NSAC

Stony Brook Network Security and Applied Cryptography Lab

Intro to Trusted Hardware

@ Oakland 2009

radu
sion
sion@cs.stonybrook.edu

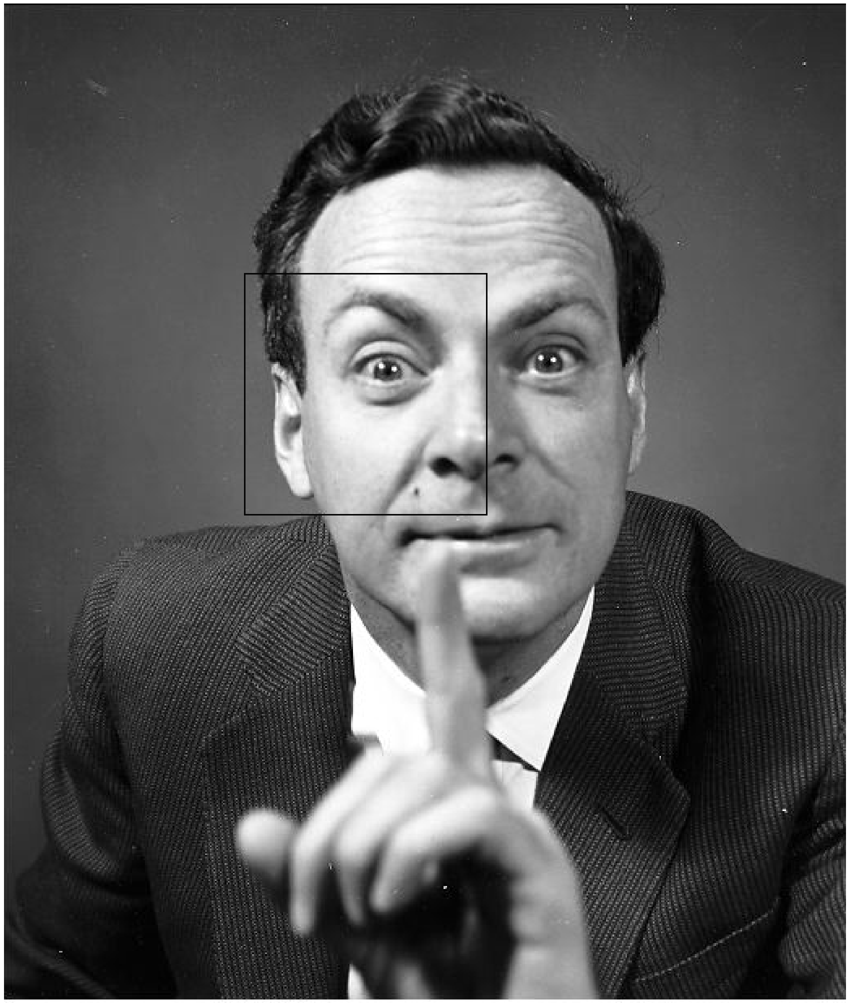
National Science Foundation
WHERE DISCOVERIES BEGIN

IBM

STONY BROOK COMPUTER SCIENCE

ver 2.1

Feynman Moment

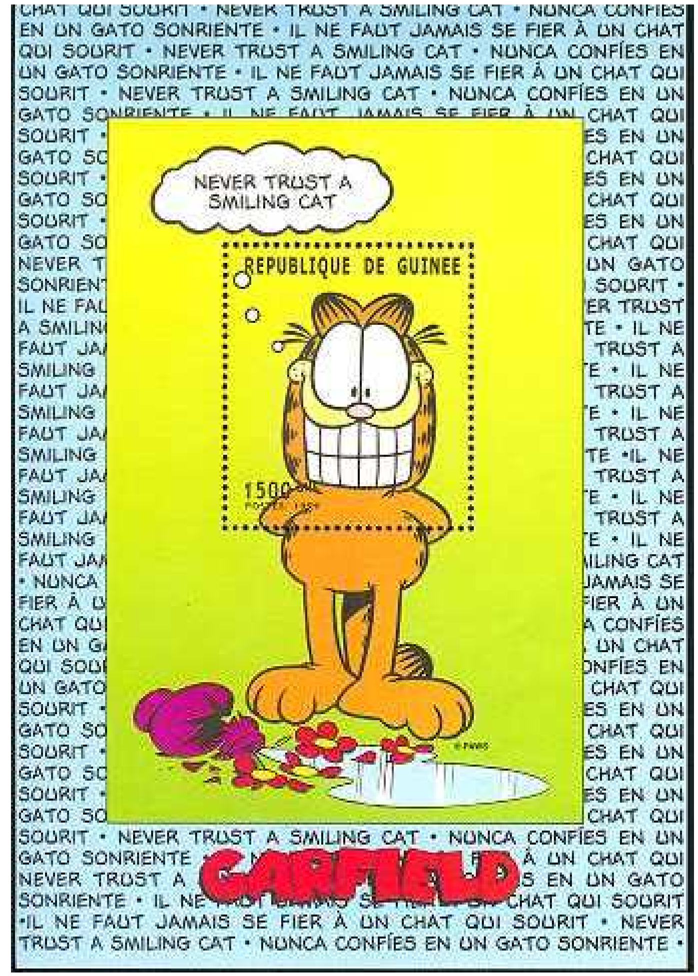


"I have much experience only in teaching graduate students [...] and as a result [...] I know that I don't know how to teach."

so: please interrupt and engage

Trust ?!?!

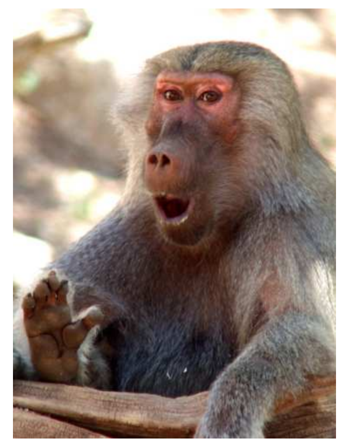
"behave in the *expected manner* for the *intended purpose*"



Usually the Monkey Gets You

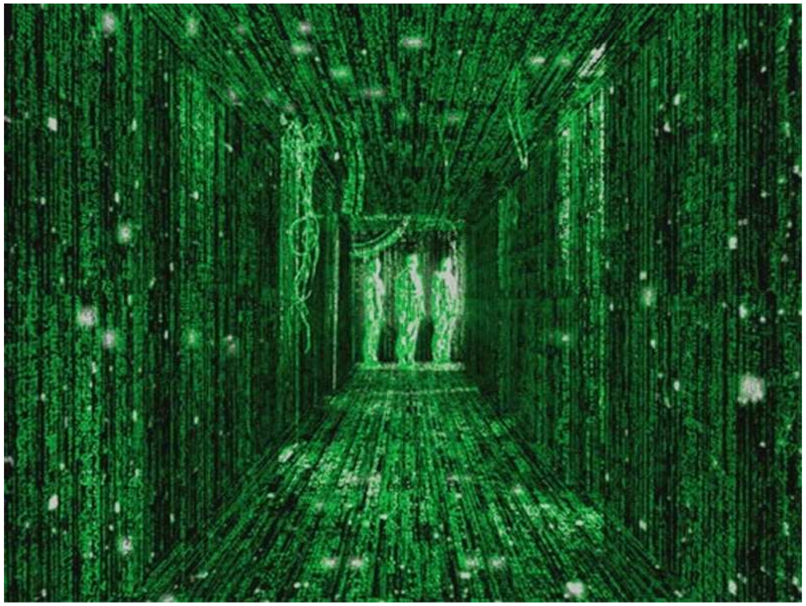
_____ Voting Machine

www._____.com: public picture of its key



Why Hardware ?

By nature, software *lives* in the Matrix but …



… hardware *makes up* the Matrix.

The Myth of Crypto Performance

Baseline.
Pentium 4. 3.6GHz. 1GB RAM. 11000 MIPS. OpenSSL 0.9.7f

DES/CBC: **70MB/sec**
RC4: **138MB/sec**
MD5: **18-615MB/sec**
SHA1: **18-340MB/sec**

Modular MUL 1024: **273000/sec**
RSA1024 Sign: **261/sec**
RSA1024 Verify: **5324/sec**
3DES: **26MB/sec**

Now we have Physical Threats

Invasive

direct access to components

damaging vs. non-damaging

Semi-Invasive

no electrical contact

Local Non-Invasive

close observation of device's operation

(consider also knowledge of attacker)

Remote

observation of device's normal i/o

Also Software Threats

Usual software suspects
External I/O Interface Drivers
Internal OS
Application Bugs

Certification Standards

Hundreds

Common Criteria (ISO/IEC 15408)

Federal Information Protection standards (FIPS)

Trusted Computing Group (TCG)

Evaluation Assurance Levels (EAL)

EAL1: Functionally Tested
EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious.

EAL2: Structurally Tested
Requires the cooperation of the developer in terms of the delivery of design information and test results.

EAL3: Methodically Tested and Checked
Maximum assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices.

EAL4: Methodically Designed, Tested and Reviewed
Maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources (Suse ES 10, RedHat 5, costs $2+ mil.)

EAL5: Semi-formally Designed and Tested
Maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering (Smart cards, IBM z/OS).

EAL6: Semi-formally Verified Designed and Tested
Applicable to the development of security for application in high risk situations.

EAL7: Formally Verified Design and Tested
EAL7 is applicable to the development of security for application in extremely high risk situations. Practical application of EAL7 is currently limited to tightly focused security functionality that is amenable to extensive formal analysis. (a single device so far, smart cards?)

FIPS 140-2 Security Levels

	Security Level 1	*Security Level 2*	*Security Level 3*	*Security Level 4*
Cryptographic Module Specification	Specification of cryptographic module, cryptographic boundary, Approved algorithms, and Approved modes of operation. Description of cryptographic module, including all hardware, software, and firmware components. Statement of module security policy.			
Cryptographic Module Ports and Interfaces	Required and optional interfaces. Specification of all interfaces and of all input and output data paths.		Data ports for unprotected critical security parameters logically separated from other data ports.	
Roles, Services, and Authentication	Logical separation of required and optional roles and services.	Role-based or identity-based operator authentication.	Identity-based operator authentication.	
Finite State Model	Specification of finite state model. Required states and optional states. State transition diagram and specification of state transitions.			
Physical Security	Production grade equipment.	Locks or tamper evidence.	Tamper detection and response for covers and doors.	Tamper detection and response envelope. EFP or EFT.
Operational Environment	Single operator. Executable code. Approved integrity technique.	Referenced PPs evaluated at EAL2 with specified discretionary access control mechanisms and auditing.	Referenced PPs plus trusted path evaluated at EAL3 plus security policy modeling.	Referenced PPs plus trusted path evaluated at EAL4.
Cryptographic Key Management	Key management mechanisms: random number and key generation, key establishment, key distribution, key entry/output, key storage, and key zeroization.			
	Secret and private keys established using manual methods may be entered or output in plaintext form.		Secret and private keys established using manual methods shall be entered or output encrypted or with split knowledge procedures.	
EMI/EMC	47 CFR FCC Part 15. Subpart B, Class A (Business use). Applicable FCC requirements (for radio).		47 CFR FCC Part 15. Subpart B, Class B (Home use).	
Self-Tests	Power-up tests: cryptographic algorithm tests, software/firmware integrity tests, critical functions tests. Conditional tests.			
Design Assurance	Configuration management (CM). Secure installation and generation. Design and policy correspondence. Guidance documents.	CM system. Secure distribution. Functional specification.	High-level language implementation.	Formal model. Detailed explanations (informal proofs). Preconditions and postconditions.
Mitigation of Other Attacks	Specification of mitigation of attacks for which no testable requirements are currently available.			

FIPS 140-2 Physical Requirements

	General Requirements for all Embodiments	Single-Chip Cryptographic Modules	Multiple-Chip Embedded Cryptographic Modules	Multiple-Chip Standalone Cryptographic Modules
Security Level 1	Production-grade components (with standard passivation).	No additional requirements.	If applicable, production-grade enclosure or removable cover.	Production-grade enclosure.
Security Level 2	Evidence of tampering (e.g., cover, enclosure, or seal).	Opaque tamper-evident coating on chip or enclosure.	Opaque tamper-evident encapsulating material or enclosure with tamper-evident seals or pick-resistant locks for doors or removable covers.	Opaque enclosure with tamper-evident seals or pick-resistant locks for doors or removable covers.
Security Level 3	Automatic zeroization when accessing the maintenance access interface. Tamper response and zeroization circuitry. Protected vents.	Hard opaque tamper-evident coating on chip or strong removal-resistant and penetration resistant enclosure.	Hard opaque potting material encapsulation of multiple chip circuitry embodiment or applicable Multiple-Chip Standalone Security Level 3 requirements.	Hard opaque potting material encapsulation of multiple chip circuitry embodiment or strong enclosure with removal/penetration attempts causing serious damage.
Security Level 4	EFP or EFT for temperature and voltage.	Hard opaque removal-resistant coating on chip.	Tamper detection envelope with tamper response and zeroization circuitry.	Tamper detection/ response envelope with tamper response and zeroization circuitry.

FIPS 140-2 Language

"The cryptographic module components shall be covered by potting material or contained within an enclosure encapsulated by a tamper detection envelope (e.g., a flexible mylar printed circuit with a serpentine geometric pattern of conductors or a wire-wound package or a non-flexible, brittle circuit or a strong enclosure) that shall detect tampering by means such as cutting, drilling, milling, grinding, or dissolving of the potting material or enclosure to an extent sufficient for accessing plaintext secret and private keys cryptographic keys …"

Instances

- Encryption disks
- USB tokens
- RSA SecurID
- TPMs
- Smart Cards
- Secure Co-processors
- CPU-level techniques
- PUFs
- misc others

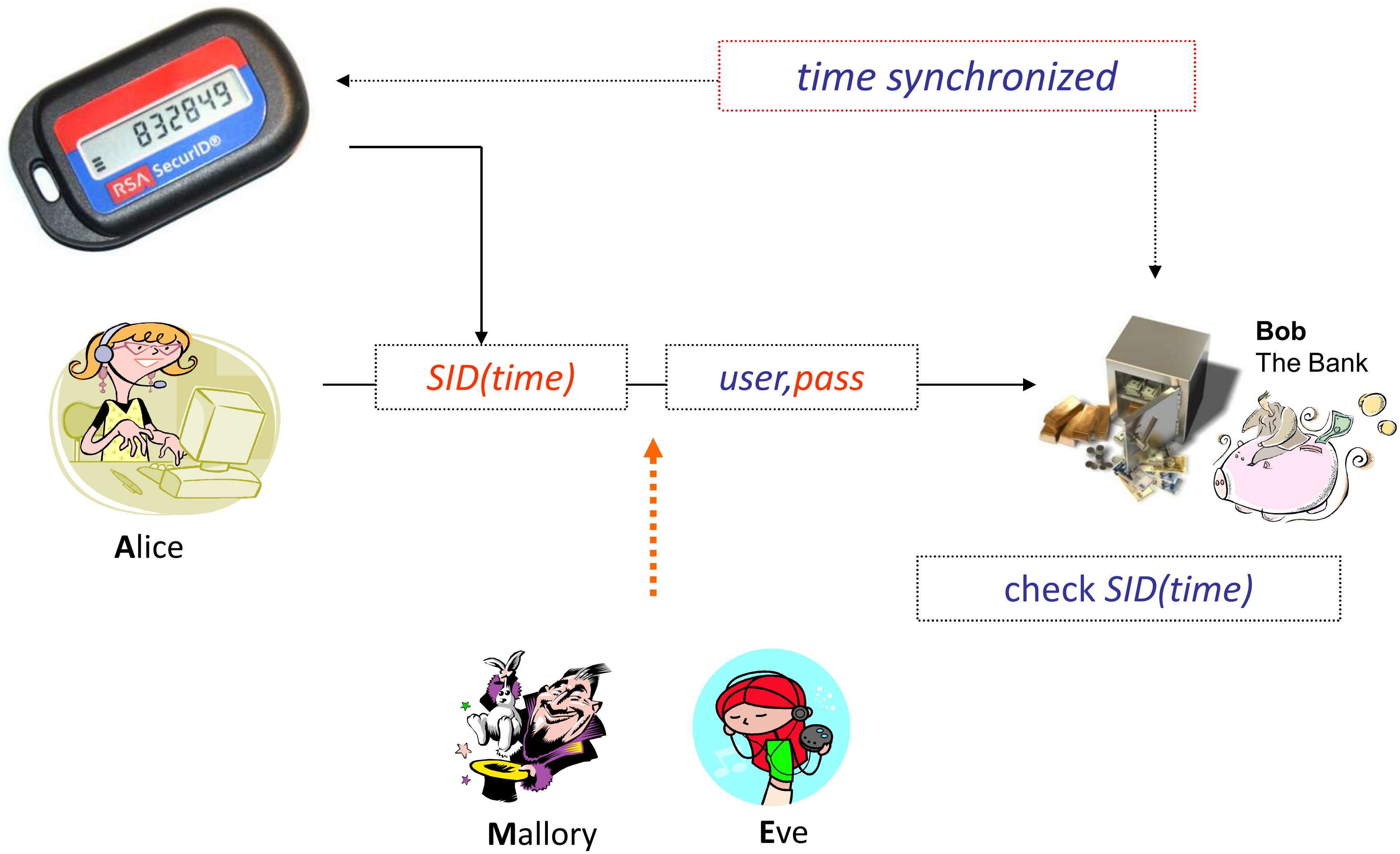
Full Disk Encryption

- **Key Management:** internal
- **Authentication:** mostly external (BIOS, or app)
  - Pre-boot authentication
  - “hashed passwords” on drive
  - emergency password recovery file outside
  - multiple users
- **Encryption**
  - On-board AES – <3% overhead / traditional drive
  - “disk erase” = change encryption keys
- **On Chipset:** Intel vPro chipsets might add encryption in the south bridge (PCI/IDE/…, not until 2010)

USB Storage

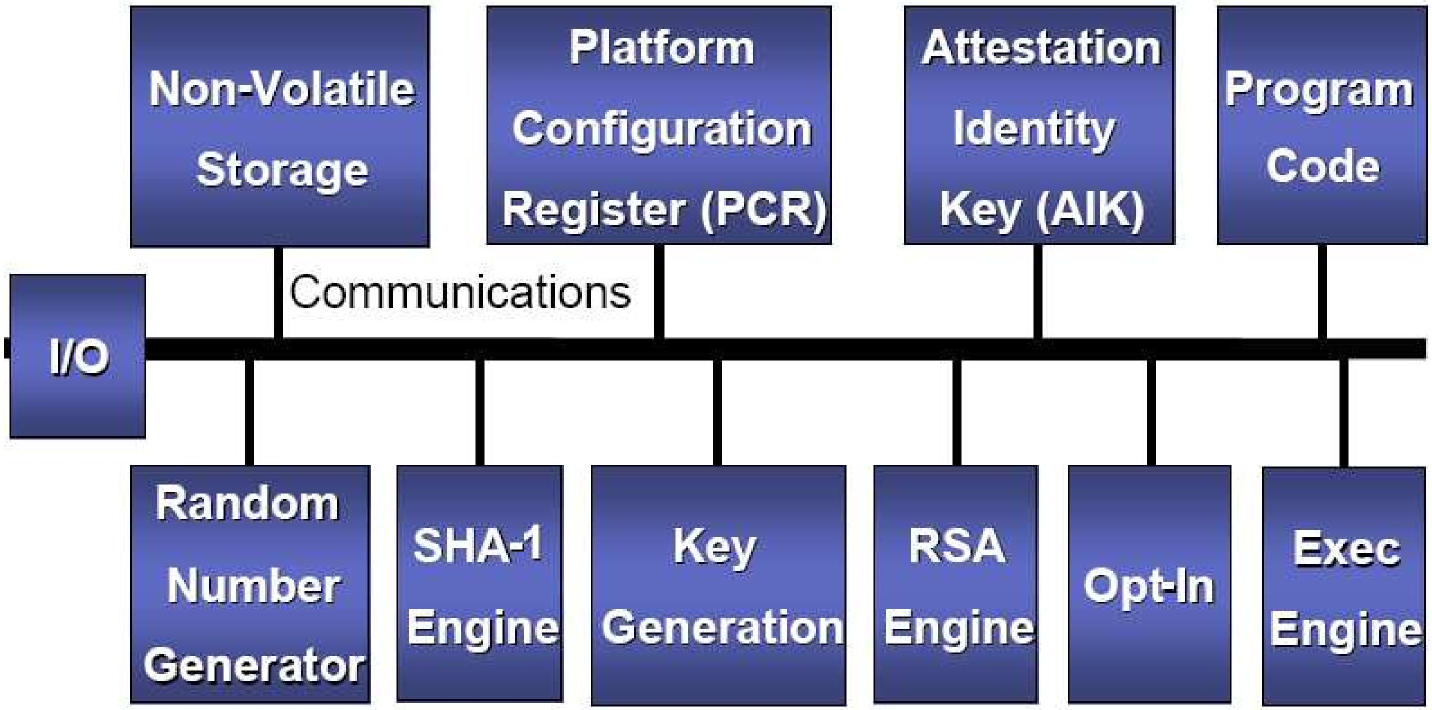
Carry secrets on USB token, often un-locked with a password. Allows for 2-factor authentication.

