Thomas J. Watson Research Center

The Importance of High-Assurance Security in Pervasive Computing

Paul A. Karger, Ph. D., Cantab.
karger@watson.ibm.com

14 March 2003

Agenda

- Secure Systems and Smart Cards Organization
- IBM4758 Secure Crypto Coprocessor
- Definitions
- Why is High Assurance Important?
- Caernarvon High-Assurance Smart Card OS
- Philips SmartXA2
- Details of High Assurance Methodology
- Unique Issues for Pervasive Computing
- Need to Improve Common Criteria for Pervasive and High Assurance
- Authenticating Attributes vs. Entities
- Conclusions

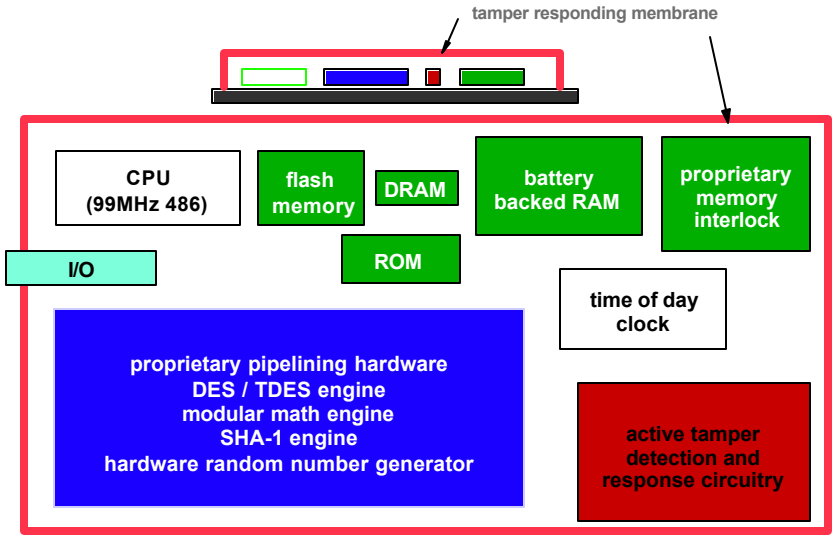
Secure Systems and Smart Cards

- Our focus: Software and tamper-protected devices that enable electronic commerce.
- Secure coprocessors
  - Physically secure devices that do fast crypto and general purpose computation.
  - IBM 4758 - First ever device to pass FIPS 140-1 Level 4
- Secure smart card operating system
  - Provable security, despite mutually hostile applications
- Side-Channel Attacks and Defenses
  - power analysis
  - RF
- Applications which require these devices

IBM 4758 PCI Cryptographic Coprocessor

- Performs high speed cryptographic operations
- Provides secure key storage
- Detects physical attacks: probe, voltage, temperature, radiation
- Programmable!
- Secure configuration and field upgrades
- FIPS 140-1 overall level 4 certified (hardware and microcode)
- PCI interface, drivers for WinNT, Linux, AIX, OS/2, OS/400, OS/390

4758 Hardware Architecture



Definitions

- Pervasive Computing
  - Lots of small computers, such as smart cards, PDAs, cell phones, appliances, etc.
- High-Assurance Security
  - A level of security sufficient to convincingly resist attacks from sophisticated, well-motivated, and well-funded penetrators
  - Built to very high standards
    - Orange Book: B3 or A1
    - ITSEC: E5 or E6
    - Common Criteria: EAL6 or EAL7
  - Systems built to lower standards will demonstrably fail against sophisticated penetrators
  - May or may not actually have a certificate
    - but the process of third-party evaluation is very important to motivate developers and managers not to take shortcuts

Who's a Sophisticated Penetrator?

- Back in the 1970s and 1980s, almost all attackers wouldn't even qualify as today's "script kiddies"
- The Orange Book B3 and A1 systems were designed to resist sophisticated attack by technically sophisticated KGB agents
- The US DoD was concerned that the KGB might
  - exploit buffer overflows or other flaws
  - might spread Trojan horses or viruses on defense networks
- Back then, typical commercial attackers guessed passwords and not much else - high assurance was only for the military
- TEMPEST attacks were classified
- Today's "script kiddies" are routinely attacking commercial entities with the worst that the DoD expected from the KGB in the 1970s and 1980s!
- TEMPEST attacks are now called DPA or side-channel cryptanalysis and Ross Anderson's students do them routinely
- Today's commercial systems face threats on a daily basis that are WORSE than what the DoD expected at the height of the cold war!

