Trustworthy Hardware

Pankaj Rohatgi

IBM T. J. Watson Research Center

CHES 2007

Acknowledgements

- **IBM 4758 team in IBM Research/Development**
- **IBM Research**
  - Caernarvon smart-card OS team
  - Zurich Research Lab: JavaCard team.
  - Secure Blue team
  - Secure Hypervisor (sHype) and Virtual TPM (vTPM) team
  - System S team
  - Colleagues: Dakshi Agrawal, Stefan Berger, Suresh Chari, Leendert van Doorn, Eric Hall, Charanjit Jutla, Paul Karger, Elaine Palmer, Reiner Sailer, Helmut Scherzer, Sean Smith,, Ronald Perez, J. R. Rao, Steve Weingart.
- **IBM's Internet Security Systems Division**
- **Prof Berk Sunar and his students: Deniz and Selcuk**
- **CHES community.**

Information Security: Once upon a time ...

FOR MOST SYSTEMS

- Few sophisticated users and developers
- Few applications, interfaces, data flows.
- Few sophisticated malevolent attackers
  - Script kiddies, publicity seekers
- Long application development and testing cycles
- Firewalls, anti-virus, secure communications considered good enough !

FEW SPECIALIZED SYSTEMS REQURING VERY HIGH SOFTWARE/HW SECURITY

• Secure OS/DBMS (Defense), Secure embedded devices with cryptographic capability, and tamper-resistance, but limited functionality (Defense, Banking, PayTV)

• Adversarial models, requirements, scenarios and techniques developed in these contexts considered esoteric and non-mainstream.

Information (In)Security – now

Information Security: Enterprise Perspective

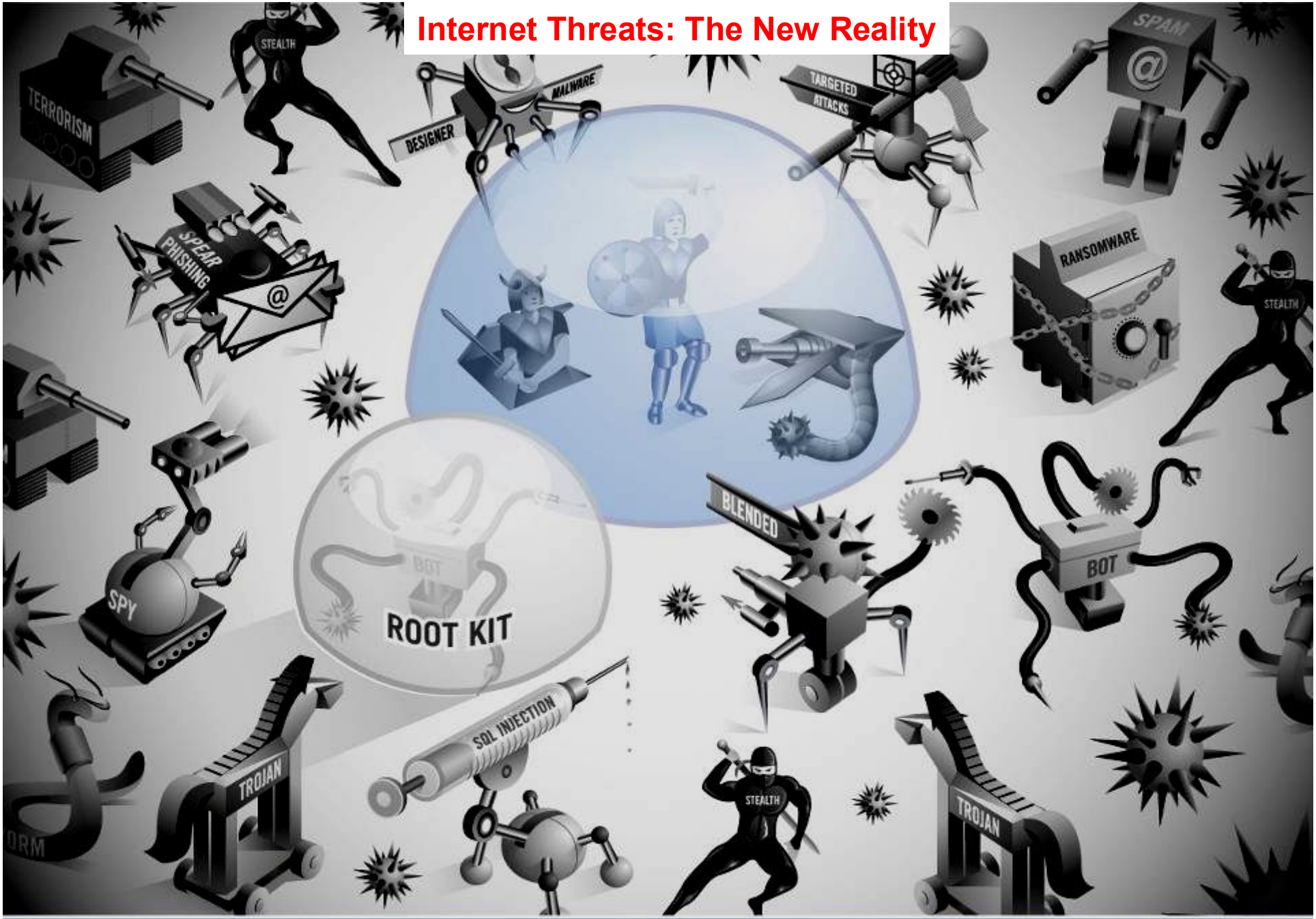
Many forms of valuable data
- Structured
- Unstructured
- Images
- Video, Voice

Accessible from/stored in multiple ways and on multiple devices…
- Cell phones
- Laptops
- PDAs,iPods
- Briefcases

All of which are subject to increasingly sophisticated internal and external attacks or accidental leakage !

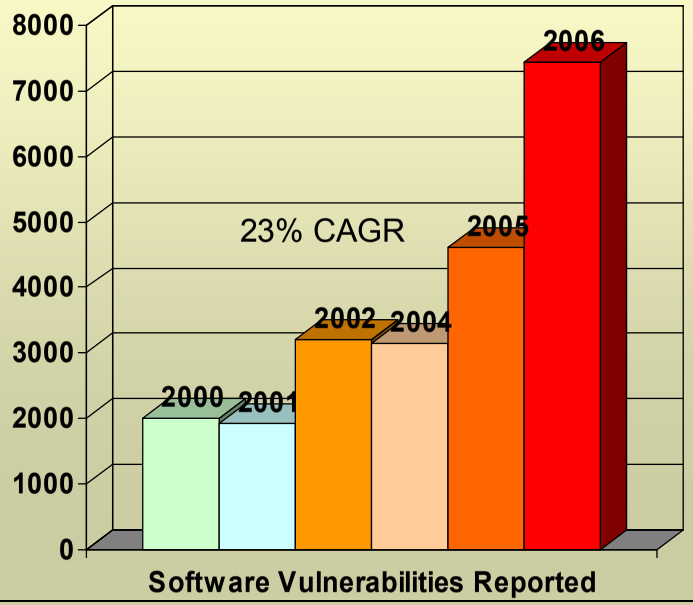
Adapted from: IBM Internet Security Systems

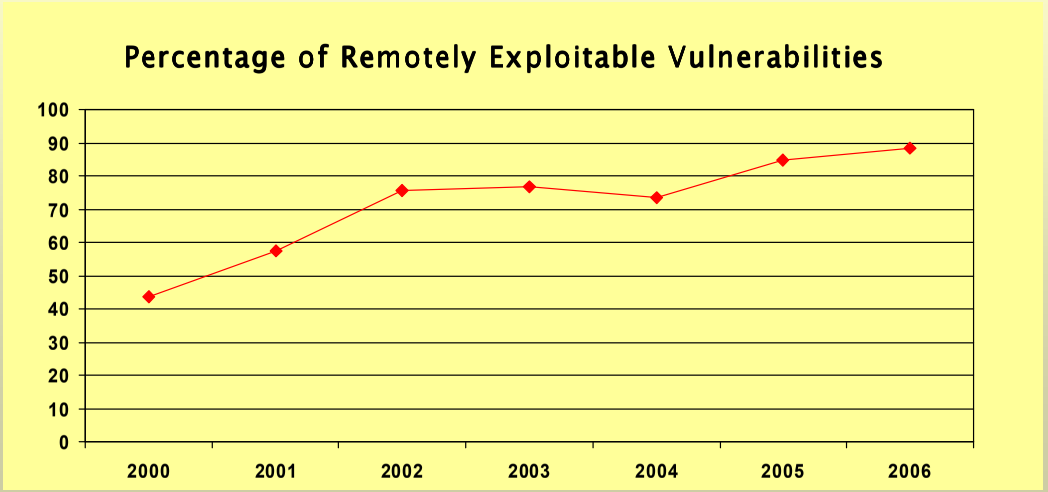
Internet Threats: The New Reality

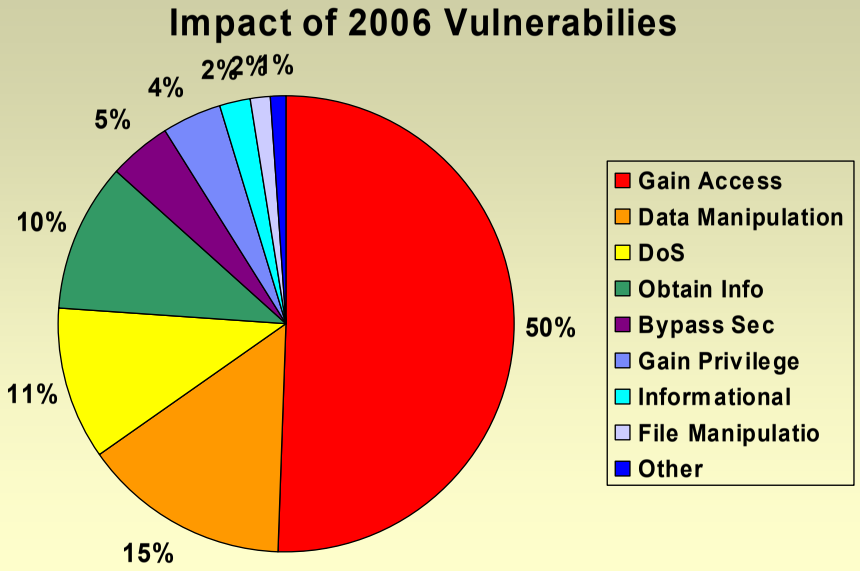


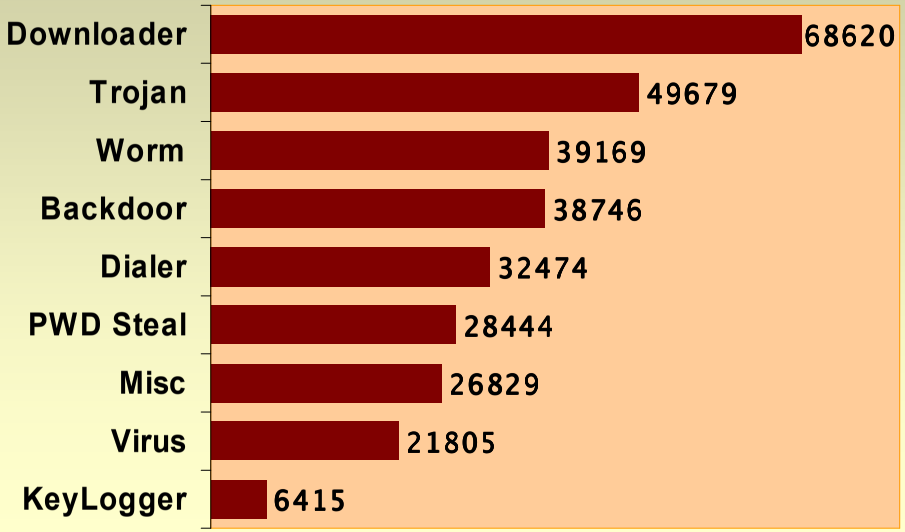
Source: IBM Internet Security Systems: http://www.iss.net/evolvingthreat/

Software (In)security Trends





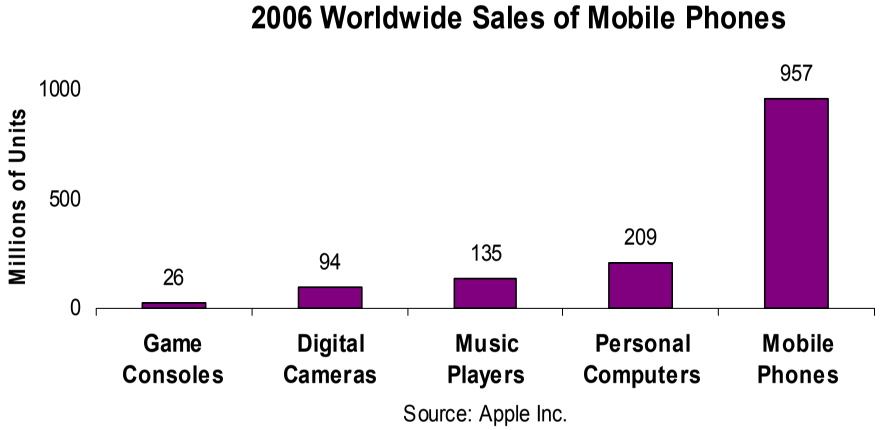




Systems, applications and attack scenarios once considered “niche” and “esoteric” are now mainstream !

