IY5606 SmartCards

implementation as well as the design
Atkers with physical as well as remote
Tampering + tamper-resistance
Range atks

- Logical/theoretrical
- Timing/side-channel
- Fault
- Physical

Try to build trusted system on trustworthy + evaluated hw

Impt = similarities + diff btwn smart cards & secure execution platforms & environments

Smart Card Properties

- Tamper Resistant Security (not impossible, but hard)
- Information Storage
- Information Processing
- Portability
- Ease of Use
- Multi Value Added Apps
- Very useful (if have)
- Achieved by engineering + cryptographic techniques

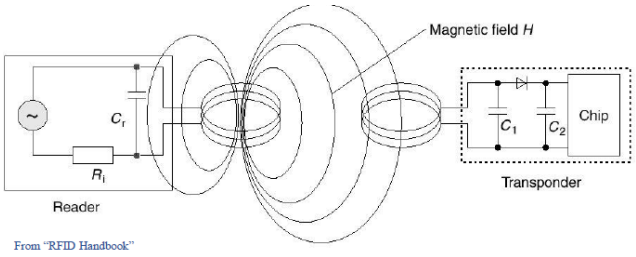
Mag Stripe Card

Cheap Widely Used suitable for Certain app (online, POS canteen) Easy to personalise No secrets	Passive No logic to protect data No Tamper resistance Relatively easy to copy One app

Smart Card

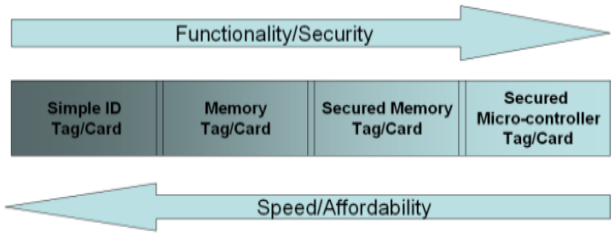
- Any pocket size card with embedded integrated circuirts
- Can store + process a lot of data
- Can be loaded with data, used for telephone calling, cash payment, and other application.

Smart Card with Contacts
Contactless Smart Cards



- A passive contact-less smart card/RFID is powered by electromagnetic induction – from a field produced by the reader

Smart Card/RFID Trade offs

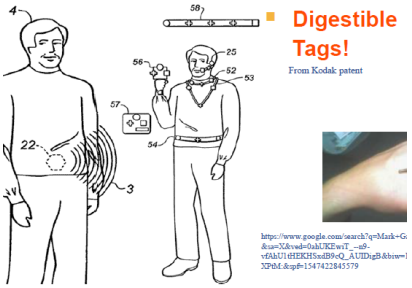


Tags Passive/Active

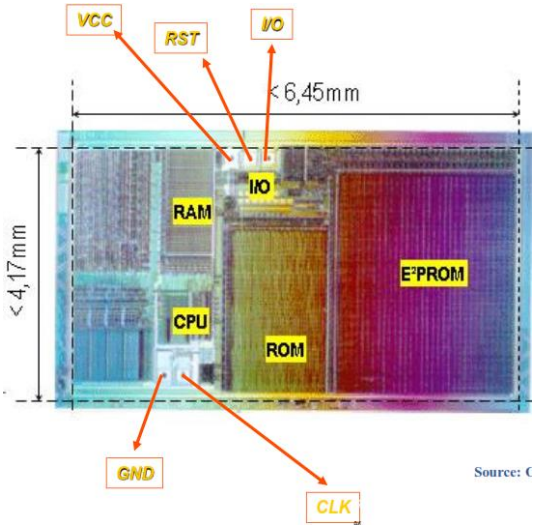
- Many diff formats
- Active = powered

NFC = contactless
Phone = behave as smart card OR token
Phone = behave as reader
Can be P2P mode

Digestible tags

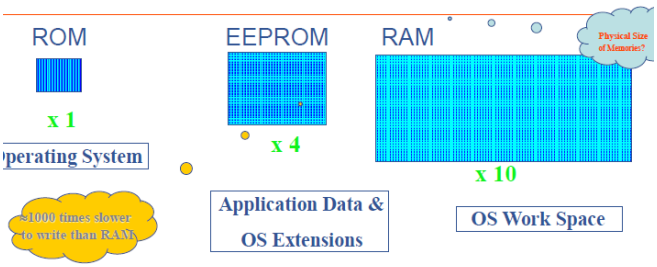


Smart Card



Gold contacts = security resistant device
Cost = important

Typical Smart Card Silicon Real Estate



	ROM	EEPROM	RAM
Low	~3Kb	~4b	~128b
High	~400Kb	~128Kb	~16Kb

RAM = needs to store variables
ROM = for OS
EPROM = Write once, read forever (initialization area)
EEPROM = Write, erase, read forever (apps, OS exts)
RAM = Write, erase, read TEMP data (working of OS and apps)
Flash = larger page size compared to EEPROM (code +data)

Smart Card memories

	RAM	EEPROM	FlashRAM	FéRAM
Persistency	No	Yes	Yes	Yes
Read acc.	0.1µs	0.15µs	0.15µs	0.15µs
Write	0.1µs	10µs	10µs	0.4µs
Erase	-	5ms	100ms	-
Granularity	-	4bytes	64bytes	-
Cycles	Unlimited	10^6	10^5	10^{10}

(1979 - 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

(1990-1996)
- **HW:** dedicated 8 bit µProcessors with powerful security mechanisms (sensors, crypto accelerators)
- **SW:** OS applications-Partial introduction of high-level languages for card programming
- **Security paradigm:** Industrial application of formal security certification

(1996-2003)
- **HW:** 16 bit/32 bit µProcessors
- **SW:** virtual machines Multos, JavaCard, .net
- **Security paradigm:** Common Criteria (CC)

(2003-2017)
- **HW:** 8-32 bit µProcessors
- **SW:** networked smart card components, IoT,
- **Security:** "Open" Industrial standards (GP, Multos), CC Evaluations

Smart Card Platform Characteristics

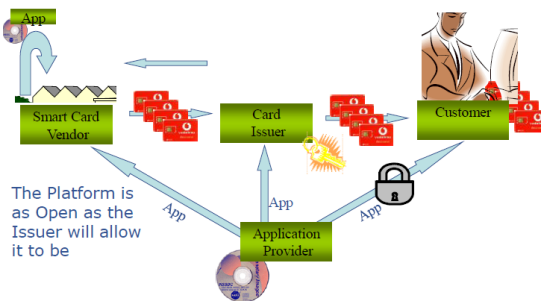
Features	Limitations ?
• CPU (>32bit) • RAM (>8kb) • ROM(>200kb) • EEROM(>64kb) • Crypto-processor option • Very Small • Low power • Low cost • Secure • Standardised • Operating Systems • Development Tools • Multiple Suppliers • **Consistent & Controllable**	• Helpless Alone – No internal power supply – Externally restrictions on power consumption – No user interface – No clock ▪ Limited (by PC comparison) ▪ memory ▪ CPU speeds ▪ Legacy cards may be inflexible ▪ New cards require deployment ▪ Controlled by the Issuer

Smart Card Features and Limitations

Features	Limitations
• CPU (<=32bit) • RAM (<=16kb) • ROM(<=400kb) • EEROM(<=128kb) • Crypto-processor (optional) • Low power (suitable for GSM phones and RFID) • Security (inherited in software and hardware) • Standardised • Active card (communicate, calculate, dynamically store data) • Relatively difficult to copy • Flexible / multiple applications for off-line purposes • On-card Processing Power – Advanced Security functions • More Storage Capability • Tamper resistance – Protect keys, passwords, numbers and other personal information • Portability • Programmable	• Lack of – internal power supply – clock • Certificate Verification – Revocation Lists • "Security" and composite evaluations • Data Storage – not any more…

