IBM Research

Lessons Learned Building a High Assurance Smart Card Operating System

Elaine R. Palmer, Senior Technical Staff Member, IBM Research

ACM Distinguished Engineer

Paul Karger (deceased), Research Staff Member, IBM Research

Suzanne McIntosh, Senior Software Engineer, IBM Research

David Toll, Research Staff Member, IBM Research

Samuel Weber, Program Director, National Science Foundation

Sep 2012

This is how we looked when we started. . .

Clip art used with permission from Microsoft

This is how we looked when we stopped. . .

Clip art used with permission from Microsoft

This talk

Shares the lessons we learned

. . . so that you might learn from our experience.

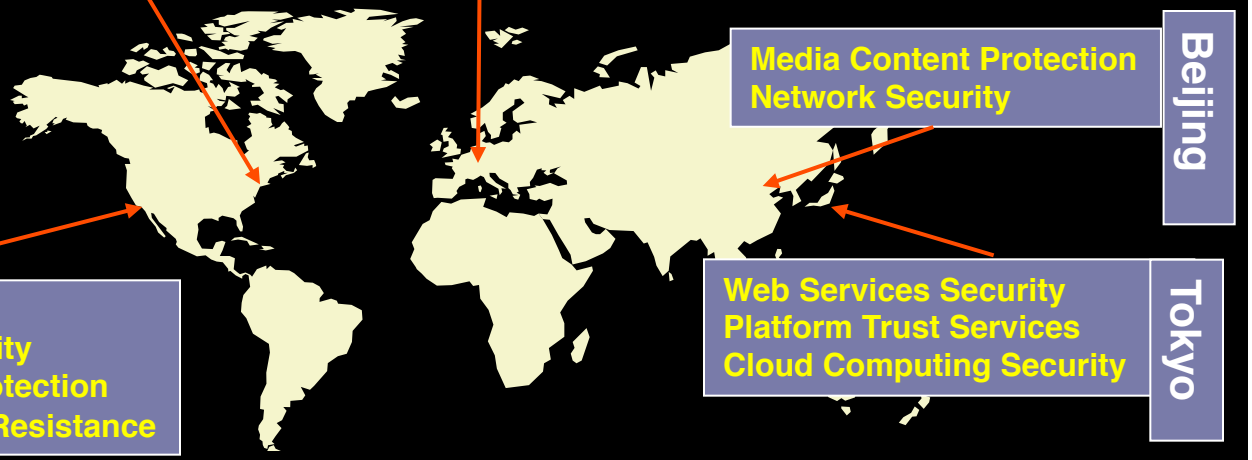
Talk Outline

- Who are we?
- Our requirements and goals
- What did we do?
- What did we learn?

Security & Privacy Research at IBM

Watson
- Enterprise Information Management
- Secure Platform Technologies
- Cybersecurity Analytics (Stored and Streaming)
- Mobile Security
- Fully Homomorphic Encryption
- Cryptographic Foundations
- Virtualization and Cloud Computing Security
- Security for the Smarter Planet

Zürich
- Authentication Solutions
- Data Storage Security
- Formal Verification Tools
- Identity Governance
- Internet Transaction Security
- Security Policies
- Virtualization and Cloud Computing Security



Beijing
- Media Content Protection
- Network Security

Tokyo
- Web Services Security
- Platform Trust Services
- Cloud Computing Security

Almaden
- Mobile Security
- Information Security
- Media Content Protection
- Software Tamper Resistance

In 1998, our colleagues asked us to solve a problem with smart cards and mobile phone id cards.

- Cards were moving from single function to multi-function
- The card operating systems weren't designed to handle this in a generic way (they could on a case-by-case basis).

Clip art used with permission from Microsoft

When we began, the state of the art was. . .

Clip art used with permission from Microsoft

- No separation of apps from each other or from the OS
- No open standard on-card interpreter like Java™
- Concerned customers used single function cards
- Multiple apps were evaluated in combination with each other
- Hardware had no support for user vs. supervisor mode

Our goal:

Develop an operating system for smart cards to make them more general purpose, and less vulnerable to programming mistakes and deliberate attacks.

