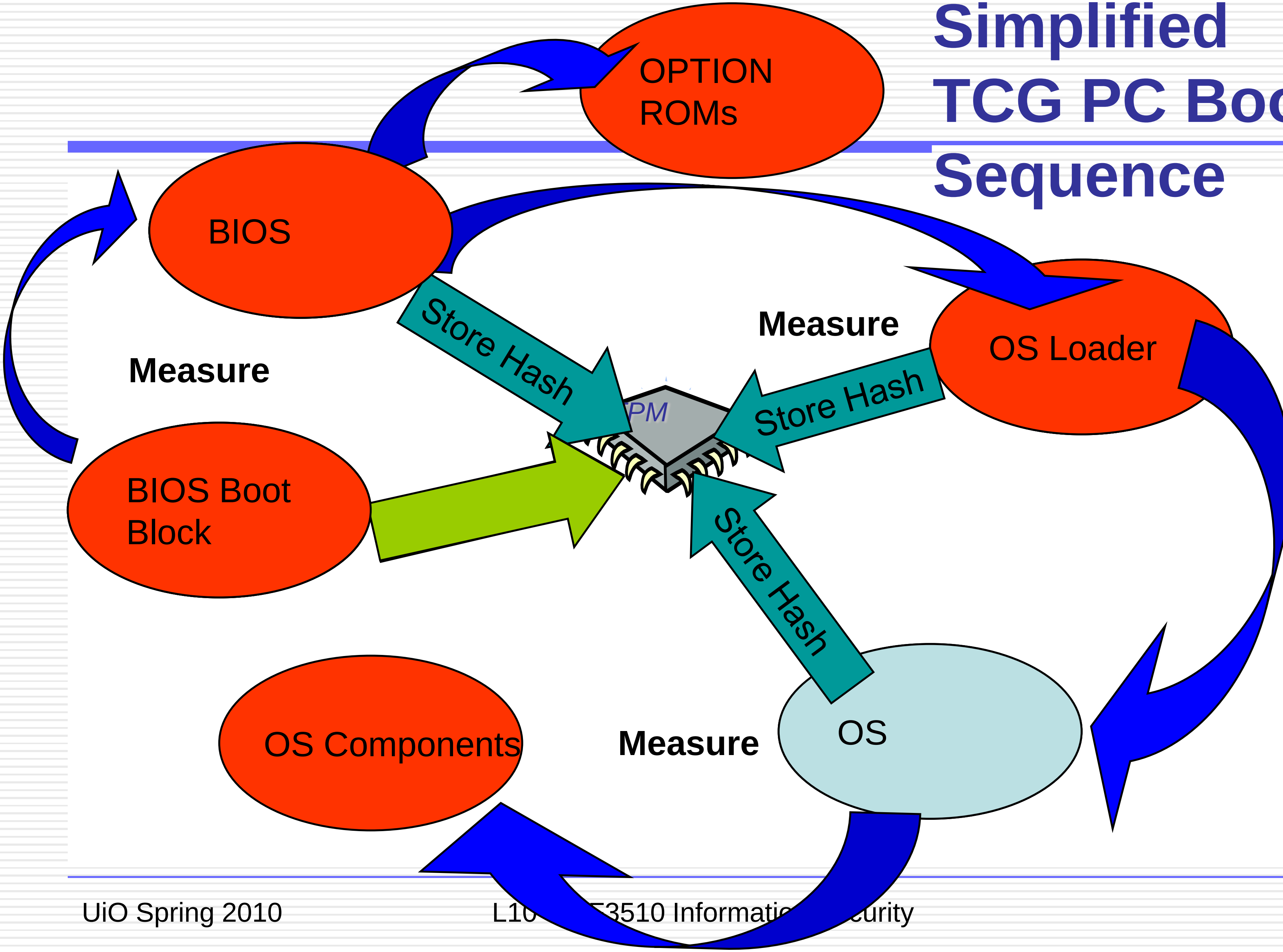
- Measurements are SHA1 hashes of code + config data:
  - called Integrity Metrics
- Measurements are stored in TPM – in Platform Configuration Registers (PCR) as platform boots
  - PCR's can't be deleted or overwritten within a boot cycle
  - They are 'update only' using a simple chained hash technique
  - UpdatedPCRValue = Hash(PreviousPCRValue || MetricToStore) (where || denotes concatenation)
  - Potentially unlimited number of measurements can be 'committed' to a fixed sized register
  - Minimum of 16 PCRs – 160 bits each