Who Needs High Assurance?

- Traditional low-assurance security systems get defeated regularly: no reason to expect that they will resist sophisticated penetration
  - Results from IBM's Global Security Analysis Laboratory experience
    - and from IBM's security consulting ethical hackers
    - and similar results from many other security consultants
- But the limited numbers of true high-assurance systems really do provide a quantum leap better security
  - not perfect - nothing is perfect
  - but demonstrably a vastly higher standard of penetration resistance
- Given the extremely sophisticated threats of today, who needs high-assurance security?
  - Anyone with high-value data to protect
  - Anyone providing critical infrastructure
  - Many classes of pervasive computing devices, although not all

Goals of Caernarvon High-Assurance Operating System Project

- Develop a secure operating system
  - for pervasive devices (smart card, GSM phone SIMs, USB tokens, etc.)
  - use hardware to enforce the security
  - allow controlled sharing of data
- provably secure - evaluated by an independent third party at a very high level under Common Criteria
- demonstrate feasibility, with eventual commercial outlet

Goals of Caernarvon High-Assurance Operating System Project *(continued)*

- the smart card must enforce separation and protection of each of these companies applications
- support both native and interpreted applications
- field loadable applications
- code written by different independent or mutually suspicious companies
- allow programming by anyone
- example: a bank issues a corporate travel smart card to employees of a large company
- has contracts with hotel chains, airlines, card rental agencies
- each of hotel chain, airline or care rental agency writes its own applications, using its own programmers, independently of the others
- the applications can be loaded as required

Caernarvon Castle

Thomas J. Watson Research Center

Caernarvon Castle

- Caernarvon is probably the most majestic of the 8 castles (Aberystwyth, Beaumaris, Builth, Caernarvon, Conway, Flint, Harlech and Rhuddlan) constructed by King Edward I, to suppress rebellions by the Welsh.
- Caernarvon castle was constructed during the period 1283-1327. It was designed to be a palace as well as a fortress, and to be the center of administration of North Wales.
- In modern times, it is the location used for the installation of the Prince of Wales.

14 March 2003
© 2003 IBM Corporation

Philips Semiconductors' SmartXA2 Microcontroller

- supervisor/user mode
- memory protection (hardware firewall)
- 16-bit CISC processor (extension of 8051 and much faster)
- lots of memory (128K ROM, 64K EEPROM, 5K RAM)
- 2048-bit modular math hardware assist for RSA, DSA
- DES/3DES hardware assist
- hardware random number generator
- UART with DMA

Philips Semiconductors' SmartXA2 Microcontroller

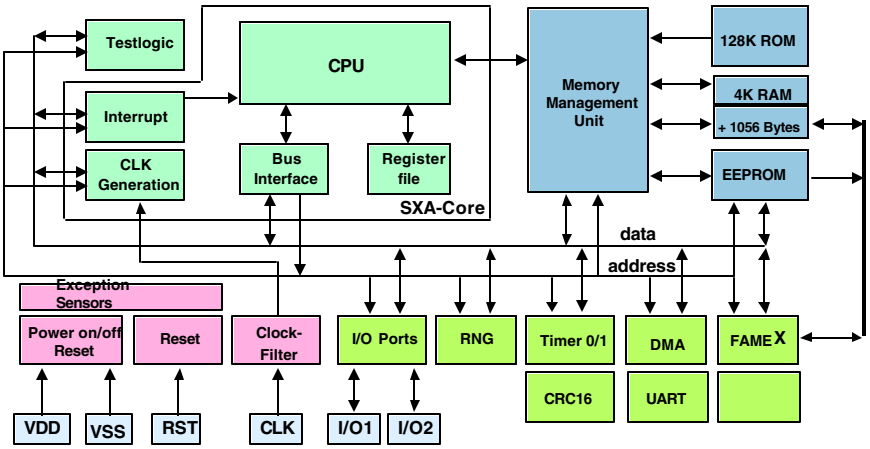
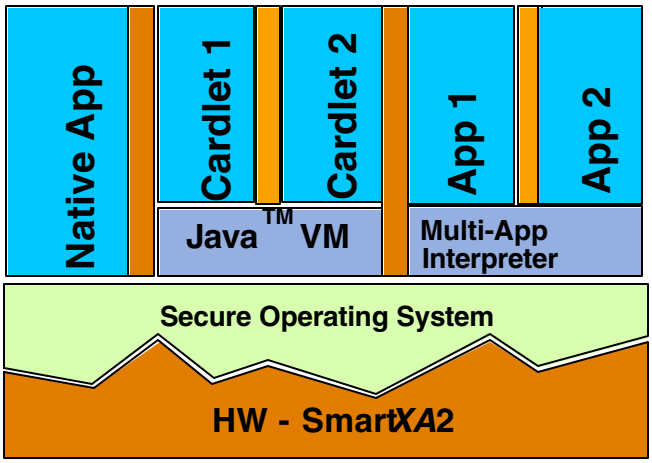