Information Residing in CARD
- Apps
- Crypto keys
- Smart card Operating System
- Smart card OS policies
- Cardholder data
- Issuer Data
- Card mgmt. data
- Card manufacturing data

Smart Card Lifecycle
- Chip + Card Manufacturing
- Pre-Personalisation OR Initialisation
- Personalisation
- Utilisation
- Termination

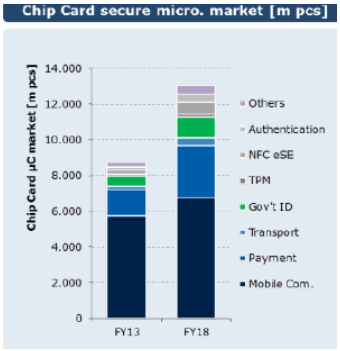


Smart Card Usage

- Banking
- Mobile comms
- Satellite tv chip card
- Transport
- Identity cards
- Physical access control
- IT access control

In 2013 over 9 Bn units shipped
By 2018 estimated to > 13 Bn units
Excludes RFID
Revenue growth in all sectors
Memories growth in Transport sector
Micros growth in Payment and embedded

urce Infineon 2014]



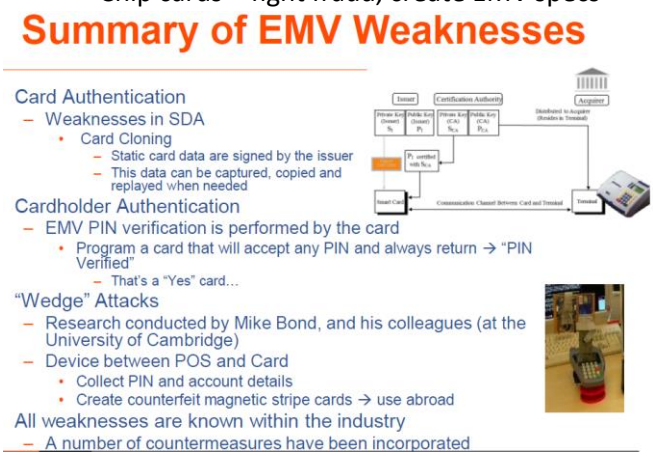
Mobile Comm

- Every GSM phone contains a smart card (SIM)
- Started as hardware security token for auth + encryption (prevent phone cloning)
- Useful for storing info (numbers, SMS)
- SIM/ME = richer, host programs, menus
- SIM today = multi-app Java Cards
  - Wide range of apps possible
  - Vendor independent dev routes

A3/8 = old standard
3G = newer standard (AES, MAC, Seq number, longer keys)
SIM/USIM Toolkit

Smart Cards in Banking

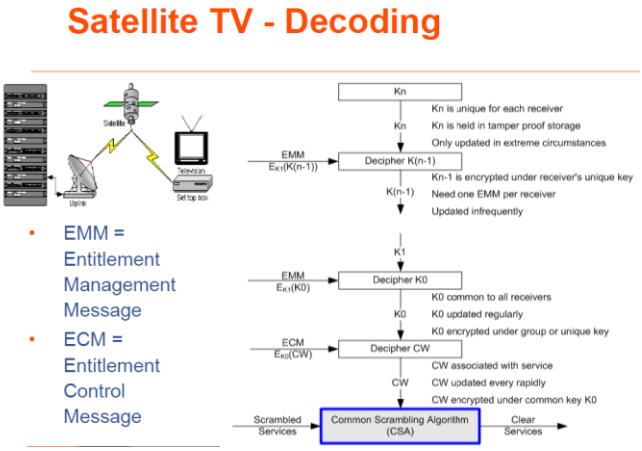
- Swipecard (easy to clone, skimming, counterfeiting)
- Chip cards = fight fraud, create EMV specs



Static Cards = replay same data
Yes Cards = always reply YES
Wedge atks = capture info in btwn transaction

Satellite TV Cards

- Card = allow decryption of satellite TV signals
- Protect video content
- Charges are significant, temptation to get free access



Security Problems

- Global secret
- Comms = one way
- Key distribution/mgmt. problems
- Security compromised
- Protection costs VS risk trade-offs

Smart Cards in Transport

- Simple + less delays
- Fraud control
- Innovative products
- Cost reduction

Big problem

- Mifare Classic = CRYPTO1 algo
- Security through obscurity
- Weak random number gen
- Reverse engineering

Countermeasures

- Kerchkoff's Principle (should not rely on crypto algo, but secret keys)

Smart Card in IDs

Interest in smartcard ID cards

- National security
- Efficient access to GOV + local services
- Fraud prevention

Card properties

- Printed picture + basic details
- Hard to forge/clone
- Electronic storage of personal details
- Stored biometrics

Problem

- Everyone has to have smart IDs, need old system as backup

- Registration
- Cost + ownership

Back to the 80's…

- When: Late 1980's
- Why: A home computer Amstrad CPC 6128
  - I wanted to protect my own computer programs
  - A Floppy Diskette copy protection mechanism introduced disks with "bad" sectors

- Game console manufacturers have attempted to fight game piracy since they came into existense
- Initially there were "catridges"
  - They were , initially, "difficult" to copy

- Xbox 360 [~2005] was designed to be "Piracy Proof with strong built in measures to defeat piracy"
  - "Legitimate" DVDs with… "Bad" sectors
    - Hang-on…. are we going back to the 80's?

- However; four months after it was released it was cracked by hackers.

General Applications

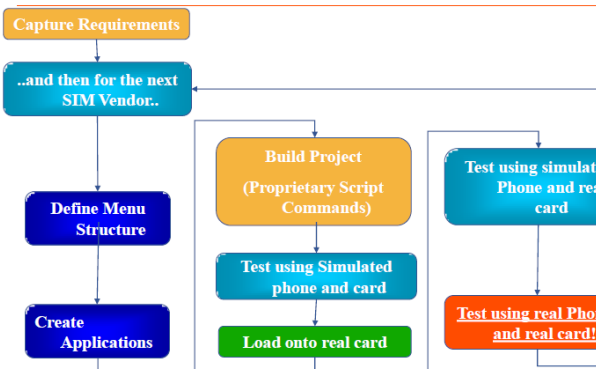
- Info services
- E-purses
- Location based services
- Auth/membership/loyalty
- Voting
- Ticketing
- DRM
- Games
- Personal & system info store/retrieve

Newer Technologies (USB interface = faster)

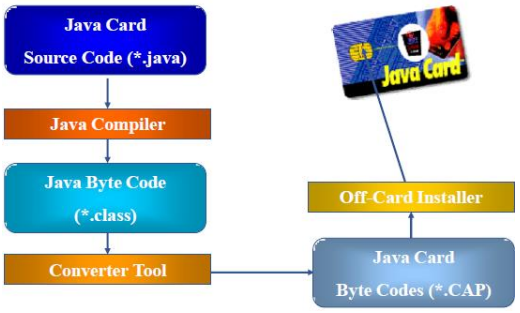
- Gem plus PC key
- Fingerprint card reader