- Security lessons from those systems now applicable

Attackers quickly follow the value – Mobile Platforms Case Study



"By 2009, 70% of knowledge work will occur in locations where workers depend on wireless and remote-access infrastructure outside the enterprise's direct control." *Gartner*

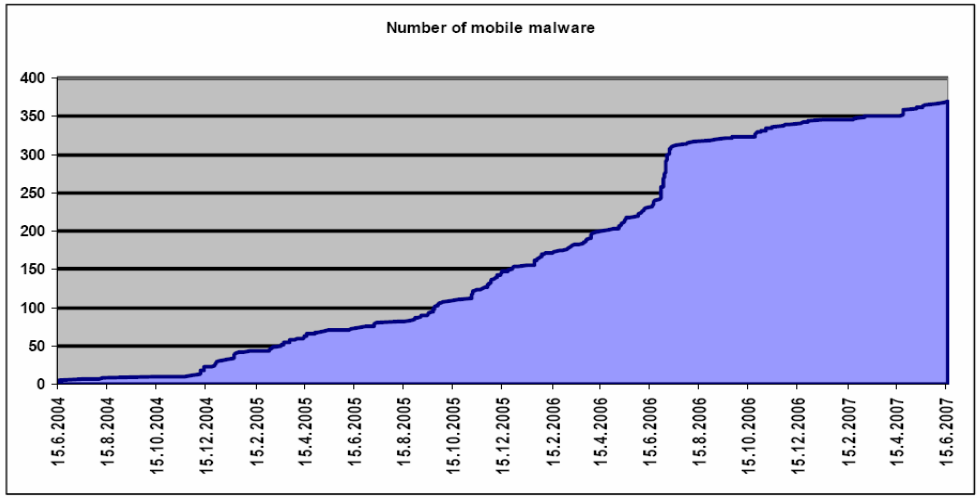
"Mobility has become a critical component of the IT and communications fabric for businesses of all sizes." *Forrester*

"Mobility remains a high-priority CIO issue that will drive steady growth in demand for mobile products and services for several years." *Gartner*

"Organizations that adopt mobile solutions… will reap significant economic benefit." *BCG*

- Mobile technology marketplace is large and growing fast
- Mobile phones poised to become the dominant personal computing platform (e.g., billions of phones already deployed)
- Mobile phones rapidly gaining capabilities and increasingly exhibit the security vulnerabilities of full-sized computers.
- Cross-platform runtime environments are facilitating the development of rich Web apps on mobile phones

Threats to Mobile platforms



F-SECURE

Data source: F-Secure

Primary threat: Loss of devices (PCs + Mobile) with sensitive enterprise data:

•# of laptops stolen in US in 2005: 750,000 – 97% never recovered – (est., Absolute Software).

• Over 6 months in London riders left behind 4,973 laptops; 5,939 Pocket PCs; and 63,135 mobile phones (source: Pointsec + Taxi Drivers Assoc.)

Malware: Whats been seen so far

• Viruses and Worms: 58; Trojans: 297; Spy tools: 9;

Whats not been seen yet

• Rootkits

• Worms that do not need user interaction for spreading

• Mobile botnets

• Large-scale profit-oriented malware (professionals)

- More than 370 mobile phone viruses so far
- Tens of thousands of infections worldwide
- Cabir and Commwarrior reports from > 30 countries
- Operator with 9 million customers: almost 5% of MMS traffic infected
- Operator with 14 million customers: Over 8000 infected devices have sent over 450000 MMS messages. Largest number of messages sent by one phone: 3500.
- Operators have given money back to customers who had Commwarrior

Source: F-Secure

Platform	2004	2005	2006	2007
Palm	3	3	3	3
PocketPC	2	2	3	4
Symbian	22	141	337	364
J2ME	0	0	2	2
All	**27**	**146**	**345**	**373**

F-SECURE

Data source: F-Secure

So what's being done now about Information Insecurity ?

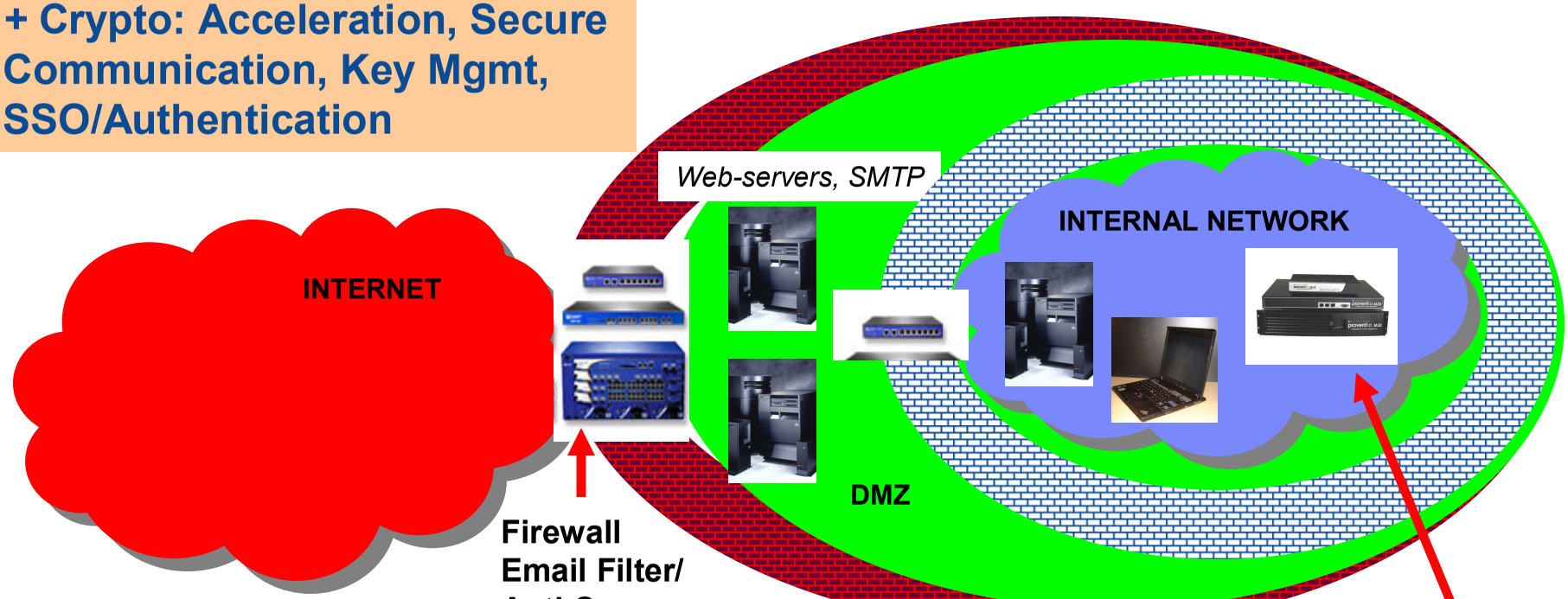
Better Security using Hardware !

This is a great time to be part of the CHES community !

Better Security Using Hardware: What's mainstream

External Hardware-Assisted Appliances "chaperone" insecure software/systems

+ Crypto: Acceleration, Secure Communication, Key Mgmt, SSO/Authentication



Firewall
Email Filter/
Anti-Spam
Anti-virus.
Virus Protection System.
Web Application Firewall
Information Leakage Protection.
Intrusion Detection System (IDS)
Intrusion Prevention System (IPS)
NAC
.....

IDS/IPS.
System Mgmt
& compliance.

+ Disk/TAPE Encryption

+ SSL/IPSEC VPN
SOA/XML/WS-Security
SSO

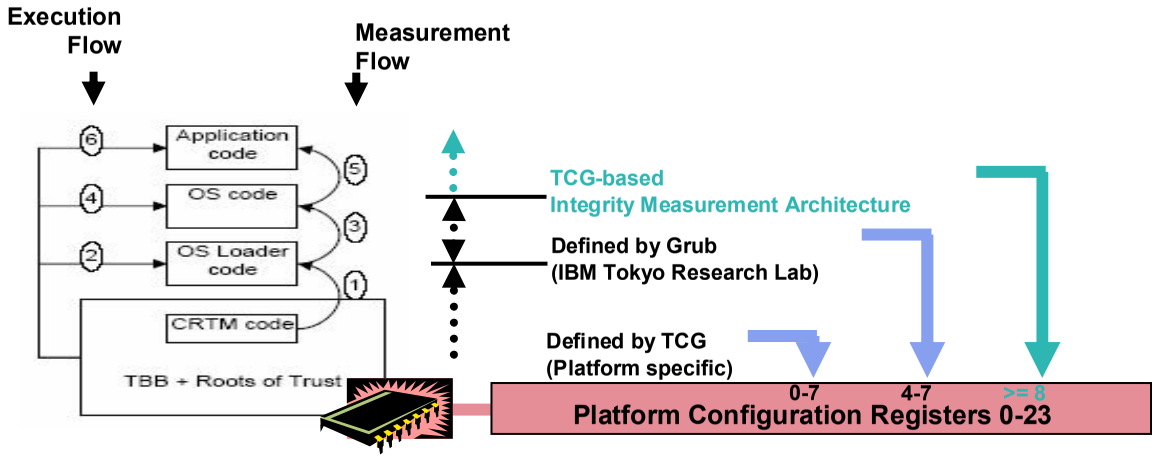
Mainstream: General purpose CPUs with Crypto Acceleration

- **Sun Niagara 2: 8 cores**
  - 2 pipelines, 4 threads, 1 FP and 1 crypto unit per core
  - Support for DES/3DES, AES. RC4, SHA1, SHA256, MD5, RSA 2048, ECC.