RSA SecurID

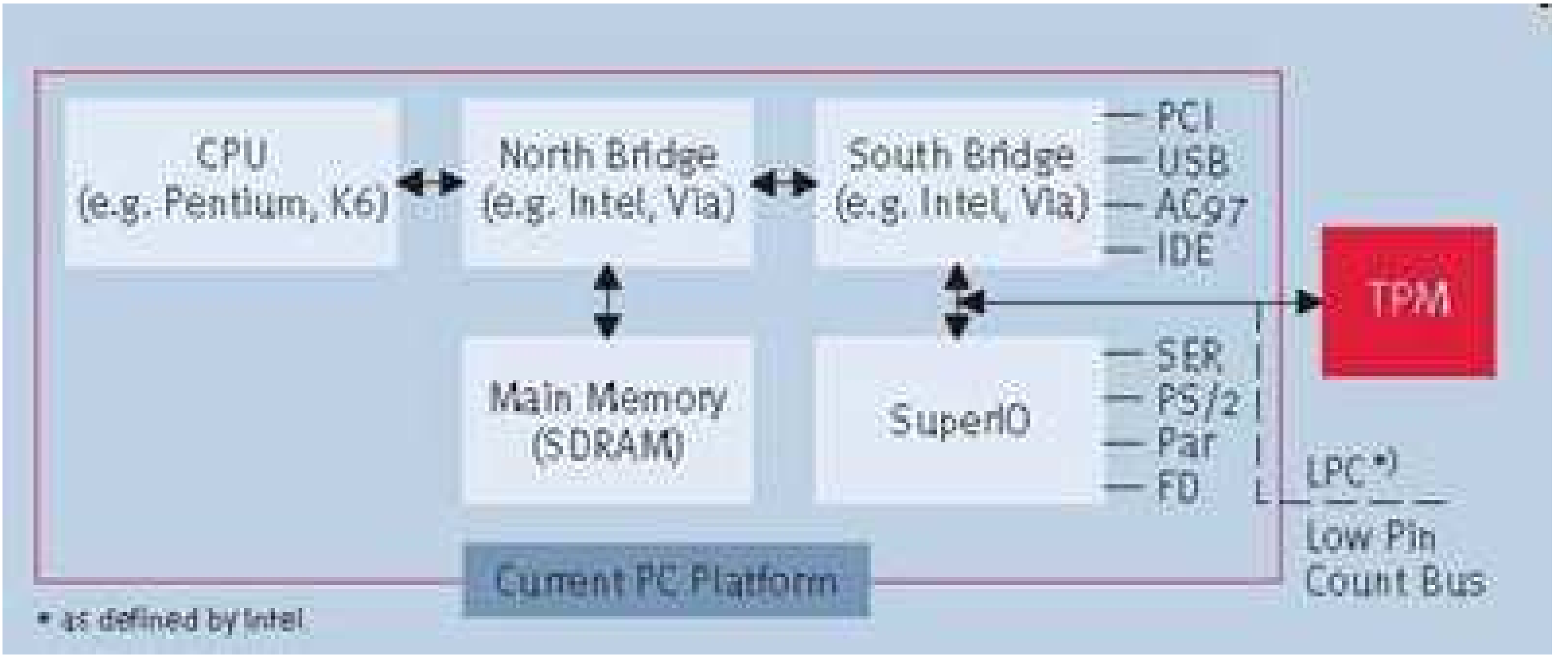

832849
RSA SecurID®
time synchronized
Alice
SID(time)
user,pass
Bob
The Bank
check SID(time)
Mallory
Eve

Trusted Platform Module (TPM)

Microcontroller that stores keys, passwords and digital certificates.



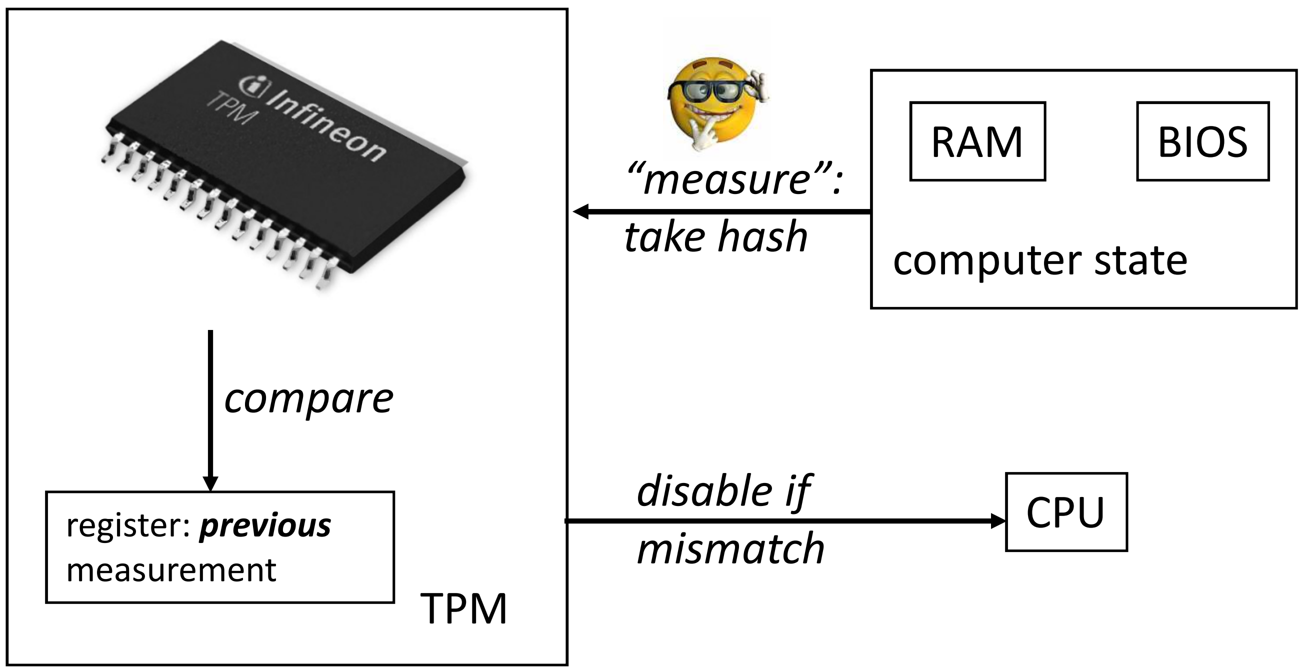
TPM Deployment



Can the Trusted Platform Module control what software runs?
No. [… it] can only act as a 'slave' to higher level services and applications by storing and reporting pre-runtime configuration information. […] At no time can the TCG building blocks 'control' the system or report the status of [running] applications.

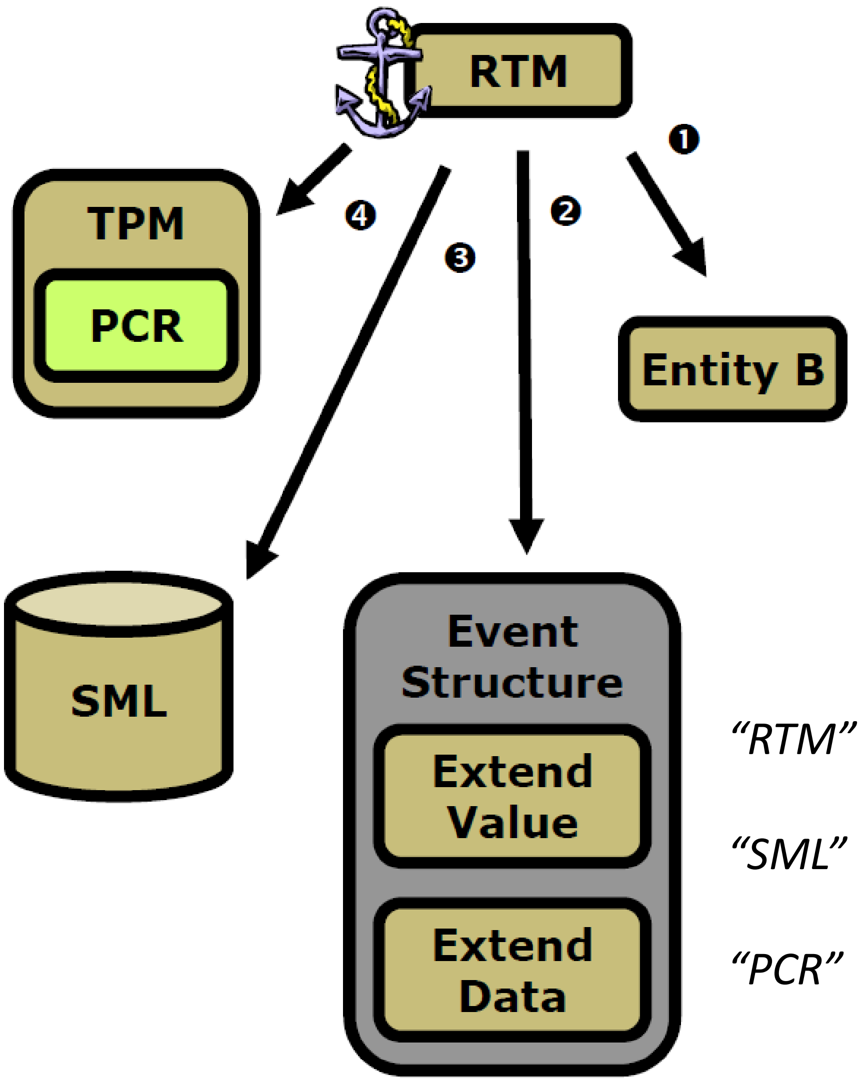
… but it can do "attestation"

Idea: authenticate next link in chain before passing control.
e.g., BIOS to OS, VMM to VM to Guest OS



"measure" = authenticate identity

Measurement

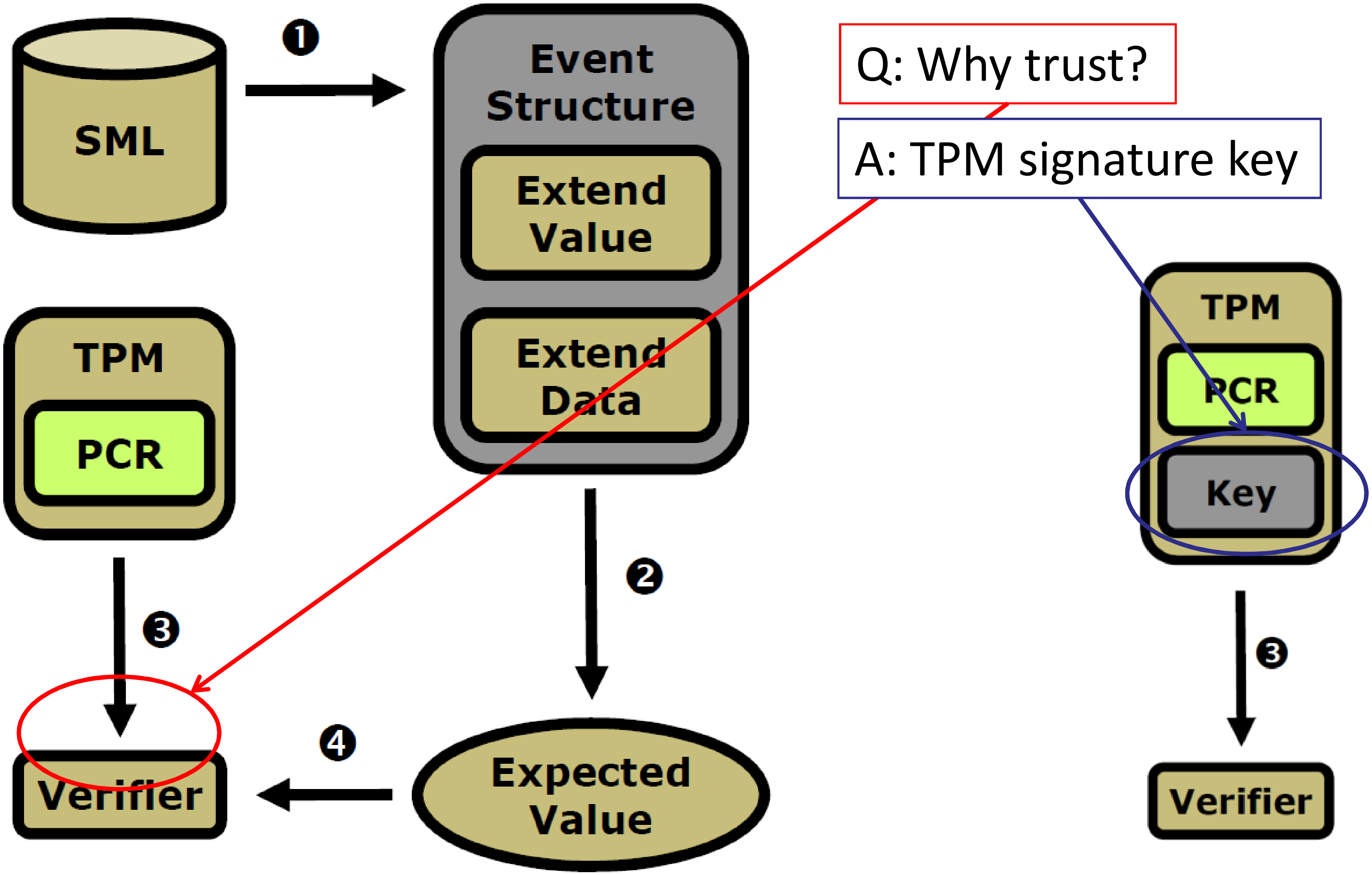


"RTM" = root of trust measurement (e.g., BIOS)

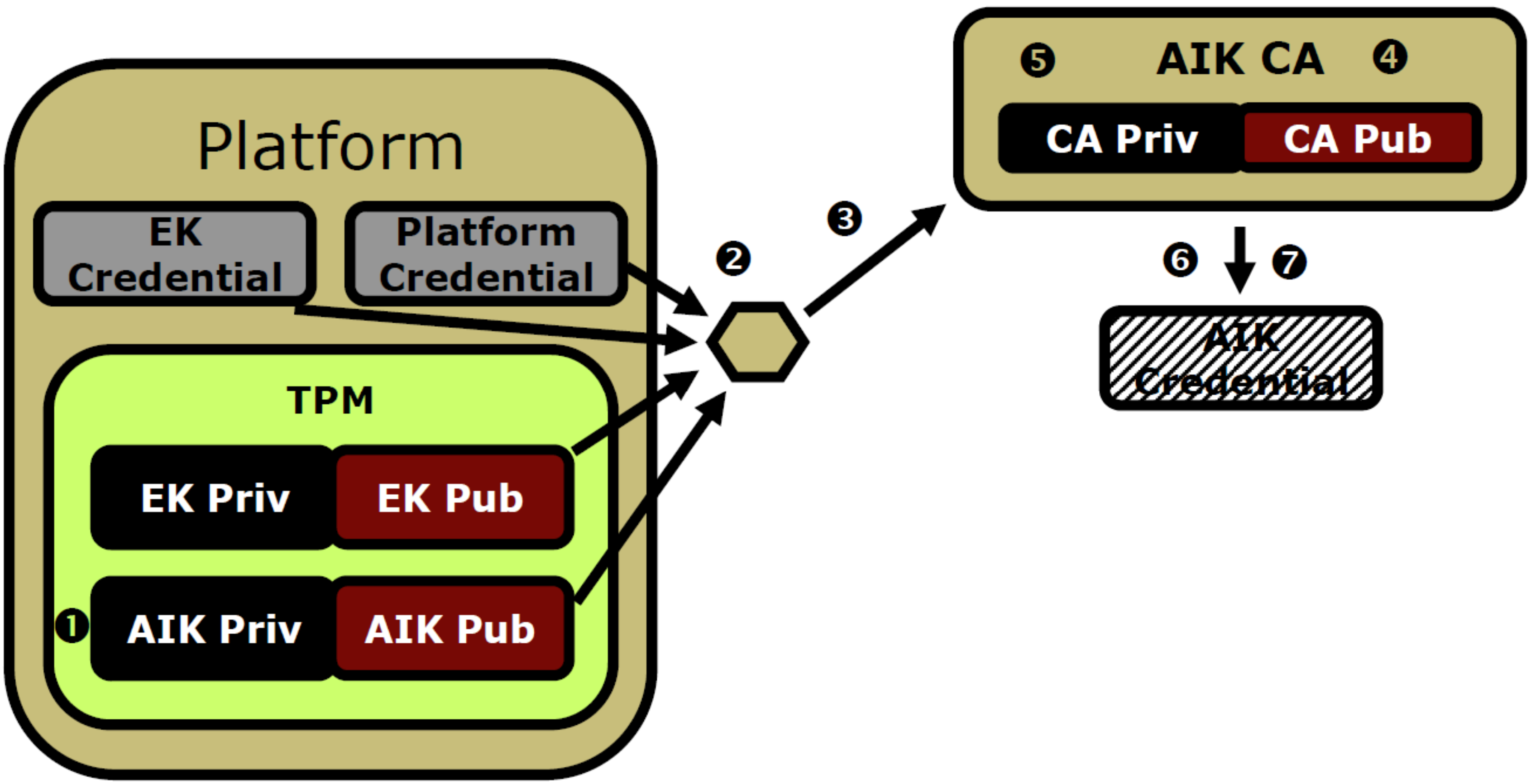
"SML" = stored measurement log (external)