Development

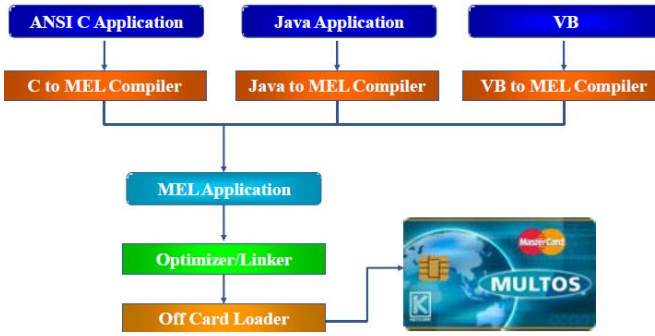
- SIM Toolkit Scripting
  Terminal compliance = inconsistent, lack of flexibility + post issue mgmt



- Wireless Browser
- Java



- Multos (any programming language)

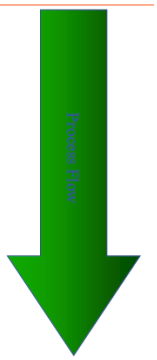


Very secure, but want to inject to earn money

- Proprietary – DIY

Typical Development Cycle

- Requirements Capture
- Card/terminal/Network split
- Vendor selection
- Mask/EEROM decisions
- Source Coding
- Compiling
- Simulation
- Test order
- Real card testing
- Real terminal testing
- System testing
- Volume order



New Service Roll-out

- The Smart Card is a very important system component but there are many components and considerations when rolling out a new Service
- Infrastructure Components e.g.
  - Communications networks and Billing Systems
  - Customer Care databases and Supply Chain issues
  - Readers/Terminals
- Business Issues e.g.
  - Marketing and Roll-out strategy
  - Legacy cards, terminals, systems and compatibility
  - Training
  - Costs v Functionality
- **Relatively speaking, the Smart Card is usually the easy bit!**

MAJOR Smart Card Attacks

- **Social Attacks**
- Hardware Attacks
  - Physical/Invasive
    - Obtaining Access to the Silicon (i.e. the microprocessor)
      - Chip Probing and Bus Reading
      - Reverse Engineering
      - Chip Rewriting Attacks
      - Drilling
  - Side Channel
    - Fault Induction Attacks
      - Glitch Attacks
      - Differential Fault Analysis (DFA)
    - Direct Memory Reading
    - Power Analysis
      - Simple Power Analysis (SPA)
      - Differential Power Analysis (DPA)
    - Radiation Electromagnetic Radiation
    - Timing Attacks
- Logical Attacks
  - Java card Platform Attacks

Attacks (Logical)

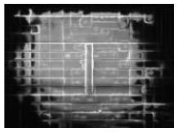
- Attacks against the design of algorithms/protocols
  - Use or eavesdrop the normal interfaces
  - Various tools available to help attacks
- RFID Sniffer

- Key cracker

Attacks (Physical)

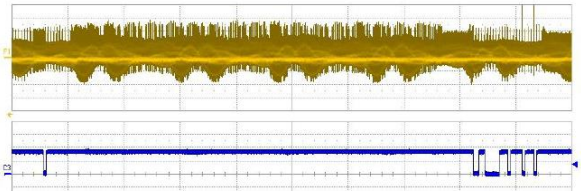
- Direct physical attack on chip/circuit to monitor or modify functionality and data
- Usually requires high skill level and specialist equipment
  - Probe station

  - FIB for track/circuit modification



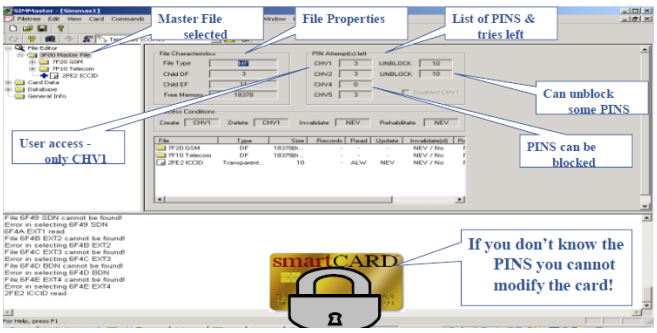
- Smart card microprocessor power consumption may leak information about processing data
  - Microprocessor logic gates consume power
  - Equipment…

Power Trace of a crypto algorithm



Security

- Mgmt. – PINS
- Modification of files on card = protected by pin
- Number of failed attempts exceeds limit = PIN blocked
- PIN unblocked if unblock code is known, unblock attempt not exceeded (unblock code exceeded = permanently blocked)
- GSM, encryption = only provided on radio
  - GSM 03.48 = additional standards for secure transfers between apps



ISO
- Cards with contacts > ISO7816-XX
- Contact-less cards > ISO14443

GSM/3GPP
- SIM/ME, STK > GSM 11.11, GSM 11.14
- App security > GSM 03.48

JAVA Card
- Java Card 2.2xx > java.sun.com

EMV
- Books 1-4 > www.emvco.com

Multos
- www.multos.com

ITSO
- Parts 1-6 > www.itso.org.uk

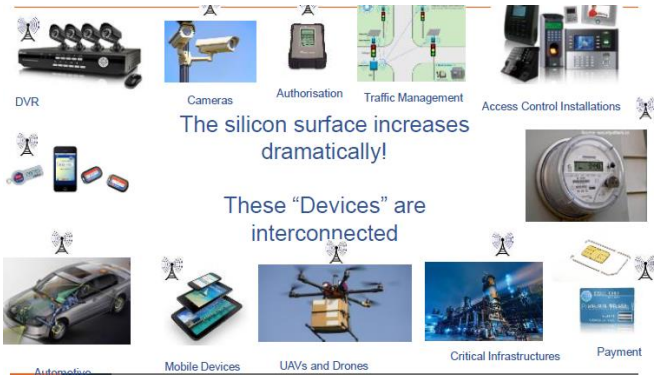
OTA

- Mobile networks have Over The Air (OTA) systems, that can update SIM contents remotely via SMS (03.48)
- File update is common and applet management is possible
  - Update the preferred roaming list to use favoured networks when roaming Change the phone book e.g. new help desk numbers
  - Change the displayed branding message
  - Add a new applet, correct a bug
- OTA systems can double-up as SIM application platforms – especially when the browser approach is used
  - Some systems allow customers to self-manage their SIMs
- OTA systems can be difficult to integrate i.e. they need
  - Connections to SMSC, Billing systems, Customer databases, customer care, IT management
  - They hold security sensitive management keys

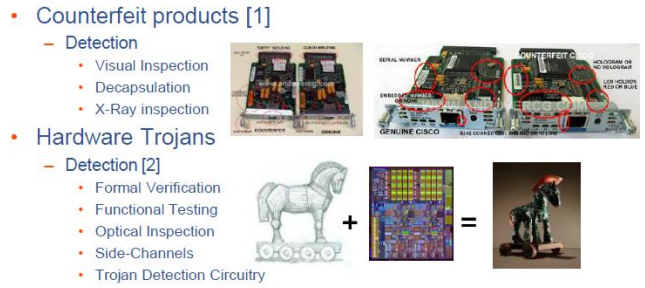
What is an Embedded/Cyber Physical System?

- Embedded systems have
  - Hardware platform that includes a processor, IO module, memory, co-processors (if needed) and other necessary modules.
  - Program, a group of instructions, that is designed to achieve a specific task.

- Small footprint computer systems controlling the actions of a bigger electronic assembly; Examples: microcontrollers in TVs, microwaves, communication devices, …



Coming from "Global suppliers" – competitive prices (cut corners)
"trusted" supply chains – security evaluations?
Recycled ?



