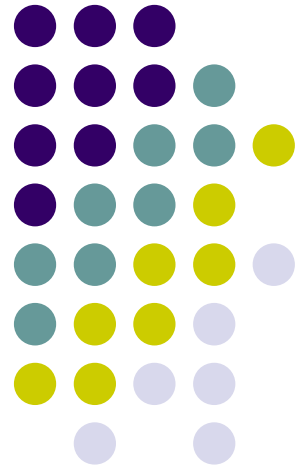
Secure Hardware PA018

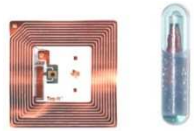
Jan Krhovják

Vašek Matyáš



Roadmap

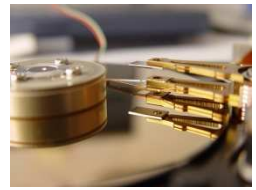
- Introduction
  - The need of secure HW
  - Basic terminology
- Architecture
  - Cryptographic coprocessors/accelerators
  - Cryptographic chip cards/smart cards
- Security categories and common attacks
  - Physical security
  - Logical security
  - Environmental security
  - Operational security
- Security requirements
  - Standards FIPS 140-1/2/3
- Secure HW and famous attacks



Why secure hardware

- Ensure (fast) secure communication and secure storage (of extremely critical data)
- Sensitive data (e.g., financial data, cryptographic keys) stored on hard disk or in memory are vulnerable
  - Adversary (with sufficient rights) can access them
  - Data in memory can be paged out to disk
  - Data in a hard disk can be backed up in unprotected storage device





- Problems with secure deletion/destruction of insecurely stored sensitive data


Where secure hardware

- Critical applications have always been banking transactions
  - Primarily due to need for secure storage
  - In 70's VISA formed worldwide banking ATM network
  - Banks can't trust themselves, their employers or customers
  - This led to evolution of so-called Hardware Security Modules and financial data networks (banking machines, sales terminals, etc.)

- Certification authorities
  - Primarily due to need for accelerating crypto operations
  - Increase in the last decade for public-key cryptography support

Basic terminology

- Hardware security modules (HSM)
  - Coprocessors
  - Accelerators
  - Cryptographic smartcards

- Host devices, API
- Attacks on HSMs
  - Physical attacks
  - Side channel attacks
  - Attacks on and with API
  - We are not interested in any form of DoS attacks!
- Top-level crypto keys – always stored inside HSM
  - Other keys can be stored outside HSM encrypted by these

Examples of Secure HW

- Multi-chip dev.
  - IBM 4758

  - IBM PCI-X

  - SafeNet PSO

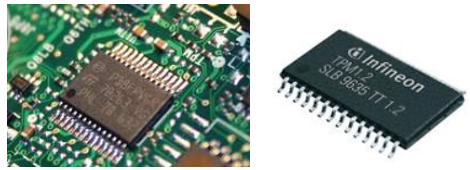
  - SafeNet PSG

- Single-chip devices
  - Gemalto SmartCards

  - SafeNet iKey

  - Dallas iButton

  - Infineon TPM chip



- VeriChip RFID TAGs

- NXP/Phillips MIFARE

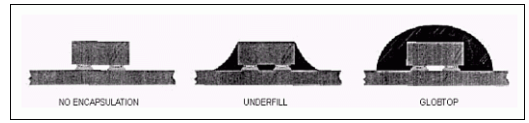
- Czech applications

Architecture of cryptographic coprocessors/accelerators

- Come out from classical von Neumann architecture
  - + Mechanisms of physical protection
    - ➢ Steel shielding, epoxy resin, various sensors
  - + Generators of true random numbers
    - ➢ Generating cryptographic material (e.g. keys, padding values)
    - ➢ Algorithmic counter-measurements against side channel attacks
  - + Special coprocessors
    - ➢ Accelerating both symmetric and asymmetric crypto
  - + Non-Volatile RAM (NVRAM) => retains its content
    - ➢ Connected to a constant power source or battery
    - ➢ Storing sensitive data (e.g. master key)
  - – I/O circuits
- Easier verification

Architecture of cryptographic smartcards

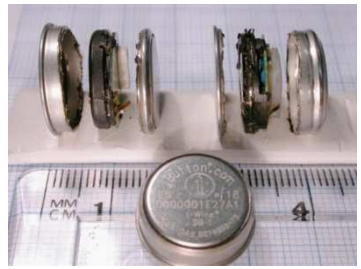
- Similar building blocks as coprocessors/accelerators
  - Everything is inside a single integrated chip
    - Limited silicon area => small size of RAM