- **IBM z-series mainframe**
  - Crypto Assist (CP Assist) per central processor CP
    - DES/3DES (encryption/mac), SHA1
  - 2 CMOS secure crypto co-processors/machine
    - FIPS 140-2 Level 4

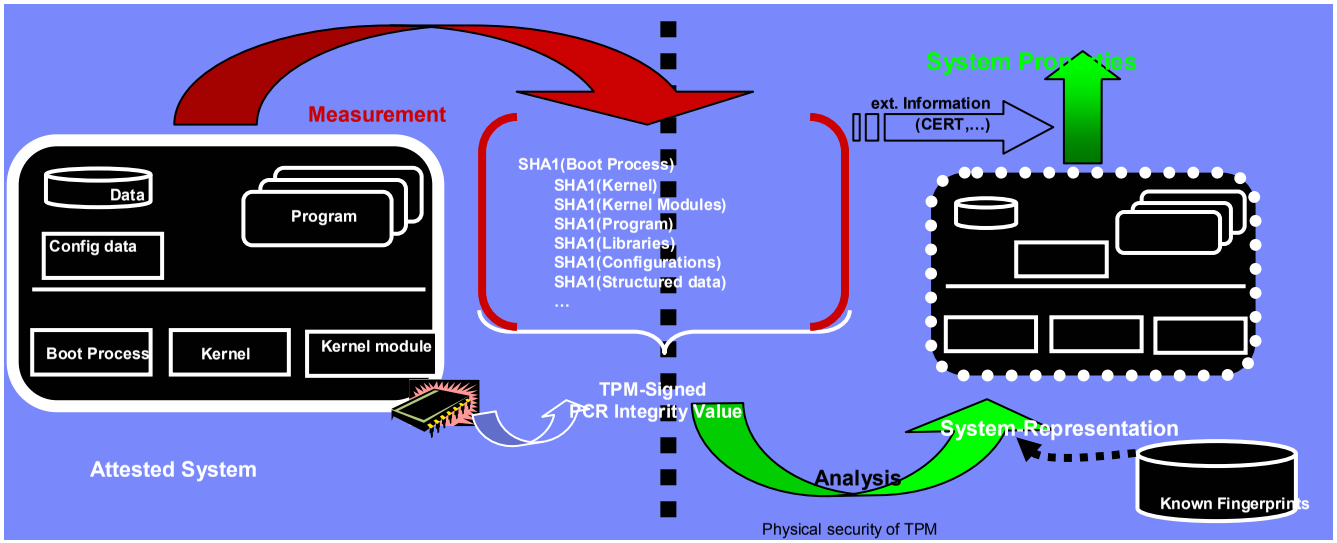
What's is becoming mainstream: Trusted computing

- **TCG/TPM**
  - Core Root of Trust
  - Trusted Boot
  - Integrity Measurement Architecture
  - Remote Attestation
  - Secure Key Storage

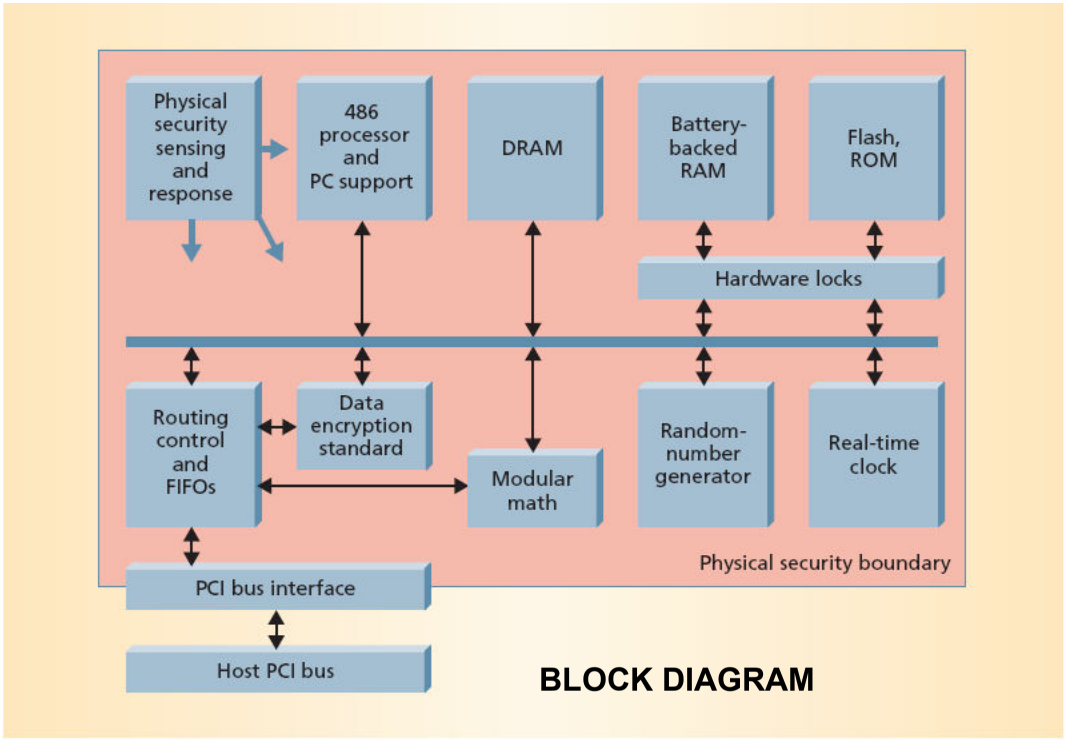


Integrity Measurement Architecture extended to OS/Applications

IBM ESS

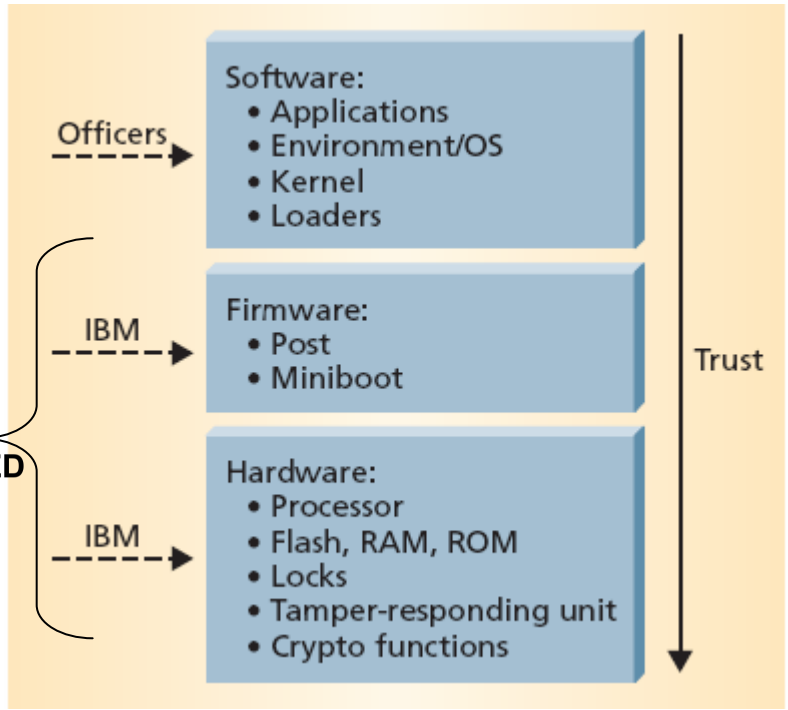


IBM 4758

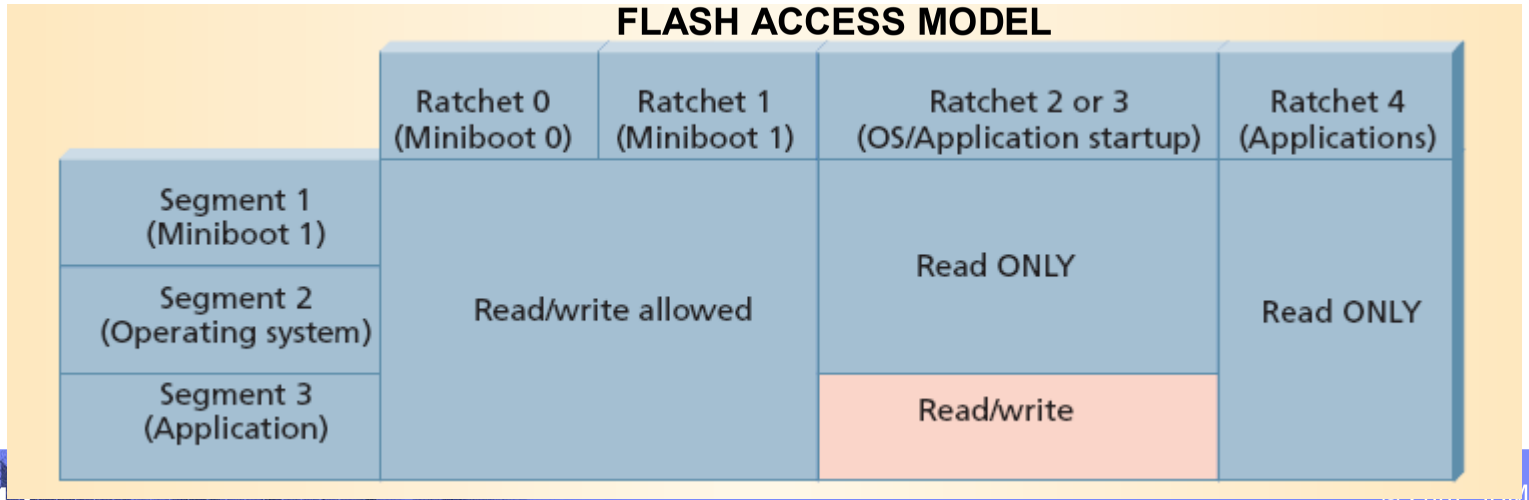


- **Programmable crypto (DES/3DES/SHA1/DSA/RSA/RNG) co-processor**
- **World's first programmable crypto co-coprocessor to obtain FIPS 140-1 Level 4 certification for Physical and Logical security of Hardware and Firmware.**
- **SEVERAL INNOVATIONS**
  - Secure Key Storage protected by a tamper detection and response mechanism certified to FIPS 140-1/2 Level 4.
  - Secure Boot. Field upgradable (including part of firmware)
  - Remote Attestation (outbound authentication)
  - Programming Model based on a Layered Trust Model

4758 Layered Trust Model for Code

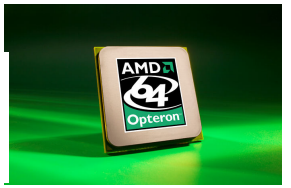


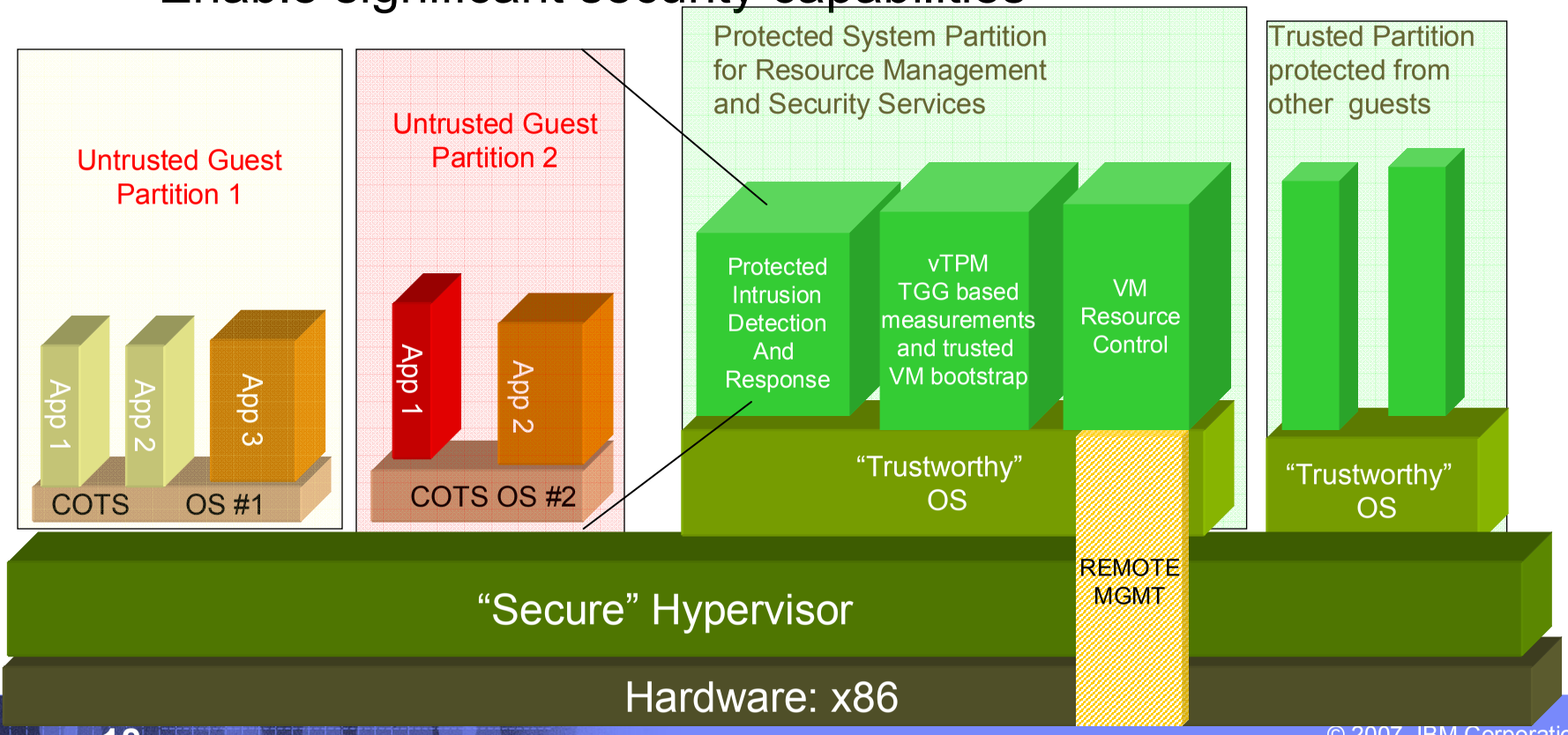
- **Initial OS/App**
  - CPQ++/CCA
- **IBM to certify key of Segment 2 owner**
  - Segment 2 owner manages/loads Segment 3 Apps
  - Could also relinquish ownership (and lose segment 2 keys)