Need for detection of insecure/counterfeit embedded devices

- Can be defeated by $2 device
- Cost of counterfeit = 7.5billion
- Problem = increases with proliferation of these systems

Target attributes

- Hardware platform
- Permanent data saved
- Runtime data (crypto, force unrandom)
- Control flow (yes/no)
- Individual instructions of program

Where to PLACE TRUST?

- Option A = Trust designer/programmers (software guaranteed to do what it meant to do)
- Option B = Trust the platform (TPM, Trusted Execution Env)
- Option C = what else?

Conclusion

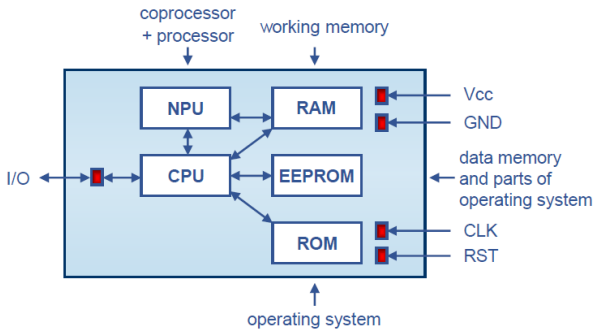
Smart card is a **cyber security asset** and a **vital enabler** & **PART OF a complex system security solution**

Smart Cards = staying (Bank cards, SIM, E-passport, ID, transports, Sat TV)
Unprotected smart cards = can be broken by adv analysis techniques
Smart card security = never ending battle
Perfect Security does not exist
Smart card = small + powerful platform to host apps that require security + consistency + control
Use of smart card = promoted + standardised + actively exploited in range of industries
New tech + form factors = appearing, evolve the smart card to smart token

1. How could you identify "smart" cards/RFIDs and readers?
2. What components would you expect to find in a smart card chip?
3. Fundamentally – why are smart cards used?
4. What are the strengths and weaknesses of smart cards/RFIDs?
5. What are the pros/cons of Issuer control?
6. What are the main commercial uses of smartcards and why?
7. Which development method would you chose to implement a card application - and why?
8. What types of security attacks have been used against smart cards & are they all relevant?
9. Why is card life cycle management important – but difficult?

Trusted Production Environment

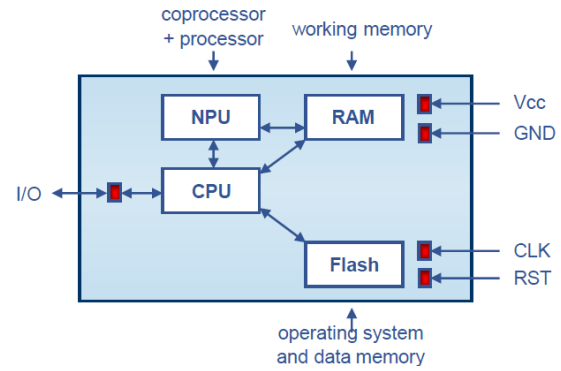
Contact Microcontroller with ROM



Put application in ROM = if app don't change in foreseeable future, cheaper

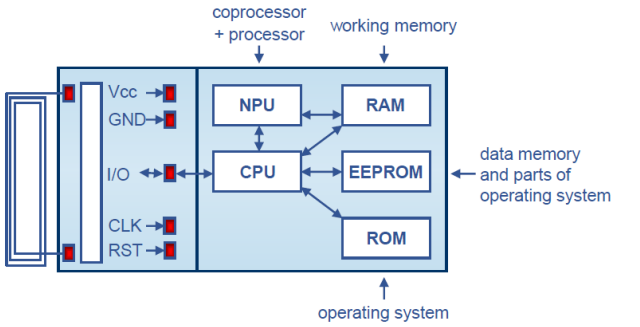
EEPROM = personalisation data

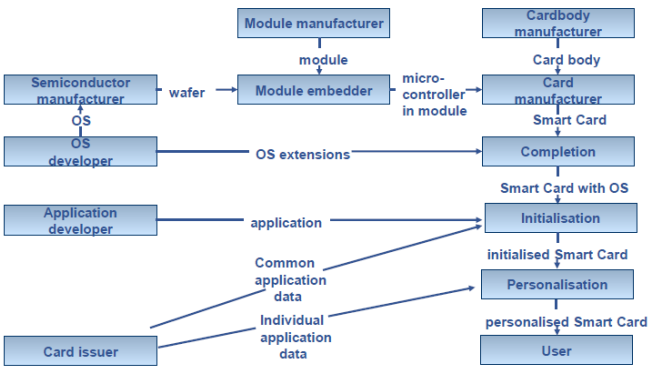
Contact Microcontroller with Flash



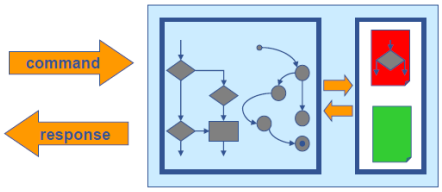
No restrictions, faster, can update

Contactless Microcontroller with ROM





File/Memory Based Applications



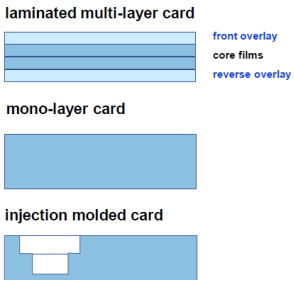
- smart card operation system
  - File/memory administration with access rights
  - construction kit
  - for applications with middle complexity
- pro/contra
  - easy application building
  - no programming
  - robust
  - less error risk

Code Based Apps:

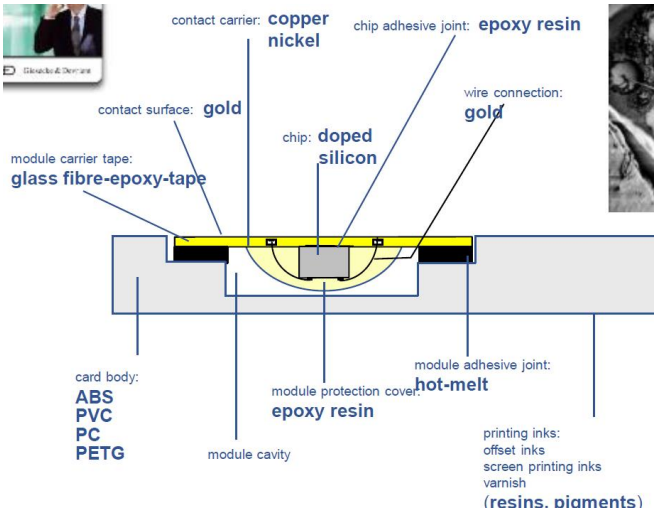
- smart card operating system
  - Interpreter for program code
  - Java Card, Basic Card
  - for applications with high complexity
- pro/contra
  - flexible, optimised for specific applications
  - complex application building
  - know-how is necessary

International Standards

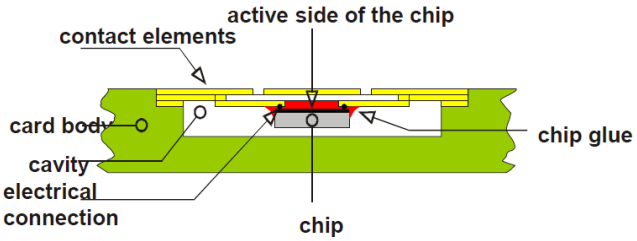
Important protocols
ISO14443 = Contact-less card
ISO7816 = Contact Card