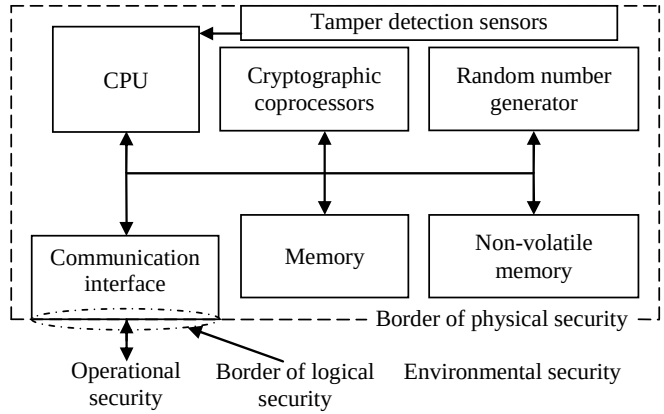
  - There is only limited power supply in mobile devices
    - New (U)SIM cards supports DES, RSA and EC cryptography
    - Their power consumption must be very small
  - Operating system is stored in ROM, applications in EEPROM
- Division according to the communication interface
  - Contact – contain contact pads
  - Contactless – contain an embedded antenna
  - Combined – single chip with both previous interfaces
  - Hybrid – more chips (and interfaces) on single card

- Super smartcard =>

Security categories



- Physical security
  - Technologies used to safeguard information against physical attack
  - Barrier placed around a computing system to deter unauthorized physical access to the computing system itself
    - Tamper: evidence, resistance, detection, response (more on the next slide)
- Logical security
  - The mechanisms by which operating systems and other software prevent unauthorized access to data
    - Access control, algorithms, protocols
- Environmental security
  - The protection the system itself
    - Access policies – guards, cameras …
- Operational security

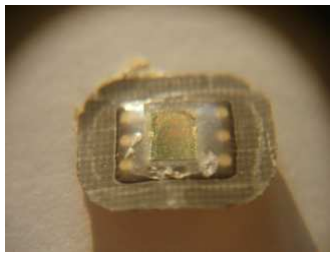


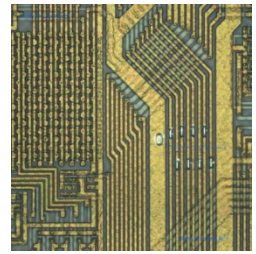
Physical security

- Tampering – the unauthorized modification of device
- Tamper evidence
  - The evidence is left when tampering occurs
  - Chemical or mechanical mechanisms
- Tamper resistance
  - Only to certain level!
  - Chemically resistant material, shielding
- Tamper detection
  - Special electronics circuits (i.e. sensors)
- Tamper response
  - Consequence of detection => destroying all sensitive information
  - Erasing/rewriting/memory destruction

Physical invasive attacks

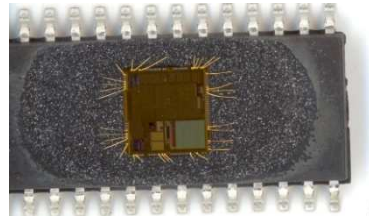
- They require a lot of time, knowledge and specialized equipment – probing station =>
- Invasive attacks (passive or active)
  - Direct access to embedded components (ALU, bus, memory …)
    - Depacking the device/chip – removing the cover (and passivation) layer
    - Micro probing – observing, manipulating or interfering the device/chip

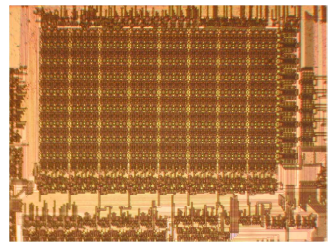
  - Reverse engineering – the process of analyzing an existing system to identify its components and their interrelationships
  - Memory readout techniques (e.g., freezing and probing)
    - Freezing by liquid nitrogen increase data retention time in RAM to hours