What's becoming mainstream: Support for "Secure" Virtualization

- **Secure Virtualization, Secure Execution Environment (Dynamic Root of Trust), Remote Mgmt**
  - Supported by Intel VPRO& AMD Pacifica
  - Enable significant security capabilities





What's considered "niche" now

- **Several application domains require usable and affordable hardware/software devices with high physical and logical security and have unique threat models**
  - Sensors and Actuators
    - Critical infrastructure control (SCADA).
    - Battlefield sensors, smart-dust.
  - Trustworthy Defense Infrastructure
    - Securing systems
    - Systems that are free from Trojans (huge emerging problem).
  - Gaming Consoles
    - E.g., Xbox 360, PS3, Wii

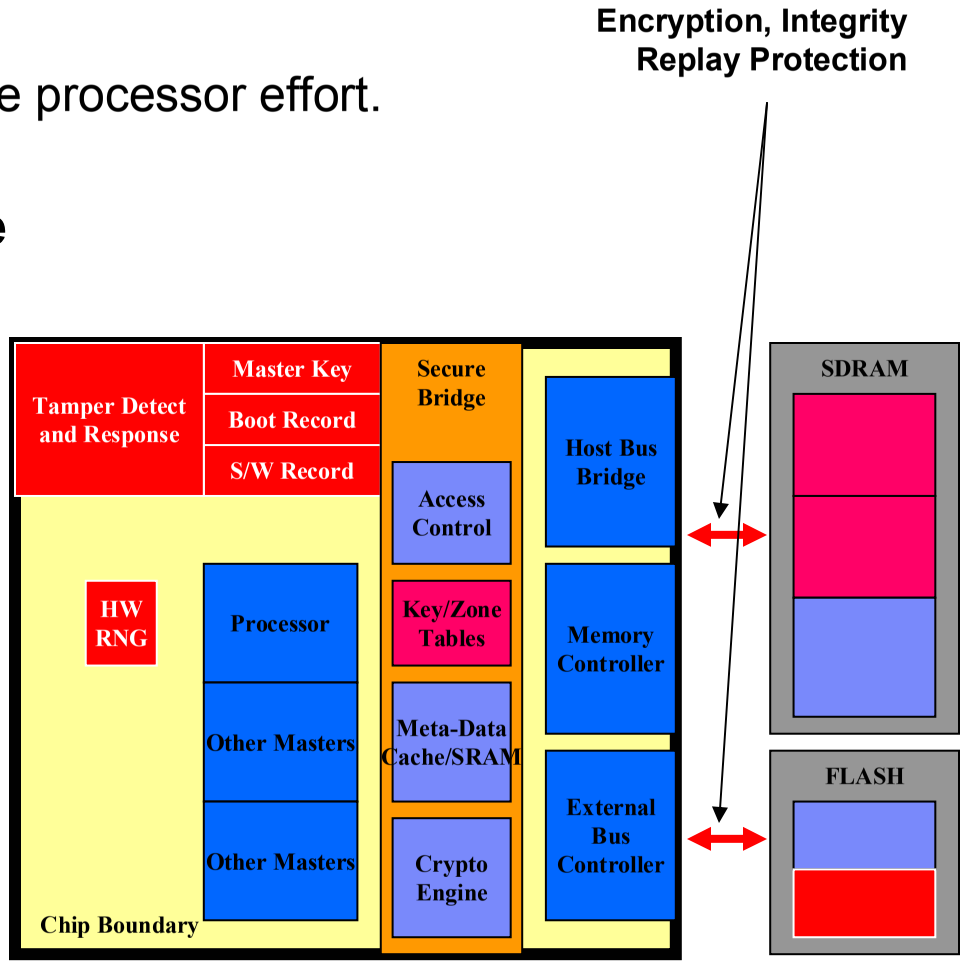
  - Content Protection/DRM
    - Video/Audio: e.g., HD-DVD, Blue-ray...

    - PayTV
  - Handheld/Mobile devices
    - Protection from malware and device loss

  - RFID Tags
  - IDs and ePassports .....

Secure Processors

- **Examples**
  - Secure Blue (IBM)
  - XOM (Toronto)
  - Princeton (PALMS) secure processor effort.
  - AEGIS (MIT)
- **Secure Blue Architecture**
  - Tamper-Resistant IC
  - Encrypted external memory
  - Affordable
  - Operates at full processor speed
  - Secure Key Mgmt for confidentiality
  - Secure Boot,
  - Secure firmware updates



Predictions

- **Applications and threats considered to be "niche" and "esoteric" today will become commonplace in the years to come.**
- **Huge challenges will need to be overcome**
- **Success will depend on new technology that aligns with business imperatives, trends and constraints**
  - Truly innovative solutions will convert what appear to be inherent disadvantages and problems into an advantage.
- **Techniques and lessons learnt in tackling these challenges will have lasting value.**
- **Ideal topics for those starting their research career in this area !**

Building a usable and affordable hardware/software device with high level of physical and logical security

- **Several challenges, many unsolved problems, many exciting opportunities**
- **Need to consider the complete lifecycle of the device**
  - Design
  - Build/Test
  - Deploy
  - Maintain
  - Decommissioning
- **Several new challenges, requirements and attack scenarios at each stage of the lifecycle.**

Decommissioning

- **Must design in the ability to efficiently migrate to a new device.**
  - Applications can live much longer than the device.
  - Virtual machines/environments require rapid decommissioning of virtualized devices (vTPM).
- **Dealing with Data Reminiscence**
  - Very hard to delete all data through software.
    - Some data never gets deleted (e.g., bad sector)
    - Information physically available after “overwrite” [Gutman]
  - Encrypt Data and Delete (Purge) Keys to erase
    - Purging Keys: Reminiscence and Imprinting problem
    - In the 4758
      - Radiation sensors
      - Temperature sensors
      - Effects of time.
        - » “screensaver” for keys, periodic scrambling of contents
  - High security systems: Many places for malicious software to hide information: e.g., device flash/e2prom
    - “Virtual” access to untrusted software may be the only solution.

Maintenance

- **Upgrades: planned and unplanned**
  - Each new application of system potentially requires an upgrade.
  - Features, Fixes, Bit-rot.
  - Secure update mechanism
  - Must know/track what upgrades have been made
  - Prevent back-leveling.
- **Handling changing users, roles, lost passwords, lost keys, etc.**
- **All these operational issues must be designed in from the beginning.**

Planning for Unexpected Upgrades

Unplanned software upgrades

- **Hardware/Firmware can offer best security but applications/system-code bugs can effectively neutralize security features.**
  - XBOX: hardware easily hacked, mod-chips possible [CHES 2002 paper]
  - Xbox 360: all external memory seems to be encrypted/mac'ed, signed system code and games.
    - Core hardware still not hacked (no mod-chips) for > 1 year
      - BUT: Xbox 360 hypervisor bug (builds 4532-4548 around Oct 2006) allowed unsigned code to execute with privilege.
  - IBM 4758
    - Core hardware and secure bootstrap still not broken
    - BUT: CCA (Application) API bug can reveal CCA keys [Bond, Anderson, 2001]
- **Secure recovery should be built in and back-leveling prevented**
  - Xbox 360: kernel patched Feb 2007.
  - IBM 4758: patched with new version of CCA.

Need for upgrading Hardware !

- **Most modern CPU's have significant "errata" that needs microcode patches.**
  - Patch is volatile and needs to be re-loaded at every boot.
  - Done by BIOS or OS.
  - Trend is worsening !
- **Hardware bugs can create vulnerabilities**
  - Multics on several Honeywell machines was subverted by a hardware bug [early '70s]
- **Secure Hardware patching must be supported and be designed for**
  - Certain versions of x86 processors accept unauthenticated patches !

Deploy

- **Identification: Is it a device and the right device ?**

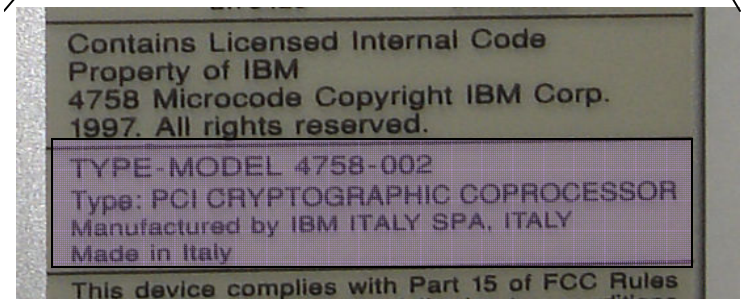
- **Personalization and Privacy**

The device identification problem (build, deploy, maintain, decommission)

- **How do you know (either locally or remotely) that you are interacting with a genuine real device and not a fake or a software emulation ?**
  - The IBM 4758 solution: Certified root secret key in hardware protected by tamper-detection and response.
    - This is too costly !
    - Tamper responsive packaging creates many problems !
  - The TCG/TPM solution: Certified EK root key in hardware, protected against software attacks but not necessarily against hardware attacks.
    - Cheaper, but software-attack only protection too weak for some applications.

Device Identification and the 4758

HOW THE IDENTIFICATION PROBLEM BECAME THE SHIPPING PROBLEM



Environmental Requirements

From the time of manufacture, the IBM 4758 PCI Cryptographic Coprocessor card must be shipped, stored, and used within the following environmental specifications. Outside of these specifications, the IBM 4758 tamper sensors will be activated and render the IBM 4758 permanently inoperable.

	IBM 4758-002	IBM 4758-023
Temp shipping	-15°C to 60°C	-15°C to 60°C
Temp storage	1°C to 60°C	1°C to 60°C
Temp operating	10°C to 40°C	10°C to 40°C
Humidity shipping	5% to 100% RH with original IBM package	5% to 100% RH with original IBM package
Humidity storage	5% to 80% RH	5% to 80% RH
Humidity operating	8% to 80% RH	8% to 80% RH
Pressure operating	min 768 mbar max 1039 mbar	min 768 mbar max 1039 mbar
Pressure shipping	min 550 mbar max 1039 mbar	not specified
Pressure storage	min 700 mbar max 1039 mbar	not specified

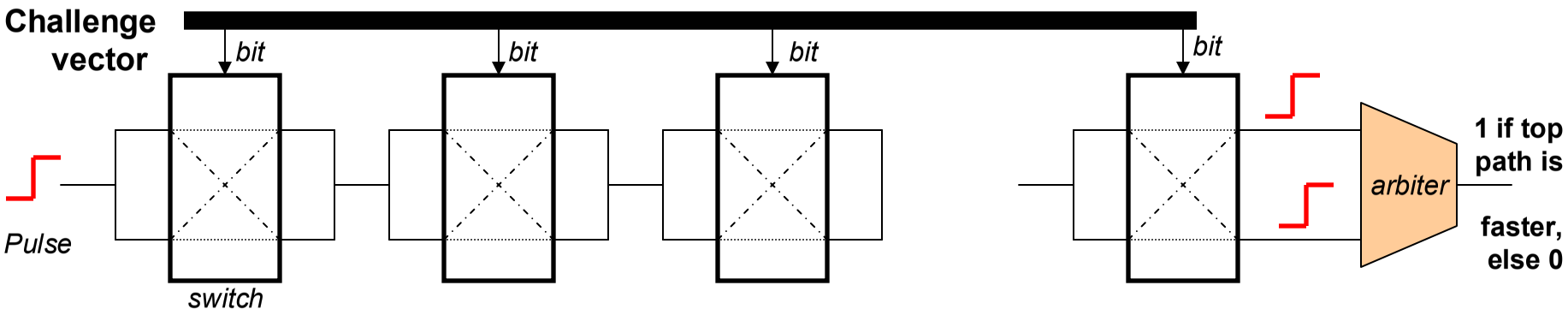
Desperate Engineers, Desperate Solutions

Device identification problem (continued)

- **Silicon Physically Uncloneable Functions (PUFs) [GCDD 02], [LLGSDD 02] have the potential to offer a good compromise for Device Identification**
  - Very clever and evolving idea with potentially multiple applications --- watch this space.
    - Example of converting a “nuisance” into an asset.