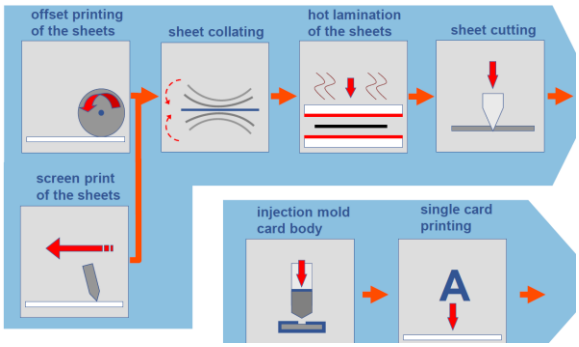
Card Type	Size	Usage
ID-1	54,0 x 85,6 mm	usual smart card
Plug-In Telecommunication	15,0 x 25,0 mm	for
Micro-SIM, Mini-UICC, 3FF Telecommunication	12,0 x 15,0 mm	for
Visa Mini	40,0 x 65,6 mm	for Payment

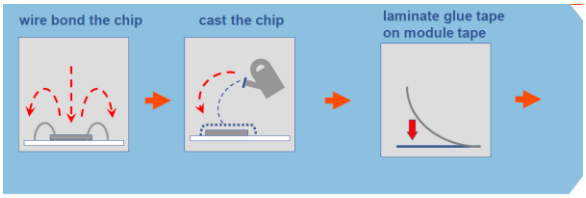


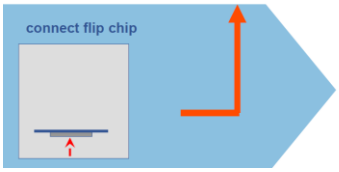
Flip Chip on Substrate

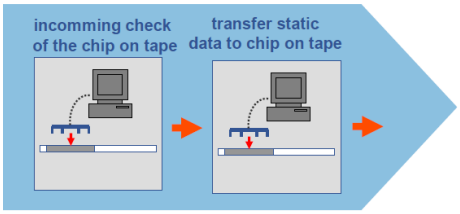


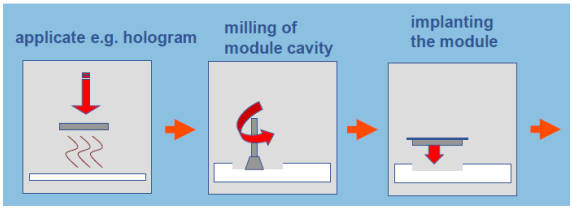
2D Barcode – printing stage

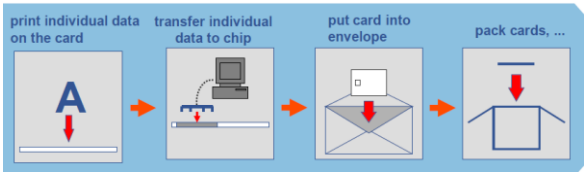




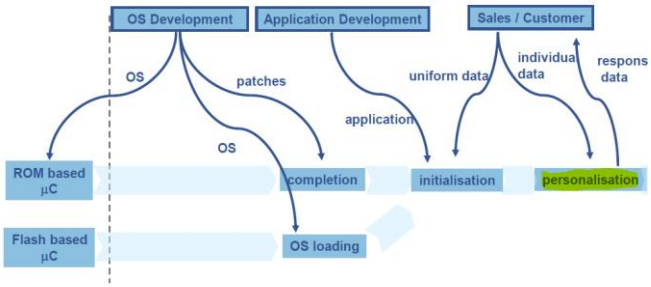








Production Flow of a Smart Card



µC Manufacturer — Card Supplier

Field of Usage	Technical Aspects	Governm. & Commercial Aspects	Order	Sample Card Production	Release Card Production	Volume Production

Card Body
- ☑ Body Size (ID1, **Plug-In**, Mini-UICC, Visa-Mini, Special format)
- ☑ Card body type (mono layer, laminated multi layer, **injection molded**)
- ☑ Interface Technology (**contact**, contactless)
- ☑ Standard Conformance (ISO, **ETSI**, ...)
- ☑ **Specification Conformance**

Microcontroller
- ☑ ROM or **Flash** based microcontroller
- ☑ Interface Technology (**contact**, contactless, dual interface)
- ☑ **Operating System** (ISO, **ETSI**, ...)
- ☑ **Java** / native
- ☐ Asymmetric Cryptography (RSA, ECC)
- ☑ **NVM size** (**Flash**, EEPROM)

Field of Usage	Technical Aspects	Governm. & Commercial Aspects	Order	Sample Card Production	Release Card Production	Volume Production

Common Card Elements
- ☑ Printing Technology (**offset print**, screen print)
- ☐ Hologram
- ☐ Security Features
- ☐ Signature Panel
- ☑ **Artwork** (who provides, special style, special colors)

Card individual Elements (delivery of card individual data necessary)
- ☐ Magnetic Stripe (incl. Data)
- ☐ Embossing
- ☐ Photo (incl. format, delivery)
- ☐ Laser Engraving
- ☑ **Thermo Transfer Print**
- ☐ Barcode
- ☐ Hologram

Field of Usage	Technical Aspects	Governm. & Commercial Aspects	Order	Sample Card Production	Release Card Production	Volume Production

Commercial Aspects
- ☑ **Price**
- ☑ **Liability**
- ☑ **Confidentiality**
- ☑ **Order Process**
- ☐ Forecast procedure
- ☐ Production on Demand

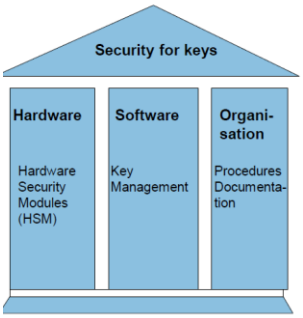
Governmental Aspects
- ☑ **Environment protection**
- ☐ Export regulation
- ☑ **Tax**
- ☐ Key escrow

Field of Usage	Technical Aspects	Governm. & Commercial Aspects	Order	Sample Card Production	Release Card Production	Volume Production

Card Data
- ☐ Delivered by purchaser (in which way and format)
- ☑ **Generated by card manufacturer**
- ☑ **Response data (delivery way and format)**

Note: Response data is very critical/important!!
- ☐ It contains all the IDs/Keys/PINs for all cards in the order
- ☐ Usually loaded into customers secure database (AuC in mobile networks)
- ☐ If attackers get access to the response files or database they may be able to create many clones!

Security - Key Management



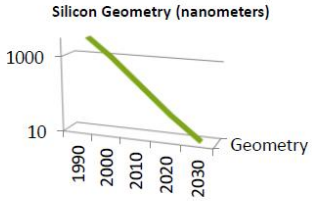
- ❑ There is a range of keys in use e.g.
  - ❑For the system sites
  - ❑For communication between sites
  - ❑For card loading/administration
  - ❑For card transactions
- ❑ They require support functionality e.g.
  - ❑Import keys
  - ❑Generate keys
  - ❑Export keys
  - ❑Transport keys
  - ❑Administration of keys

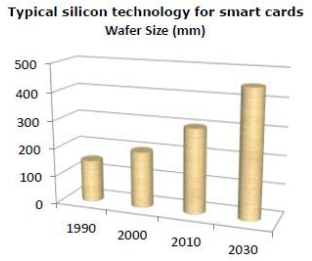
Physical security very impt. And Key mgmt servers

Cryptographic keys for different parts of the manufacturing process

Silicon Trends in Security

- Security ICs trails semiconductor logic design...
  - Needs embedded NVM
  - Secure layout
  - Secure design
- Drivers for design:
  - Function
  - Cost
  - Trust