"PCR" = hash(prev,extend), cannot be forced

Verification



Breaking Correlation: AIK CA



"AIK CA" = external certificate authority for AIKs

"AIK" = attestation identity key (2048 bit RSA generated by TPM)

Dynamic vs. static PCRs

Static PCRs: 0-16
Reset by reboot only

Dynamic PCRs: 17-23
Can be reset to 0 without reboot
Reboot sets them to 1 (can remotely distinguish reboot from dynamic reset)

Special PCR 17
Only hardware CPU command can reset it.
SKINIT instruction can trigger that.
Software cannot reset PCR 17

Attacking the TPM

TPM Reset Attack
Sean Smith et al.,
www.cs.dartmouth.edu/~pkilab/sparks/

also
Bernhard Kauer, "OSLO: Improving the security of Trusted Computing", USENIX Security 2007

Programming the TPM

Trusted Software Stack (TSS) Libraries

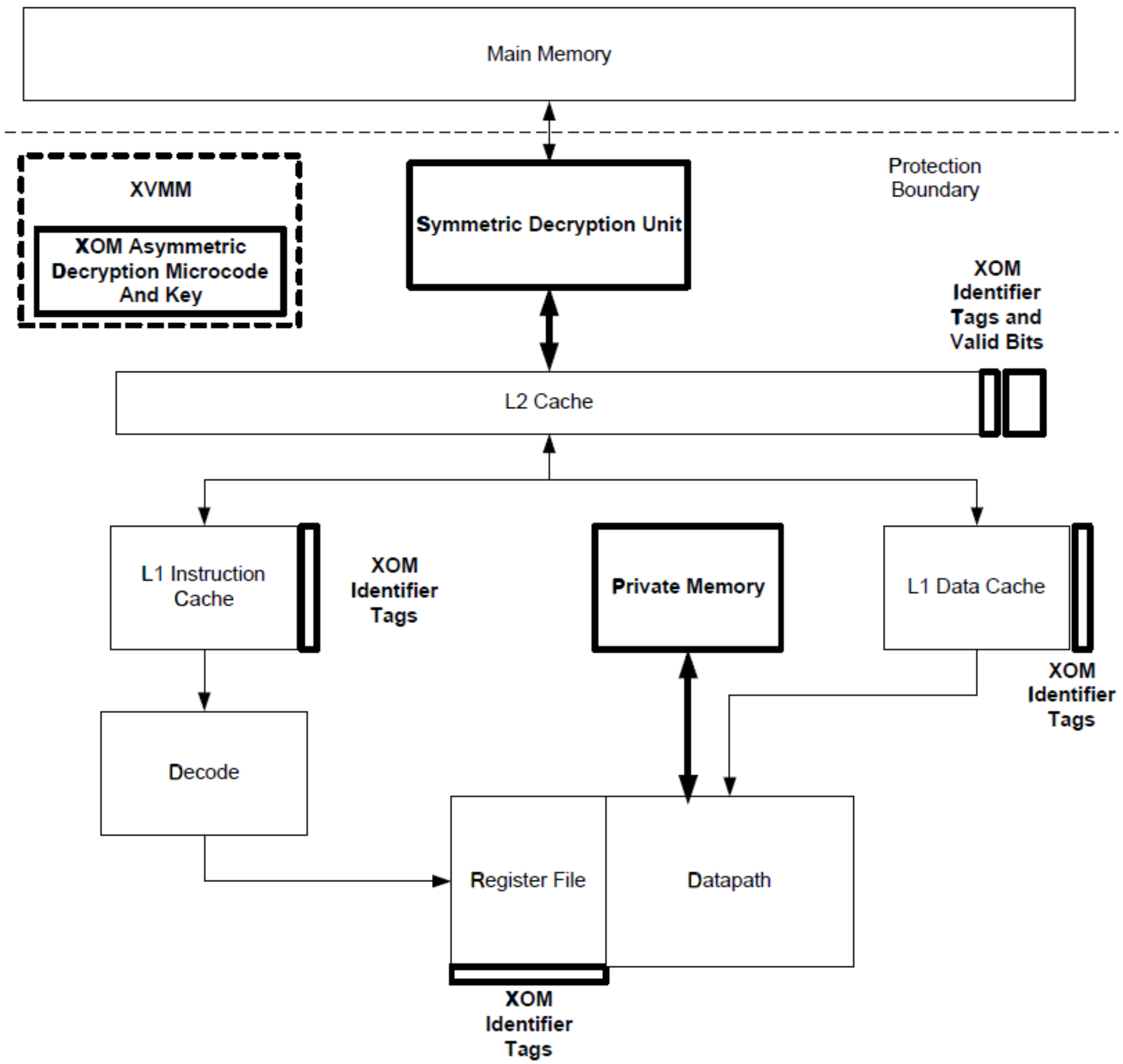
Use Windows TSS dll

Linux TSS SDK

Developer Support and Software

http://www.trustedcomputinggroup.org/developers/

eXecute Only Memory (XOM)



Lie, Thekkath, M. Mitchell, Lincoln, Boneh, J. Mitchell, Horowitz, “Architectural support for copy and tamper resistant software”, ASPLOS 2000.

Smart Cards/Chips

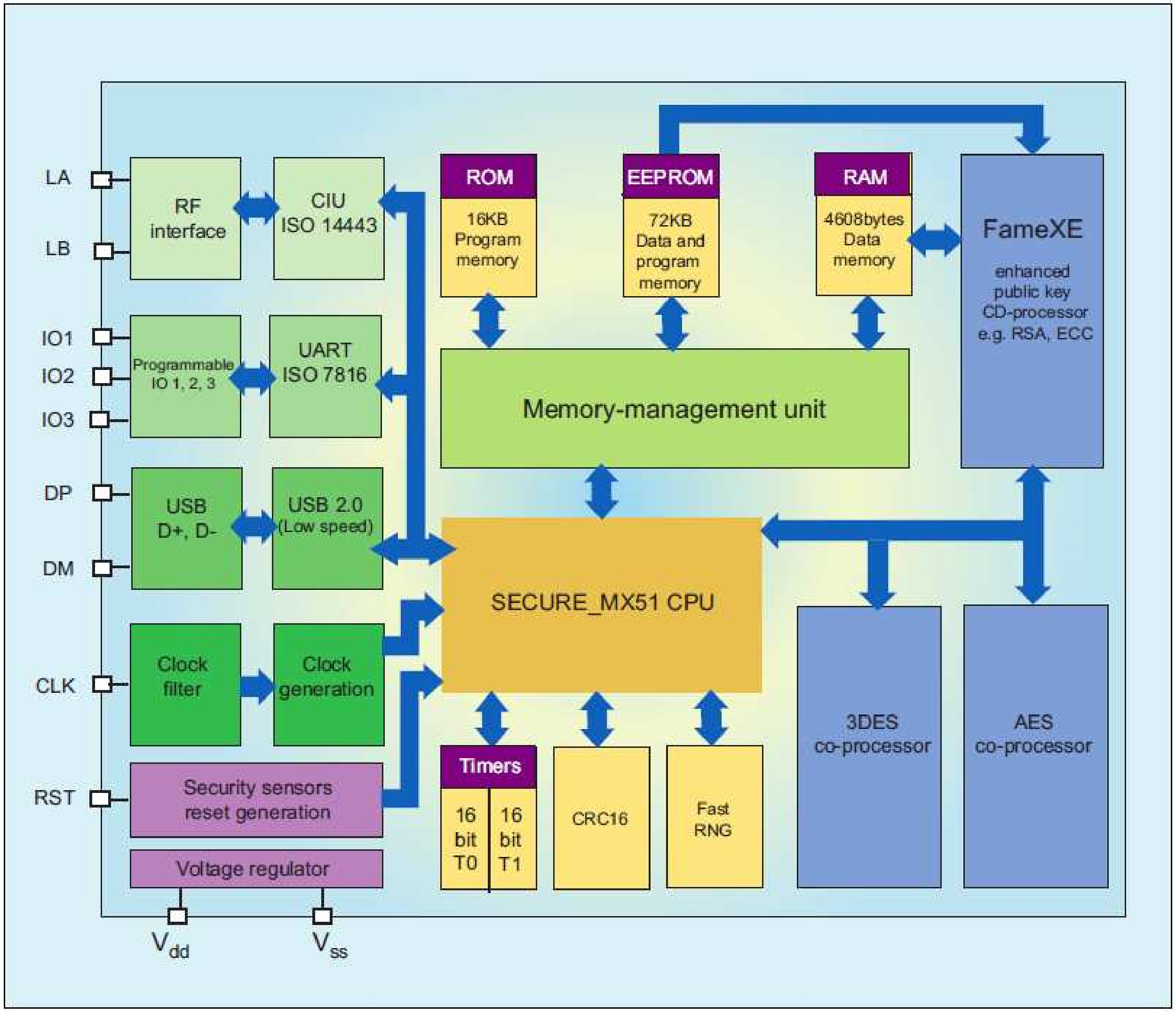
Contact smart card

RFID smart card

Functionality

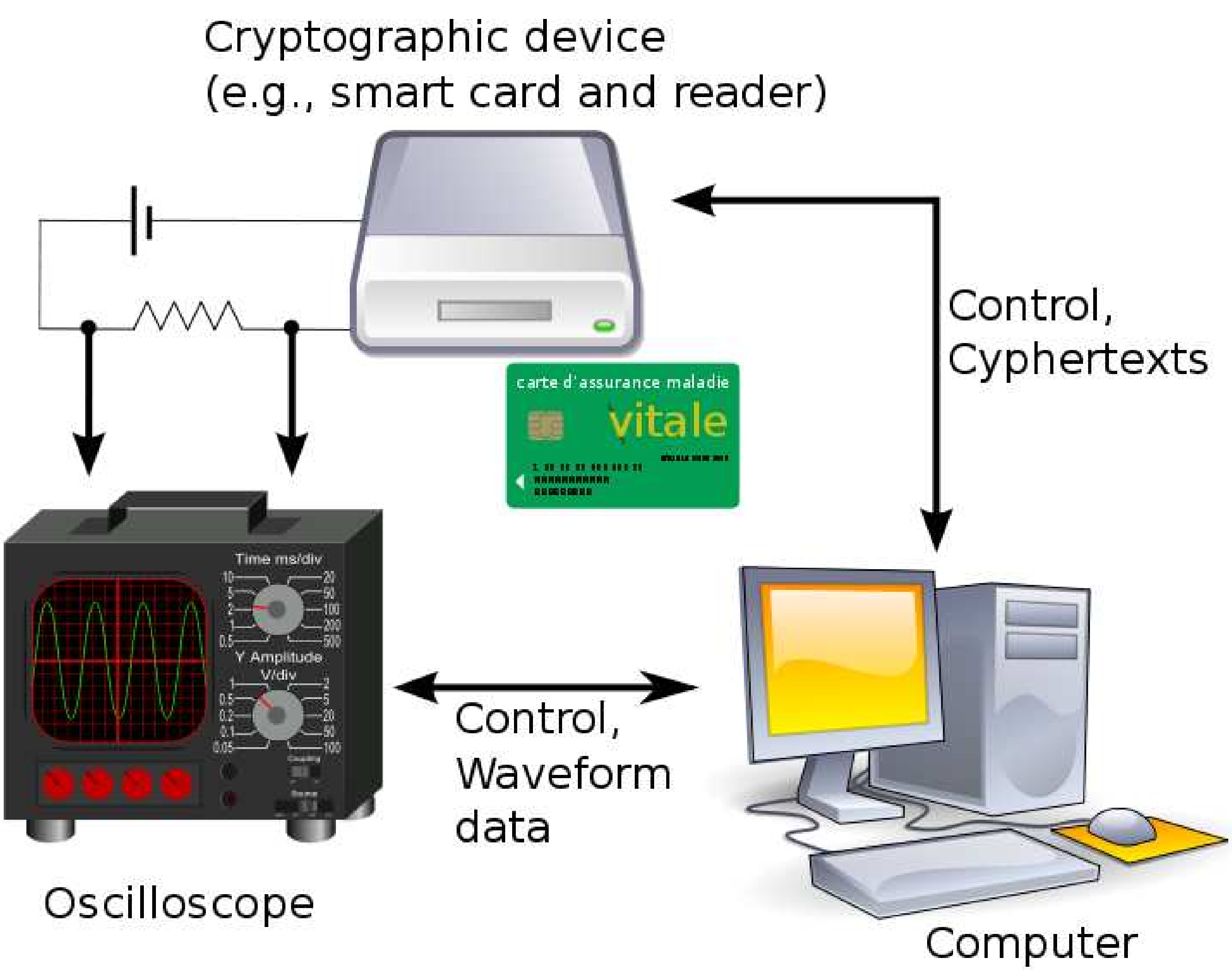
DES, RSA(?), MD5, SHA-1, 4-16kb ROM/RAM, soon 1MB (!), 16bit 10-30MHz CPU, 10-80kbps (source: Sharp)

Architecture



Philips Smart MX

Power Analysis



US Passport

RFID

Made by Smartrac (Netherlands) and shipped to the US from Europe via Thailand. In 2007 China allegedly stole the RFID chip.

Heat and Acids

Polishers and Microscopes

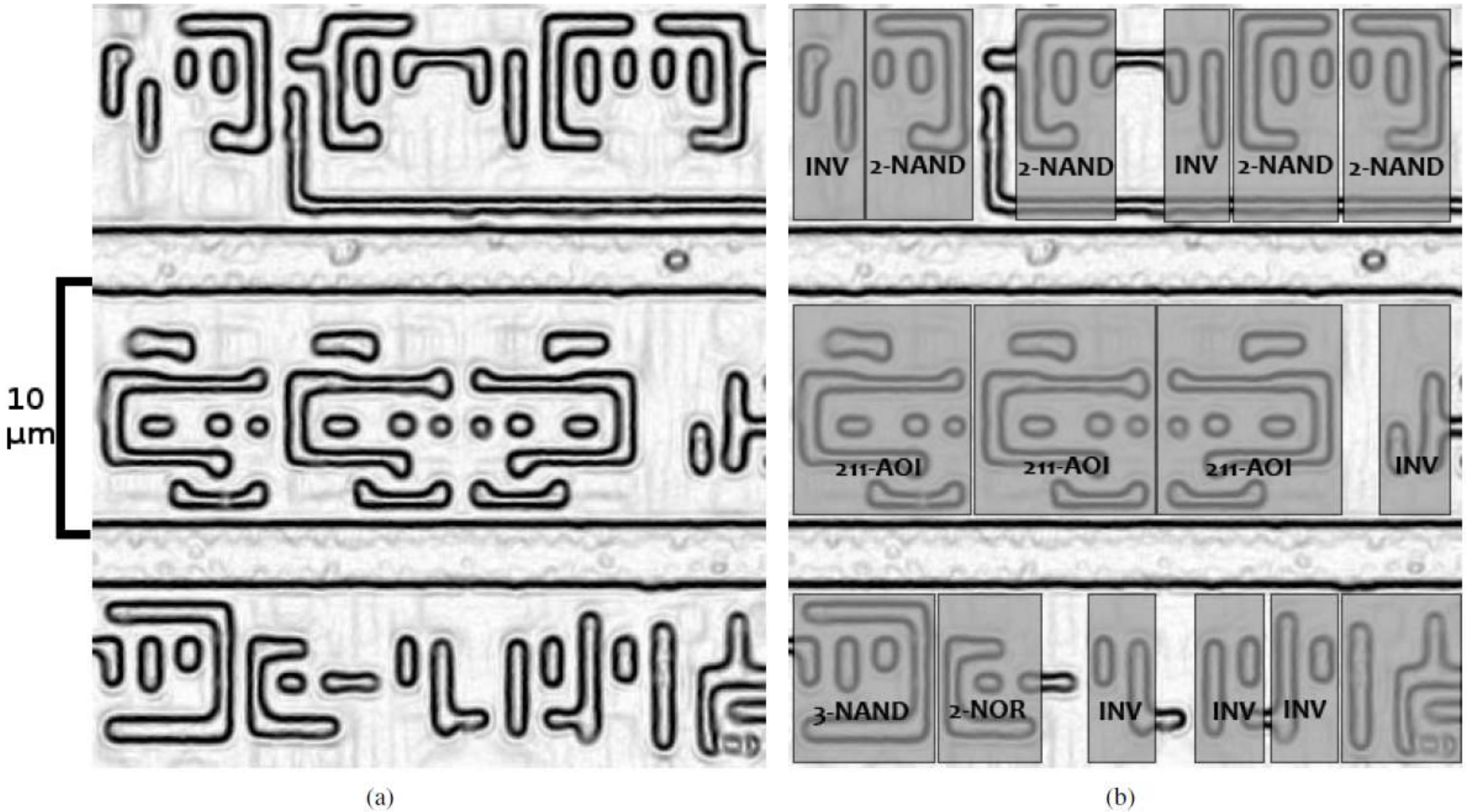


Figure 1: (a) Source image of layer 2 after edge detection; (b) after automated template detection.