Physically Uncloneable Functions (PUFs)

BASIC IDEA



- **Responses to different challenges, a random function of manufacturing process variation, is unique to device and not duplicatable.**
  - Propagation has non-linear effects, feedback can make modeling attacks harder.
  - Circuit could be embedded such that active probing alters function
- **Over time, the basic technique has been improved and evolved to create unique cryptographic keys for each device [G. Edward Suh, Srini Devadas, DAC 2007].**
- **Further reading: Using the PUF concept for Intellectual property protection**
  - Prevent manufacturer from making extra hardware for black market [Alkabani, Kousahanfar, USENIX Security 2007]
  - Paper on PUFs and IP protection from Philips in CHES this year.
- **A concern: impact of side-channels on PUFs not studied**

Design/Build/Test

- **Large number of unsolved problems in this space**
- **Some long-standing problems**
  - Formal verification of designs, as designs get larger.
  - Secure composition of H/W and software components.
  - Automated testing for security and high assurance.
  - Tamper Resistance, side-channel resistance.
  - Intellectual property protection, Anti-piracy
- **Relatively new problem**
  - Loss of trustworthiness of the entire design/build/test process for Hardware and Software
    - Significant immediate impact for defense applications.

Composition Problem (Design/Test)

- **Rare to have software/hardware designed simultaneously by same team.**
  - Software and Hardware usually built using pre-existing components from different teams and/or vendors!
- **Lack of complete information about components makes it impossible to make a security assertion about composed system**
  - Infeasible to re-evaluate every component for each new application.
- **Problem has a very long history**
  - Purple book (Trusted Database Interpretation for TCSEC) tried to formalize this problem when arguing when a high assurance OS and a high assurance DBMS that runs on a high assurance OS could be evaluated separately.
    - Defined the notion of TCB subsets which rarely occurs in practice
      - 4758 hardware ratchet mechanism is one example !
  - Chicken and egg problem: Don't know what information about each component will be needed until the vulnerabilities of the entire system are being analyzed.
    - The fact that individual components are "secure" in some context is meaningless, Security does not compose !

Composition Problem: An example from Caernarvon

- **Many hardware RNGs have been evaluated and certified with the following guidance about use**
  - The RNG should be tested before use for sensitive operations.
- **Wanted to use a certified hardware RNG from a partner for seeding a pseudorandom number.**
  - BUT, extensive randomness tests in software on a smart-card could reveal the random numbers.
    - Template Attacks !
  - So, tested random numbers must be different from the random numbers used for seed.
    - e.g., Test a first set of random numbers, use a second set as seed
    - OR: create three sets, test first and third and use second etc.
  - BUT that requires a very specific analysis of the RNG failure mode that was not part of the evaluation !

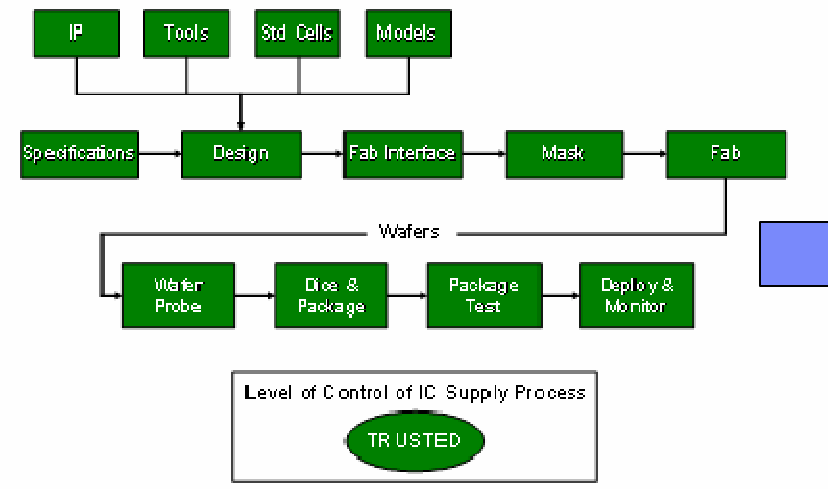
Other examples of composition problems

- **Can a particular secure operating system be implemented on a processor where certain illegal instructions result in unspecified state left in some machine registers ?**
- **Can a secure virtual monitor be built without detailed knowledge of a platform's micro-architectural attack vulnerabilities ?**
  - These could change at each stepping.
- **Can a security protocol using crypto algorithm A be implemented on a platform which has side-channel leakage, but a side-channel resistant implementation of A ?**

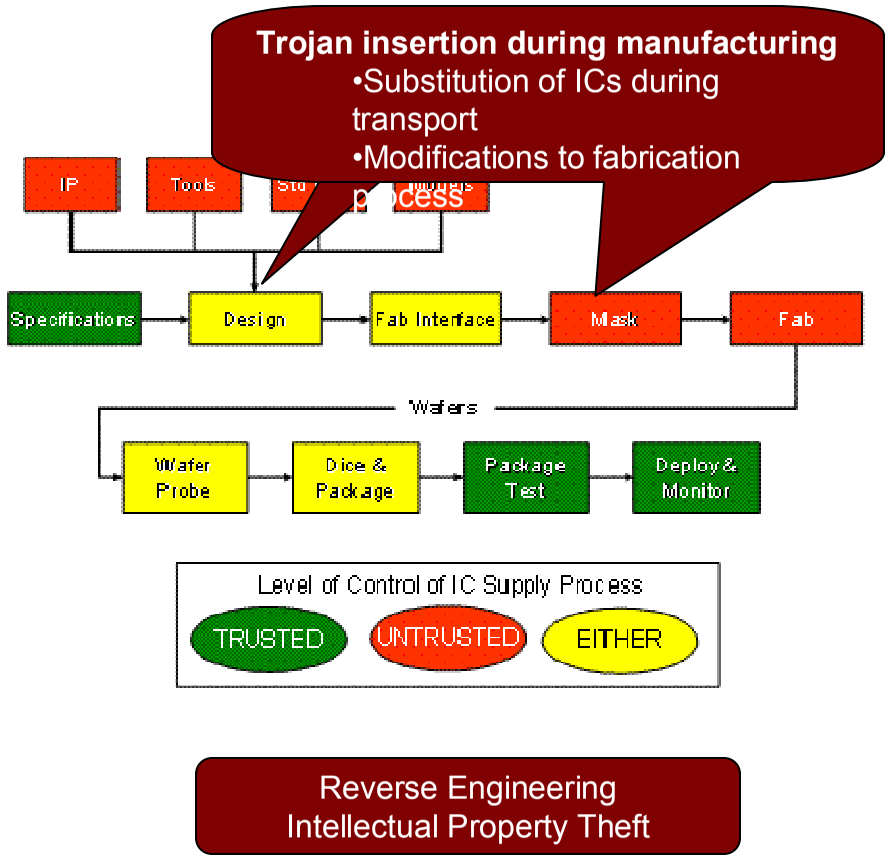
Trustworthy design and manufacturing

The Problem: Trustworthiness of current IC supply

Old IC Supply Process



Current IC Supply Process



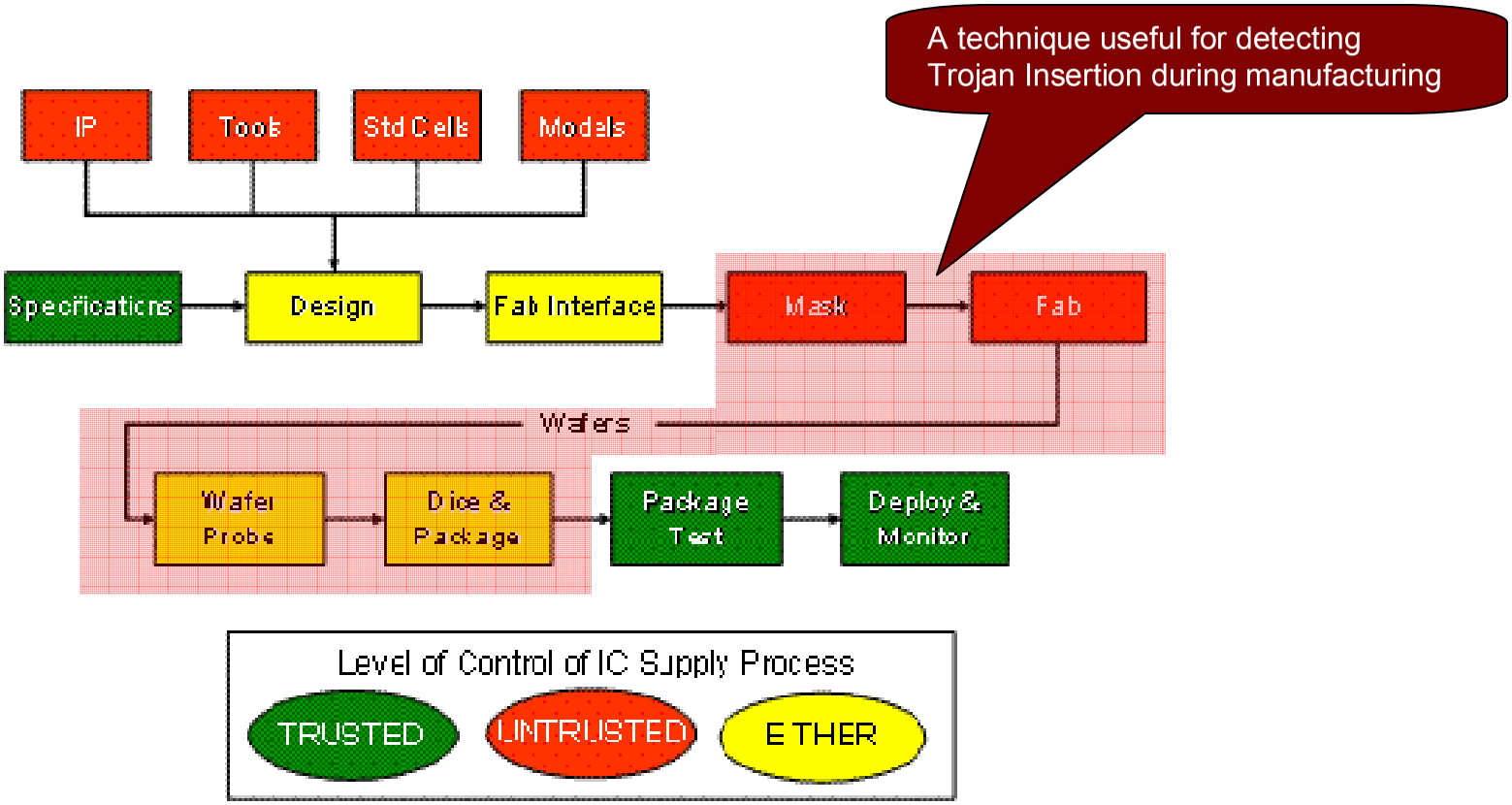
Reverse Engineering
Intellectual Property Theft

From: DARPA BAA 06-040, Trust for Integrated Circuits

IMPACT: ICs FOR SENSITIVE COMMERCIAL/DEFENCE APPLICATIONS

How to detect Trojan insertion [**ABKRS 2007**]