Physical semi-invasive attacks



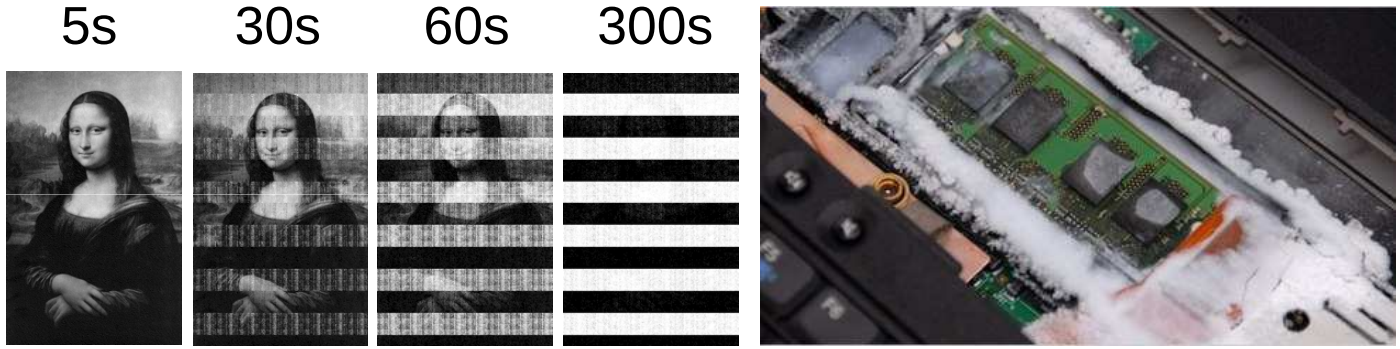
- They require only low-cost equipment
  - Easy reproduction of prepared attack particular device configuration
- Depackaging the chip – passivation layer remains
  - Utilizing electromagnetic field, UV light, X-rays, laser, (local) heating or freezing, irradiation

  - Optical fault induction
    - Illumination of SRAM can change its content
    - SRAM memory with 80x magnification =>



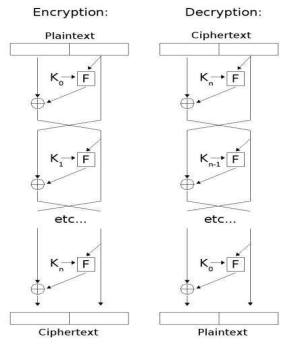
Cold boot attacks on encryption keys

- DRAMs memory retention after power is lost
  - Retain their contents for seconds to minutes



  - Horizontal bars – due design of the memory chip
- Sucessful attacks on popular disk encryption systems
  - BitLocker, FileVault, dm-crypt, and TrueCrypt
  - This is no attack on secure HW (it is classical DRAM)
- For details see: http://citp.princeton.edu/memory/

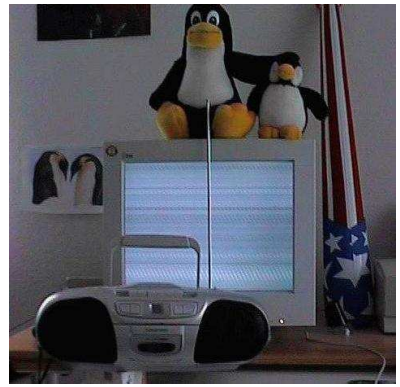
Logical security



Feistel Cipher

- Access control
  - The assumption is existence of trusted environment
- Cryptographic algorithm
  - Mathematical functions – only keys should be secret
  - Ensuring confidentiality, integrity, authentication …
- Cryptographic protocols
  - Distributed algorithms – sets of three to ten messages
    - Their single steps are created by calling of API functions
  - API is the only one (exactly defined) communication interface between HSM and the host application
    - Economy prevails security – too many supported standards in APIs
    - API of HSM thus contains hundreds functions with many parameters
      => very big space for errors and formation of new attacks

Logical (non-invasive) attacks

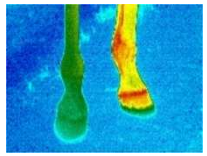


- No physical damaging of device
- Monitoring/eavesdropping
  - TEMPEST attacks (e.g., TEMPEST for Eliza =>)
    - Electronic devices emits electromagnetic radiation
    - Reconstructing data (images, keystrokes) from el.-mag. radiation
    - For more see: http://www.eskimo.com/~joelm/tempest.html or
    - http://lasecwww.epfl.ch/keyboard/
  - Side channel attacks
    - Timing analysis – measuring the time of cryptographic operations with respect to input data and algorithm implementation
    - Power analysis – measuring the fluctuations in the consumed current when the device is performing specific operations
    - Fault analysis – generating of glitches (in voltage, clock signal …)
- Software attacks on and with API
  - No specialized equipment needed, can be perf. remotely
  - Very fast and dangerous – taking only a couple of seconds!