[Nohl, Starbug, Plötz, and Evans, "Reverse-Engineering a Cryptographic RFID Tag", USENIX Security 2008]
[Garcia, van Rossum, Verdult, Schreur, Wirelessly Pickpocketing a Mifare Classic Card, Oakland 2009]

Weak: LFSR Cipher, RNG

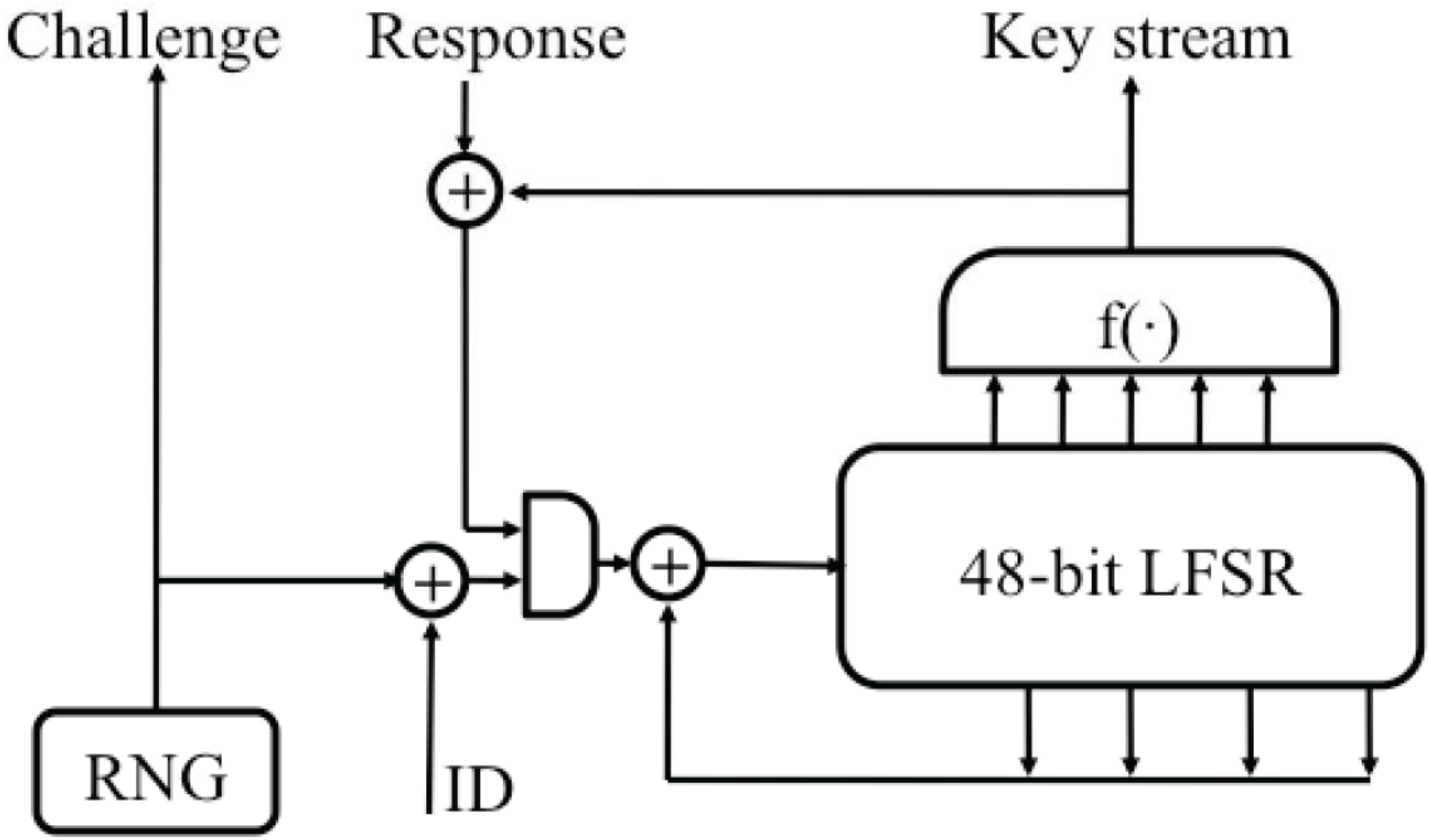
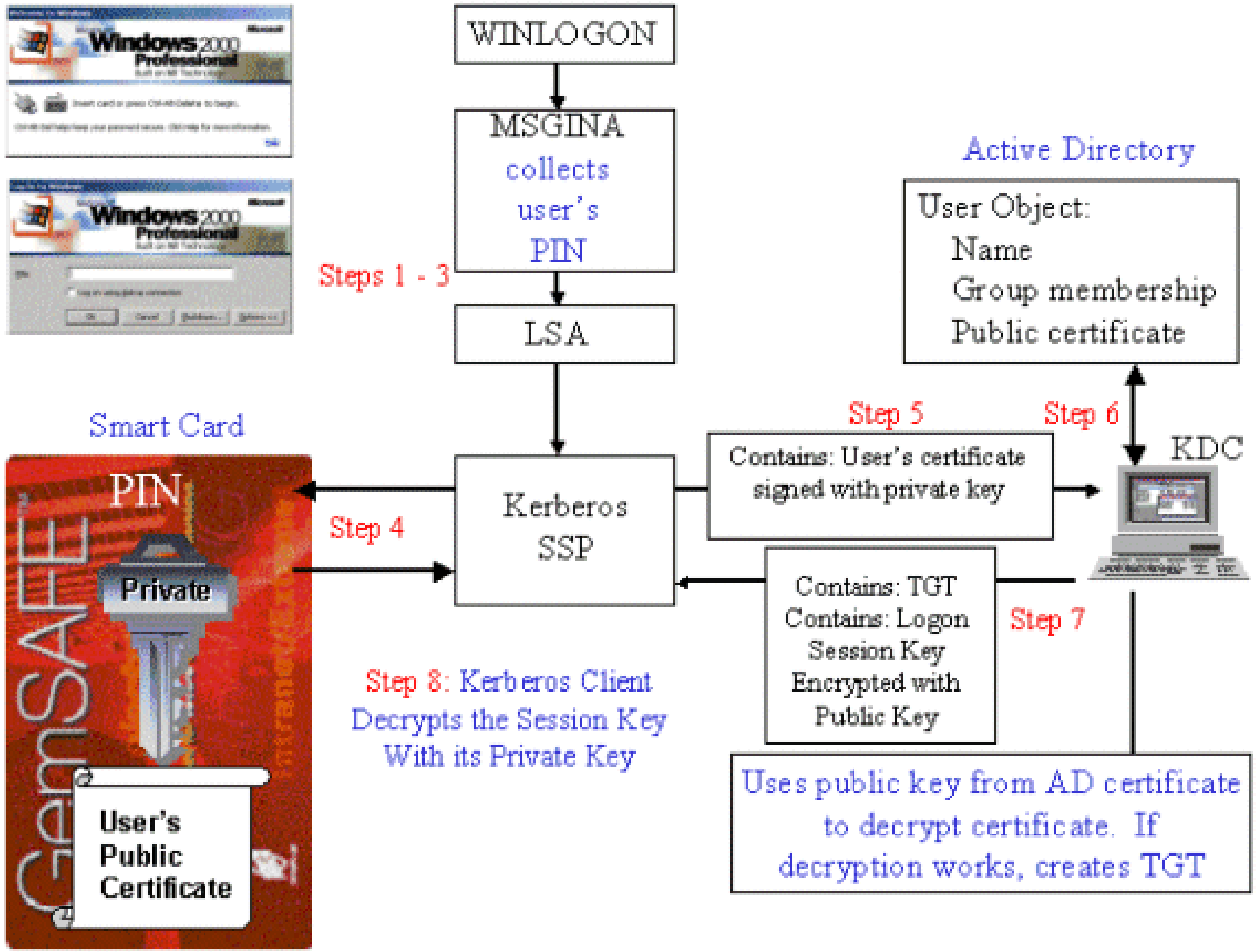
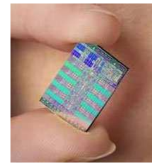


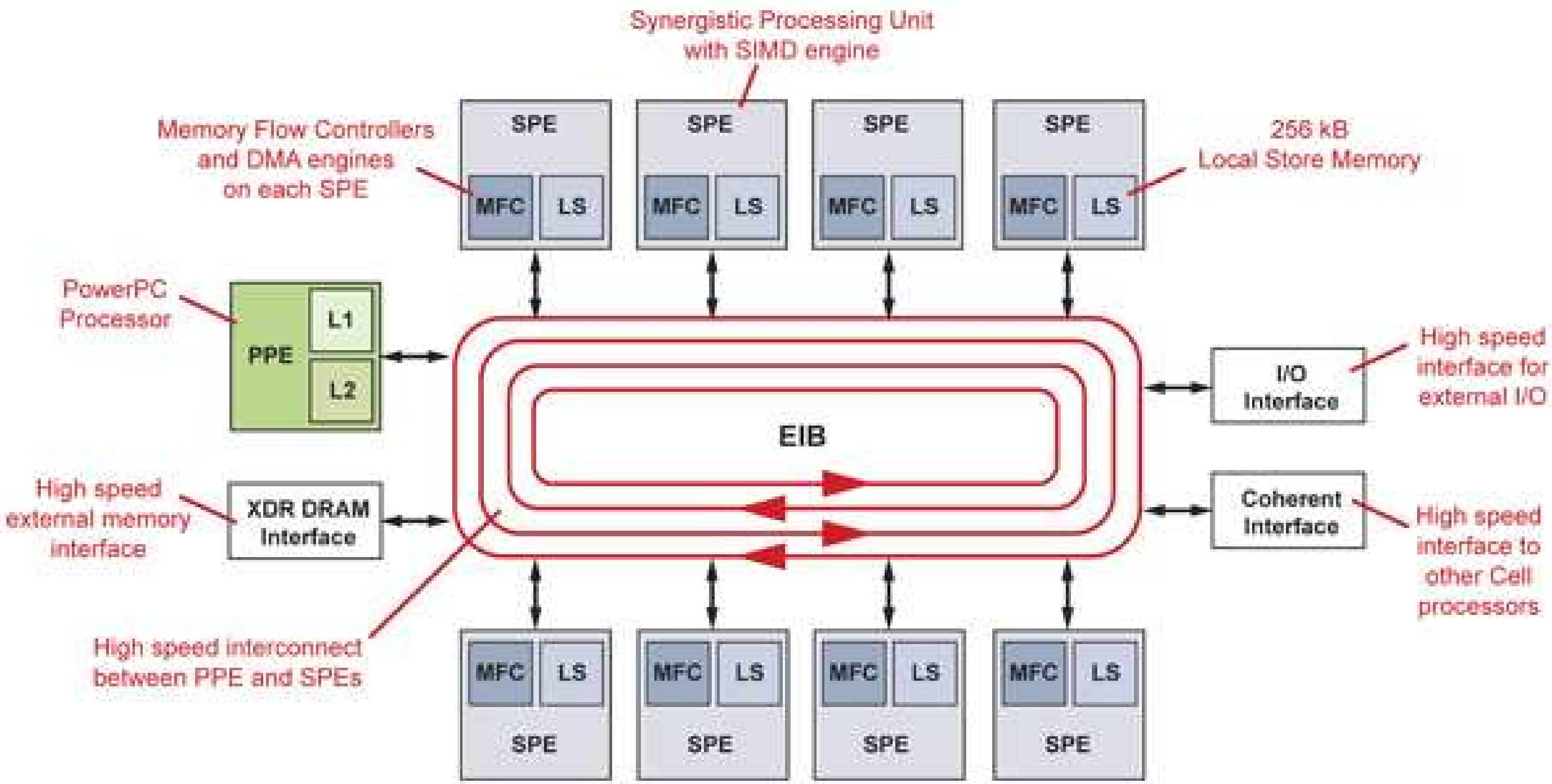
Figure 2: Crypto-1 stream cipher and initialization.

Smart Card: Windows login



Cell Broadband Engine

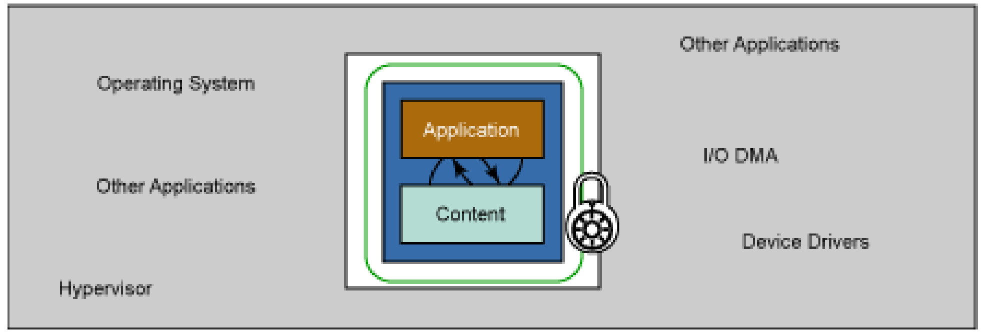




Apps: PS3, Xbox 360, IBM BladeCenter, HPC, Video cards etc.

Cell BE: Secure Processing Vault

Idea: isolate application.

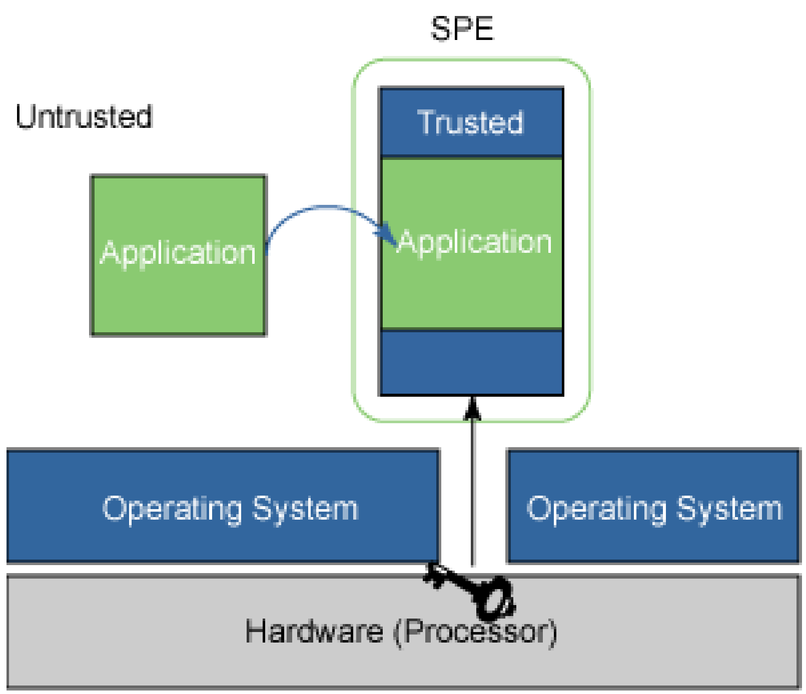


Isolated SPE

- Disengaged from the bus
- SPE LS contains app code + data
- PPE-SPE control mechanisms are disabled
- Only external action possible is cancel: all information in the LS and SPE is erased before external access is re-enabled.
- All LS reads and writes from units on the bus (PPE, SPEs, I/O) have no effect on the locked-up region of the LS.
- Dedicated area of the LS is left open to data transfers.
- Any number of SPEs can be in isolation mode at any given time.

Cell BE: Runtime Secure Boot

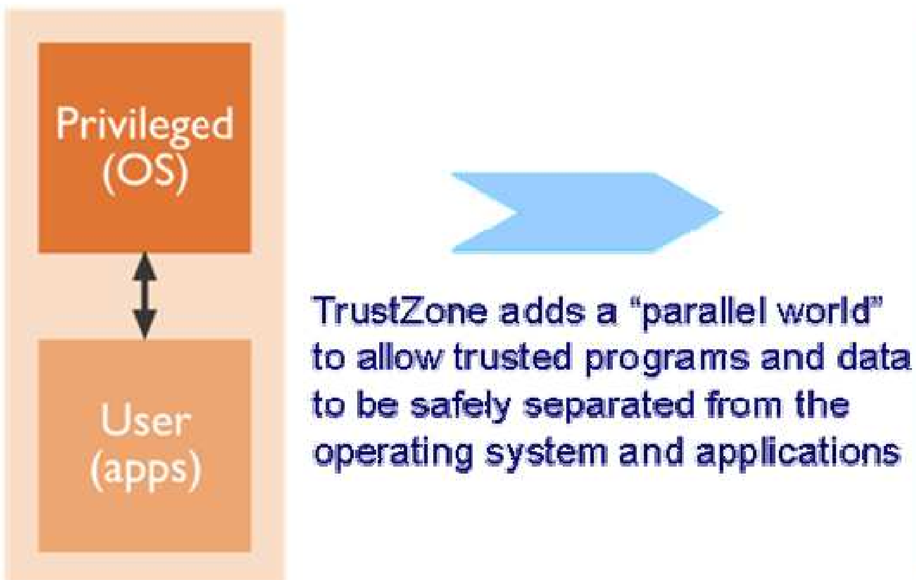
Idea: verify application. Cool: hardware auth.

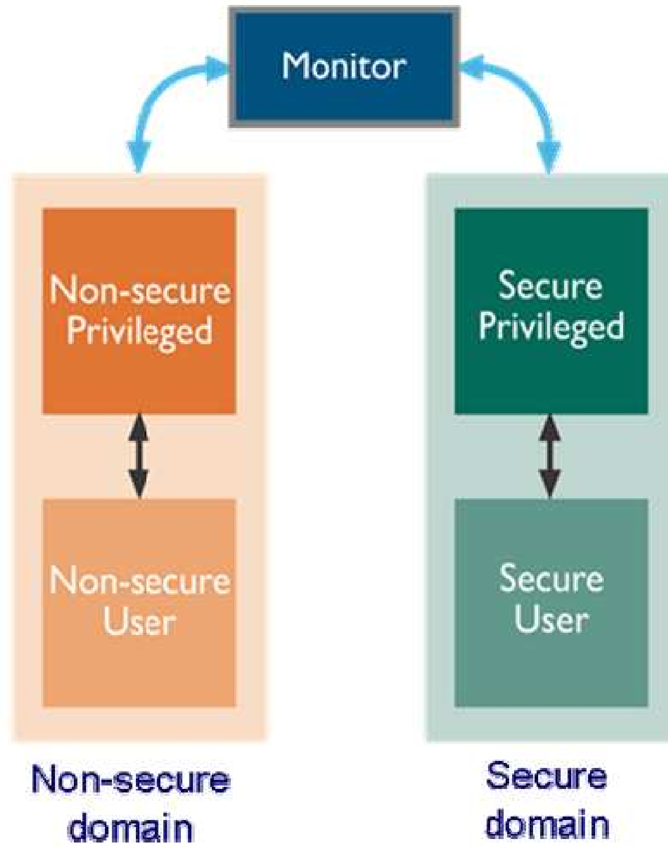


1. Isolation mode is initiated.
2. Previous application is stopped and cancelled.
3. Application is fetched in and checked by the hardware authenticator
   - based on a hardware key and cryptographic algorithm
4. Integrity check fails; execution stopped
   - Application was tampered
5. Check succeeds; will kick-start the application's execution in isolation mode.