Modified components can't hide the modification because they are 'fingerprinted' *before* they are given control of CPU

Integrity Protected Booting 3

- First link in the trust chain is the **R**oot of **T**rust for **M**easurement (**RTM**):
  - For PC's this is a modified BIOS Boot Block
  - Trusted implicitly
- RTM relies on TPM to store and report integrity metrics in a reliable non-forgeable way

Simplified TCG PC Boot Sequence



TCPA PC Specific Implementation Specification v. 1.0

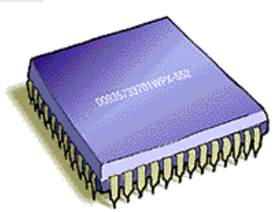
TCG supports two modes of booting

- Secure boot
  - the platform owner can define expected (trusted) PCR values that are stored in special non-volatile Data Integrity Registers (DIR) in the TPM.
  - If a PCR value does not match the expected value for that stage of the boot process, TPM can signal a boot termination request.
- Authenticated boot
  - does not check measured values against expected values – just records in PCRs

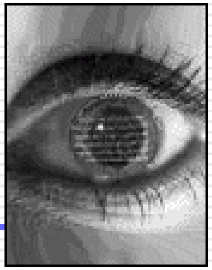
TPM – A Passive Security Enabler

- Note that TPM is passive:
  - It doesn't *decide* which software can and can't run.
  - It provides a way to reliably report the post-boot state of the platform
  - TCG aware *application or OS* <u>can be designed</u> to not start unless platform is in a particular state (no malware etc)
  - TCG aware *application or OS* can be designed to require a TPM mediated online authorisation from a vendor before starting (check for current license etc.):
    - TCG can be *used* to build systems where somebody else decides whether software can or can't run

      TCG does not provide this functionality – it merely enables it



Platform Identity and Privacy



- TPM is uniquely identified by single key pair called *Endorsement key* pair :
  - Generated during manufacture
  - Optional support for EK reset
  - TPM has no way to release private part of endorsement key
- Manufacturer provides a certificate to verify that the public key identifies a genuine TPM
  - Called an *Endorsement Credential*
  - **only** use is to request certified pseudonyms (**identity credentials**) from Privacy CA's – can't be used in any other transactions
  - *Identity credentials* used for remote attestation
  - This is TCG's privacy protection mechanism
- Identity credentials allow a third party to trust that they are dealing with genuine TCG platform without knowing ID

Proposed TCG Applications

- Secure VPN access
- Credential/identity management
- Stronger user authentication
- IT policy compliance checking
- Secure corporate document handling
- Digital Rights Management (DRM)
- Secure e-commerce (online banking, share trading, shopping etc.)
- User privacy protection

Microsoft Vista & Windows 7 BitLocker

- Disk volume encryption
- Off-line protection only
- Protects against data loss in case of lost/stolen computers
- Can be based on TPM, but not necessarily

Spectrum of Protection

BitLocker offers different types of protection, depending on needs

TPM + USB
"What it is + what you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key

User Must:
Protect USB key

USB Only
"What you have"

Protects Against:
HW attacks

Vulnerable To:
Stolen USB key
No boot validation

User Must:
Protect USB key

TPM + PIN
"What it is + what you know"

****

Protects Against:
Many HW attacks

Vulnerable To:
Hardware attacks

User Must:
Enter PIN to boot

TPM Only
"What it is"

