New challenges in Smart Card design

July 13th , 2005

MPSOC 05

Margaux

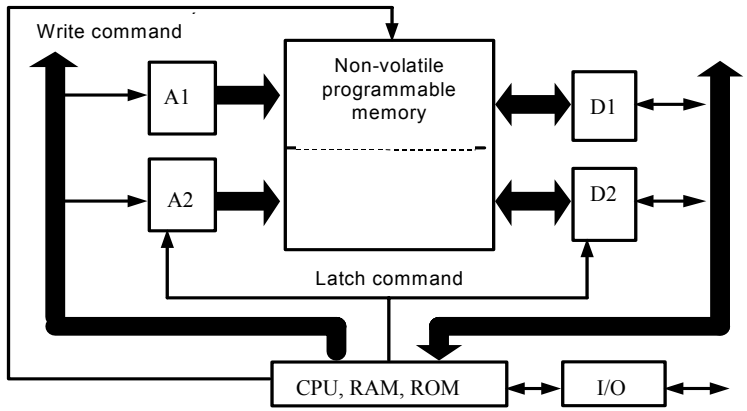
Jean-Pierre Tual

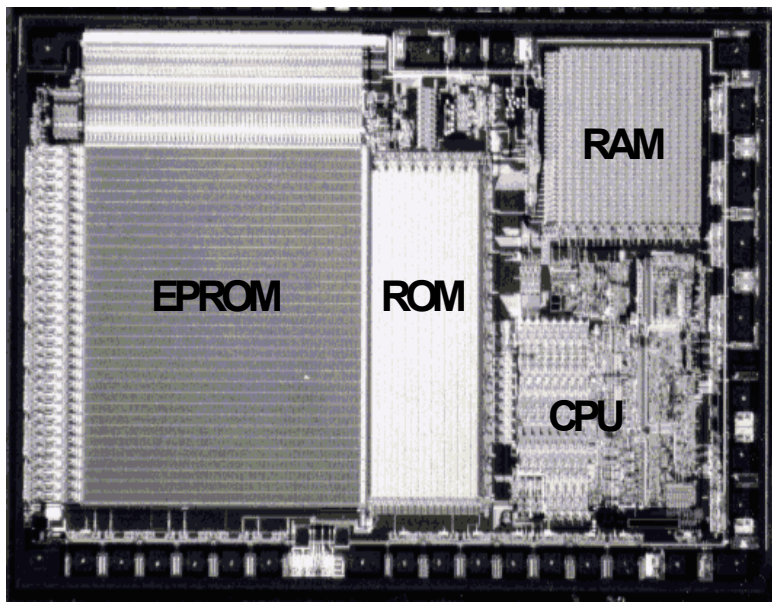
Agenda

- Smart-card key concepts
- Smart card design methodology short story
- Key trends in smart cards environment
- What needs to be addressed by the Smart Card industry
- HW evolution and challenges
- SW evolution and challenges
- Some more global concerns
- Conclusions

Smart card key concepts

- **Autonomous computer**
  - SPOM architecture
  - CPU check of all logic/electrical signals before sending them to NVM
- **Trusted Personal Device**
  - Guards privacy
    - Tamper resistance
    - No information disclosure during communication
  - Does exactly what is expected
    - Certification
    - User interaction and control
  - Refuses to perform what illegal users expect it to do
- **Portability/Usability**
  - Small footprint
    - 20-25 mm2 max for classical plastic embedding
  - Low cost
  - User customizability





axalto

Smart card design methodologies short story

- **Heroic times (1979 circa 1990)**
  - **HW** : “recycled” 4-8 bit µProcessors from other industrial applications re-enforced by ad-hoc measures
  - **SW**: low-level assembly code, intermixing basic OS and applications
  - **Security paradigm:** Security by obscurity-ad-hoc measures
- **Early productivity times (1990-1996)**
  - **HW:** dedicated 8 bit µProcessors with powerful security mechanisms (sensors, crypto accelerators)
  - **SW:** development of a new level programming paradigms with clearer cut OS applications- Partial introduction of high-level languages for card programming
  - **Security paradigm:** Industrial application of formal security certification
- **Industrial times (1996-2002)**
  - **HW:** smart-card industry to embark on Moore’s law- 16 bit/32 bit µProcessors appearing in the landscape. NMV doubling about every 18 Months
  - **SW:** new shift of smart-card SW paradigm with JavaCard and later .net RTEs
  - **Security paradigm:** Common Criteria with large public exposure of security stakes, requirements and threats

Smart Cards: new applications

- **Interoperable multi-application Citizenship or Governmental applications**
- **Secure/seamless payment of goods over the internet, mobile or multimedia networks**
- **Airport, seaports and large mass-transit area security solutions**
- **Premium Multimedia services over Mobile Networks (CAS for Pay-TV)**
- **Protection/development of eEuropean content industry through the deployment of secure DRM solutions**
- **Interactive, high-speed, multimode, multi-standard service roaming**
- **Highly secure intranet or extranet applications**
- **Home network and ambient intelligence**

No more selling KB's of memory but value !!

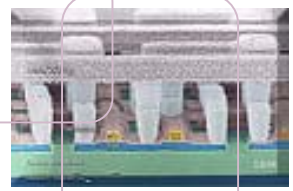
⇒Existing design paradigms not sustainable in the long term!