Goals and Requirements

- Provide a level of security sufficient to resist sophisticated, well-motivated, and well-funded attackers
- Build to very high standards

  Common Criteria: EAL6 or EAL7

  Orange Book TCSEC: B3 or A1

  ITSEC: E5 or E6

  The process of third-party evaluation motivates developers not to take shortcuts

- Contrast with the standard for commercial avionics software, DO-178B level A, which emphasizes reliability and safety, but not security

Common Criteria Evaluation Assurance Levels	
EAL1	functionally tested
EAL2	structurally tested
EAL3	methodically tested and checked
EAL4	methodically designed, tested, and reviewed
EAL5	semi-formally designed and tested
EAL6	semi-formally verified design and tested
EAL7	formally verified design and tested

Some things required of high assurance systems

- Application of best available software engineering techniques
  - Third-party evaluation and certification of design, implementation, tests, documentation
  - Extensive documentation
  - Exhaustive testing
  - Formal models and proofs of correspondence
  - Systematic search for covert channels and vulnerabilities
- Note - Common Criteria does not mandate specific software engineering tools, such as static analysis

A little known fact

High-assurance systems are extremely reliable.

Talk Outline

- Who are we?
- Our requirements and goals
- What did we do?
- What did we learn?

Caernarvon Castle – North Wales

Caernarvon Smart Card Operating System

- targeted at Common Criteria EAL6 or EAL7
- functions correctly despite hardware and software attacks
- supports multiple field-downloadable applications from mutually distrusting and potentially hostile sources
- uses hardware protection to separate the OS from the apps and the apps from each other (including “native” and interpreted apps)
- establishes one end of strong two-party authentication
- mandatory access controls enforce controlled sharing of on-card data and applications
- supports post-issuance ad-hoc coalitions across organizations

Layered Assurance or High Assurance?

Hardware and the OS are one tightly integrated layer, protecting

- applications from each other
- the OS from the applications

Our goal was to layer ANY arbitrary combination of applications on one high assurance OS and hardware platform. Without a high assurance OS, specific combinations of applications are usually evaluated together.

Caernarvon does the heavy lifting

It relieves applications and virtual machine of the hard and expensive work.

- crypto
- access control
- communications
- authentication
- vulnerability analysis
- certification

Clip art used with permission from Microsoft

Seven enterprises in five countries worked on Caernarvon

IBM

Philips Semiconductors (now NXP)

Atsec Information Security

German Federal Office for Information Security (BSI)

German Research Center for Artificial Intelligence (DFKI)

University of Augsberg

A Common Criteria evaluation lab

It was a multi-disciplinary development endeavor

- Hardware design and implementation
- Software design and implementation
- Test framework and testing
- Common Criteria documentation
- Formal modeling
- Vulnerability and covert channel analysis