ARM TrustZone

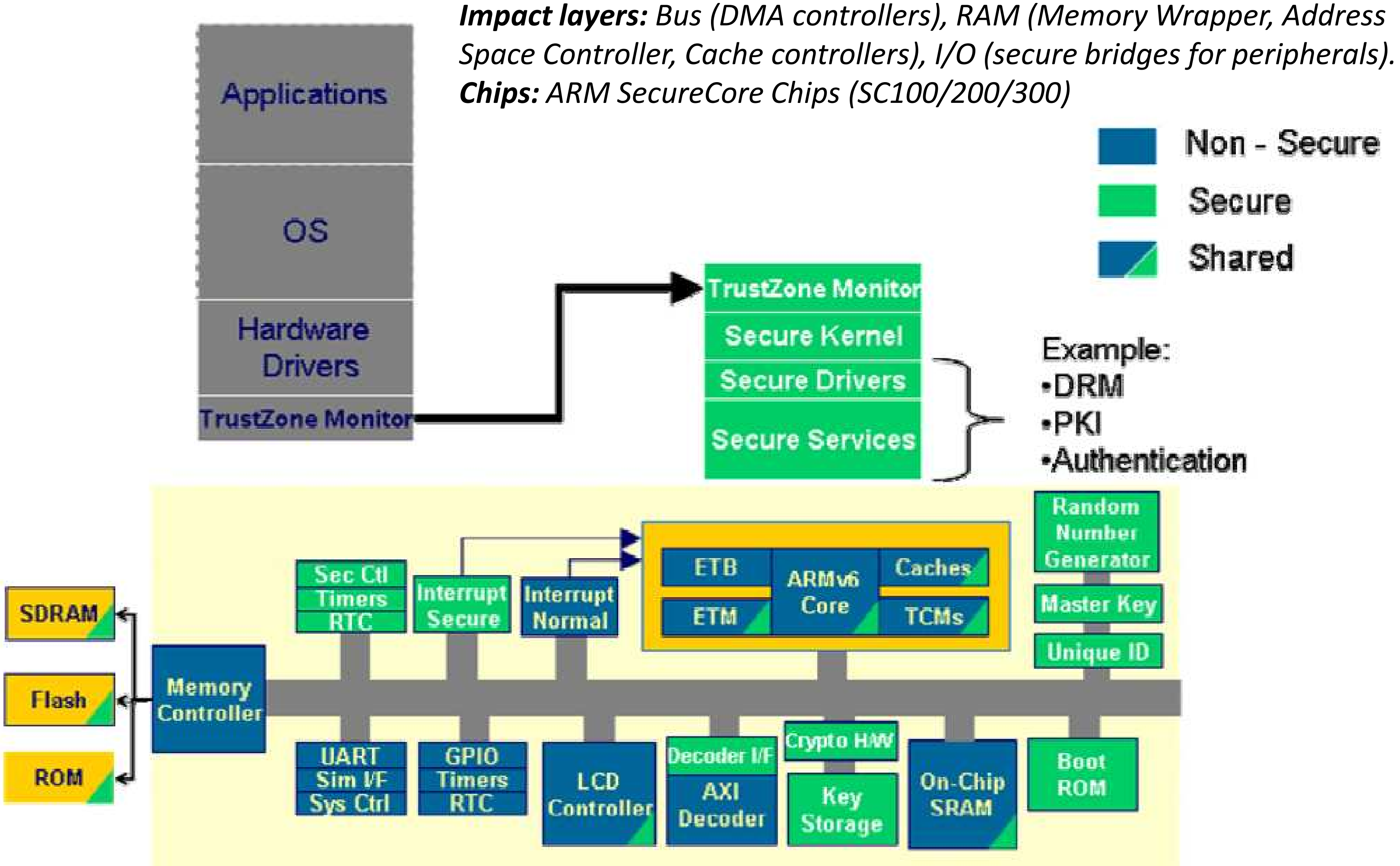
ARM TrustZone: "allows the system to be more easily partitioned for security while maintaining hardware-backed protection for the security sub-system."



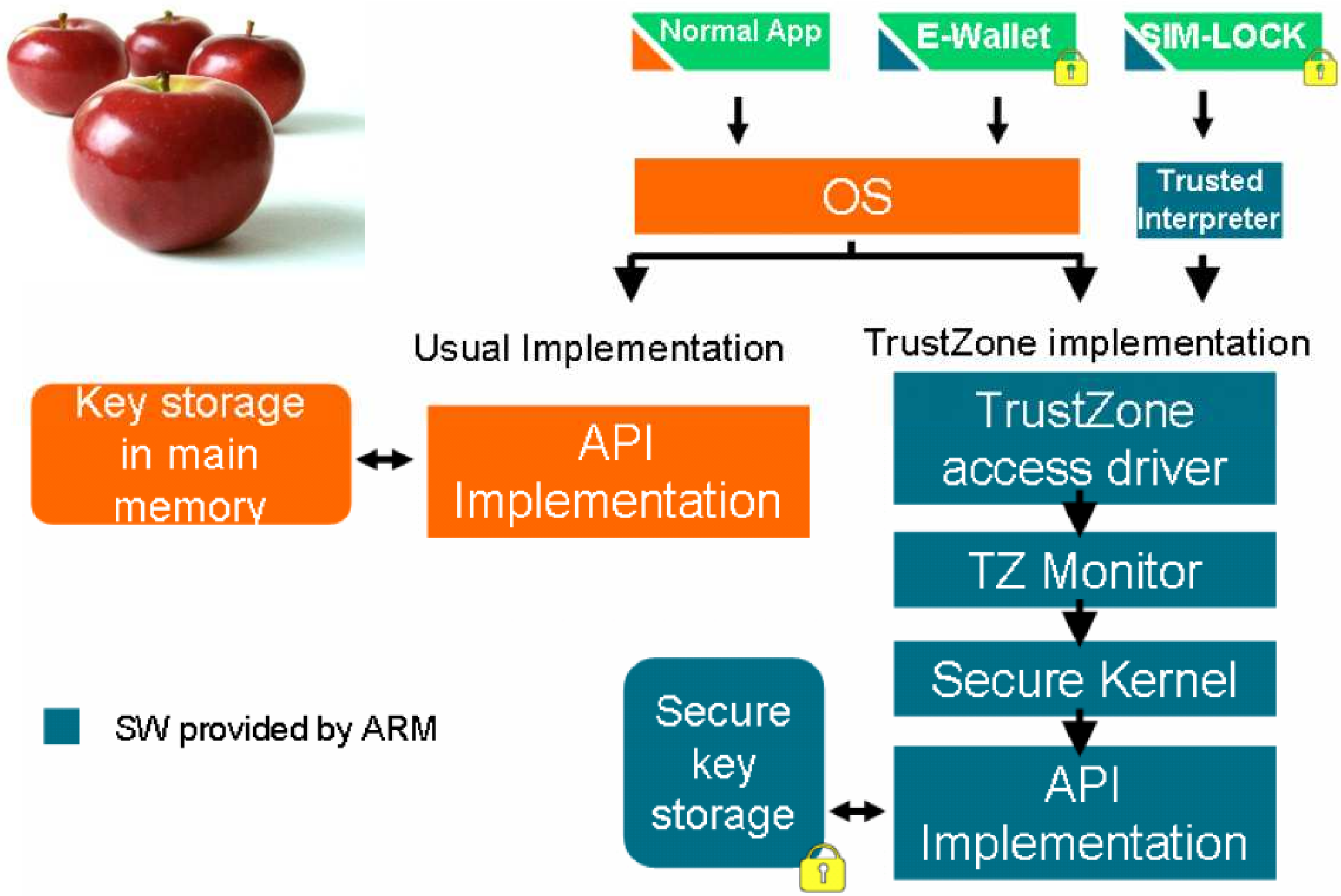


TrustZone: Partitioning

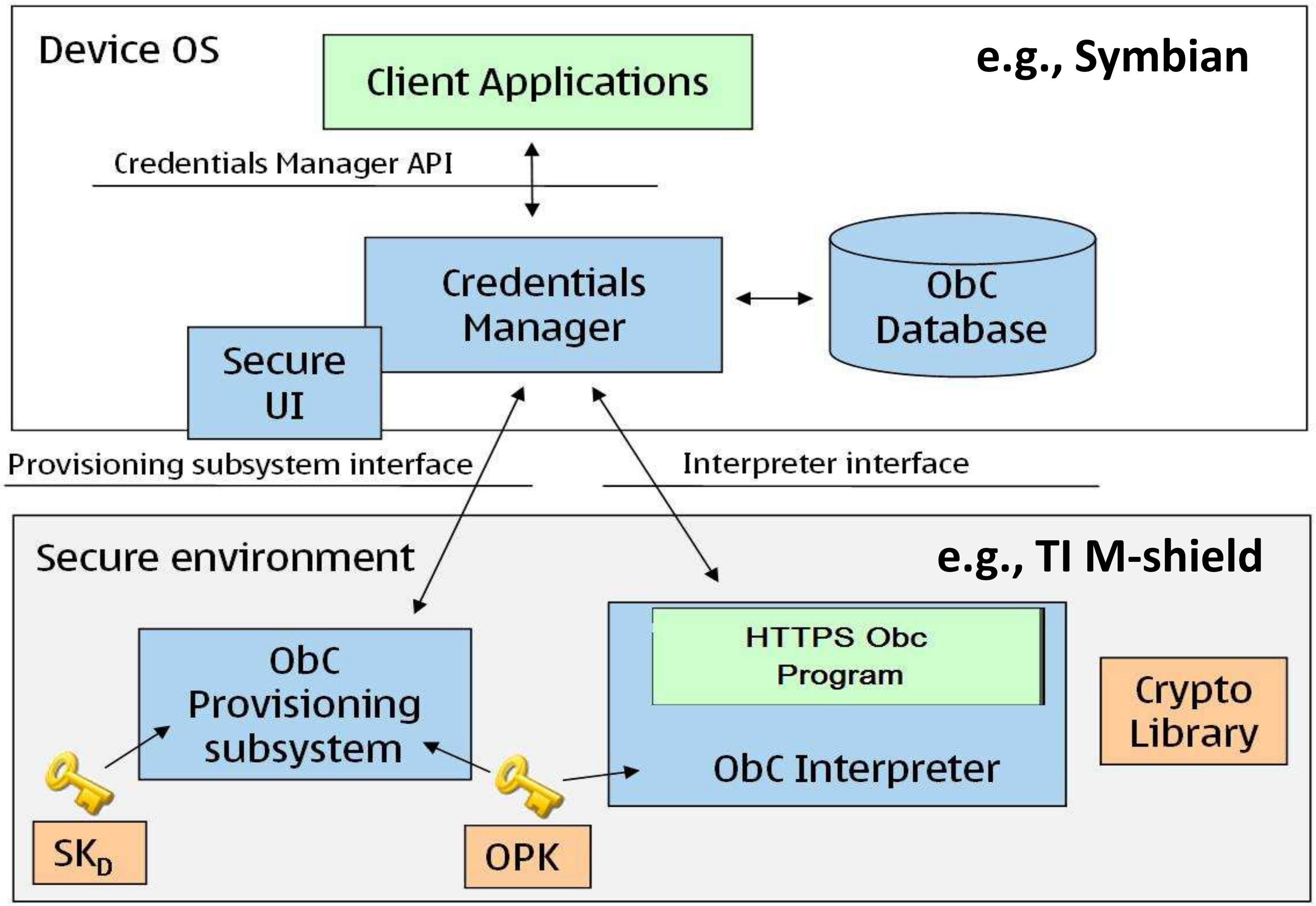
Impact layers: *Bus (DMA controllers), RAM (Memory Wrapper, Address Space Controller, Cache controllers), I/O (secure bridges for peripherals).*
Chips: *ARM SecureCore Chips (SC100/200/300)*



TrustZone: Writing Apps

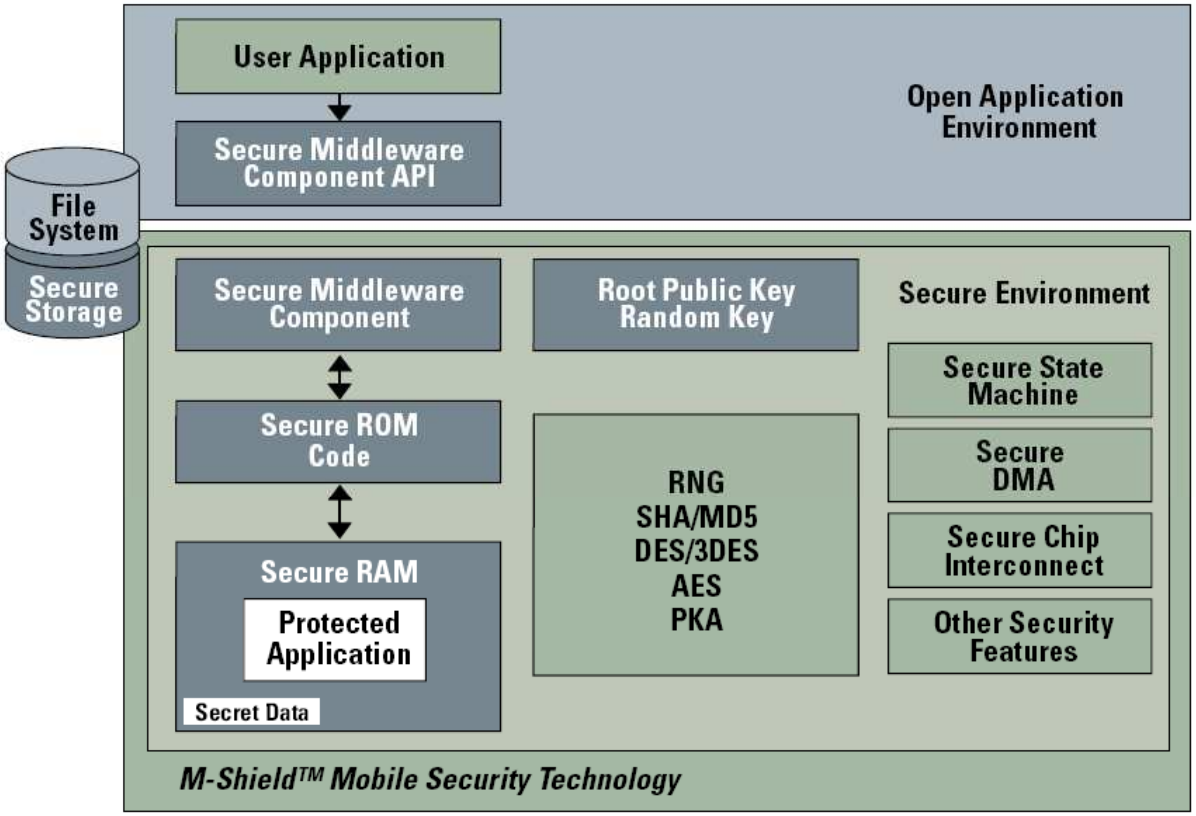


Nokia ObC



Device OS
Client Applications
e.g., Symbian
Credentials Manager API
Credentials Manager
ObC Database
Secure UI
Provisioning subsystem interface
Interpreter interface
Secure environment
e.g., TI M-shield
HTTPS Obc Program
ObC Provisioning subsystem
ObC Interpreter
Crypto Library
SK_D
OPK

Texas Instruments M-shield



Secure State Machine "guarantees policies while entering / executing / exiting secure environment", automatic **secured DMA** transfers (bus-level encryption ?), **secure chip interconnect**, **hardware crypto, ARM TrustZone**

SKINIT (AMD)/SENTER (Intel)

kernel says

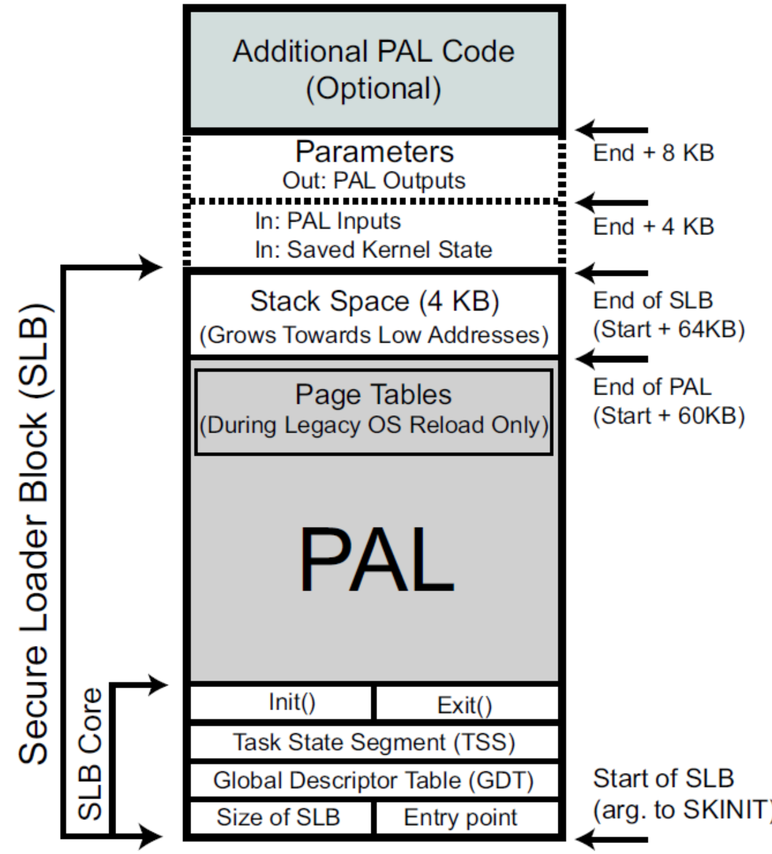
"SKINIT <address of SLB>"

CPU

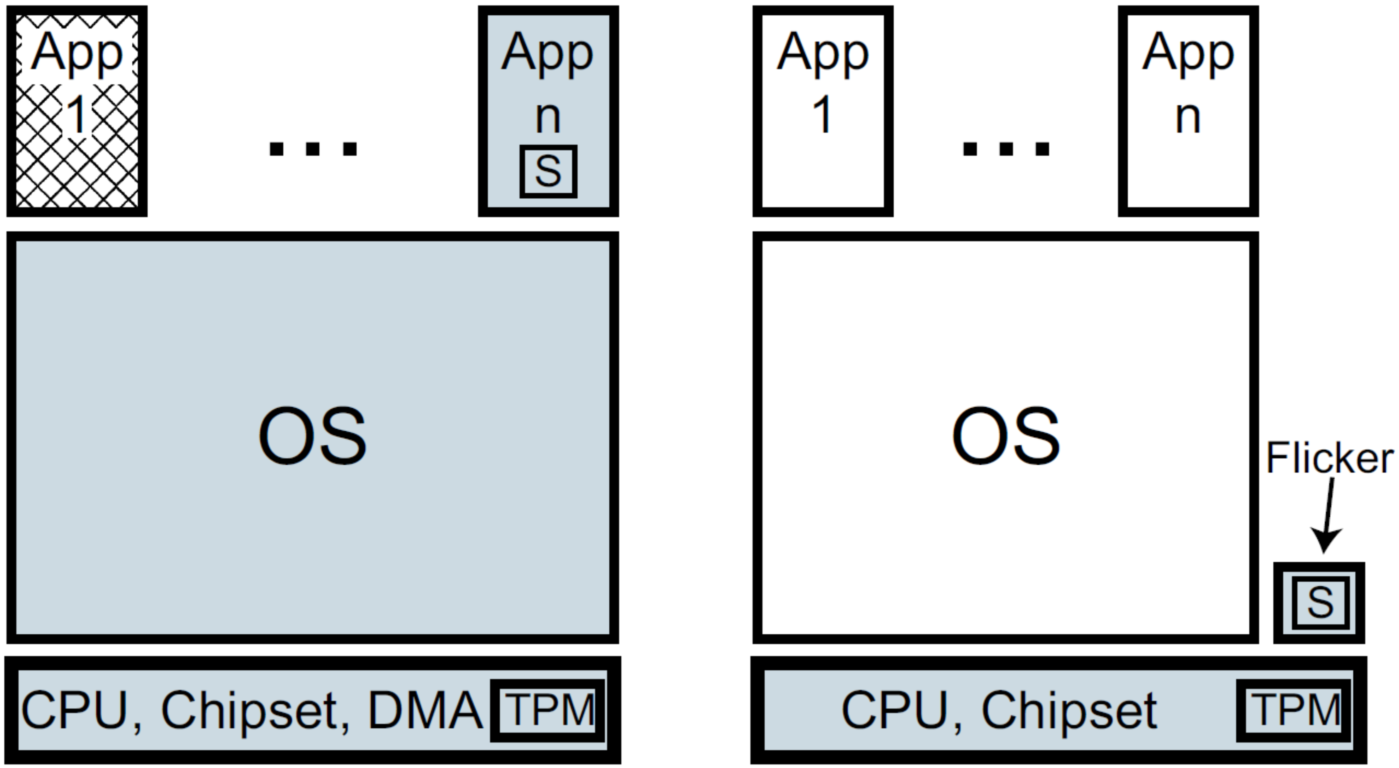
disables DMA to SLB,
interrupts, debugging
resets PCR 17-23
transfers SLB to TPM
enters flat 32-bit addressing
jumps to entry point

TPM

measures SLB into PCR 17



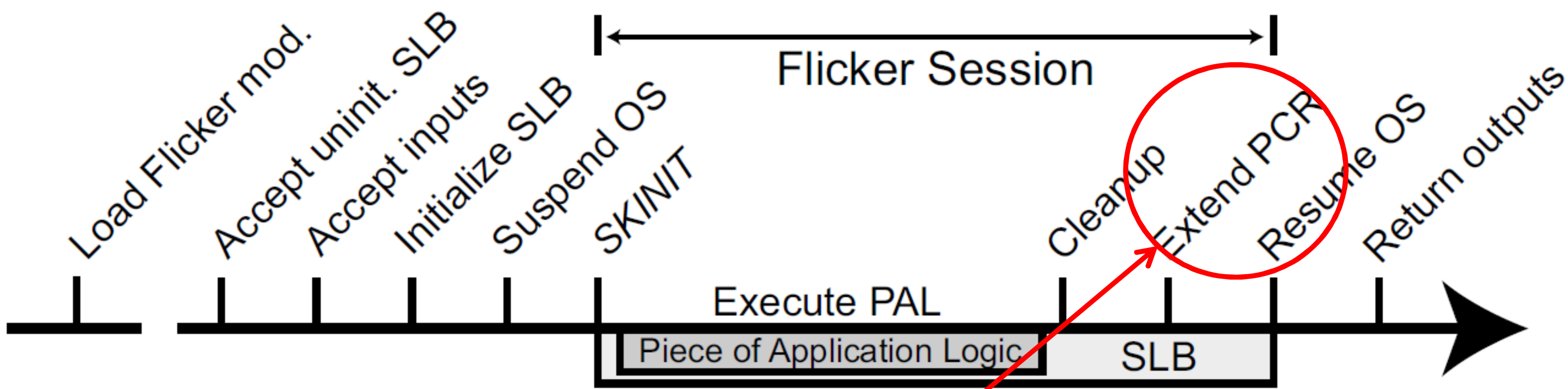
Flicker: using SKINIT (AMD)



Left: a traditional computer with an application that executes sensitive code (S). **Right:** Flicker protects the execution of the sensitive code. Shaded portions represent components that must be trusted; applications are included on the left because many run with super-user privileges.