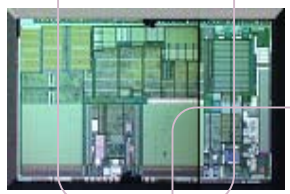
axalto

The key role of security

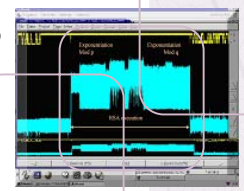
Over 25 years of race between designers and hackers on HW/SW protection

- **Invasive attacks**
  - RNG corruption
  - Bus probing
  - Focus Ion Beam
  - Pico-second Imaging
  - …

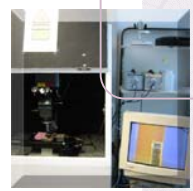




- **Observation attacks**
  - Timing
  - DPA, SPA
  - Electromagnetic
  - Alpha-particles
  - …



- **Side channel attacks**
  - Glitches injection
  - Light attacks
  - Laser pulses
  - …



- **SW attacks**
  - Trojan horse
  - Differential Fault analysis
  - Protocols
  - Cryptography

- **Independent certification**
  - FIPS
  - Common Criteria

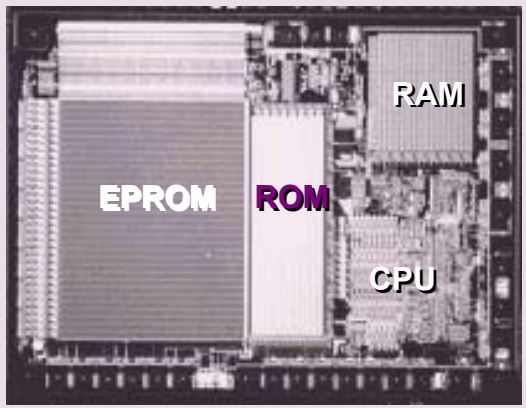
axalto

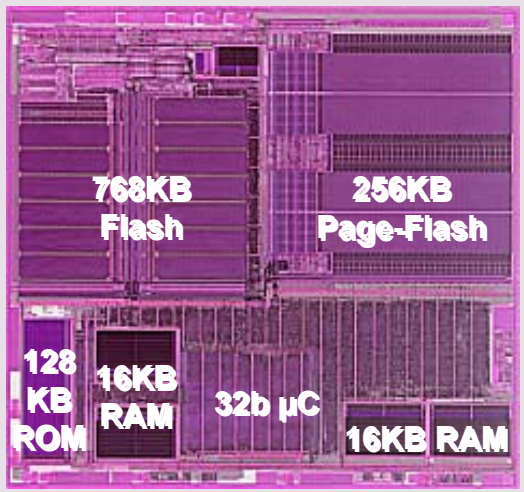
What needs to be addressed by the Smart-card industry

- **Homogeneous evolution of**
  - CPU architecture
  - Internal memory architecture and Secondary Storage
  - Communication bandwidth and Multiplicity of supported interfaces
- **Evolution of SW design paradigms**
  - Openness of programming interfaces: High-level RTE, standard communication stacks
  - Embedded SW internal architecture and design flow
  - Richer OS capabilities, Real Time, Multi-thread, Multi Application, Power-aware
  - SW validation and test: formal techniques, automation
- **Security management**
  - From security to TRUST (HW/SW cooperation, incorporation of privacy)
  - Fast certification granting, maintaining and updating cycle times
- **HW/SW co-design**
  - Incorporate best practices from embedded system world (productivity)
  - Export best-practices to secure embedded system world (Return On Investment)
  - Develop trust-constrained HW/SW design flows
- **Design Environment “invariants”**
  - Cost
  - Security and “out of spec” stressing
  - Constrained resources

CPU evolution- tremendous progress made by the industry

- **From 1981…**
  - First industrial Spom (SC µP)
  - RAM: 36B, ROM: 1,6 KB, EPROM: 1 KB
  - NMOS 3,5 µ- 42 K-transistors
  - 600 lines of embedded assembly code
- **…To 2005**
  - New generation of smart-card IC's
  - RAM: 16KB, 1MB Flash, 128 K ROM
  - Full 32 bit architecture
  - ACL memory protection
  - CMOS 0,18µ - over 1,5 M transistors
  - Embryonic "MPSOC": dedicated crypto-processor
  - 300 K-lines of embedded C-code
  - .NET Web services enabled framework
- **What are the next challenges?**
  - New internal communication schemes for new applications
  - Powerful memory protection schemes
  - More and more dedicated "co-processors": crypto, biometry,RTE accelerators
  - Multiple protocols support (USB, NFC, SPI, UWB?….)
  - Memory stacking, external memory support





axalto

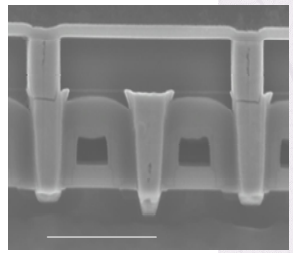
Internal Memory architecture

- EEPROM not sustainable on long term (cost, shrink-ability)
- RAM will remain a rare and costly resource
- Alternative choices for NVM

