

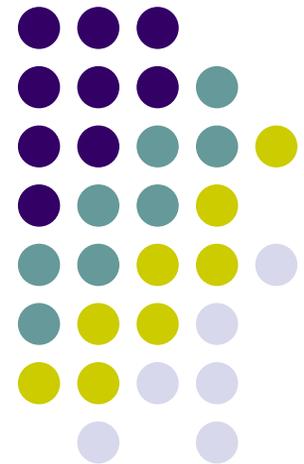
Secure Hardware

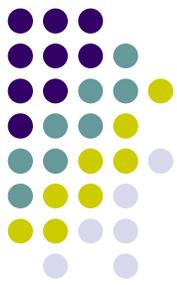
PV018

Masaryk University
Faculty of Informatics



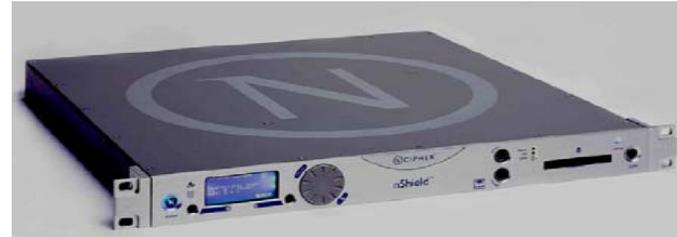
Jan Krhovják
Vašek Matyáš

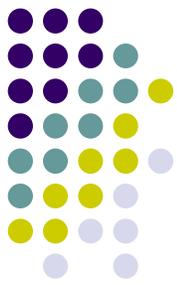




Roadmap

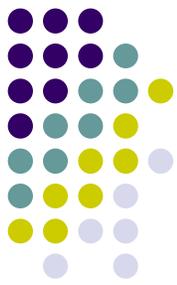
- Introduction
 - The need of secure HW
 - Basic terminology
- Architecture
 - Cryptographic coprocessors/accelerators
 - Cryptographic chip cards/smart cards
- Security categories and common attacks
 - Physical security
 - Logical security
 - Environmental security
 - Operational security
- Security requirements
 - Standards FIPS 140-1 and FIPS 140-2
- Programmable cryptographic coprocessor IBM 4758





Why secure hardware

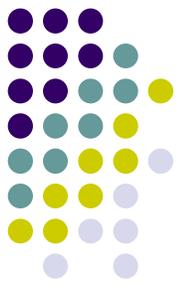
- Ensure (fast) secure communication and secure storage (of extremely critical data)
- Sensitive data (e.g. financial data, cryptographic keys) stored on hard disk or in memory are vulnerable
 - Adversary (with sufficient rights) can access them
 - Data in memory can be paged out to disk
 - Data in a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



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



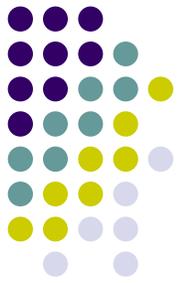


Basic terminology

- 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



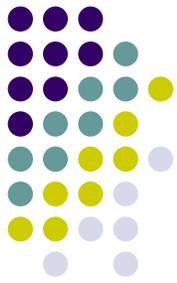
Architecture of cryptographic coprocessors/accelerators



- 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



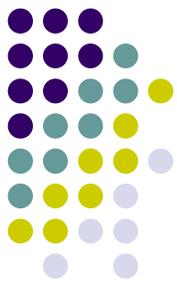
Architecture of cryptographic smartcards



- Similar building blocks as coprocessors/accelerators
 - Everything is inside a single integrated chip
 - Problems with limited silicon area => only 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

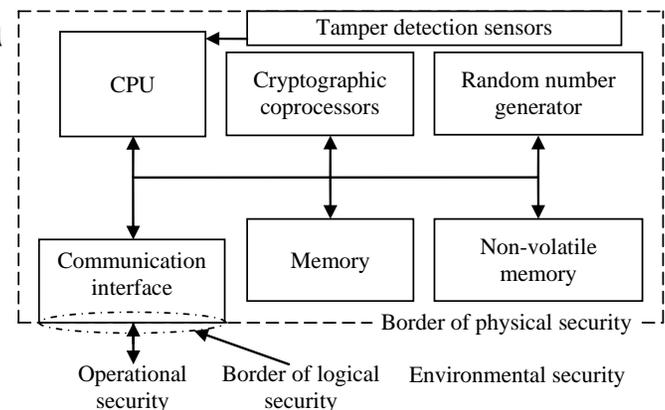
- Super smartcard =>

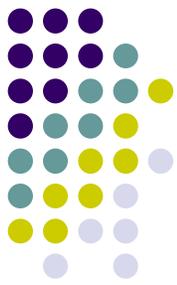




Security categories

- 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

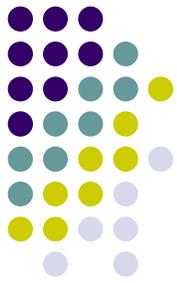
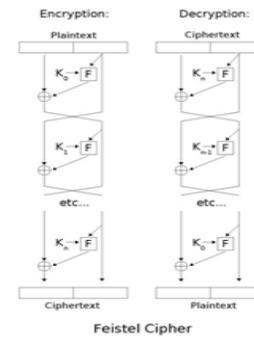