Historical limitations

- **High Costs**
  - Not just the chip & card, but issuance and lifecycle management
- **Low Interoperability**
  - Non-standard design (France v RoW), IST17 started in 1993
  - Proprietary software, limited application development e.g. Java 1.0
  - Generic card edge specifications were rare e.g. ETSI GSM 11.11
- **Security**
  - Legacy IT systems compatibility with smartcards e.g. Windows NT
  - Card security difficult to quantify
  - Designs subject to hacking as many based on standard CPU

- **Memory IC's**
  - Contact and Contactless
    - Embedded authentication
    - Transport ticket, Access,
  - **Microcontroller IC's**
  - Contact and contactless (dual interface)
    - GSM SIM, eSIM, eID Cards
    - Transport, Payment, Enterprise

Contact cards:

- **Function: UID, Counter & Data Carriers**
- **Current options:**
  - Memory: 0.1K to 2 Kbit :typ 0.5 K bit
  - Product lifetime <5 year
  - Chip size: typical <2mm²
- **Security**
  - Traditionally low security moving to ECC
  - Challenge-response keys matching
  - On-line authentication,
- **Market: consumables**
  - Printer cartridges
  - Consumer accessories
  - IoT authentication.
- **Trends:** demand driven by anti-counterfeiting & IoT product security

Contactless cards:

- **Format :Multiple Options, all secure proximity cards**
  - ISO standard (ISO14443) typ 106kb/sec
  - Type A e.g. Mifare™, my-d™ move, Legic®, HID I-class®
  - Type B e.g. Caplypso®
  - Non-ISO, Sony–Felica™,
  - Product lifetime> 5 years
  - Chip size: typical <1mm² ~ 1Kbyte
  - Vicinity RFID – ISO15693 e.g HID I-class®
- **Security**
  - Cipher Stream= was proprietary schemes, or 3DES
  - **Market growth 10%/yr**
  - Transport – pre-paid & period pass, ID/access cards ,NFC Tags, loyalty cards

Contactless Memory:

- **More Memory** : up to 100K bytes (typical ~<1K)
- **More** Security : towards open standards & AES
  - NB> Mifare hack etc.
- **More Formats**: paper, fob, clip-on, watch, wristband, implant!
  - low cost designs: UID tags, small memory, in the long term printed antenna
  - embedded tags on PCB.
- **Growing Market** > 1 Bn units per year
  - 30% Access control, 30% Transport, growing use for online authentication
  - Growth markets: BRICS nations

Microcontrollers

- **Function:** Secure Data & Authentication
  - Options : Contactless I/O or Dual
  - Data retention up to 10 Years or more
- **Security**
  - Challenge-response (RSA/ECC)
  - Symmetric and asymmetric key exchange
  - Encrypted sessions – 3DES – AES 128/256 On-line or Offline authentication(with SAM)
- **Technology :**
  - 8/16/32Bit CPU
  - Memory ROM/EEPROM/Flash/RAM
  - Typical chip size <4 mm2 (e.g. SIM, EMV)
  - Typical chip family lifetime 5-7 years
  - Typical Process lifetime 3 years – geometry shrinks to <50nM

Microcontrollers -Contact

- **Technology**
  - CPU 8Bit, 16Bit or 32Bit (+Crypto co-processors)
  - Memory (ROM /EEPROM/Flash)
  - 2K to 1M BYTES (optional GB stacked chips)
  - 4 major silicon suppliers
  - Typ product option lifetime 2-3 years (family > 10ys)
- **Security** – depends on application
  - H/W up to CC EAL 6+ high
  - S/W up to CC EAL 7
- **Market Trends**, CAGR >~5%/yr
  - Telecom – USIM (ETSI) typ 256 KB (data size) – M2M
  - Gov't ID/Health Cards (PIV, CEN) typ 32-80KB
  - Pay TV typ 64KB
  - Banking cards (EMV -> DDA) typ 8KB
  - Embedded – IoT- trust anchor – e.g. TPM
  - Increasing memory, faster performance,
  - multi-applications, SCWS (JC 3.0)

Microcontrollers - Contactless

- **Technology**
  - CPU: 8Bit, 16Bit and 32 bit (+crypto)
    - Options with Contact & Contactless I/O or Dual
  - I/0 :ISO14443 (A/B) up to 848Kb/sec
  - 4+ major silicon vendors,> 10 Bn cards issued
    - Product Lifetime 3+ years
  - Memory (ROM & EEPROM) 4K to 512K BYTES
  - Data retention up to 10 Years
- **Market Trends :> 3B** units per year- growth 6 %/yr
  - Electronic Ticketing (ITSO, CEN-IOPTA, CIPURSE)
  - ID Cards & ePassport (ICAO, CEN)
  - Banking cards (EMVco, China Unipay)
  - NFC SE (with or w/o modem chip)
  - Larger Flash memory, faster performance- VHBR,
  - Multi-applications, biometrics/images in ePassports, EMV cards , open standards

Duel Benefits:

- Multi-applications on one card
  - Increases service options e.g. payment
  - Consumer convenience increases
  - Functions can kept separate and secure
  - Improves customer relationship
- Hybrid cards can offer issuer independence
  - e.g. a public transport e-ticket with loyalty card or EMV
  - An existing access system can have PKI functions added
  - Preserves integrity with convenience

Flash:

Technology Features

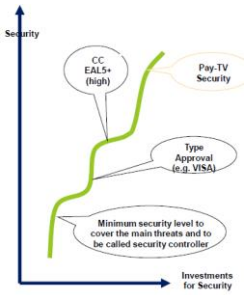
- **Flexibility**
  - Single memory technology with sophisticated security mechanisms
  - Economies of scale, one size fits all, offers supply chain choices
  - Accepted for Payment and Government ID applications
- **Performance**
  - Outstanding contactless performance 40% faster than conventional products
  - Flash Memory : fast page write/erase 32KB/sec

Card Manufacturer Benefits

- **Delivery times** chips delivered faster.
- **Development:** Rapid error-checking and correction during software development reduces effort and costs
- **Logistics:** Reduces planning expense, storage costs and market risks.

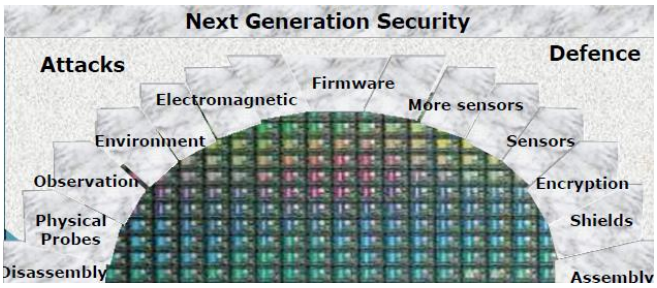
Drivers for Security:

- IP Theft
- Fraud, Service Denial
- Reputation Dmg
- Illegal use by criminals

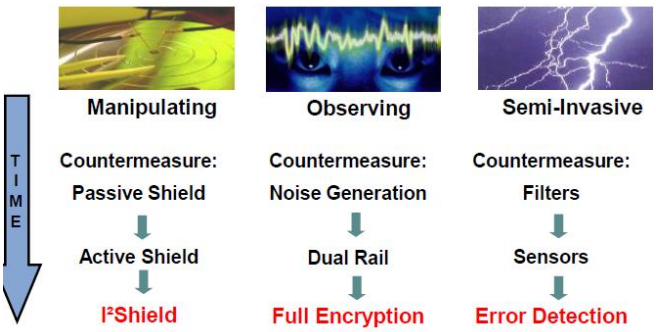


*Enables new services

* Security protects whole value chain, HW, SW, IP provider, network operator, consumer



Attacks

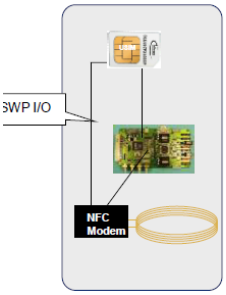


NFC Security

Applications – secure to insecure(location, info from posters)

Use Case NFC: UICC
- SWP Connection from NFC modem to UICC

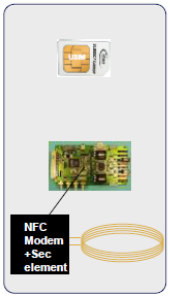
Authentication for NFC service from UICC.
Operator is a core element in business model.