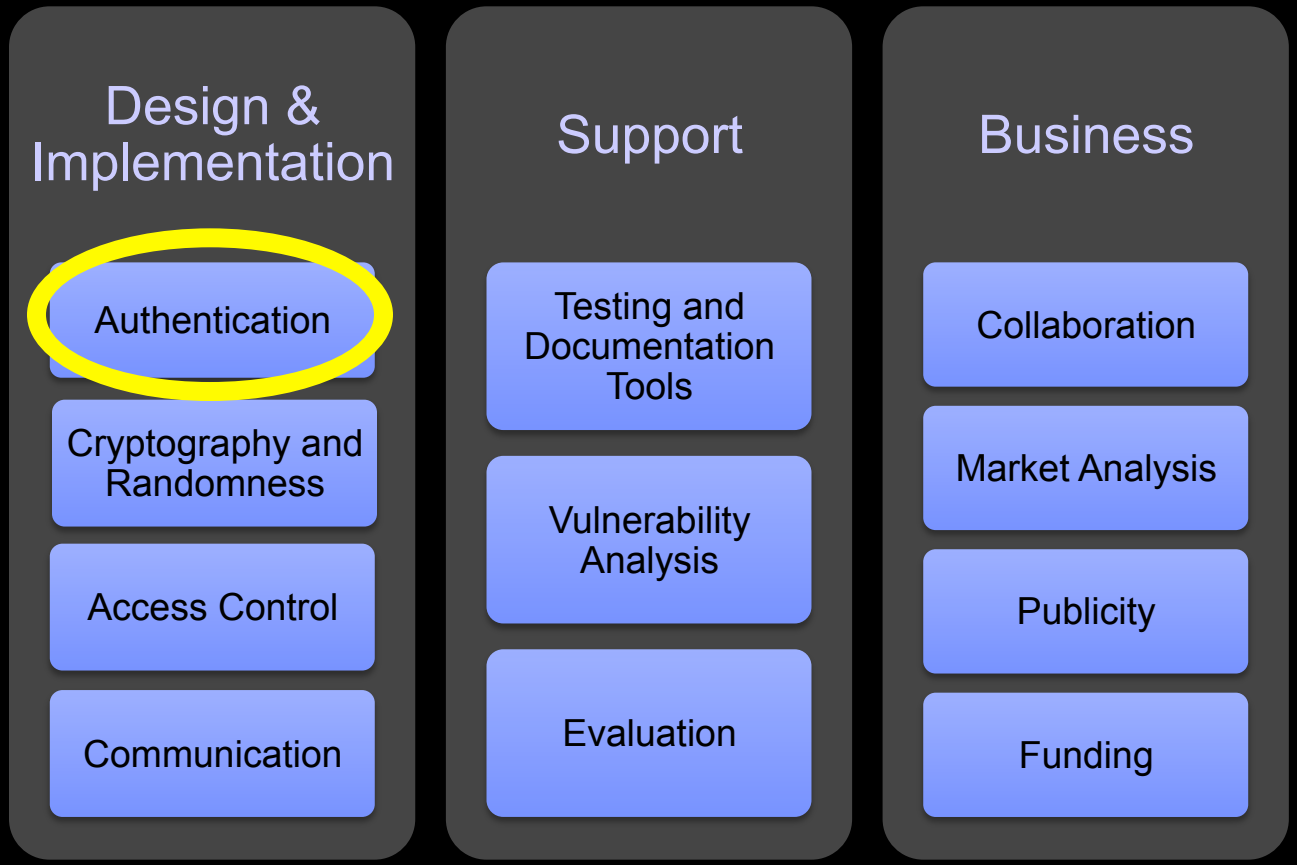
Status and Successes

- Alpha system implementation on hardware emulator complete
- Semi-formal specification complete (12 volumes, 2000 pages)
- Cryptographic Library certified at EAL5+ in Germany, and incorporated in several products
- Authentication protocol is part of European standard for Electronic Signature Creation Devices
- Mandatory Access Control Policy incorporated into Trusted Linux Client and IBM Infosphere Streams
- Security policy model is part of computer security courses worldwide
- Formal model of early system in public domain

Talk Outline

- Who are we?
- What is high assurance?
- What did we do?
- What did we learn?

Important components of high assurance OS development



Q. Which party should disclose its identity first, card or reader?

Clip art used with permission from Microsoft

A. the reader. The card has personal information on it.

Clip art used with permission from Microsoft

Q. In a multi-application card. . .
what software should authenticate the reader?

A) the operating system
B) the application that is running
C) the strongest application

Hint: Would you trust your rapid transit app to authenticate you to your bank?

A) the operating system
B) the application
C) none of the above

It does it for all of the others.

Caernarvon's privacy-preserving authentication protocol

- Standardized (in European standard for digital signature creation devices)
- Uses a Diffie-Hellman key agreement scheme
  - Based on the SIGMA protocols, part of the Internet Key Exchange (IKE) Protocol
  - Rigorously analyzed, proven correct
- Performed by the OS to guarantee that authentication was completed
- Protects the cardholder's identity by requiring the card reader to authenticate first
- Protects from eavesdroppers by exchanging session keys early in the protocol

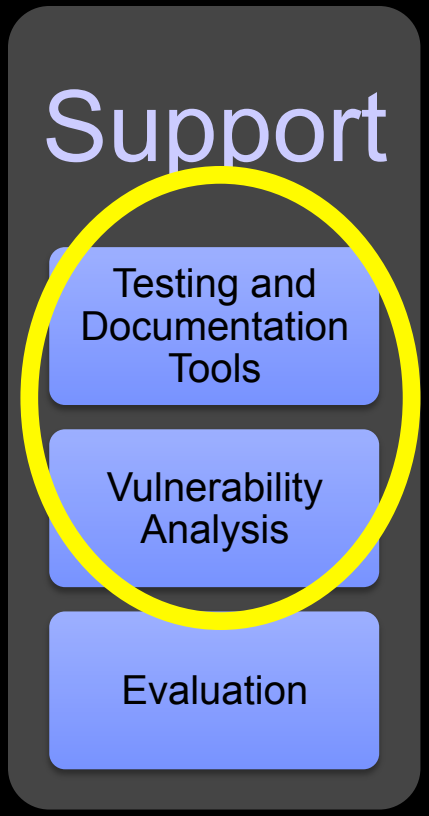
Technical Challenges

Common Criteria is not explicit in requiring modern knowledge / tools such as fuzz testing, static analysis, formal analysis, threat modeling