Attacks on and with API

- Examples of commonly used API
  - Public Key Cryptographic Standard (PKCS) #11
  - Common Cryptographic Architecture (CCA)
- Three major problems of cryptographic API
  - Insufficient ensuring integrity of keys
    - Problems with backward compatibility (e.g. support of DES or RC2)
    - Meet in the Middle Attack, 3DES Key Binding Attack, Conjuring Keys …
  - Insufficient checking of function parameters
    - Banking API and working with PINs => PIN recovery attacks
    - Decimalisation Table Attacks, ANSI X9.8 Attacks …
  - Insufficient enforcing of security policy
    - PKCS #11 – only set of functions, designed for one-user tokens

Environmental security



- The asset is the device itself (not the stored information)
  - At least interesting aspect of security from analysis perspective
  - The goal is to limit attacker's opportunity to initiate an attack
    - Creating layers of hindrance (e.g. access control&policies)
    - Monitoring – using (infra/thermal) cameras

- Not necessarily applicable to HSMs operating in hostile environments (typically highly physically secured)
  - Administrators of HSMs (i.e. security officers) have a certain amount of power over a HSMs that can be misused
  - To prevent single security officer from compromising the system, the principle of dual control policy is enforced
    - At least two security officers (e.g. from different banks) must agree to change the device configuration (e.g. installing/changing of keys)
    - At least two security officers must collude to circumvent the security
  - Administrative/procedural controls should be the part of security policy whenever is it possible

Operational security

- HSM can be operated only trough functions of API
  - With API functions can programmer interact by keyboard
  - Some devices allows the user to execute limited number of exactly defined API commands (e.g. ATMs by PINpad/keypad)
- The security risks related to proper manipulation with cash machines and their interfaces are growing
  - The user should be able to recognize the fake
    - Payment terminal, ATM, card reader =>
  - The user should know what to do with keypad
  - The user should operate cash machine alone
  - The user should be aware of latest attacks as
    - Transparent overlay of keypad, Lebanese loop =>
  - The user should safeguard his PIN

Classes of adversaries I

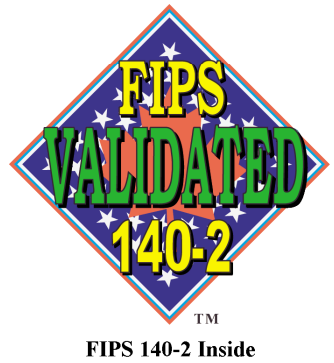
- Class 0 (script kiddies)
  - No knowledge of the system
  - Exploit existing tools (trial-and-error method)
- Class 1 (clever outsiders)
  - Often very intelligent
  - Insufficient knowledge of the system
  - Access to only moderately sophisticated equipment
  - Exploit existing weakness in the system
- Class 1.5 (well-equipped outsiders)
  - Very intelligent with basic knowledge of the system
  - Low-cost equipment to build new attacks
  - Specialized laboratories in universities, etc.

Classes of adversaries II

- Class 2 (knowledgeable insiders)
  - Specialized technical education and experience
  - They understand the parts of system + typically have access to most of it
  - Access to sophisticated tools and instruments for analysis
- Class 3 (funded organizations)
  - Teams of specialists (can be from Class II)
    - Related and complementary skills
    - Capable of in-depth analysis of the system
  - Use of the most sophisticated analysis tools
  - Design of new sophisticated attacks

Security requirements on HSM: FIPS 140-1/2 (I)

- FIPS 140-1 (11.1.1994), FIPS 140-2 (25.5.2001)
  - Related to design and implementation of HSM
- Some of 11 areas of security requirements:
  - Cryptographic module specification
  - Cryptographic module ports and interfaces
  - Role, services, and authentication
  - Physical security
  - Operational environment
  - Cryptographic key management
  - Mitigation of other attacks
  - …
- Testing and independent rating in each area
  => 4 overall levels of security (level 4 = best)



Security requirements on HSM: FIPS 140-1/2/ (II)

- Standard defines 4 levels of security
  - Level 1 – no physical security required
    - At least one approved security function
    - Classical example – cryptographic software for normal computers
  - Level 2 – temper evidence required
    - Role-based authentication
    - OS must be evaluated
    - Classical example – smart card
  - Level 3 – tamper detection & response required
    - Authentication based on identities
    - Example – Chrysalis-ITS Luna CA^3
  - Level 4 – environmental failure protection/testing
    - Example – IBM 4758 or IBM PCIXCC

FIPS 140-2 in detail

	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 or physically 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.			