Protects Against:
Most SW attacks

Vulnerable To:
Hardware attacks

User Must:
N/A
No user impact

Ease of Deployment / Maintenance

BitLocker life Cycle

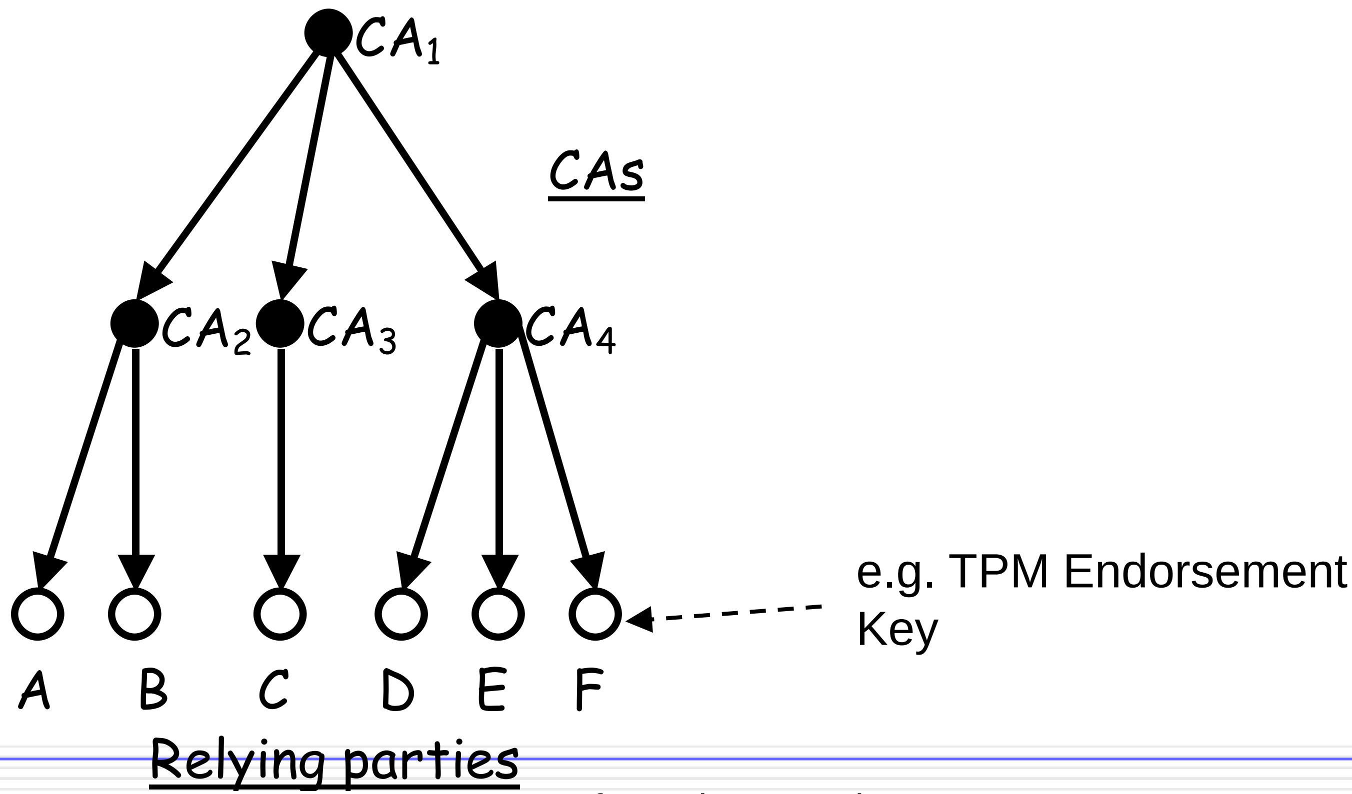
- Installation
  - Select protection
  - Select recovery password or key
- Operation - 4 different modes:
  - TPM only, TPM+PIN, TPM+USB, USB only
- Decommissioning
  - Remove keys by formatting volume
  - Remove BitLocker key protectors
  - Reset TPM