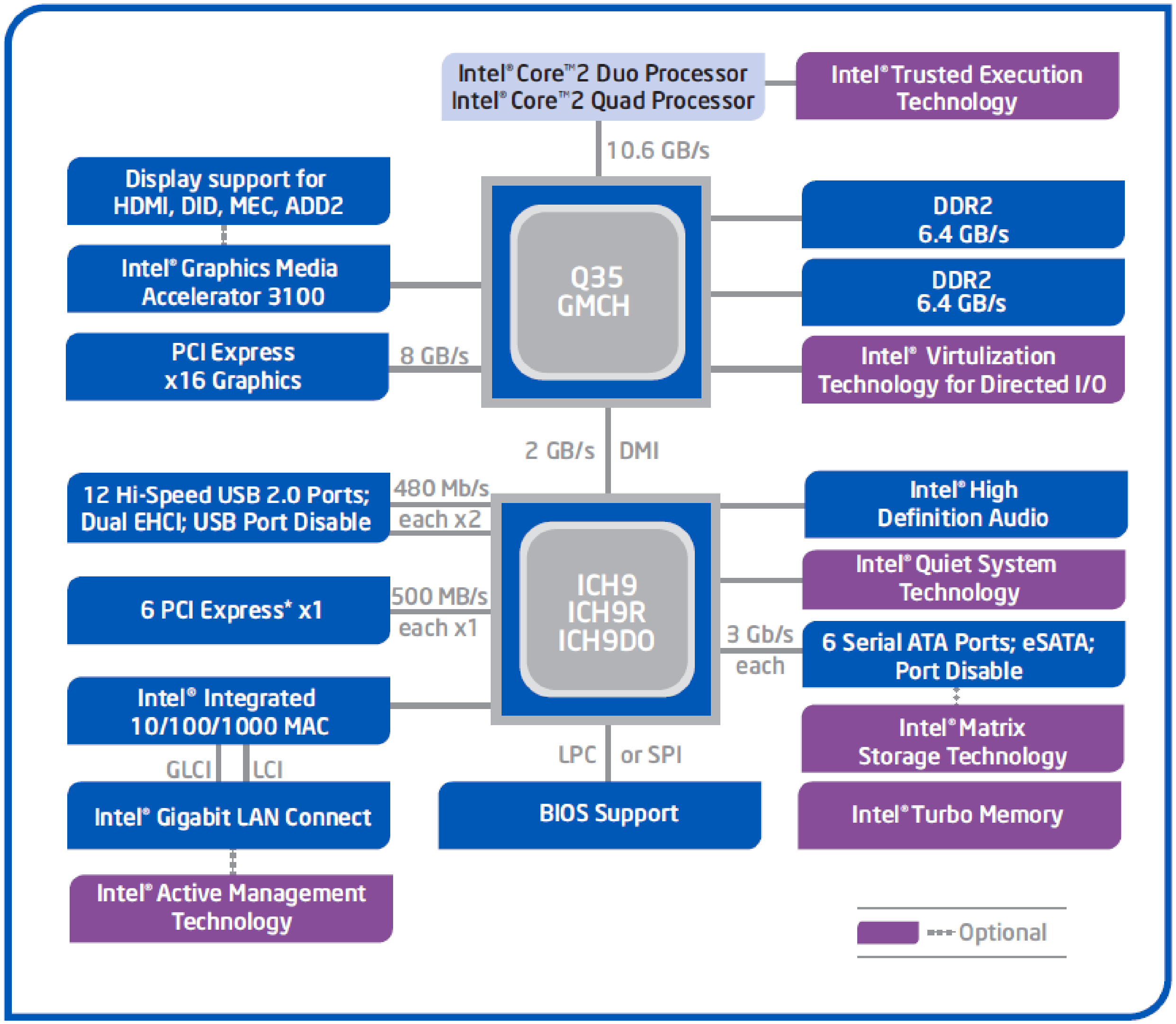
[McCune et al., "Flicker: An Execution Infrastructure for TCB Minimization", EuroSys 2008]

Flicker Session

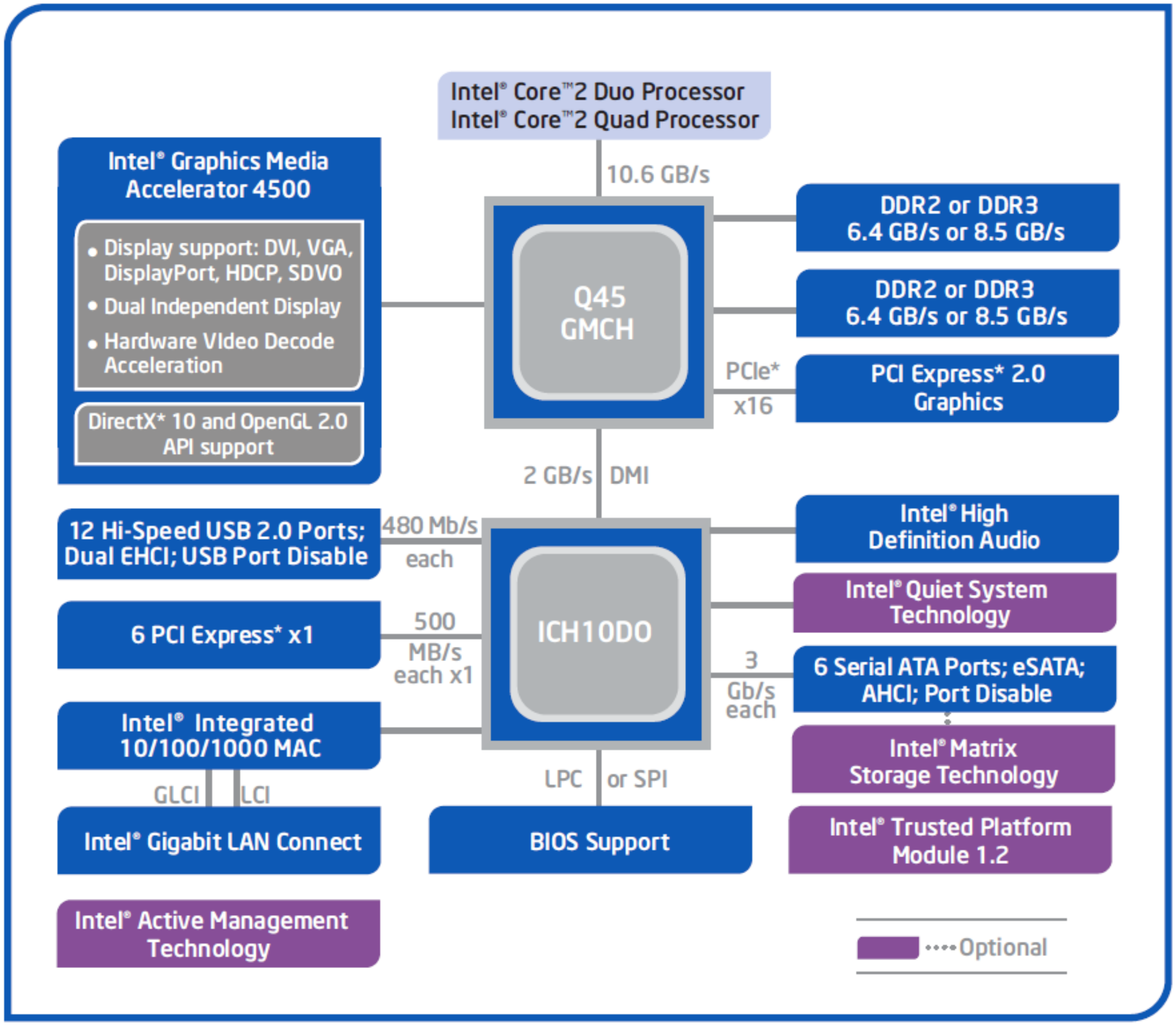


SLB core extends a well known value (function of input/output values of PAL + random nonce from remote party) in PCR 17: allow remote party to distinguish between values generated by the PAL (trusted) and those produced by the resumed OS (untrusted)

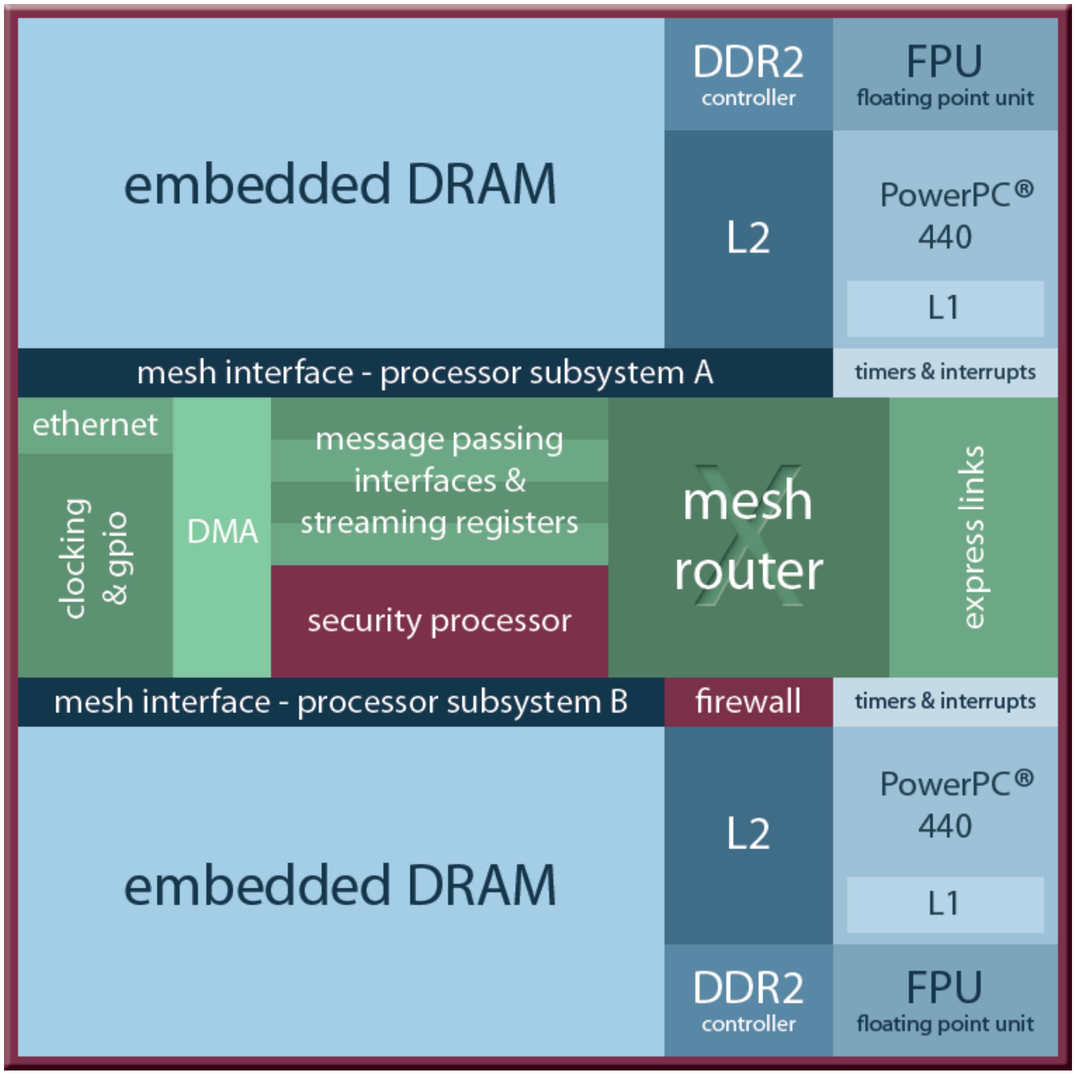
Intel Q35: Trusted Execution (TXT)