	EEPROM	FLASH	FRAM (Ferroelectric)	MRAM (Magnetic)	PCM (Phase change)
Relative cell size*	5 - 10	0.25 - 1	3 - 10	1 - 3	0.8 - 2
Scalability	Poor	Fair	Poor	Poor	Good
Granularity (E)	Byte / Word	Block	Bit	Bit	Bit
Endurance	10^6	10^5	10^{10} (destructive read)	$>10^{14}$	10^{12}
Write time (P/E)	ms / ms	µs / s	< 100 ns	< 100 ns	< 100 ns
Write Power	10V x 100µA *(1000)*	5V x 1mA *(5000)*	3V x 100µA *(300)*	1.8V x 10mA *(1800)*	3V x 1mA *(3000)*
Maturity (target volume date)	Volume prod.	Volume prod.	Limited prod.	Test chips > 2004	Test chips >2004

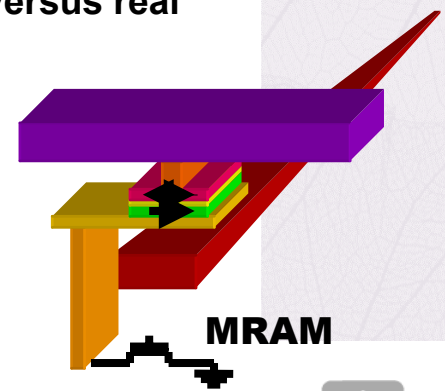
Smart-card specific impact of the internal NVM

- Smart-card may become a more and more “memory-dominated” device
- On-chip NVM limitation may require different RAM usage
  - Availability of a RAM Cache or Buffer
  - CACHE still limited by:
    - Initial loading of program in fast applications
    - Cache management vs. security
  - BUFFER can be used
    - Cost trade off



P-Flash

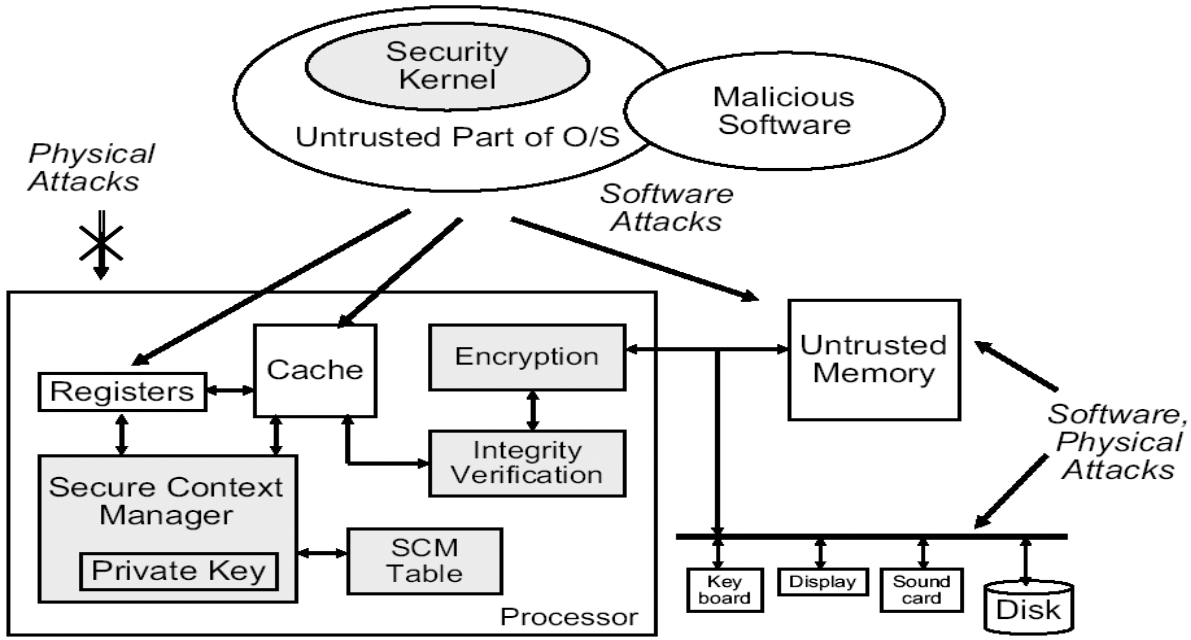
- Direct application programmability in NVM
  - Data must be safely stored during operations
  - House keeping functions may be required for clearing (e.g. FLASH)
  - NVM program/erase timing variable (needs optimised algorithms) versus real time
- Direct applications execution from NVM
  - Optimization may be needed for speed / power
  - May require many standard primitives to be stored in ROM
  - Instruction cache or buffer usage may be considered