Trusted Computing

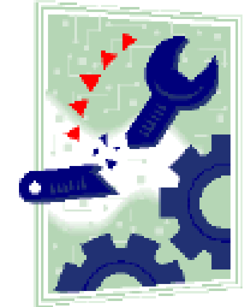
- Practical Issues

TCG Issues

- Requires a PKI – deployment and scaleable certificate revocation unresolved



TCG Issues

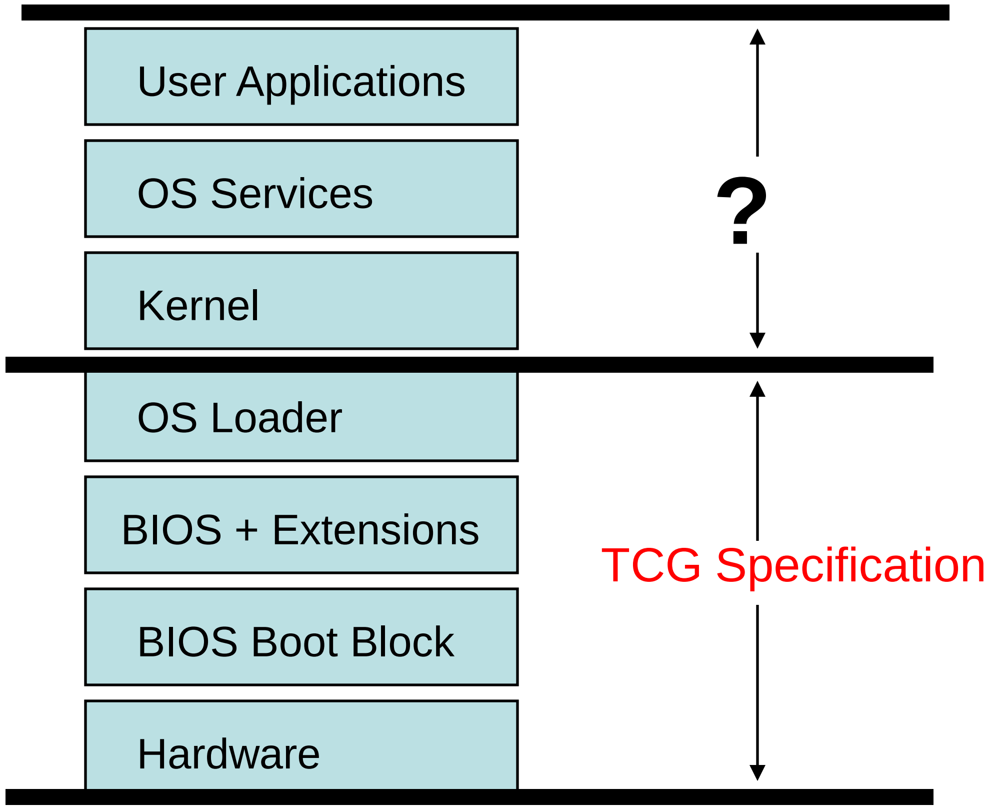


- TCG assumes a system can be trusted if:
  - PCR match values ‘trusted’ by a relying party
  - Why should a particular fingerprint be ‘trusted’?
  - Expected values must of a known configurations
- Problem: what if an OS or system component is inherently insecure due to design or implementation flaws?
  - TCG will not make the system secure
  - PCR will match values of an insecure system
  - TCG does not solve code quality problems

TCG Issues (contd.)

- TCG establishes trust through hashing SW and detecting variations from trusted values
  - For robust maintainable systems we need to go beyond code hashing and signing
  - Robustness depends on a sound OS architecture
  - Requires reference monitor, least privilege etc.
- TCG does not provide the features necessary for trust:
  - trusted input and output path
  - ability to enforce a security policy through reliable memory and process separation
- TCG is only a (small?) part of the solution

TCG & Operating System Integration?