Physical attacks

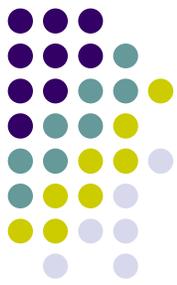


- Invasive attacks (passive or active)
 - Direct access to embedded components (ALU, bus, memory ...)
 - Micro probing – observing, manipulating or interfering the device/chip
 - Reverse engineering – the process of analyzing an existing system to identify its components and their interrelationships
 - Memory readout techniques (e.g. freezing and probing)
 - Freezing by liquid nitrogen can increase data retention time in RAM to hours
 - They require a lot of time, knowledge and specialized equipment
- Semi-invasive attacks (only on integrated chip cards)
 - Depackaging the chip, but the passivation layer remains
 - Utilizing UV light, X-rays, laser, electromagnetic field, local heating
 - Optical fault induction – illumination of SRAM can change its content
 - They require only low-cost equipment
 - Easy reproduction of prepared attack for the same HW, FW, SW

Logical security

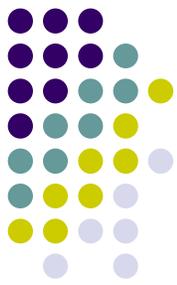


- Access control
 - The assumption is existence of trusted environment
- Cryptographic algorithm
 - Mathematical functions – only keys should be secret
 - Ensuring confidentiality, integrity, authentication ...
- Cryptographic protocols
 - Distributed algorithms – sets of three to ten messages
 - Their 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 attacks



Logical attacks

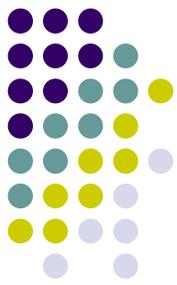
- Non-invasive attacks
 - No physical damaging of device
 - Monitoring/eavesdropping
 - TEMPEST attacks
 - Electronic devices emits electromagnetic radiation
 - Reconstructing data from electromagnetic radiation
 - Side channel attacks
 - Timing analysis – measuring the time of cryptographic operations 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
 - They are very fast – taking only a couple of seconds



Attacks on and with API

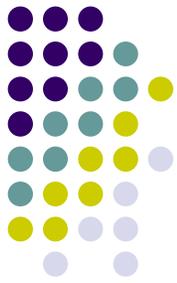
- Examples of commonly used API
 - Public Key Cryptographic Standard (PKCS) #11
 - Common Cryptographic Architecture (CCA)
- Three major problems of cryptographic API
 - Insufficient ensuring integrity of keys
 - Problems with backward compatibility (e.g. support of DES or RC2)
 - Meet in the Middle 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

Example of attack on API: Conjuring Keys From Nowhere

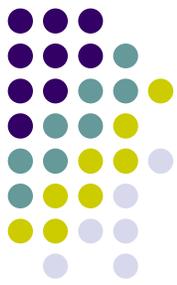


- Unauthorized generating of keys stored outside HSM
 - Random value of encrypted key is given to HSM
 - Older HSMs used this technique to legitimate key generation
 - Today is it considered as attack
 - After decryption is the value of key also random
 - In the case of DES has with probability $1/2^8$ good parity
 - DES key is stored with odd parity – LSB in each octet is parity bit
 - In the case of two-keyed 3DES-2 has a good parity with probability $1/2^{16}$ (and this is still achievable)
 - These keys can served to form more complicated attacks
- The defense lies in carefully designed key formats
=> e.g. add before encryption checksum + timestamp

Environmental security



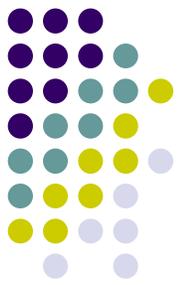
- The asset is the device itself (not the stored information)
 - At least interesting aspect of security from analysis perspective
 - The goal is to limit attacker's opportunity to initiate an attack by creating layers of hindrance (e.g. access policies, controls)
 - Not necessarily applicable to HSMs operating in hostile environments (they are typically highly physically secured)
 - The exception are the administrators of HSMs (i.e. security officers)
 - They 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

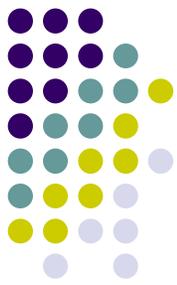
- HSM can be operated only through functions of API
 - With API functions can programmer interact by keyboard
 - Some devices allows the user to execute limited number of exactly defined API commands (e.g. ATMs by PINpad/keypad)
- The security risks related to proper manipulation with cash machines and their interfaces are growing
 - The user should be able to recognize the fake
 - Payment terminal, ATM, card reader =>
 - The user should know what he 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