- **Advantages:**
  - + High security (EAL 5+)
  - + Phone independent power management and usage
  - + NFC applications are OS-phone independent
  - + Ease of use for MNO due to interfacing with UICC (limited requirements and interfacing to handset)
- **Disadvantages:**
  - Longer implementation due to new standards
  - Longer time to market requirements
  - TSM services
  - ETSI Standards are still changing
  - enhanced s/w stack on UICC.

Use Case NFC: Secure Element
- No direct NFC connection to UICC

SIMless authentication.
Authentication for NFC service from security element of baseband processor. Operator MAY participate in business model.



- Advantages:
  - + Easy implementation, no I/F requirements for UICC.
  - + No standardization work for SIM to be done.
  - + Faster time to market
- Disadvantages
  - Security of application processor is proprietary
  - API security of phone is proprietary
  - 3rd party security management (TSM) restricted
  - NFC applications OS- Mobile Phone dependent
- Requirements
  - Compatible API
  - broad availability of specific NFC SE applications

NFC Tag Types

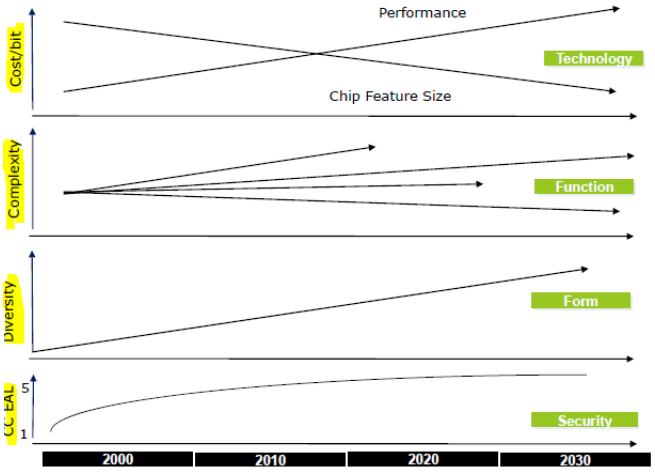
NFC Type 2
- Plain memory typ 0.1KB memory^4 Kbytes
- high volume "pairing" market, audio, toys, wearables

NFC Type 4
- Microcontroller (smart card)
  - Non-standard applications (authentication, payment)
  - Optional 2nd interface (I2C/SPI/USB)

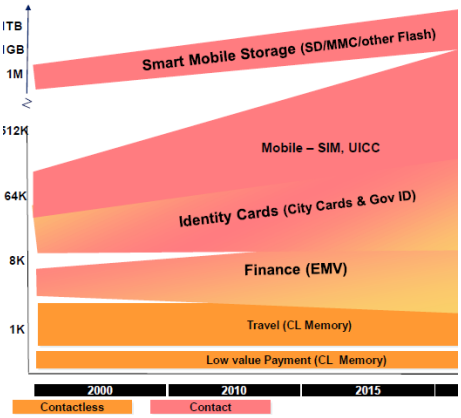
NFC Type 5? Due from 2019
- ISO15693 "long range" with crypto
- item tags e.g. Library books

Performance = Speed - Processing

1996	2020
CPU = 8bit with 16KByte EEPROM	• CPU = 32bit with 1MByte memory
700 nM geometry	40 nM geometry
Typical 20K ROM	Typical 500K FLASH NVM
4 K EEPROM, 0.25K RAM	Up to 16K RAM
Clock freq = 5MHz	• Clock freq = 100 MHz max
Typ 3.56MHz ext only	typ 50 MHz
Max I/O = 76Kb/s	• Max I/O = 8 Mb/s USB
Typ 19.2Kb/s	Typ 106 Kb/s CB or 848Kb/s CL
Crypto – rare	• Crypto -common
512bit RSA (with CRT) = 66mS	2048 bit RSA (with CRT) < 50mS
No formal security evaluation	• Certified > CC EAL 6+ high (for chip)
	Typ Common Criteria EAL4 for card

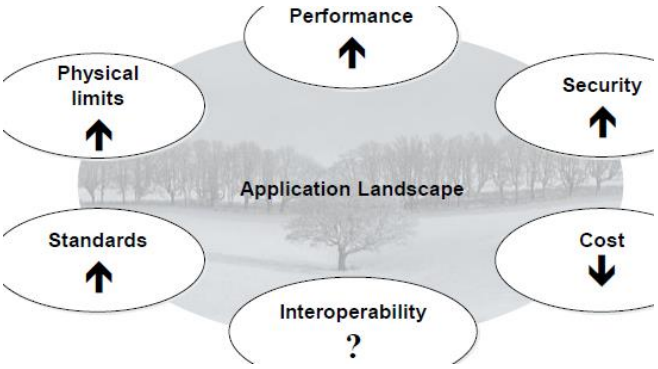


Smart Chip Market Trend – Memory

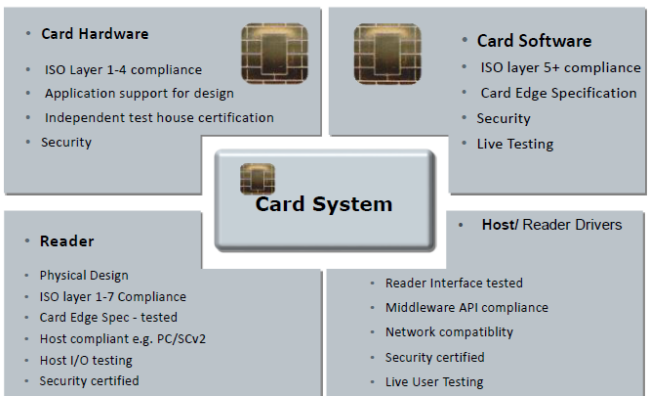


TRENDS for Smart Chip Technology

- **Design Complexity is increasing (Moore's Law)**
- **Physical Formats**
  - Non-card formats e.g. for contactless systems
    - passports, watches, wristbands, keyfobs, rings
- **Embedded Security (IoT)**
  - Automotive
    - Secure Engine Management Units, eCall, EV charging
  - Entertainment
    - Authentication of accessories, home hubs, wearables
  - IoT (Machine to Machine)
    - e.g. Utility/Industrial telemetry, healthcare, "smart cities"
  - Personal Information Technology
    - e.g. notebook TPM chips, secure USB keys, FIDO tokens
- **Integrated**
  - Separate core in SoC with secure execution environment
    - Secured hardware?