MRAM

Secondary storage support

- **Challenge: Need for a secure Private Tamper Environment**

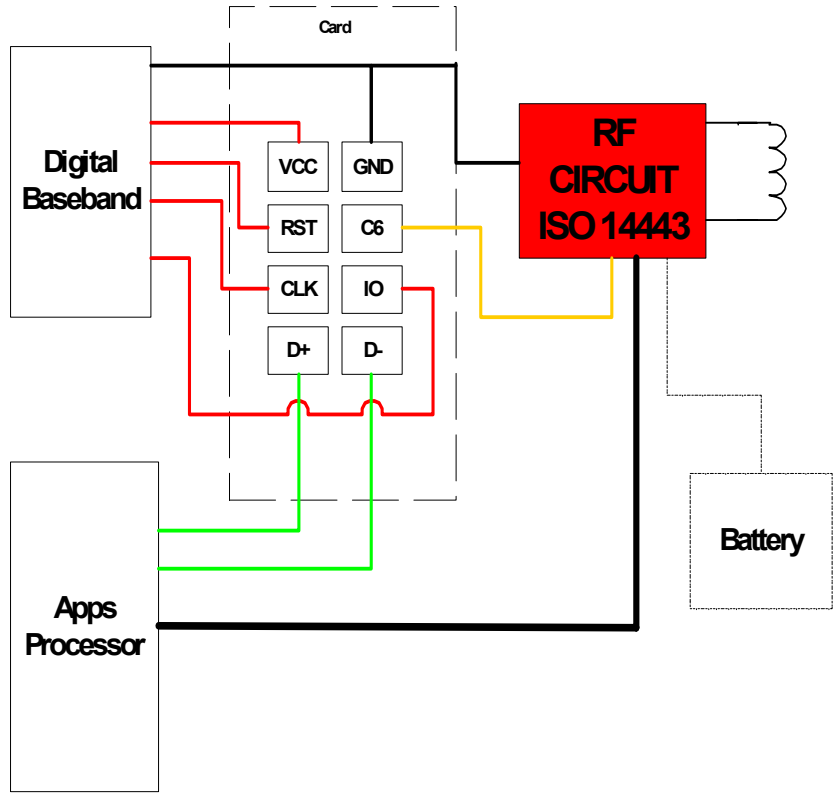


- **Supporting solutions still to validate**
  - MAC
  - Hash-tree (HW, SW or mixed)
  - Encryption

axalto

Communication bandwidth

- ISO protocols and current I/O structure limited
- New fast protocols requested
  - USB (1.0, 2.0, OTG/Interchip)
  - NFC
- Multiple interfaces needed
- Standardization essential
- New packages?
- Support of protocols for external memory wishful
  - SPI, other?
- Where can we go with further on-card integration of wireless protocols?
  - Wi-Fi?
  - UWB?



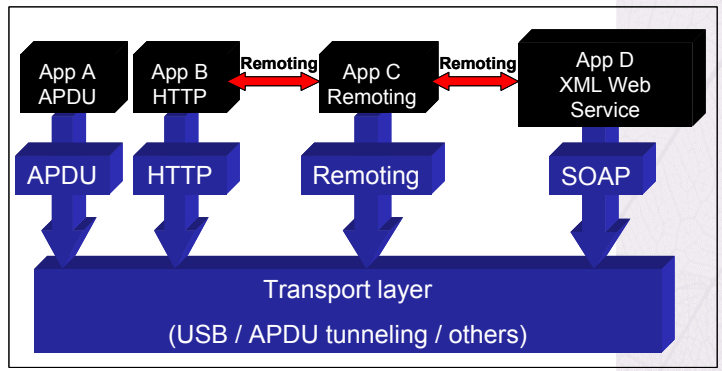
Example in the Mobile Com area

Openness of programming interfaces: High-level RTE, standard communication stacks

- **High-level RTE**
  - JavaCard RTE paved the way but…
  - Still very different from Java flavors need extensions for becoming closer from
  - ….other framework maybe legitimate in the smart card environment (.net, OSGI,…)
  - Security model should support secure , perfect firewall, “post-issuance” and hopefully remote secure personalization
- **Standard communication stacks**
  - Get rid of Current ISO 7816 “master-slave” and APDU centric limitations:
  - Use of an I/O stream allows a more natural coding
  - Read and Write access can be mixed without protocol clutter
  - Structures can be locally declared and complex mapping can be read/write in one shot
  - Support Multiple Message Wire Formats no more sustainable
  - On-card implementation of TCP/IP sockets mandatory (IVP4 and IPV6), including the security related aspects (e.g. IPSEC)
  - On-card Web/server capabilities mandatory for interactions with the environment

Embedded SW internal architecture and design flow

- **From « application support » to «service delivery» shift of paradigm**
  - SW Framework to deliver Card services over LAN/WAN to card supporting devices (terminals, servers)
  - SW Framework to administrate platform and services
  - SW Framework to support legacy applications
- **Integration architecture enabling easy/cheap deployment of smart-card services in an service oriented infrastructure**
  - On-card Web service support and remoting
  - Peer to Peer
- **Smart card SW design flow and methodology to be completely revisited**
  - Component oriented approach
  - Closer connection with HW design flow mandatory to optimize use of new HW resources



Richer OS capabilities

- **Major requirements to support**
  - Multi-thread, multi-tasking (Call handling in Mobile Pay-TV)
  - Real time operations (e.g. biometry, streaming)
  - Availability (e.g; fleet management)
  - Concurrency
  - Privacy enhanced technologies (anonymity, pseudonymity,…)
- **Underlying SW challenges**
  - Preemptive concurrency model in reduced resource environment
  - Dynamic resource control and (de)-allocation in dynamic multi-application support
  - Scheduling for real time application
  - Optimal resource allocation to applications
- **Micro-kernel potential in this respect is very high**
  - Feasibility of very small µ-kernel footprint demonstrated (e.g. L4)
  - Various scheduling strategies, w/ or w/o pre-emption easily testable
  - Abstraction of devices & enhancement of interrupt subsystem
  - MMU/MPU management is encapsulated in the µkernel
  - Fast IPC support
  - Flexibility
  - Correctness….up to the limit induced by HW parallelism and SW concurrency