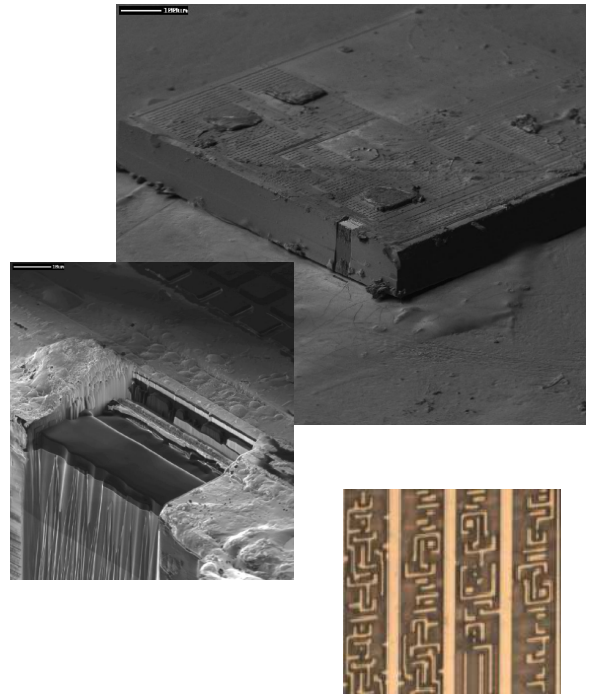
First draft of FIPS 140-3 (13.7.2007)

- Each of 11 security areas redefined + added fifth security level
- Strengthening of several requirements
  - Physical & software security (new area)
  - Protection against non-invasive attacks
- Most important changes
  - Authentication
    - Definition of roles and services (1); role-based or identity-based authentication (2); identity based operator authentication (3); two-factor authentication (4,5)
  - Non-invasive attacks
    - No requirements (1,2), protection against time analysis (3), protection against simple/differential power analysis (4), protection against electromagnetic emanation (5)
  - Self-tests
    - Continuous RBG testing, RBG entropy source test

Second draft of FIPS 140-3 (11.12.2009)

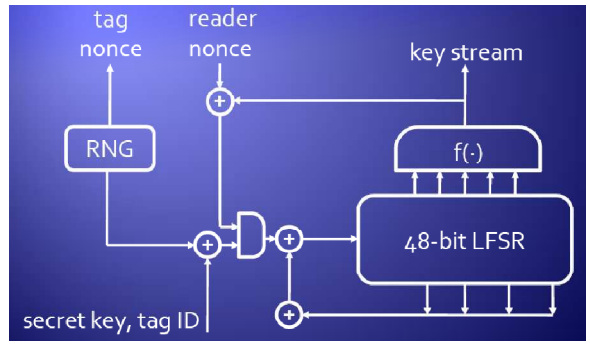
- Changes based on comments received on the first public draft
  - Reverts to 4 levels of security as specified in FIPS 140-2
  - Reintroduces the notion of firmware cryptographic module
    - Defines the security requirements for it
  - Limits the overall security level for software cryptographic modules to Level 2
  - Removes the formal model requirement at Level 4
  - Requirements for mitigation of non-invasive attacks at higher security levels
  - Elimination of the requirement for formal modeling at Level 4
  - Modified conditions for pre-operational/power-on self-tests, and strengthened integrity testing
- Official deadline for comments: 11.3.2010
  - Link: http://csrc.nist.gov/publications/drafts/fips140-3/revised-draft-fips140-3_PDF-zip_document-annexA-to-annexG.zip
  - Development: http://csrc.nist.gov/groups/ST/FIPS140_3/

Famous attacks I



- Security of Mifare RFID tags
  - Reconstruction of integrated circuit from photos of chip
  - Chip has several thousand gates, but only ~70 different types

- 16-bit RNG based on LSFR
  - Seed derived from value of read
    - Time delay between power on and the reception of message data from the contactless card reader
  - No non-linear component in feedback
  - Output derived only from fixed subset of bits