diagram courtesy of Philips Semiconductors

Software Layers on the SmartXA2



Thomas J. Watson Research Center

Why not just get a software-based interpreter validated?

14 March 2003
© 2003 IBM Corporation

Proving that a software-based approach is secure

- In a nutshell, it's a harder problem.
  For example, for JavaCard:
- To prove that a software-based JavaCard security model is secure, we have to prove that:
  - There is no way to generate a bogus address in downloaded code. This is very hard.
  - There is no way to download a bogus address into the card.
  - The compiler must run in a secured system like the IBM 4758.
  - The byte code verifier and the code signer must also run in a secured system like the IBM 4758, and must be formally verified.

Proving that a hardware-based approach is secure

- To prove that a hardware-based approach is secure we have to prove that:
  - The hardware memory management and supervisor state work.
  - The OS makes proper use of the hardware protection features.
  - Even if there is a bogus address in downloaded code, it can do no damage.
- Most high assurance systems use this approach.

CC High Level Assurance Requirements

- Third-party evaluation
  - all steps of design reviewed and approved by independent evaluators
  - evaluation checked by certifying body, before certificate issued
- Define the Target Of Evaluation (TOE)
  - Thoroughly specify/document entire design
  - Security Policy must be stated
- Formal Model of Security Policy
  - proven correct and consistent
- Top Level Specification of System Design
  - must correspond to the formal model of security policy

CC High Level Assurance Requirements *(continued)*

- Apply best available software engineering techniques
- Extensive Documentation of All Code Modules
  - informally shown to correspond to the top level specifications
- Extensive Testing
  - both module by module and full system
  - extensive test documentation is required
  - security penetration testing

Security Target and Functional Specification

- Target of Evaluation (TOE) defined by the Security Target (ST)
  - contains a description of the TOE
  - specifies the TOE Security Functions (TSFs)
- Highest level description is the Functional Specification (FSP)
  - defines external APIs and their correspondence to TSFs
  - must be formally modeled
  - must demonstrate correspondence with ST, and completeness of representation

High Level and Low Level Designs

- Next level description is the High Level Design (HLD)
  - describes the structure of the TOE as subsystems
  - must be formally modeled
  - EAL7 requires formal proof of correspondence with model of FSP
- Next comes the Low Level Design (LLD)
  - describes the structure of the TOE in terms of modules
  - describe all interfaces to the modules
  - must demonstrate correspondence to HLD, and completeness of representation

Testing

- must test everything for EAL7 - the TSFs, the various functions, etc.
  - documentation must demonstrate the coverage and depth of testing
- Vulnerability Analysis
  - EAL6/7 requires a systematic search for covert channels and vulnerabilities
  - documentation must detail these, and demonstrate that the search was systematic
  - must document all insecure states, such as may provide opportunities for misuse

Thomas J. Watson Research Center
IBM

Development and Delivery

- Development Facilities
  - document tools etc.
  - document development facilities and security measures
  - document development methodology, configuration management, etc.
- Delivery Procedures
  - must define and document delivery procedures, to ensure adequate secrecy and integrity of the delivered product
  - must use encryption etc. tools that are acceptable to the certification authority
  - it appears that each certification authority accepts only its own proprietary tools

14 March 2003
© 2003 IBM Corporation

Security Evaluations: Advantages and Advice

- "Never trust the vendor!"
- Legal requirements
  - German Digital Signature Law (ITSEC E4, CC EAL5)
  - US Government (FIPS 140-1)
- Know what level of security is enough to protect your asset.
- "Designed to meet" does not equal "Meets" or "Validated".
- Ask to see evaluation reports. Note that evaluation reports of specific products are often considered confidential and may not be available to you.