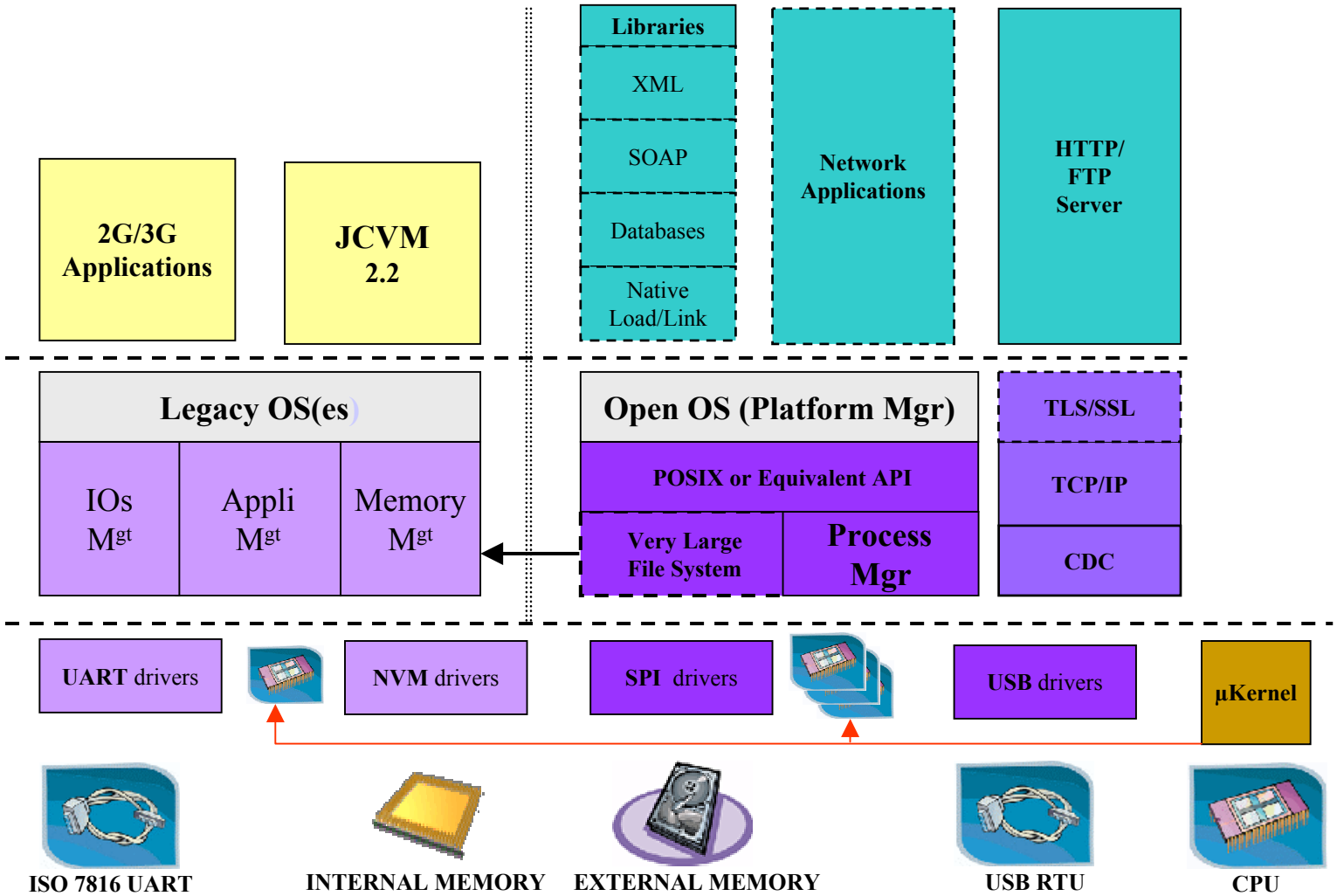
Adaptable security supported by intelligent cooperation with HW

- **Several problems to address**
  - Absolute level of security required
    - Strongly depends on the environment (E.G. Pay-TV vs EMV payment)
  - Security update at reasonable cost
    - E.G. crypto algorithm update, security policy updates
  - Access to critical HW resources and critical Assets
    - E.G. crypto keys, user credentials
  - Simultaneous management of Trusted and non Trusted applications
    - E.G. SIM application with Games, Multimedia applications
- **Some related underlying challenges**
  - Security intelligence sharing/cooperative models between Cards (HW/SW), Terminal and System
  - Level of HW “re-programmability”
  - Secure “over-the-air” or “over-the-wire” personalization update capabilities
  - Security assessment of the underlying application segregation mechanisms

Software validation and test, formal techniques, automation

- **In general, Smart card development SW is not fundamentally different from other SW development activities**
  - Error prone
  - Complex
  - Costly
- **Classical SW innovation and Challenges have nevertheless to be tuned to some specifics items**
  - HW/Trusted SW design flows
  - Trusted embedded SW factory
  - Powerful formalisms and tools for
    - Security modeling
    - Security validation
  - Security constrained test generation and fault modeling
  - Test coverage