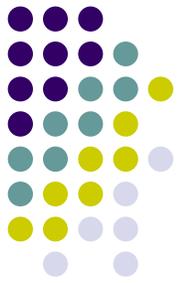
- Class 0 (scripting kiddies)
 - No knowledge of the system
 - Exploit existing tools (trial-and-error method)
- Class 1 (clever outsiders)
 - Often very intelligent
 - Insufficient knowledge of the system
 - Access to only moderately sophisticated equipment
 - Exploit existing weakness in the system
- 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

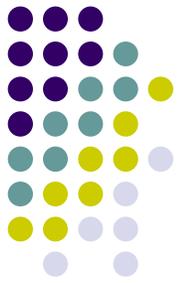
- Class 2 (knowledgeable insiders)
 - Specialized technical education and experience
 - They understand the parts of system + typically have access to most of it
 - Access to sophisticated tools and instruments for analysis
- Class 3 (funded organizations)
 - Teams of specialists (can be from Class II)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



- 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)

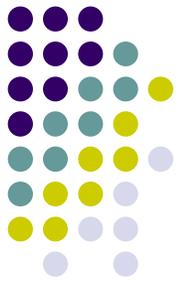
Security requirements on HSM: FIPS 140-1(2) (II)



- 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 – tamper 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 CA³
 - Level 4 – environmental failure protection/testing
 - Example – IBM 4758 or IBM PCIXCC



FIPS 140-2 in detail



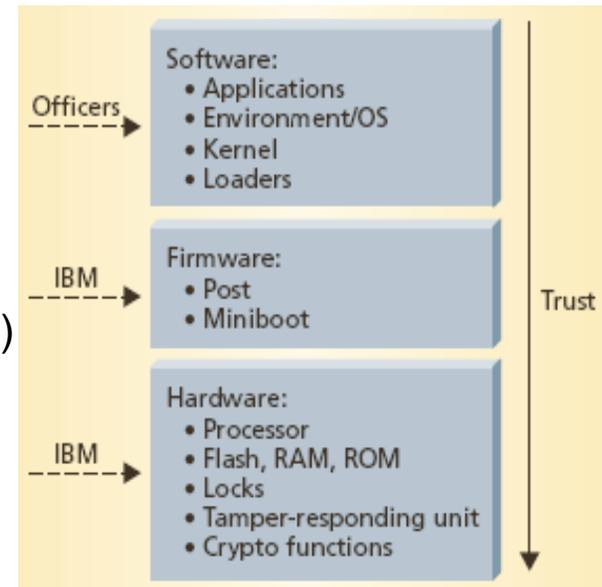
	<i>Security Level 1</i>	<i>Security Level 2</i>	<i>Security Level 3</i>	<i>Security Level 4</i>
Cryptographic Module Specification	Specification of cryptographic module, cryptographic boundary, Approved algorithms, and Approved modes of operation. Description of cryptographic module, including all hardware, 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.			

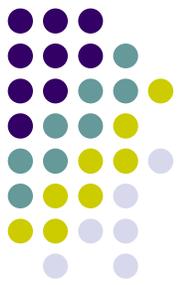
IBM 4758 PCI CC

IBM PCI-X CC (new)



- Programmable cryptographic coprocessors
- HW & FW are certified at Level 4
- Layered design
 - Provided SW is a sample application (without guarantees)
 - Many customers use it (to their damage 😊)
 - Higher layers confide in lower layers
 - HW and FW are under control of IBM
 - SW controls the owner



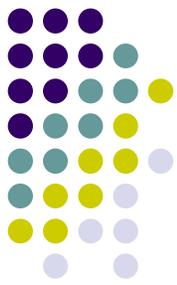


Performance of IBM CC

- Comparison of IBM cryptographic coprocessors =>

<i>Function</i>	<i>Performance (MB/s)</i>
DES (56-bit key)	200
TDES (168-bit key)	67
AES (128-bit key)	185
AES (192-bit key)	156
AES (256-bit key)	136
SHA-1	198
MD5	239
RSA (1024-bit CRT key)	3300 operations per second

<i>Operation</i>	<i>Performance (operations per second)</i>		<i>Ratio</i>
	<i>PCICC (4758)</i>	<i>PCIXCC</i>	
Generate 1024-bit RSA digital signature	104	1172	11.3
Generate 2048-bit RSA digital signature	42	458	10.9
Encipher 1024-byte blocks of data (single-DES) [†]	166	1183	7.1
Encipher 1024-byte blocks of data (TDES)	168	1145	6.8
Generate RSA key, 1024-bit CRT ^{††} format	0.28	1.8	6.4
Generate wrapped DES key	125	1007	8.1
Export (wrap) DES key	190	1214	6.4
PIN translate	106	1075	10.1
PIN encrypt	147	1488	10.1
Derive diversified key	131	1048	8.0



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