Trojan Detection using IC Fingerprinting

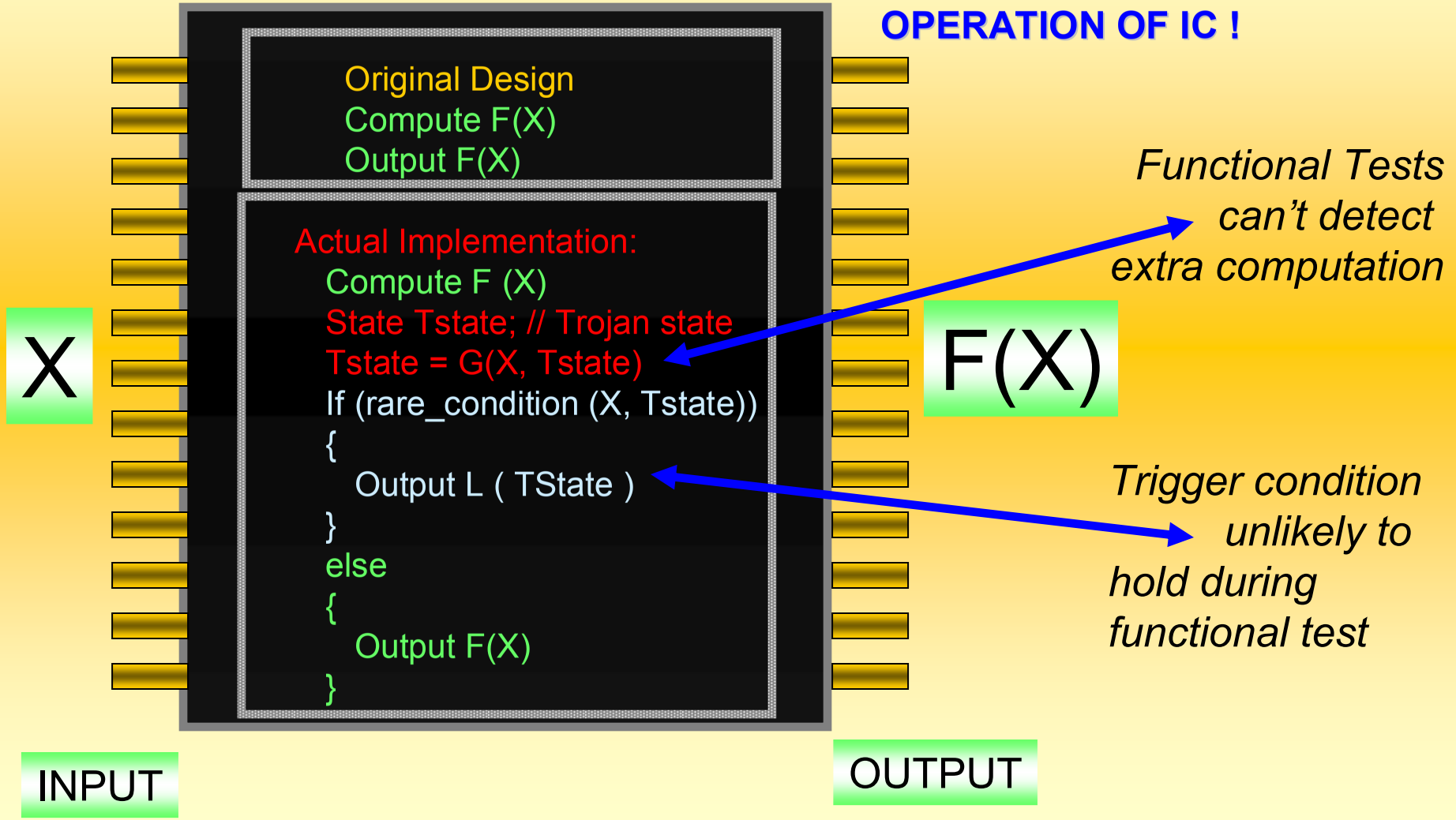


Existing Techniques usable for Trojan Detection

- **Detecting IC substitution during transport**
  - Identification problem.
    - Anti-Tamper Techniques and PUFs
- **Mitigating problem of Trojan insertion during manufacturing ?**
  - What will not to work
    - Functional testing of ICs before deployment.
    - Destructive validation of a sample of ICs

The Problem: Why functional tests don't work

NO VISIBILITY INTO INTERNAL OPERATION OF IC !



Idea: Use Side-channels

- **Side channels (power, EM) have been used to glean information about inner workings of hardware devices and extract secret keys**

- **Can side-channels be used to detect Trojan activity within an IC ?**
  - Convert a highly successful attack technique into a defensive technique ?

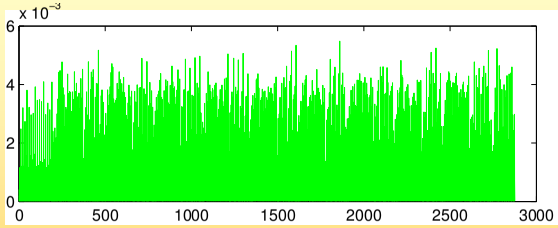
Functional Test + Side Channel Information

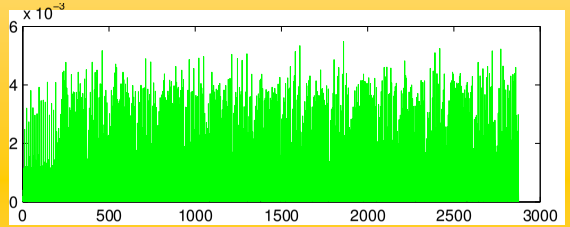
X

F(X)

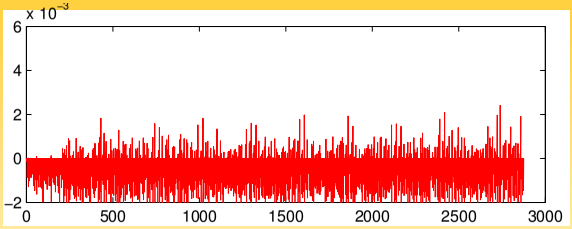
Original Design:
Compute F(X)
Output F(X)

Actual Implementation:
Compute F (X)
state Tstate; // Trojan state
Tstate = G(X, Tstate)
If (rare_condition (X, Tstate))
{
Output L (TState)
}
else
{
Output F(X)
}

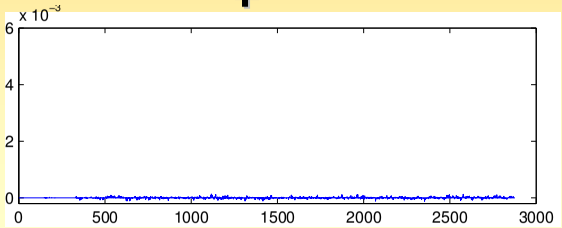




+



+



INPUT

OUTPUT

Trojan Detection using Side Channels

- **Collect side channel signals during exhaustive functional testing of genuine ICs and create a side channel "fingerprint"**
  - ICs can be later destructively validated.
    - Companies like Taeus International, Semiconductor Insights, Portelligent do such IC "teardowns".

- **Large Trojan activity can be easily detected.**
  - Large signal differences.

Challenge: Small, minimally active Trojans

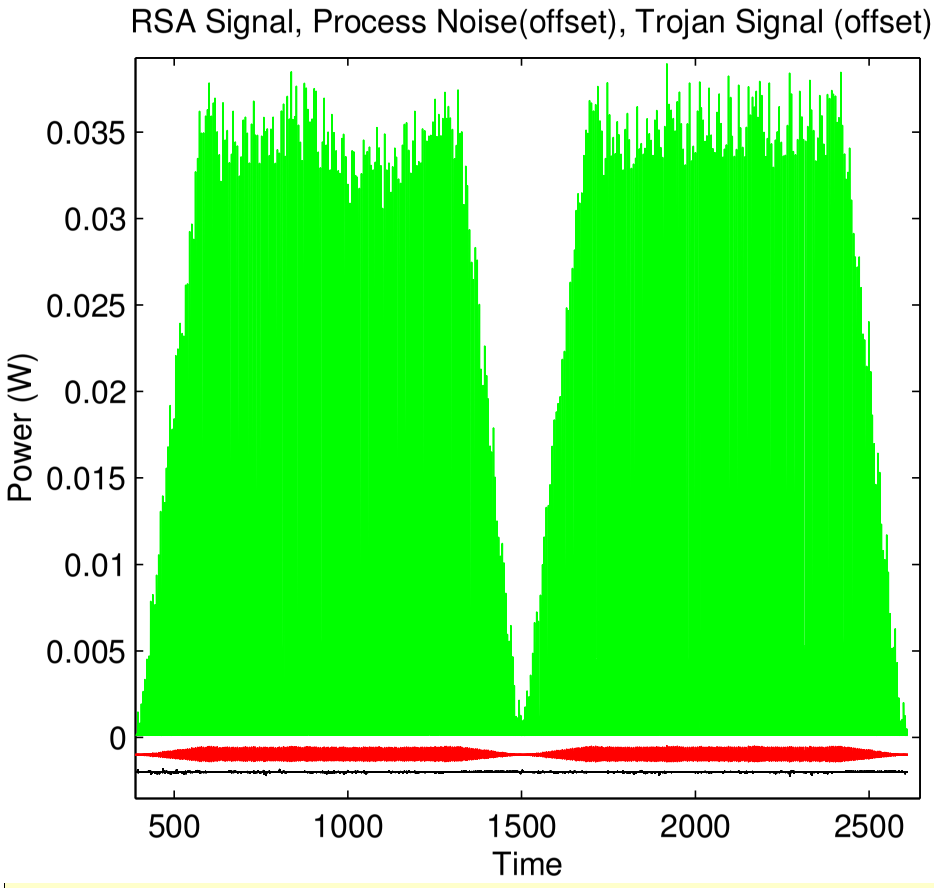
- **E.g.: Trojan tests a trigger condition to activate.**

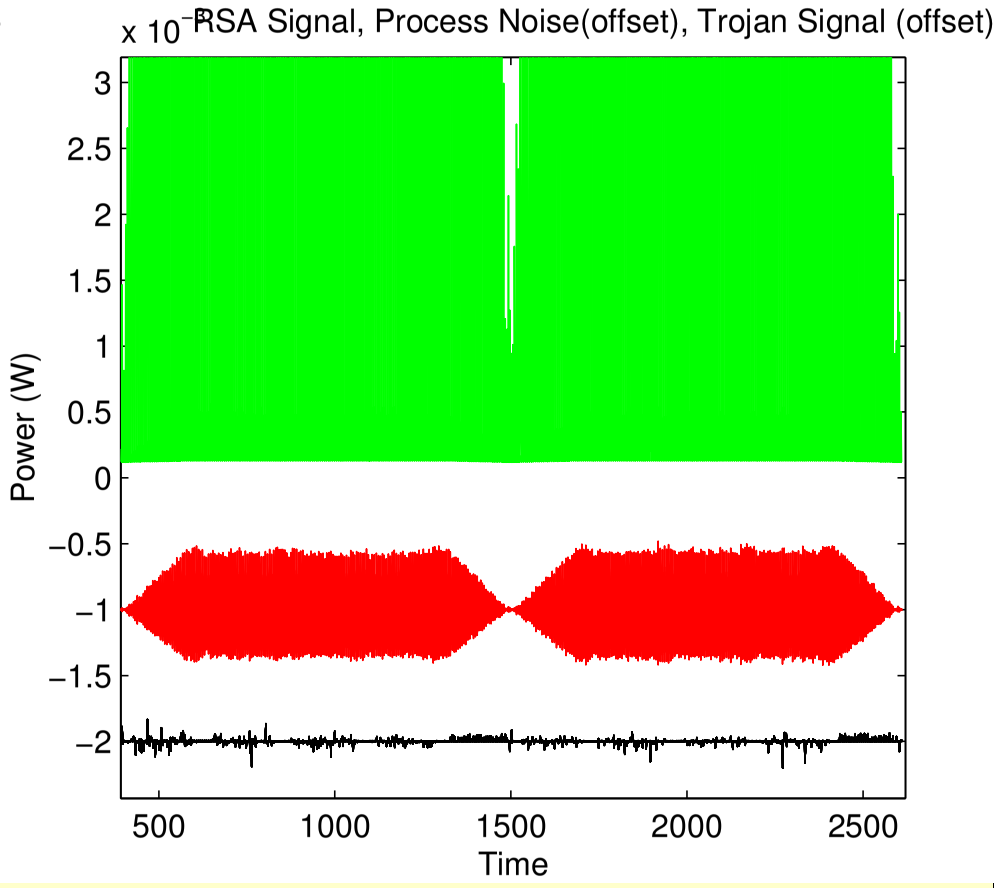
- **Trojan signal much smaller than the normal side-channel signal variations between good ICs.**
  - variations in manufacturing process !