Target Smart Card SW architecture



axalto

Design difficult challenges (HW& SW)

- **Productivity to avoid exponentially increasing design costs. Re-use.**
- **Power management.**
- **System-level integration of heterogeneous technologies**
- **Error tolerance relaxing for cost reduction ?**
- **Development of SOC test methodologies including Security constraints (from DFT to DF for Secure T)**

Productivity to avoid exponentially increasing design costs. Re-use

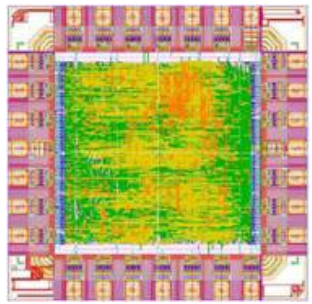
- **About 90 % design methodologies still serial**
  - Little interaction between SW/application developers and HW developers
  - Causes long design cycle
  - Consistency of security chain (HW/SW) hard to maintain all across the design flow, even in a security constrained methodology (CC, FIPS,..)
    - Need for for more generic co-design methodologies taking security in account
- **Technology costs**
  - Current NVM technology not mainstream (EEPROM)
  - Security overhead costs very high
  - Cost/performance ratio not optimal versus market expectations
  - NVM disruption mandatory
    - Flash? PCM?
- **Re-use**
  - HW/SW re-use concepts start to diffuse in the industry
  - Security re-use still a wish but not a reality. Certification costs prohibitive

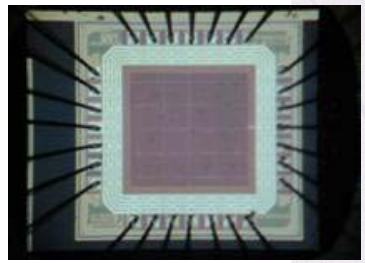
Power management

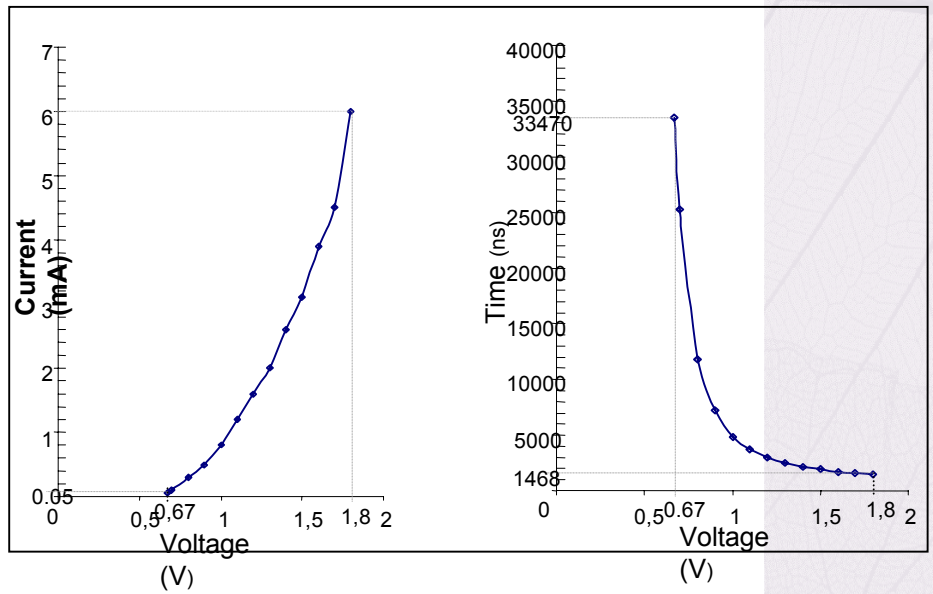
- **Two major types of requirements**
  - Functional requirements
    - Device working at spec in multi-application contexts
      - e.g. Mobile Pay-TV and GSM/UMTS
    - Low power
    - User acceptance
      - Autonomy
  - Security requirements
    - Resistance to various types of attacks based on electrical signatures
- **Need for a global system approach**
  - Need for dedicated HW and SW primitives
  - Need for a Power Management Framework enabling HW and SW handshake at functional and security levels

HW primitive example: Asynchronous Logic

- **Objective**
  - Reduce electrical signature
  - Improve DPA resistance
- **Approach**
  - Definition of a formal model of the asynchronous circuits (logical and electrical level)
  - Use the model to analyze circuits leakage and define the DPA countermeasures
  - Design flow specified for DPA resistant asynchronous circuits
  - General DPA resistance criterion specified to compare the prototypes resistance against DPA
- **Potential of ASL in terms of DPA resistance demonstrated**
  - e.g crypto operators by TIMA in the MEDEA+ Esp@ss project
- **Concept extension to a larger extent still to be assessed**







axalto

SW primitive example: Energy masking in RSA computations

- **Square-and-multiply always**
- **Algorithm**
  - **Input**: x,n
  - **Input**: d= $d_{m-1}2^{m-1}+\ldots d_1 2+d_0$
  - **Output:** $y_0 = x^d \bmod n$
  - $Y_0 := x$
  - **For** l=m-2 down to 0 **do**