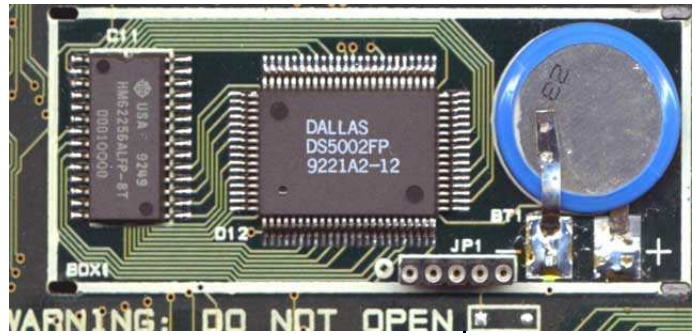


- For details see

  http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html

  http://www.smartcard.co.uk/MifareInSecurity.pdf

Famous attacks II



- Cipher Instruction Search Attack on the Bus-Encryption Security Microcontroller DS5002FP
  - Effective and cheap attack => class of adversary I
  - Encryption of data bus and external address
    - Secret key is stored in battery buffered register inside chip
    - External RAM contains encrypted data on encrypted address
  - Sending suitable instructions and observing their effects
    - MOV 90h, #42h (encoded as 75h 90h 42h) outputs byte value 42h on parallel port P1 (address 90h)
      - 2^{16} combinations for the first two encrypted instruction bytes
      - Testing 2^8 values for first byte => decryption for one address
    - NOP followed by MOV increases the address from which is MOV fetched

- For details see

  http://csdl.computer.org/dl/trans/tc/1998/10/t1153.pdf

  http://www.cl.cam.ac.uk/~mgk25/tamper.pdf

Famous attacks III

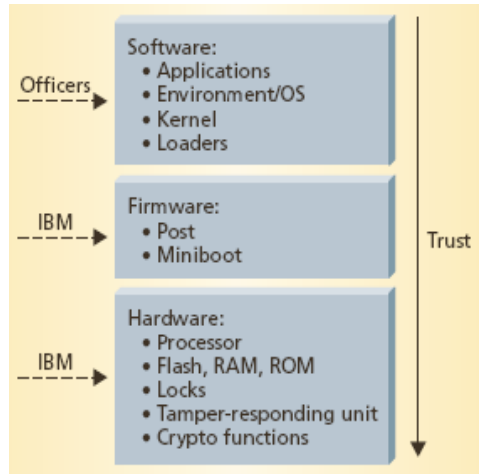
- Reverse engineering of Chrysalis-ITS (Safenet) Luna CA^3
  - Certified at FIPS 140-1 Level 3
  - Examination of vendor-specific functionality (cloning protocol)

- Interesting findings
  - Disassembling shows no potting material
    - Only red glue – maybe for structural integrity or head dissipation
  - Proprietary Luna API (translated to PKCS#11 by higher-level lib.)
    - Found number of undocumented Chrysalis PKCS#11 extensions
    - Owners of key material stored in device can extract private key material (to the clear) or migrate them into other LunaCA^3 devices

- For details see

  http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-592.pdf

Famous attacks IV



- IBM 4758 (with CCA API)
  - HW & FW are certified at FIPS 140-1 Level 4
  - Layered design
    - Higher layers confide in lower layers
    - HW and FW are under control of IBM
    - SW controls the owner
  - Surprisingly easy logical attacks on CCA API

- For details see

  http://www.cl.cam.ac.uk/~mkb23/research/PIN-Cracking.pdf
  http://www.cl.cam.ac.uk/~mkb23/research/Thesis.pdf
  http://www.cl.cam.ac.uk/~mkb23/research/CCA-EMV.pdf
  http://www.cl.cam.ac.uk/~mkb23/research/Clulow-Dissertation.pdf

Conclusions

- Secure hardware
  - Limited functionality – easier to verify – better security (than multipurpose hardware)
  - Dedicated circuits – faster than software implementation
- Secure hardware doesn’t guarantee absolute security
  - Any secure hardware can be reengineered
  - Main reason of its usage is increased cost of attack
    - And also better performance of demanding crypto operations
- Bad design and integration imply attacks
  - The security of current generation banking APIs is really bad with respect to insider attacks
  - Number of standards implemented ensures interoperability but also causes errors