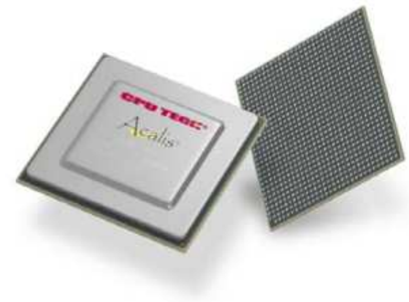


Intel Q45 express chipset



Acalis CPU 872 Secure Processor

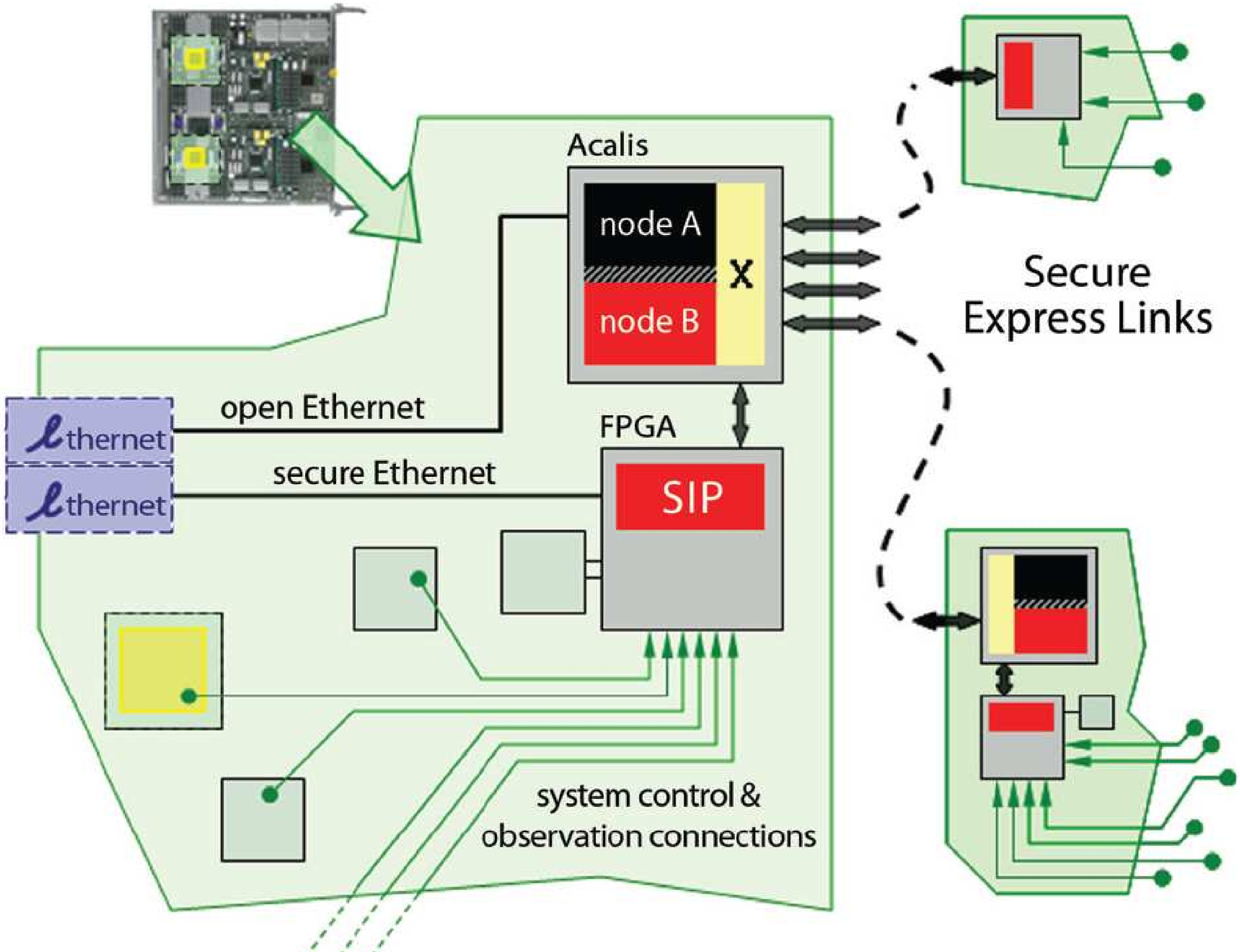




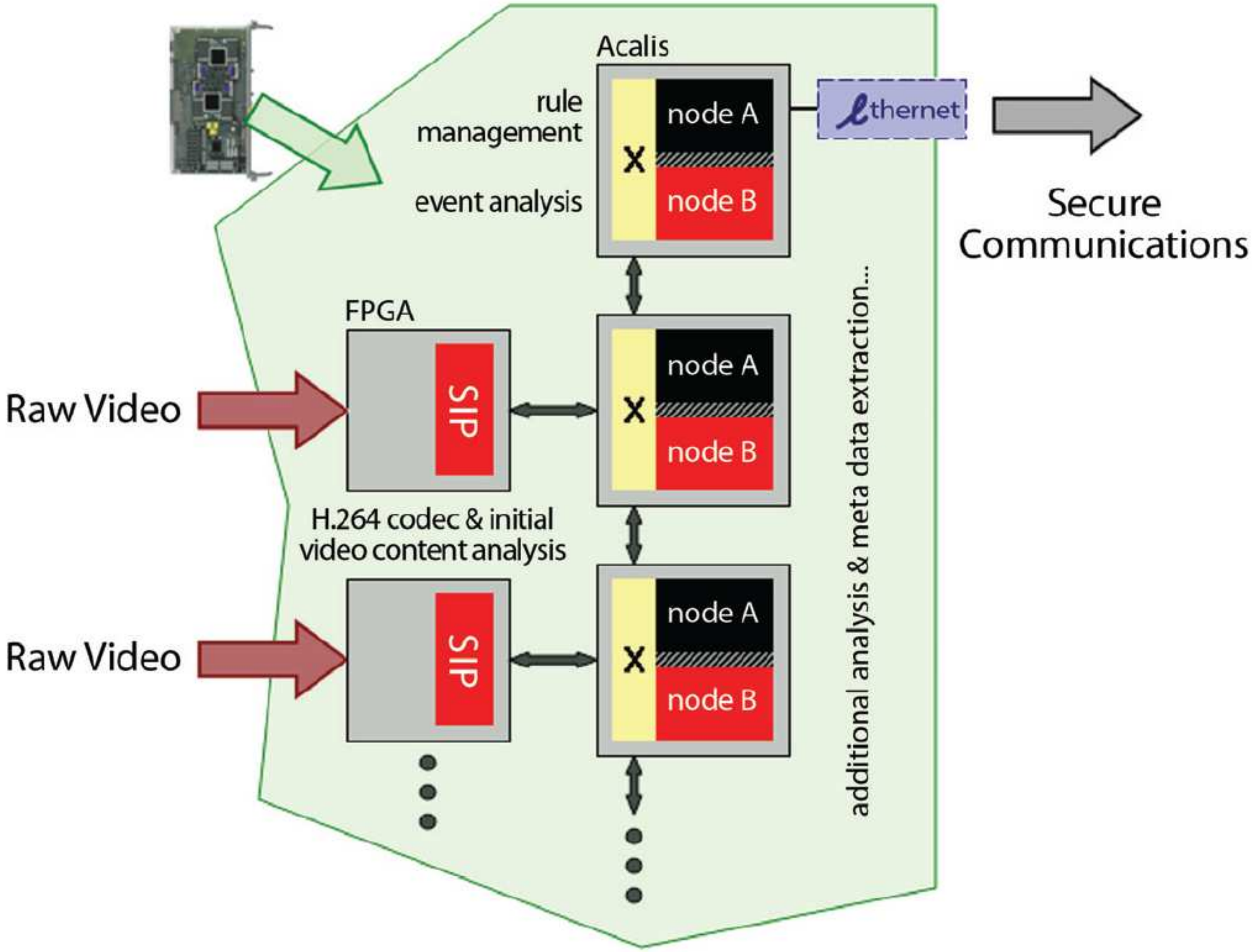
Secure boot
Encrypt/decrypt
Secure interconnect
Hardware firewall
Triggered zeroization signal
Unique serial code

More? Ryan?

CPU872: "Secure Anchor"



CPU872: "Secure Mobile Comp.E."



Secure Co-Processors

"A secure coprocessor is a general-purpose computing environment that withstands physical and logical attacks.

The device must run the programs that it is supposed to, unmolested. You must be able to (remotely) *distinguish between the real device and application, and a clever impersonator*.

The coprocessor must remain secure even if adversaries carry out destructive analysis of one or more devices. "

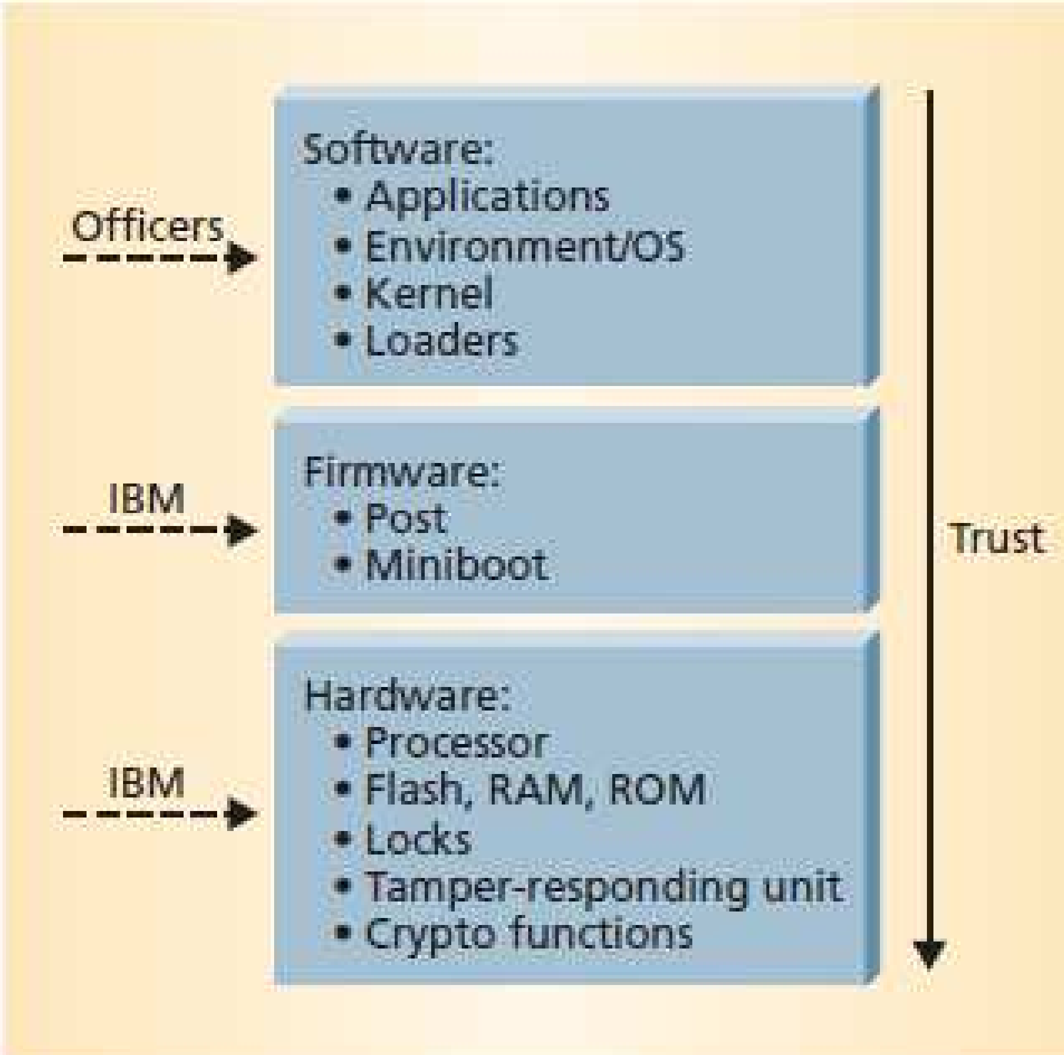
Dyad and Strongbox

J. D. Tygar, Bennet S. Yee, “Strongbox: A System for Self-Securing Programs”, 1991

Bennet S. Yee, “Using secure coprocessors”, PhD thesis, CMU, May 1994 (with Doug)

Trust Chain

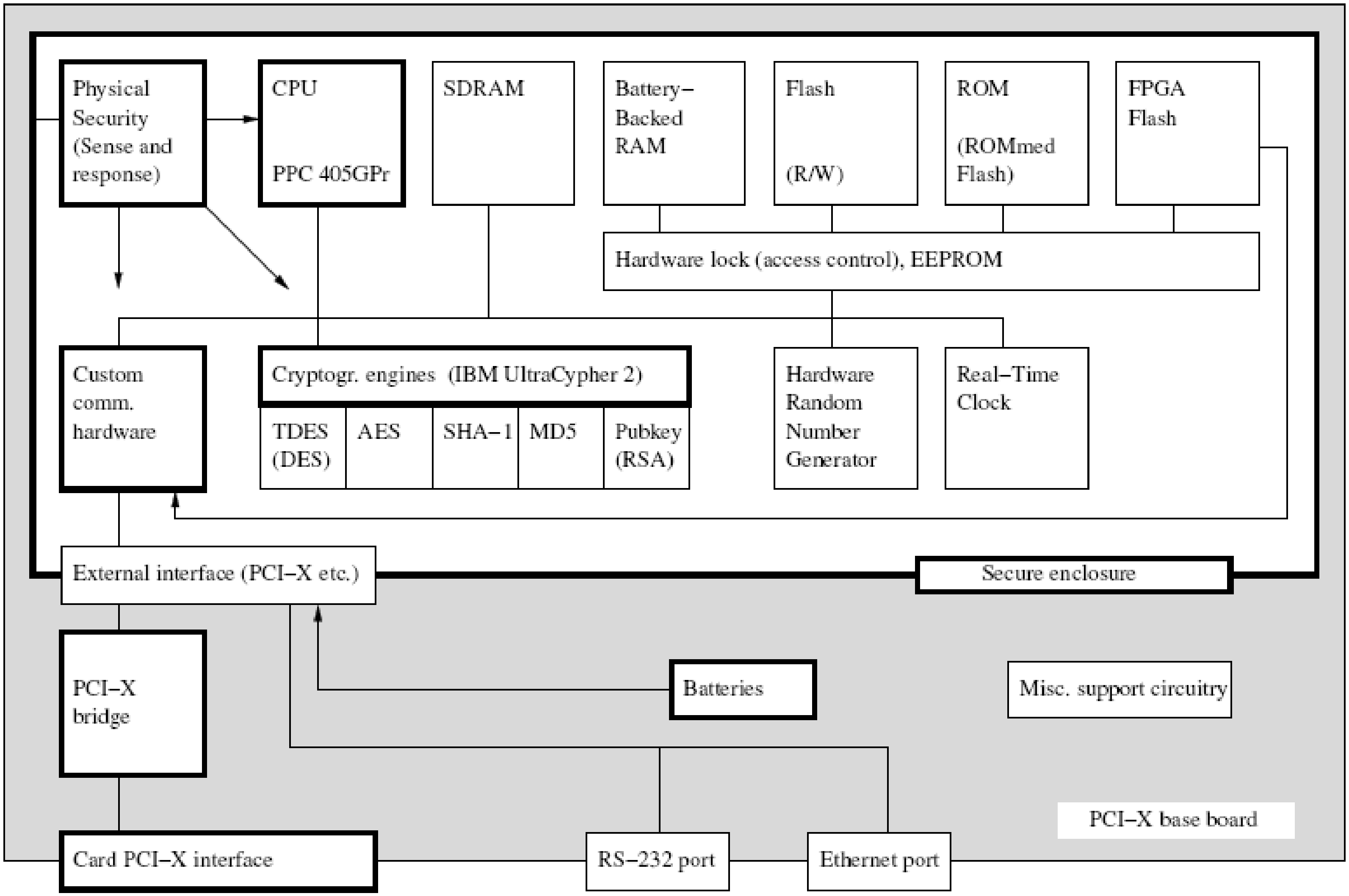


Officers
IBM
IBM
Software:
• Applications
• Environment/OS
• Kernel
• Loaders
Firmware:
• Post
• Miniboot
Hardware:
• Processor
• Flash, RAM, ROM
• Locks
• Tamper-responding unit
• Crypto functions
Trust

SCPU: IBM 4764-001 PCI-X

266MHz PowerPC. 64MB RAM. 64KB battery-backed SRAM storage. Crypto hardware engines: AES256, DES, TDES, DSS, SHA-1, MD5, RSA. FIPS 140-2 Level 4 certified.

IBM 4764-001 Architecture



IBM 4764-001 Segments

Segment	Description
0	Basic code The basic code manages coprocessor initialization and the hardware component interfaces. This code cannot be changed after the coprocessor leaves the factory.
1	Software administration and cryptographic routines Software in this segment: • Administers the replacement of software already loaded to Segment 1. • Administers the loading of data and software to segments 2 and 3. • Is loaded at the factory, but can be replaced using the CLU utility.
2	Embedded operating system The coprocessor Support Program includes the operating system; the operating system supports applications loaded into Segment 3. Segment 2 is empty when the coprocessor is shipped from the factory.
3	Application software The coprocessor Support Program includes a CCA application program that can be installed into Segment 3. The application functions according to the IBM CCA and performs access control, key management, and cryptographic operations. Segment 3 is empty when the coprocessor is shipped from the factory.

Performance

Function	Context	IBM 4764	P4 @ 3.4Ghz
RSA sig.	512 bits	4200/s (est.)	1315/s
	1024 bits	848/s	261/s
	2048 bits	316-470/s	43/s
RSA verif.	512 bits	6200/s (est.)	16000/s
	1024 bits	1157-1242/s	5324/s
	2048 bits	976-1087/s	1613/s
SHA-1	1KB blk.	1.42 MB/s	80 MB/s
	64 KB blk.	18.6 MB/s	120+ MB/s
	1 MB blk.	21-24 MB/s	
DMA xfer	end-to-end	75-90 MB/s	1+ GB/s
CPU freq		266MHz	3400Mhz
RAM		16-32MB	2-4GB

Observed: 43MB/s

Table 3: Hardware Performance Overview. SCPUs (e.g., IBM 4764-001 PCI-X) are orders of magnitude slower for general purpose computation than main CPUs (Pentium 4, 3.4Ghz, OpenSSL 0.9.7f). On the other hand, the crypto acceleration in the SCPU shows in direct speedup of crypto operations. Also optimized key setups might result in slightly different numbers for the main CPU.

Limitation: Heat

Dissipating heat while being tamper-proof.

Attacks

Possible Attacks

Probe Penetration

Power Sequencing (filter on power supply)

Radiation

Temperature Manipulation

Improper battery removal

Response (on tamper detection)

Zeroes its critical keys

Destroys its certificates

Is rendered inoperable

4764: Ranges

	Operating environment	Storage environment	Shipping environment
Temperature	+10° C - +40° C (+50° F - +104° F)	+1° C - +60° C (+33.8° F - +140° F)	-15° C - +60° C (+5° F - +140° F)
Relative humidity	8 - 80%	5 - 80%	5 - 100%
Wet bulb	+27° C (+80.6° F)	+29° C (+84.2° F)	+29° C (+84.2° F)
Pressure (minimum)	768 mbar	700 mbar	550 mbar

Think Break

relationship between "tamper-evident", "tamper-resistant", "tamper-proof" etc.

Miscellaneous “SCPU”s

netHSM

Networked shareable cryptographic resource for multiple servers. Just crypto, no tamperXXX CPU.

nShield

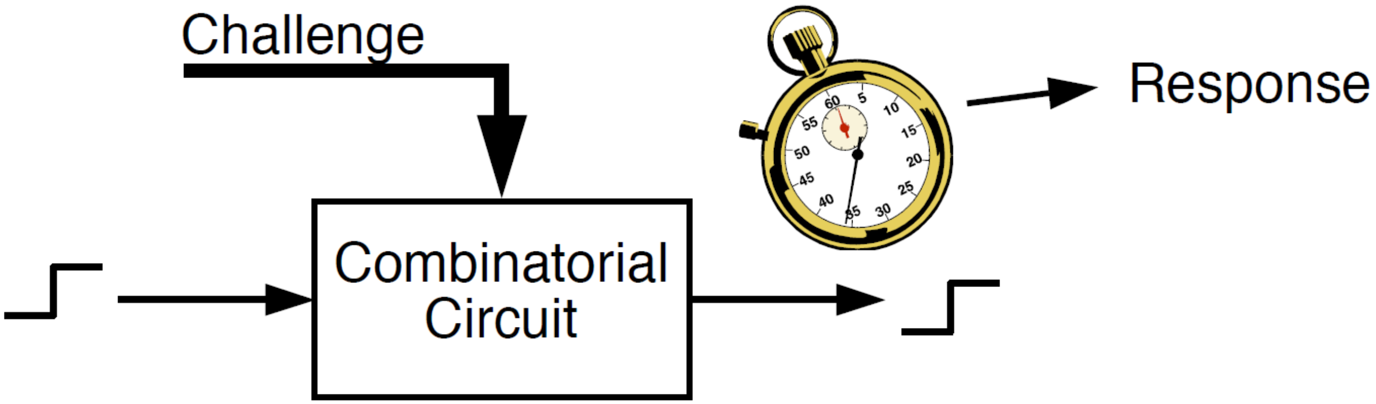
FIPS 140-2 level 2/3 TPM/SCPU

miniHSM

FIPS 140-2 level 3 mini SCPU

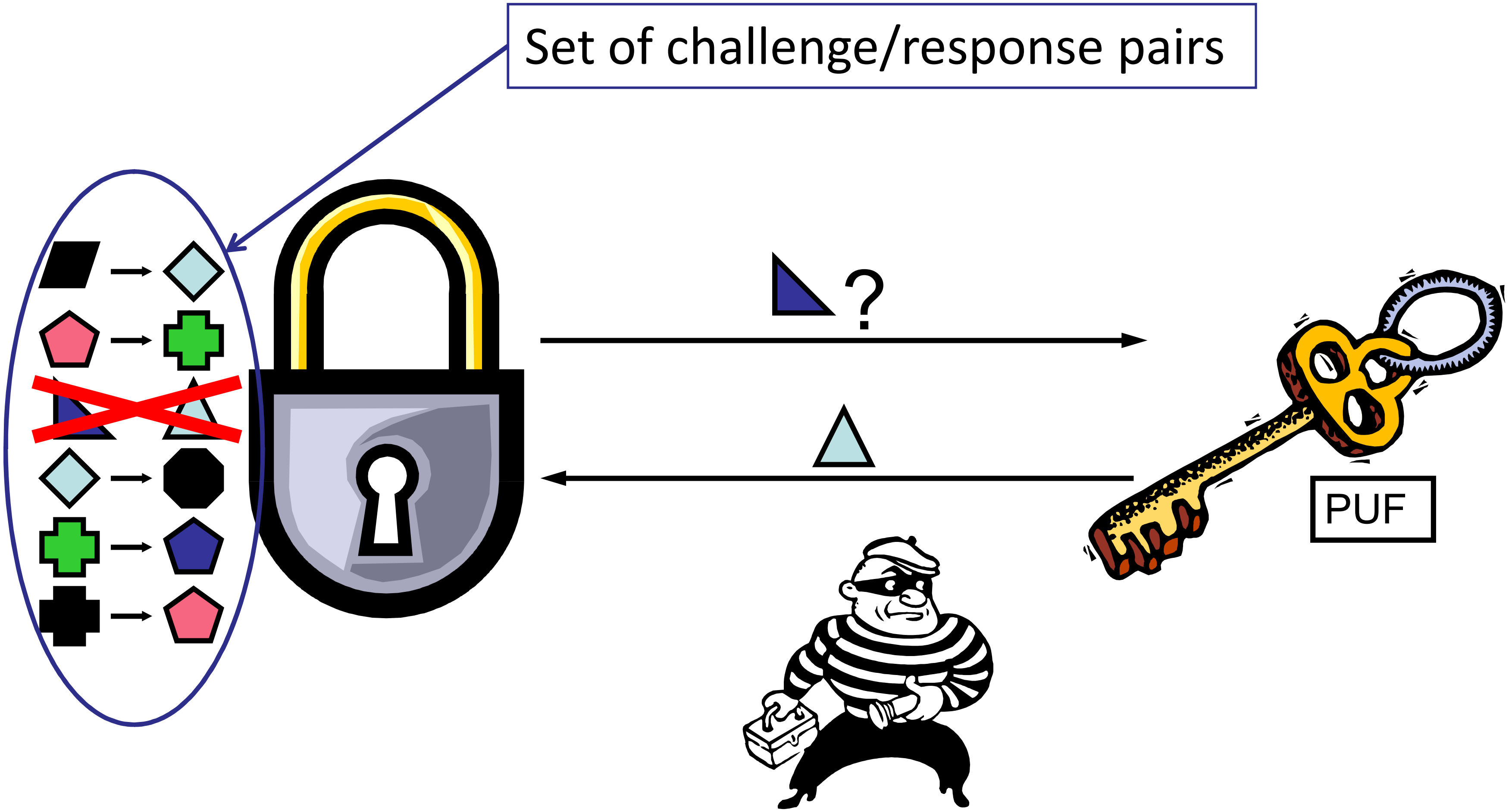
Physically Unclonable Function

Based on a physical system
Easy to evaluate (using the physical system)
Its output looks like a random function
Unpredictable even for an attacker with physical access



Silicon PUF: no two ICs are the same

PUFs as Unclonable Keys



PUFs: Applications

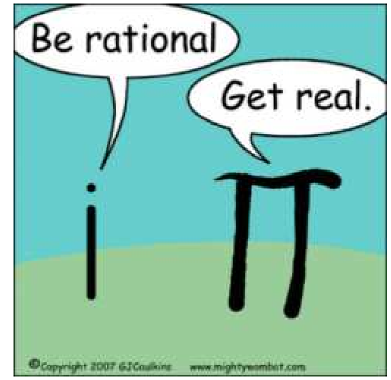
Anonymous Computation

Run computations remotely and ensure correct results. Return a certificate showing they were run correctly.