Modern technology poses additional challenges

- Compilers
  - optimizations break source / target correspondence
  - infamous for optimizing away security checks
  - Optimizations change code structure, introduce “dead” code
- Side-channel attacks now sufficiently high-bandwidth to require attention

Tooling Support in Non-Standard Development Environments



Smart cards have a sparse set of development and test tools from a handful of vendors.

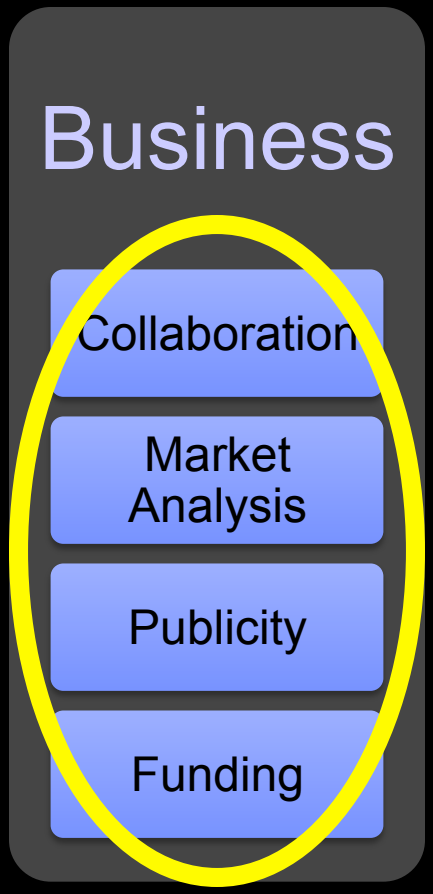
Tools available decades ago for other platforms are unavailable.

We often had to roll our own

Custom tools helped us meet Common Criteria requirements

- Documentation generator for thousands of pages of documentation embedded in the source code, including cross volume, cross references
- Testing framework for
  - internal and external functions
  - automated verification of results (success, specific errors, failures)
- Code coverage trackers, with object-to-source matching (beware of optimizers!)
- Source code static analysis tools to search for bugs (beware of non-standard C)
- Automated test generators

It's not just the technology that's hard



market analysis

market development

requirements gathering

relationship management

project management

budgeting

funding procurement

public and media relations

. . .

Collaboration is essential on a High Assurance OS project

- deep skills from multiple disciplines
- skills seldom found in one organization
- interdependency of hardware and software
- full disclosure of proprietary documents

Clip art used with permission from Microsoft

Customers request the moon. . . (we thought >=EAL6)

"secure and reliable"

"strongly resistant to terrorist exploitation"

"strongly resistant to identity fraud"

. . . but settled for the earth

“EAL4+”

Interim (Lower Assurance) Deliverables

Yes, they slow you down, but. . .

they provide funding and milestones for ongoing development.

Long term commitment is a challenge

- The time to complete commercial-off-the-shelf high assurance systems exceeds the typical funding horizon of organizations.
- Changes in one organization's goals or priorities can imperil the entire project

Our Advice for Those Seeking High Assurance

- Make buddies in the business school and government security organizations
- Find a tester who loves to automate everything
- Create interim deliverables, even if they slow you down
- Document, document, document - in the code, in the tests
- The hardware and software are inseparable.
  Build a team that supports this notion.
- Stay passionate about what you do.
  It will get you through the tough times.

"There are no shortcuts to evolution."

-Louis D. Brandeis,
former American Supreme Court Justice

There are no shortcuts to evaluation either.

Summary

- High assurance is essential, achievable, and demonstrated
- For more details, see

  Karger, McIntosh, Palmer, Toll, and Weber, “Lessons Learned Building the Caernarvon High-Assurance Operating System”, *IEEE Security & Privacy*, Jan/Feb 2011.

This talk is dedicated to the memory of our friend and colleague, Dr. Paul A. Karger.