- **We developed an approach that works at least in simulations.**

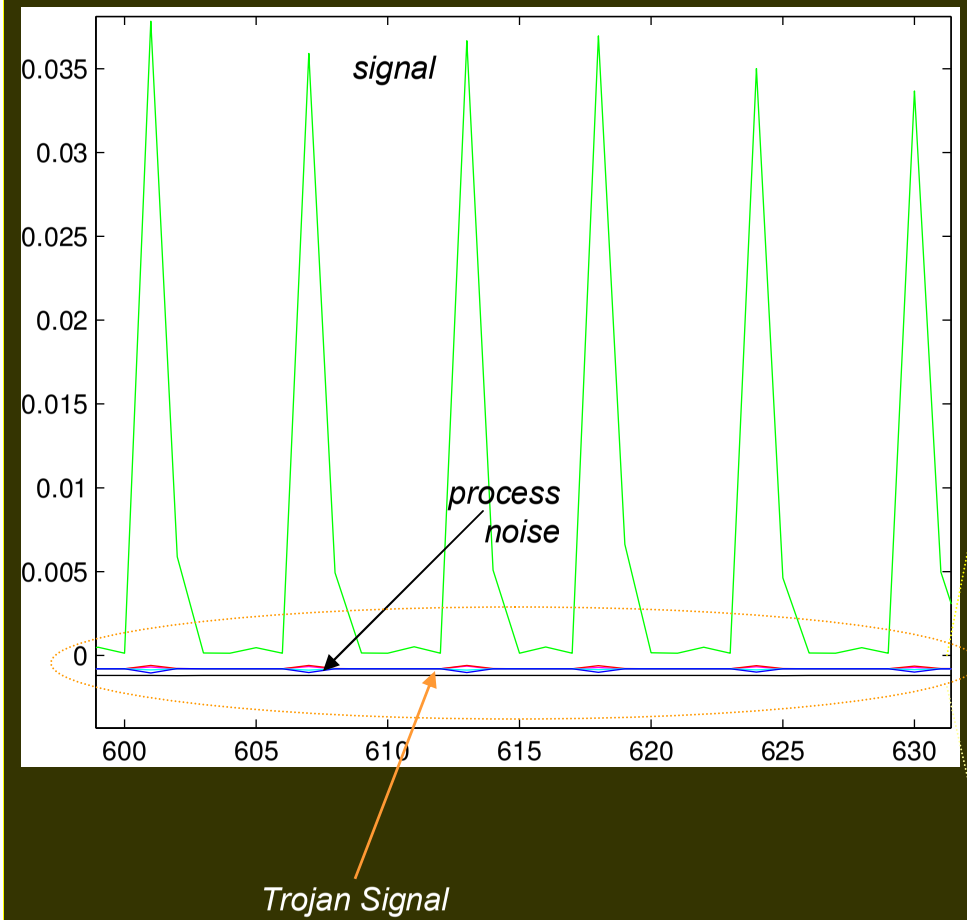
Small, Minimally Active Trojan

Main Signal
Variation due to Process Noise
Trojan Signal (checking trigger)

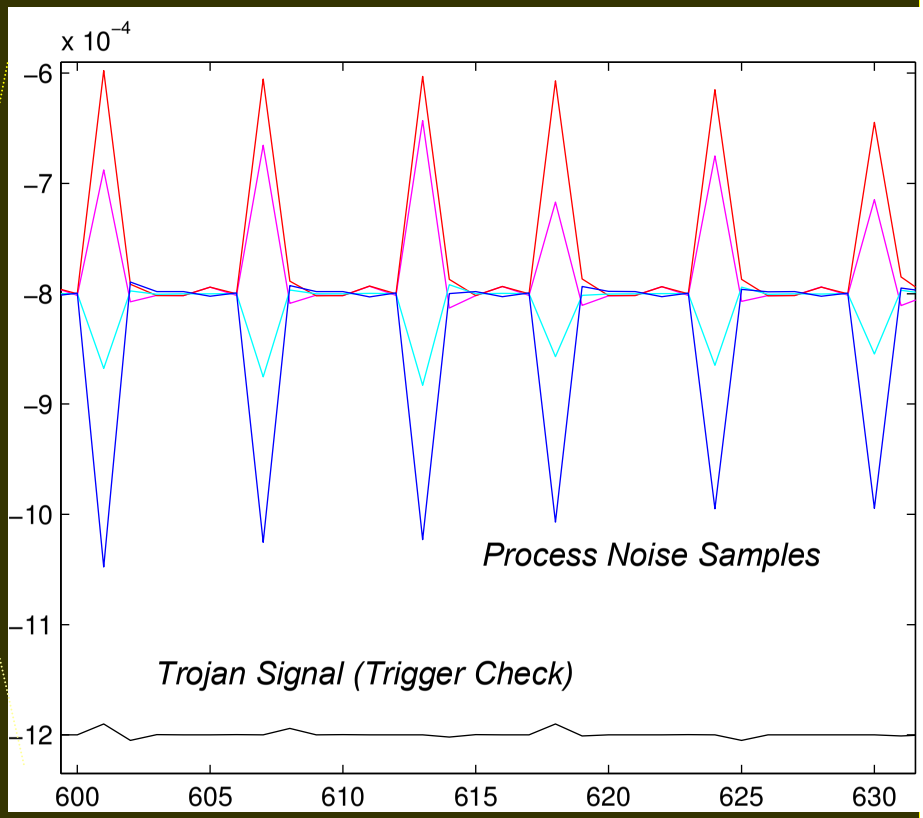




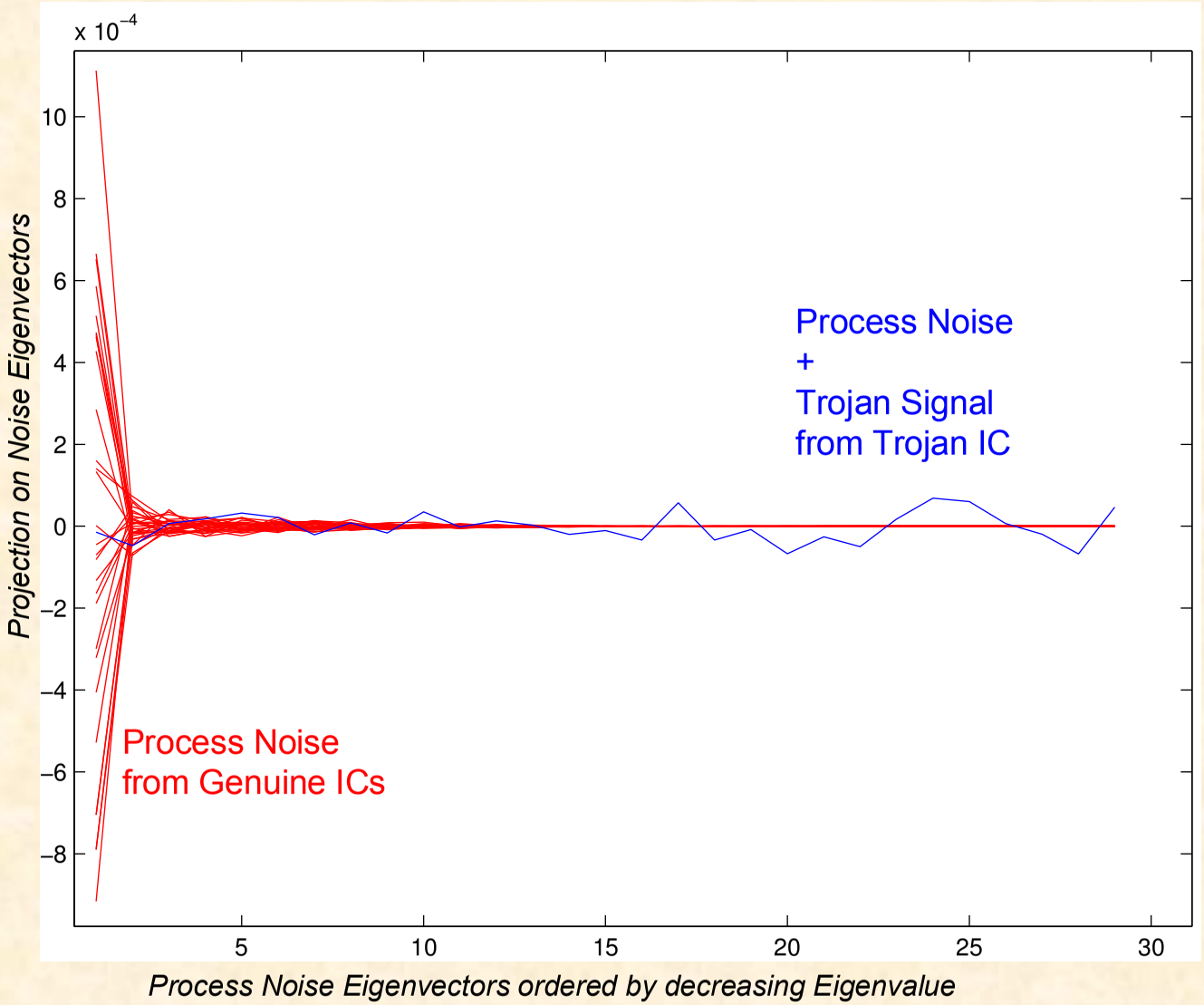
Signal, Noise, Trojan: A closer look in a time-window



- Process noise highly correlated over time window, limited degrees of freedom (much less than 30)
- Trojan contribution unlikely to correlate the same way over time window.



Distinguishing Trojan signal from process noise: a technique



- **Principal Component Analysis or Karhunen Loeve (K-L) analysis** of process noise samples identifies main orthogonal dimensions (eigenvectors) where process noise lies.
- Noise samples then projected on eigenvectors ordered by eigenvalues.
- Variation due to process noise concentrated along first few eigenvectors.
- Trojan Signal unlikely to lie solely in those dimensions
- Trojan signal likely to show up in dimensions where there is little process noise.

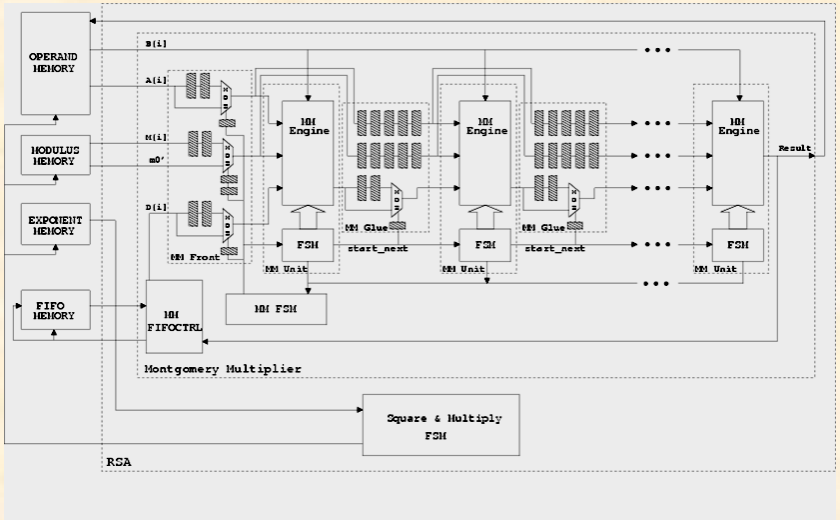
Experimental Results (via Power Simulations)

Simulation: RSA circuit

The Circuit:

RSA encryption macro

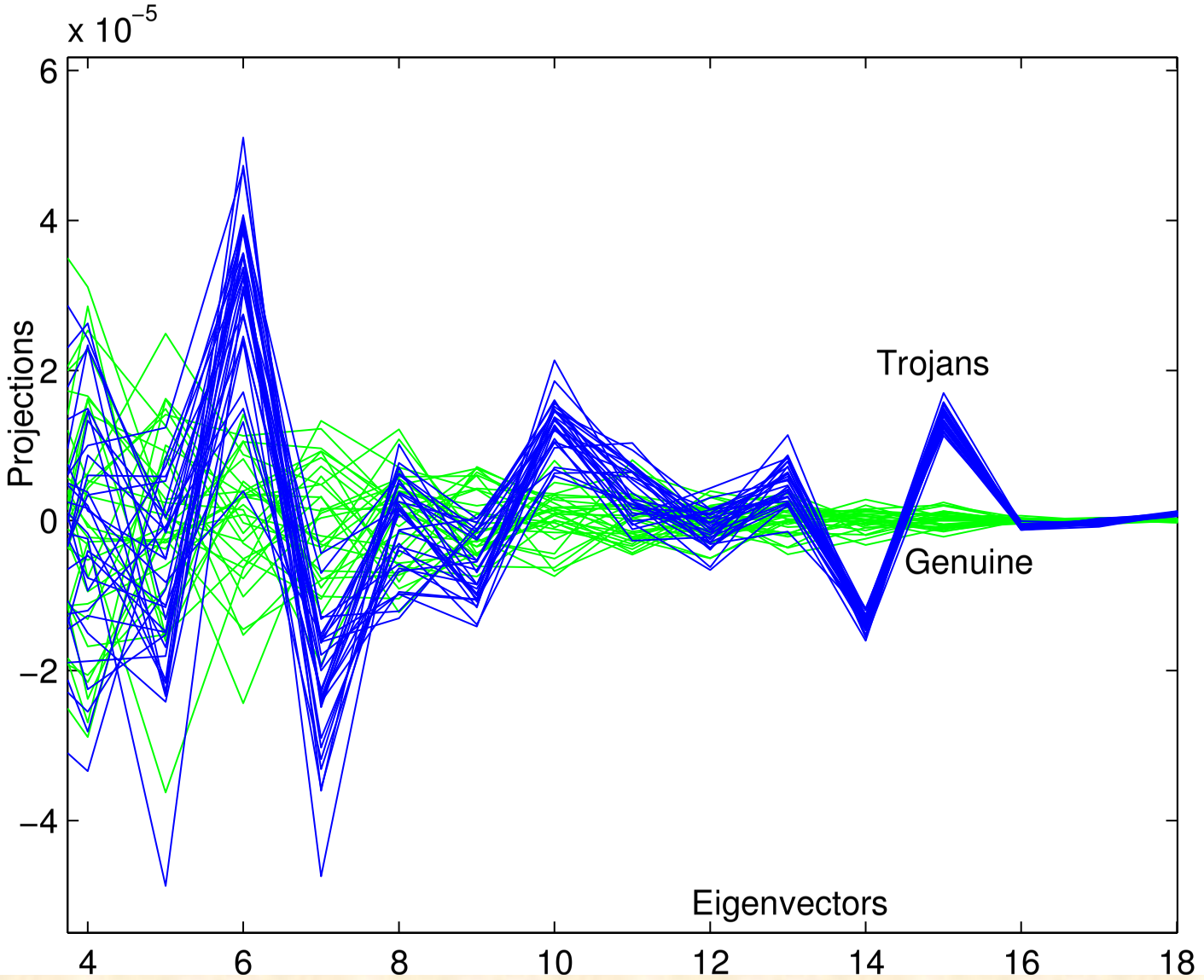
- **512-bit: equivalent area of 27914 gates, average power consumption: 3.001 mW, Maximum clock frequency of 617MHz, 5-15% process variation (delays, capacitances etc).**



- **Multiple Trojans**
  1. **Counter based Trojan:** 16-bit counter, Area: 406 gates and approx **1.43%** of the total circuit ares, Causes error upon reaching particular counter size
  2. A small but "real" **8-bit comparator based Trojan,** that creates error upon successful comparison. (NEVER ACTUALLY TRIGGERED)
     - 33 gates **(0.1% size)**
  3. A somewhat unrealistic 3 bit comparator Trojan sufficient to test the technique
     - Unclocked, **3 gates (0.01% size),** (NEVER ACTUALLY TRIGGERED)

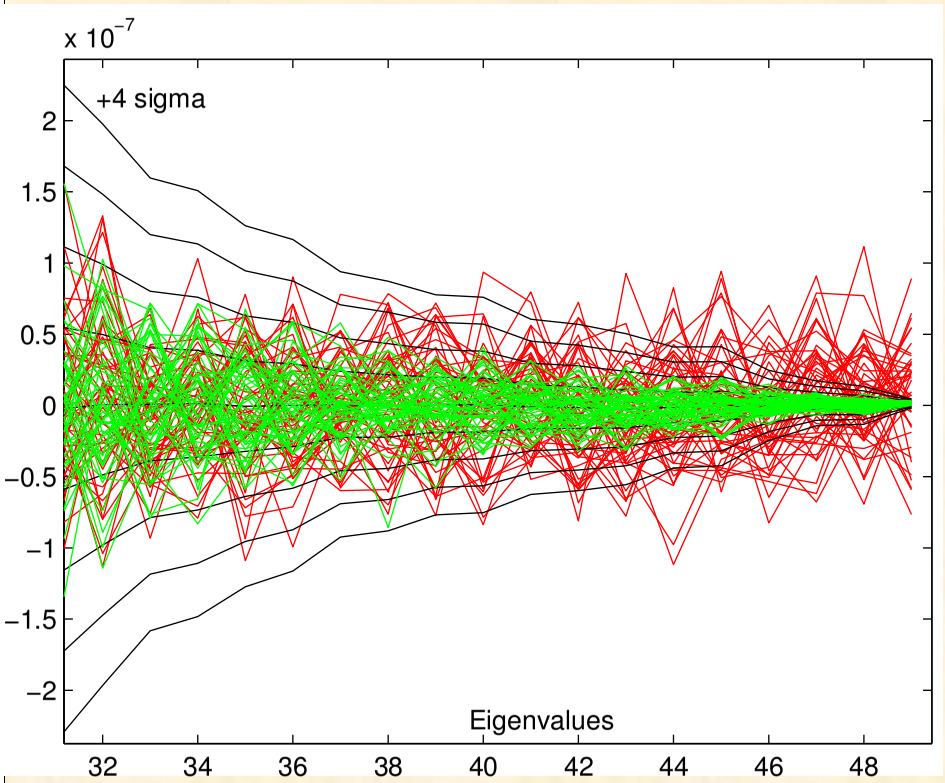
Results for different Trojans

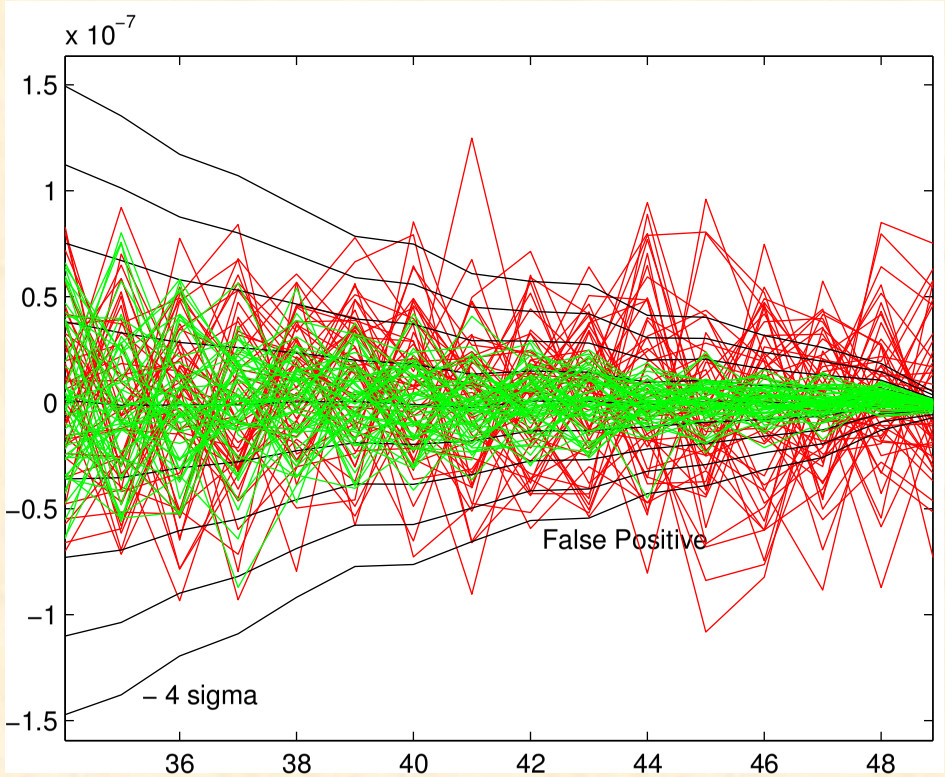
Larger Trojans easily separate from process noise



Even the tiny trojan 3 (0.01%) can be detected using statistics

- Trojan Signals
- Non-Trojan Signals
- μ +/- i*σ plots

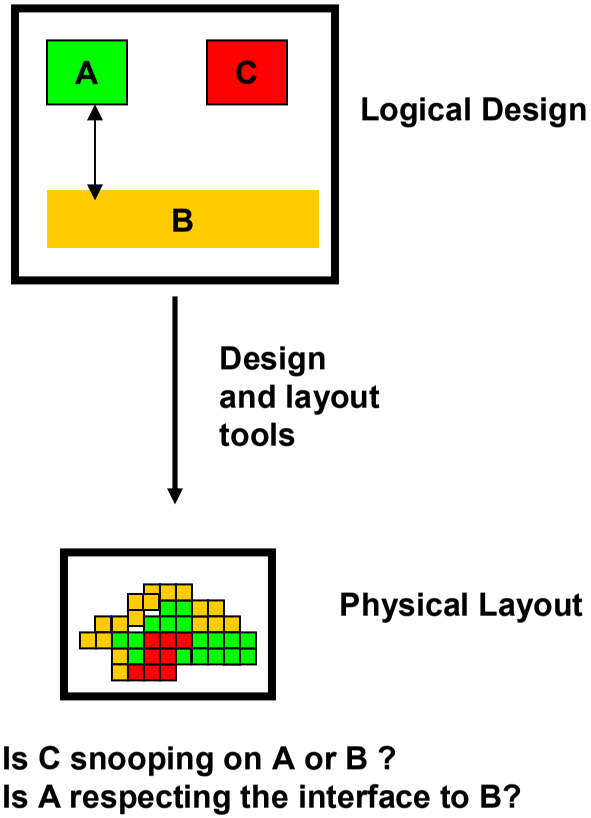




Trust Issues with FPGA Design Flows

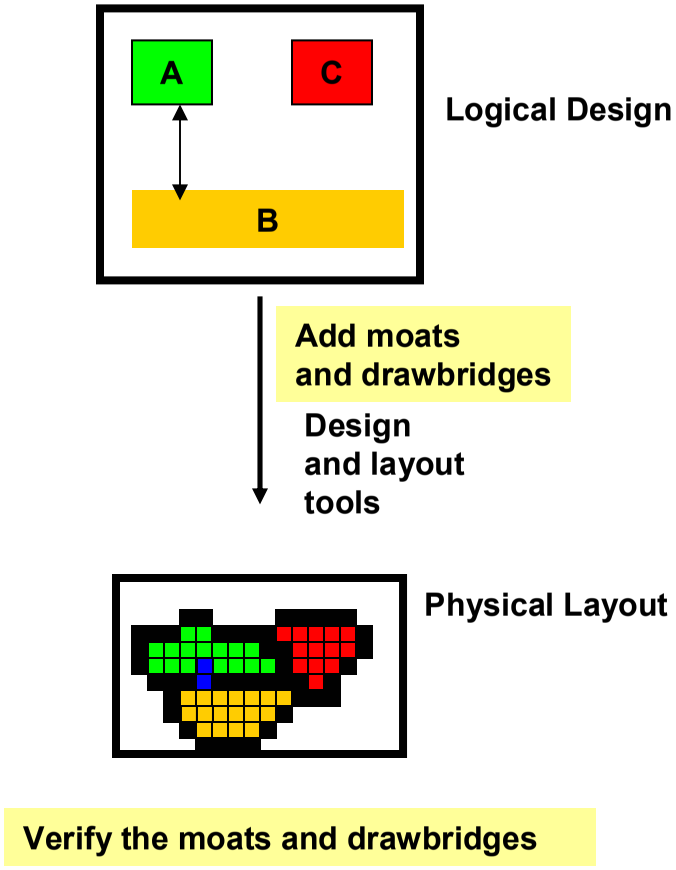
Trust issues in FPGA Design Flow

- **Many commercial and defense applications use FPGA's**
  - Need to combine many "cores" from multiple vendors/pedigrees into a single FPGA
  - Cores usually obfuscated to protect IP.
  - Major trust issues
    - One core may be designed to snoop/interfere with others !
    - Design tool may be untrustworthy



How to ensure a trustworthy FPGA design ?

- **Recent research at UC Santa Barbara & Naval Postgraduate School: Paper at IEEE S&P this year [HBWSKLNI '07]**
  - Title: Moats and Drawbridges: An Isolation Primitive for Reconfigurable Hardware
  - Moats: Basic Isolation of cores
  - Drawbridges: Interfaces between components



Epilogue

Success can bring its own problems

- **A successful product/device always gets used in ways that were unanticipated by designers and creates unanticipated problems**
  - Technical and Non-technical !

Questions