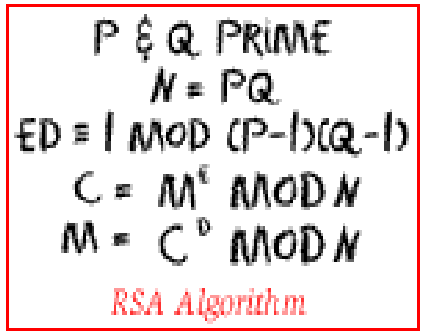
    $y_0 := y_0^2 \bmod n$

    $y_1 := y_0.x \bmod n$

    $y_0 := y_{d_i}$

    End for

    Return y_0
- **SPA safe…..but**
- **First values of y_0 depend only of few bits of d => DPA attack feasible**
  - 8 highest weight bits deducible => $d_{0\ th}$
  - Check d_{0th} by power consumption
  - Iterate to find all bytes of d by blocks of 8



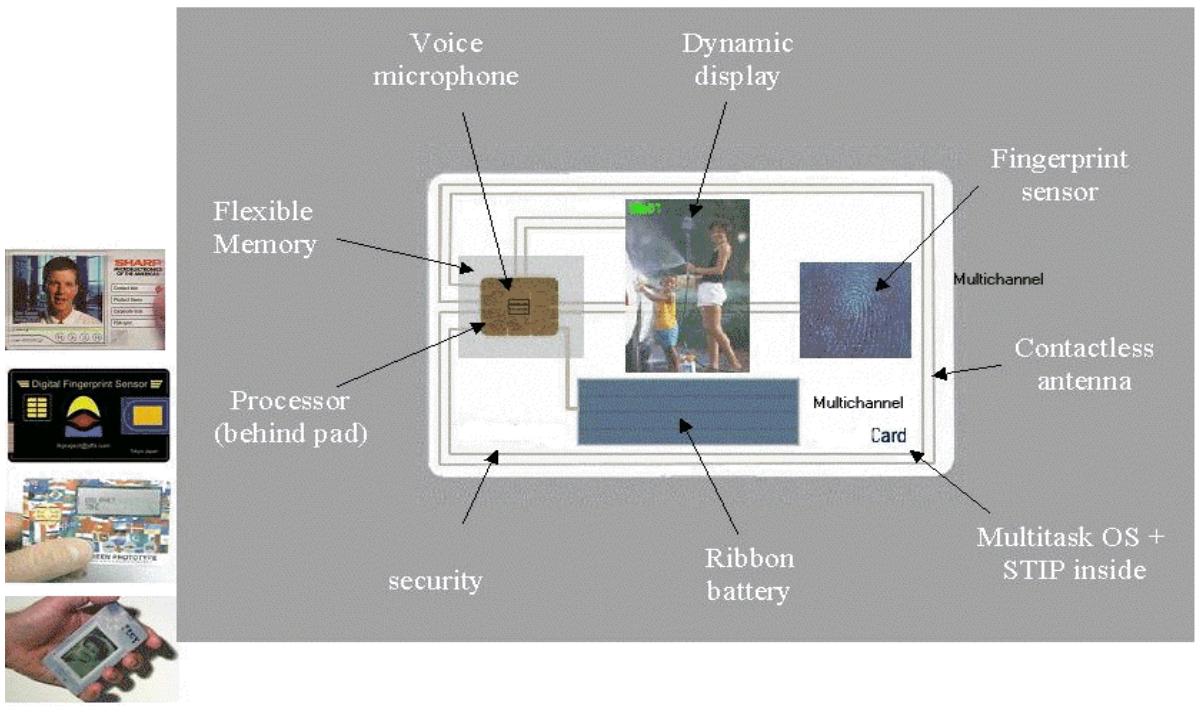
- **DPA-safe modification**
  - $f(v_1,\ldots,v_2) = v_1 \cdot v_2 \pmod n$
  - x= x1.x2 (randomly)
  - $y_1 = x_1^d \bmod n$
  - $y_2 = x_2^d \bmod n$
  - $Y_0 = y_1.y_2 \bmod n$

Power Management Framework example: device signatures

- A lot of critical effects to consider
  - Timing signature
  - Current, EMC signature
  - Other…..(radiation, SEU,…)
- Strongly influenced by the design choices
  - Architecture
  - Logical
  - Electrical
  - Physical
- Hard to control with classical design tools
  - Synthesis tools may force wrong RTL/Gate choice
  - Electrical implementation may create unbalanced paths
  - Physical effects hard to predict until final layout completed
- Severity may be strongly affected by the SW under execution
- No global supporting tool available
  - Addressing multi-level and back-annotation support
  - No guided support for absolute/random signature generation/control process
  - Signature evaluation coupling SW execution with parasitic electrical simulation
  - Proprietary solution and experience is the only solution today

System-level integration of heterogeneous technologies

- **Next generation od smart-cards applications will need « on-card » integration of « MEMS » components….**
- **….in addition to the remote control of some other**