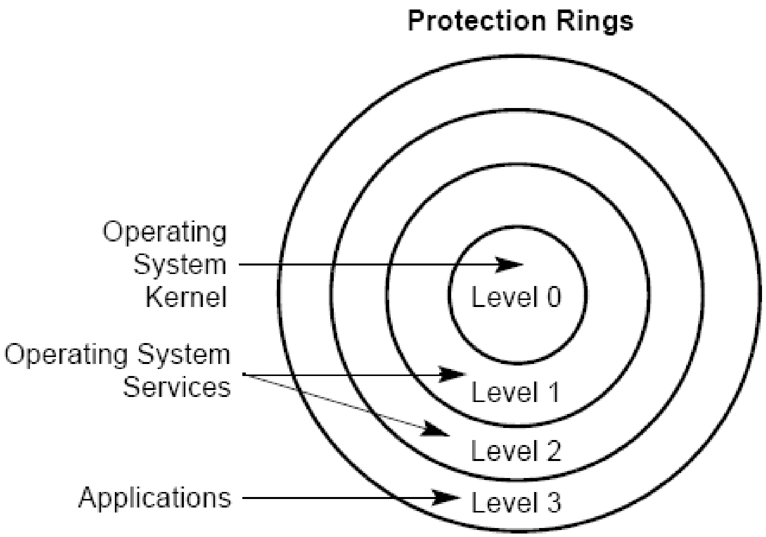


TCG Issues (contd.)

- To be useful, remote attestation must report the *current* configuration
- Without OS modifications TCG measures boot components up to the kernel image
  - Measurement of the kernel is not enough!
  - Security relevant changes can be made at any time post boot
    - e.g. dynamically load a kernel debugger, device driver/kernel module, privileged code
  - OS itself must be modified to ensure all security relevant events are recorded in PCR's post boot
  - Can we agree on what a 'security relevant change' is?

What is a ‘Security Relevant Change’ in a DAC OS ?

- In an nutshell, just about anything!
- Any process executing with supervisor privilege (ring 0) can effect a ‘security relevant change’ at any time.
  - Why? Because there is no way to enforce mandatory isolation within ring 0



More TCG Issues

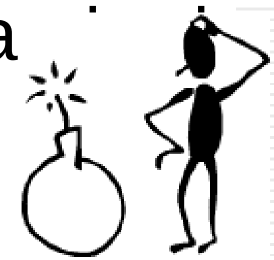
- Config. changes can make data inaccessible:
  - Granularity and fragility of PCR register values
  - May not be able to return to required config – data lost
- Data Loss - Recovering sealed data after a machine crash is very complicated. An (optional) procedure that requires contact with the TPM manufacturer is required. **So complicated that manufacturers may be reluctant to support it.**
- Denial of service risk?

Trusted computing

– why all the controversy?

Trusted Systems, Politics & Policy

- Remote attestation is a double edged sword
  - Enables privacy protecting services e.g. in P2P
  - Enables anticompetitive business models
- Risks – TCG/NGSCB becomes defacto sta
  and therefore not really “opt in”
  - e.g. online banking requires it



- Technology has significant potential for abuse
  - DRM models that erode ‘fair use’ and privacy
  - Active SW license condition enforcement
  - Document censorship (disappearing documents)
  - Forced viewing of advertising

Main Source of the Controversy

- The trust model is potentially directed against the computer's owner
  - Unforgeable remote attestation service is required for DRM, SW Licence enforcement etc.
  - **However**, this service has the potential to significantly shift the balance between the interests of users/consumers and SW HW suppliers
  - EFF thinks users should be able to control the content of attestations through *owner override*– "*fix the problem by restoring others' inability to know for certain what software you're running*"

Legal Issues

- Potential for monopoly abuse and lock in
  - User data locked into proprietary formats that hinder migration to competing products
- Automatic *enforced* license revocation:
  - Safety risks for critical applications
- Can you meet other legal obligations (privacy, financial) if you don't and can't know what software is running on your system?
- Who gets to trust who?
- Liability for Privacy CA's and other certification bodies in TCG structure

End of lecture