Software Licensing

Sell software which only runs on specific PUF-identified chip.

Areas



Finance

Online banking, ATMS

Commerce

Energy, Smart-grid, Healthcare

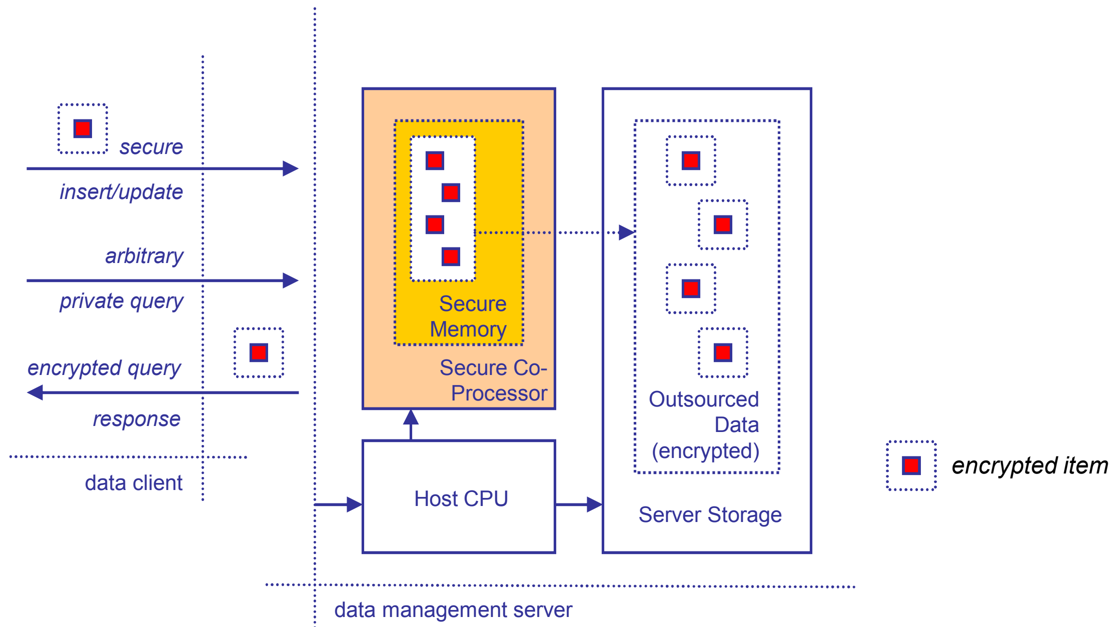
Government

Regulatory compliance

Military

Secure battle-field devices

Server-side SCPU

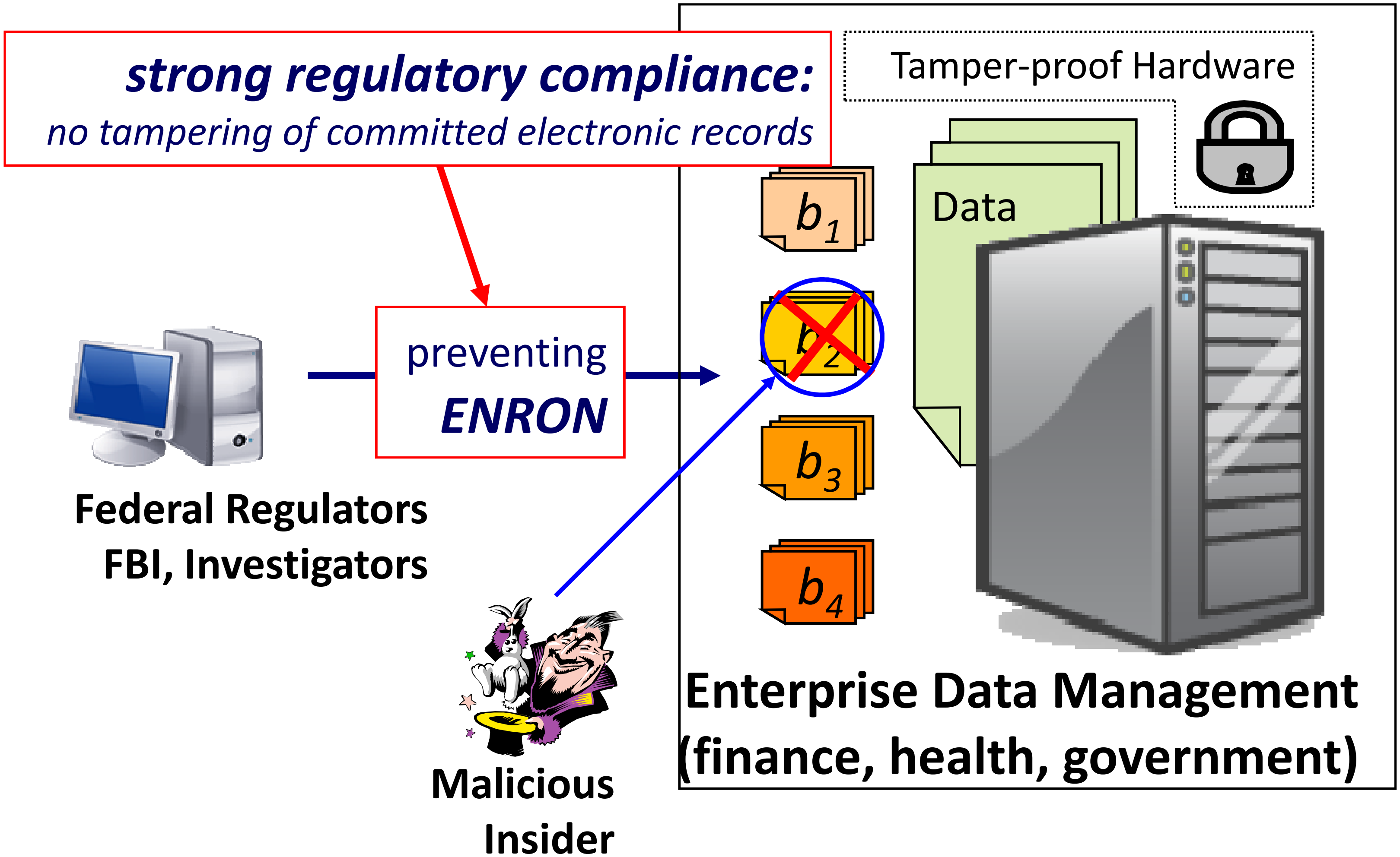


A secure co-processor on the data management side may allow for significant leaps in expressivity for queries where privacy and completeness assurance are important.

Selected Apps

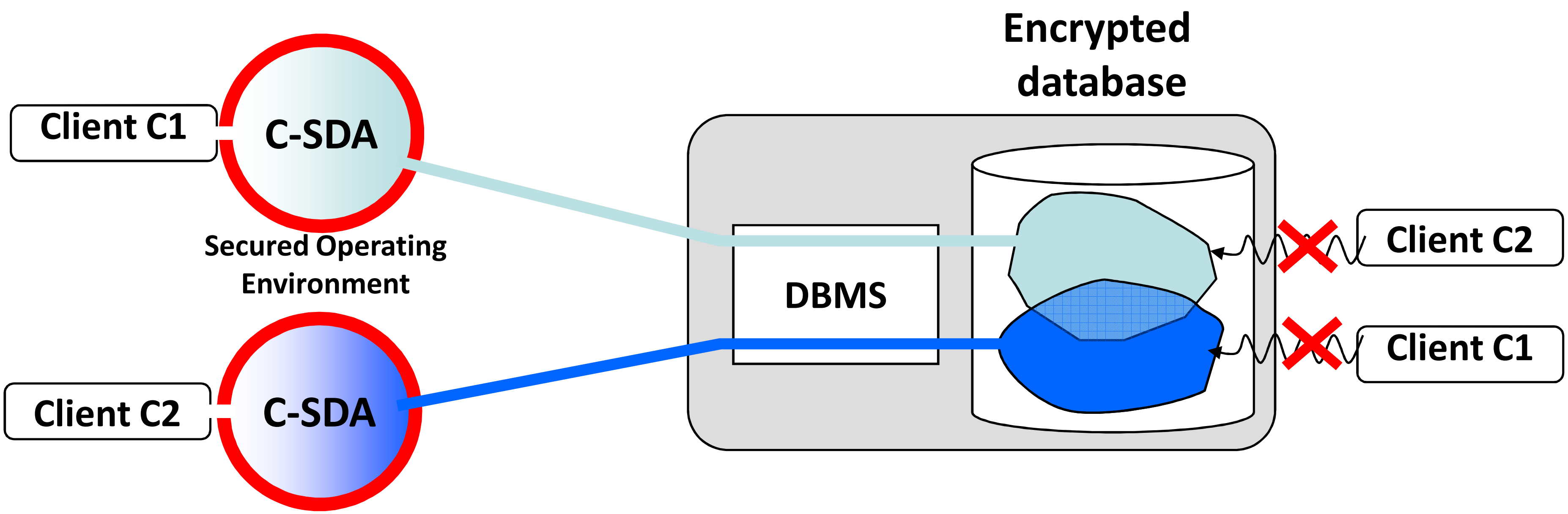
Regulatory Compliance Systems
Relational data processing
Oblivious Data Access ("practical PIR")
Chip-secured data access
Secure Data Storage

Regulatory Compliance



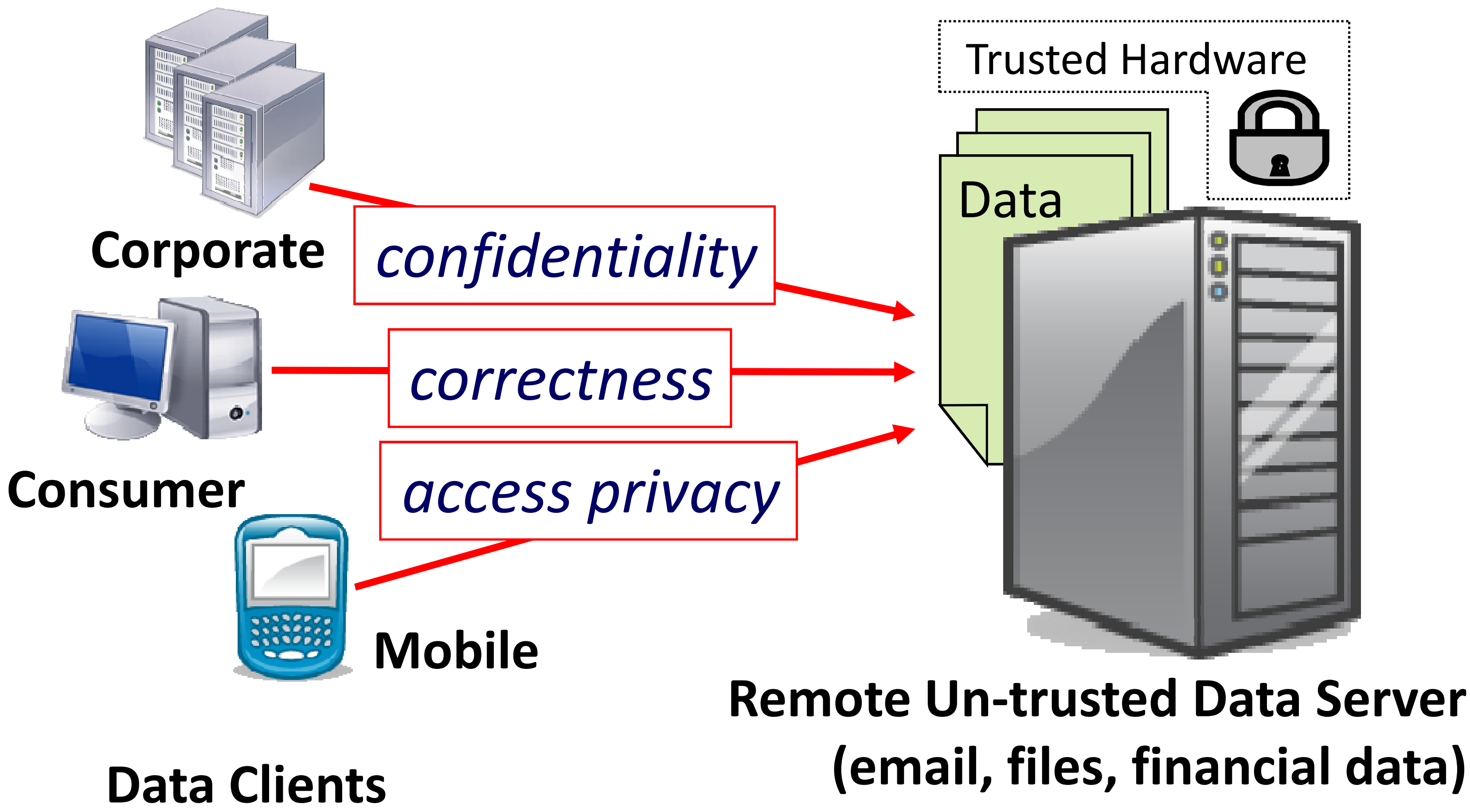
Chip-Secured Data Access

[Bouganim, VLDB 2002]



Smartcard: 32 bit RISC processor (≈ 40Mips), limited communication bandwidth (10 to100 Kbps), tiny RAM, writes in EEPROM very costly.

Secure Data Outsourcing



take home stuff

Understand *your* adversary

e..g., physical, insider vs. software-only, remote

Understand defenses and cost of attack

$\$10^1$ of overcoming defenses should not protect $\$10^6$

/bin/yes > /dev/null


Thanks ☺

some rare references

G. Edward Suh, Dwaine Clarke, Blaise Gassend, Marten van Dijk, and Srinivas Devadas. The aegis processor architecture for tamper-evident and tamper-resistant processing. Technical Report LCS-TM-460, MIT Laboratory for Computer Science, February 2003.

Benjie Chen and Robert Morris. Certifying program execution with secure processors. In HOTOS'03: Proceedings of the 9th conference on Hot Topics in Operating Systems, pages 23–23, Berkeley, CA, USA, 2003. USENIX Association.

David Lie, Chandramohan Thekkath, Mark Mitchell, Patrick Lincoln, Dan Boneh, John Mitchell, and Mark Horowitz. Architectural support for copy and tamper resistant software. In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), pages 168–177, November 2000.

Bouganim L., Pucheral P., "Chip-Secured Data Access: Confidential Data on Untrusted Servers", Int. Conf. on Very Large Data Bases VLDB 2002.

Ernesto Damiani, S.De Capitani di Vimercati, Sushil Jajodia, Stefano Paraboschi, Pierangela Samarati, "Balancing Confidentiality and Efficiency in Untrusted Relation DBMS", ACM CCS 2003

E. Mykletun and G. Tsudik, On using Secure Hardware in Outsourced Databases, International Workshop on Innovative Architecture for Future Generation High Performance Processors and Systems IWIA 2005.

IBM 4764 PCI-X Cryptographic Coprocessor (PCIXCC). Online at http://www-03.ibm.com/security/cryptocards/pcixcc/overview.shtml.

B. Bhattacharjee, N. Abe, K. Goldman, B. Zadrozny, V. Reddy, M. del Carpio, C. Apte,"Using secure coprocessors for privacy preserving collaborative data mining and analysis", Second ACM International Workshop on Data Management On New Hardware (DAMON) 2006

Kenneth Goldman, Enriquillo Valdez: "Matchbox: Secure Data Sharing", IEEE Internet Computing 2004

"Practical server privacy with secure coprocessors", IBM Systems Journal 2001, S. W. Smith, D. Safford

J. Marchesini, S.W. Smith, "SHEMP: Secure Hardware Enhanced MyProxy", Technical Report TR2005-532, Department of Computer Science, Dartmouth College, February 2005.

A. Iliev, S.W. Smith, "Protecting Client Privacy with Trusted Computing at the Server", IEEE Security and Privacy, March/April 2005

A. Iliev, S.W. Smith, "Private Information Storage with Logarithmic-space Secure Hardware.", 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems.

A. Iliev, S.W. Smith, "Prototyping an Armored Data Vault: Rights Management on Big Brother's Computer.", Privacy-Enhancing Technology 2002

E. Mykletun and G. Tsudik, "On using Secure Hardware in Outsourced Databases", International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, January 2005

Maheshwari, Vingralek, and Shapiro, How to build a trusted database system on untrusted storage, OSDI 2000

S. White, S. Weingart, W. Arnold, and E. Palmer. Introduction to the Citadel architecture: security in physically exposed environments. Technical Report RC16672, IBM Thomas J. Watson Research Center, March 1991.

B. Yee. Using secure coprocessors. PhD thesis, Carnegie Mellon University, May 1994.

S. Weingart. Physical security for the uABYSS system. In *Proceedings of the IEEE Computer Society Conference on Security and Privacy, pages 38–51, 1987.*

B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas. Controlled physical random functions. In *Proceedings of the 18th Annual Computer Security Applications Conference,* December 2002.

Karsten Nohl, Starbug, Henryk Plötz, and David Evans. Reverse-Engineering a Cryptographic RFID Tag. USENIX Security. August 2008.