Unique Issues of Pervasive Computing for High Assurance

- Traditionally, high assurance evaluations have assumed that the system in question is physically protected. Therefore, only the logical software issues have mattered.
- Reality of Pervasive Computing is that devices may be stolen, may be subject to sophisticated physical attacks, and may have to inter-operate with large numbers of other devices, any of which might be actively hostile.
- Hardware evaluation is just as important as software evaluation in the pervasive environment.
- France, Germany, the Netherlands, and the UK have developed supplementary documents to assist in Common Criteria evaluation of smart cards and similar pervasive devices
  - not all countries are party to these documents
  - these documents do not focus on high assurance
  - some technical problems remain

Interaction of Covert Channels with Side-Channel Attacks

- Covert channel attacks have historically been software-based attacks that could allow Trojan horses to leak information in violation of mandatory security policies
- Traditional covert channel countermeasures have all been based on clever software approaches in the operating system
- Side-channel attacks (such as power analysis) have focused efforts on countermeasures to prevent the leaking of cryptographic keys, and have not considered covert channels, as lower assurance systems don't have to consider covert channels
- Combining a Trojan horse covert channel attack together with a physical side channel attack (such as DPA or RF emanations) means that traditional covert channel countermeasures are suddenly insufficient, and the side-channel countermeasures that have focused exclusively on protecting keys are ALSO insufficient.
- Research is needed in this area.

Hardware Random Number Generators

- Hardware Random Number Generators have been a problem with evaluations.
  - FIPS 140-1 didn't permit their use at all
- BSI has issued AIS 31, "Functionality classes and evaluation methodology for physical random number generators"
  - provides good information on how to handle statistical failures of hardware random number generators and strongly recommends the use of FIPS statistical tests to ensure that the hardware random numbers are good
  - BUT -- AIS 31 doesn't provide any guidance at all for dealing with side-channel attacks on the testing process
    - the act of carrying out the FIPS tests on the random numbers is likely to leak the value of the random numbers
    - this is worse than not testing the numbers at all, since there is only a small probability that a hardware random number generator will fail, but there is a very high probability that a power analysis attack would succeed
- Problem is not unsolvable, but AIS 31 needs to be revised

Composite Evaluations - Hardware and Software

- The supplementary documents from the 4 European Certifying Bodies try to deal with how to combine separate hardware and software evaluations for smart cards and similar pervasive devices
  - address legitimate concerns that hardware evaluation results may be proprietary and may even be done in a different country than the software evaluations
  - define what information must be released to software developers and evaluators and what may be withheld
- Problem is that these composite evaluation guidelines did not really consider the needs of a high assurance evaluation
- At high assurance levels, the software developers and evaluators need LOTS more information about the hardware, in order to deal with issues, such as covert channels and highly sophisticated attacks on the hardware.
  - information that may be suitable to withhold at low assurance levels may be absolutely essential at high assurance levels
  - strictly following these existing composite evaluation guidelines can seriously jeopardize the success of a high assurance evaluation
- Much more work is required here

Authentication of Entities or Attributes?

- On Thursday, paper by Creese, Goldsmith, Roscoe, and Zakuddin raised important issues that authenticating the security attributes of a pervasive device may be more important than authenticating its identity.
- IBM's work on authentication for Caernarvon confirms this result
- Caernarvon authentication not only identifies who the device is, but also the mandatory security attributes of the device.
  - Caernarvon OS then uses those attributes to make access control decisions on files, etc.
  - Caernarvon mandatory access control policy described in papers at 2000 ESORICS and at 2000 EUROSMART Security Conference
- Papers on Caernarvon authentication approach are under preparation

Conclusions

- High assurance security is becoming more and more important, not just for pervasive computing, but all networked computing of any kind
- The attackers are getting MUCH more sophisticated, and only high assurance techniques have been shown to resist such sophisticated penetrators
- Pervasive computing raises a number of new issues for high assurance evaluations, and only some of them have been addressed so far.

For more information

Paul A. Karger
IBM, T. J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598, USA
karger@watson.ibm.com