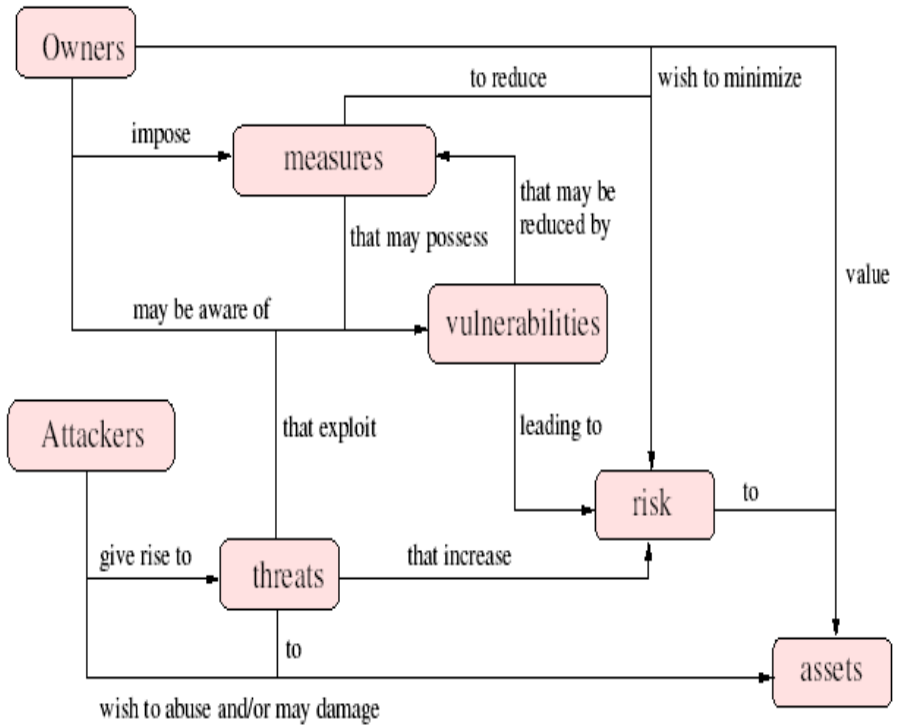


- **Besides technology and manufacturing (new form factors) problem secure remote control will be a key challenge**

axalto

Error tolerance relaxing for cost reduction

- **Common criteria has created a common language to speak about security, but**
  - Complex
  - Costly
  - Hardly coping with TTM constraints
- **More research needed to develop**
  - Incremental certification
  - Efficient maintenance schemes
  - Related support tools

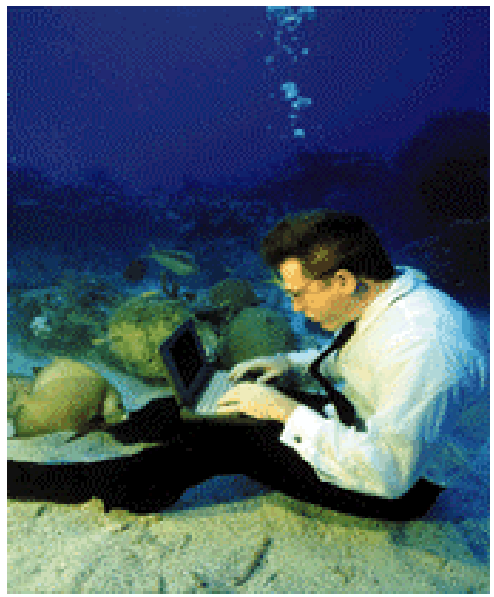


Development of SOC test methodologies including Security constraints (from DFT to DF for Secure T)

- **DFT strategy should not allow chip to work under out of band perturbation**
  - Clock or power supply glitches
  - Radiation, Light, laser…
  - Besides known “perturbation” effects large room for unknown
- **DFT strategies can be of two types**
  - Detect source of propagation and protect against perturbation effects
  - Detect faults effects of perturbation and stop propagation
- **No global solution exist but fault detecting effects very complex**
  - Requires redundant state encoding => illegal state detection
    - Redundancy level? Cost issue? Coverage issue?
    - Modelization and simulation are complex
    - Few tools exist and bring too little coverage
    - Verification and test of detection still a problem

A synthesis of previous material: Ambient Intelligence

- **Smart-card role in ambient intelligence scenarios (examples)**
  - Storage/management of user profile and preferences, privacy attributes
  - Seamless transfer to ambient network of user-neutral information
  - User –controlled transfer to ambient network of user-sensitive environment
  - Context sensitive user Identity/credential management
  - Authentication Framework management
  - Transaction management
  - Context-aware control of user's environment
- **Smart-card relevant design challenges**
  - On-card wireless interfaces: NFC, Zigbee-UWB,..
  - Embedded µ-sensors
  - Low power design, on-card batteries
  - Native of remote control of user CE appliances peripherals (Storage, MMI and Visualization,….)
  - IPV6 permanent addressing capabilities/IPV6 Mobile
  - Complete IPSEC security framework compatibility
  - Network-neutral, WEB-service enabled framework
  - On-card mobile agent SW technology



Conclusions

- Smart cards are just entering a new phase of their development
- Changes in the environment may create numerous business opportunities
- Design paradigms for smart card must be revised dramatically to potentially exploit those new opportunities
- There remain some difficult HW and SW technical challenges in front of us, due to some specific constraints of the constraints
- Tighter HW and SW co-design flows, tuned to smart card specifics can be one of accelerating success factors
- Cultural exchanges (in and out) with other domain may benefit to the whole industry as smart card part of the pie (1% semi conductor market) may not give the adequate Return on Investment
- Very active participation of University and Research is mandatory