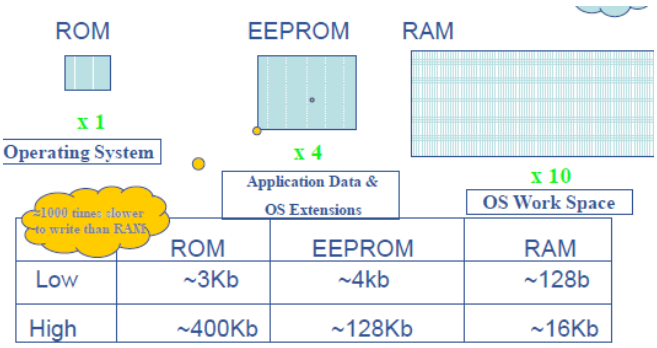
Interoperability



Summary

- Smart card ICs are evolving
  - Card have had a long and varied existence, which will continue
    - Over 50 Bn smart cards have been produced (95% have expired)
    - In early 00's > 50 % of cards issued were still memory cards
    - Today Microcontrollers cards make up >80% market share
- Existing Trends continue
  - **Cybercrime**, ID theft and counterfeiting are increasing
    - who controls the chip controls the revenue
  - **Physical limits** are being overcome with new technology
  - **Electronic limits** are being extended
    - designs with Flash memory offers new solutions
    - power consumption determines performance
  - **Legacy infrastructure limit** communication and usage
    - contactless already 40X faster , USB over 500 times
  - **Contact & Contactless cards** support different needs
    - The future will be driven by contactless services and smart phones
  - **Technology** influence
    - Higher memory, higher processing power, and new applications

Multi-Application Smart Card OS + Platform



	ROM	EEPROM	RAM
Low	~3Kb	~4kb	~128b
High	~400Kb	~128Kb	~16Kb

if store EEPROM, Flash, will be slower, affecting overall performance

ISO7816

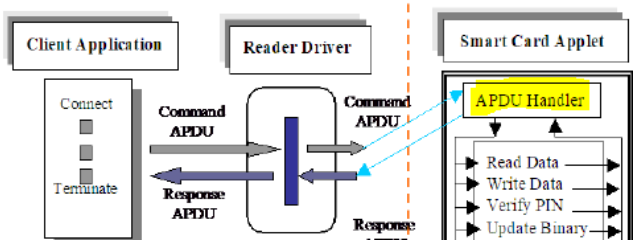
-1 physical char

-2 dimensions and location of contact

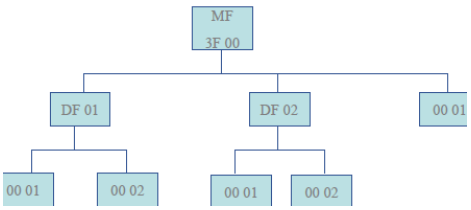
-3 electronic + transmission protocols

-4 file struct, secure messaging, application protocol data units (APDU)

- + Applications could be written in C, C++, VB, Java, etc.
- + Easy to program



File system



Command and Response APDUs

- **Select Command (INS) APDU Format :**

CLA	INS	P1	P2	Lc	Data	Le
A0	A4	00	00	02	DF 01	02

A4 = Select Command

P1 + P2 = 00 00 = Selection by file ID.

Data = Selection by : file ID = *DF 01*, Parent (no bytes).

- **Response APDU Format :**

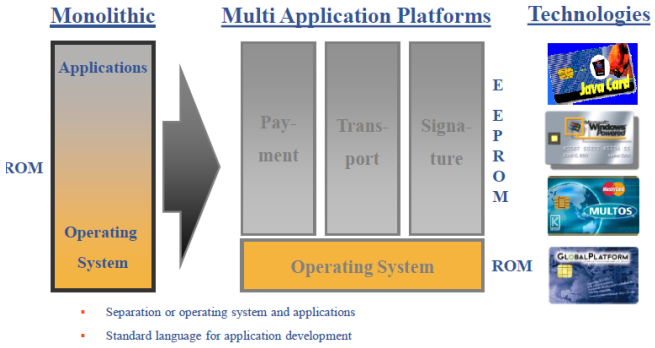
Data	SW1	SW2
---	2 bytes	2 bytes

90 00 = Successful Ending of the command, 6A 82 = File not found

Must be correct to work

Before Multi-app cards

- Various smart card OS (SCOS)
- They claim multi app support
  - Agreed in advanced, installed in ROM
  - Specific smart card microprocessors, embedded in SCOS



Memory = less of an issue

Java Card Forum Objectives

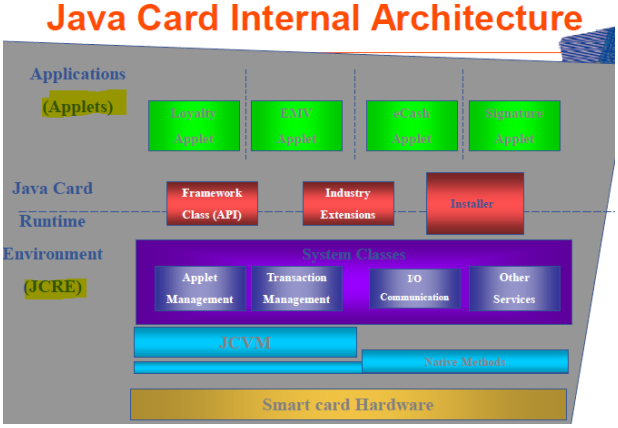
- Promote Java Card Language + assc. Standards
- Prep technical doc
- Exchange technical info
- Propose improvement to API specs
- Provide technical input to chip suppliers
- Dialogue among software suppliers regarding Java Card API

JAVA – why?

- **Standard lang** (extensive doc, OOP, Java lang protection, off the shelf devkits)
- **Security**, well defined sec arch (sandbox, no pointers)
- **Portability + interoperability** ("write once, run everywhere")

Java Card Language
– *Subset* of Java Language.
Java Card Virtual Machine
– *Subset* of JVM.
Java Card API
– Little resemblance to the traditional Java API.

Java Card Internal Architecture



JAVA CARD API v1.0

Presented the `java.iso7816` package
12KB ROM, 4KB EEPROM, 512B RAM
Supported:
- Boolean,
- byte and short types,
- object oriented scope and binding rules,
- flow control statements,
- uni-dimensional arrays, and
- operators (e.g. "=") & modifiers (e.g. "/", "+")

JAVA CARD API v2.0

16KB ROM, 8KB EEPROM, 256B RAM
C Ver 1 =12KB ROM, 4KB EEPROM, 512B RAM
packages:
- `javacard.framework` → *APDU, System and Util,*
- `javacardx.framework` → *ISO 7816-4 compatible file system*
- `javacardx.crypto` → *Export-controlled crypto functionality*
- `javacardx.cryptoEnc` → *Basic crypto functionality*

Portability and interoperability was an issue
Class file conversion & download were vendor specific

RAM decreased = OS becomes more efficient

Java API v2.1

- Applet firewall and object-sharing model redesigned
  - The applet firewall was more restrictive and there is a well-defined *Shareable Interface* for object sharing.
- Applet install method was interoperable
  - Install method used a byte array as an input parameter instead of the APDU object
- New Exception Class hierarchy
  - *Throwable*, *Exception*, and *RuntimeException* classes in the *java.lang* package to be strict subsets of the standard Java *java.lang* versions.
- Multiple instances of single applet class possible
  - This allowed a single applet class to be registered as multiple applet instances.
- AID class has a more general purpose
  - Compare AIDs, etc.

- javacardx.framework (file system) extension package deleted
  - Sample applets were able to use internal objects to represent ISO 7816-4 type files and records efficiently.
- Cryptography extension packages restructured
  - `javacardx.security`
    - Signature, Digest, Suitable for export restrictions (i.e. no strong encryption).
  - `javacardx.crypto`
    - Cipher Class, Subject to export restrictions.

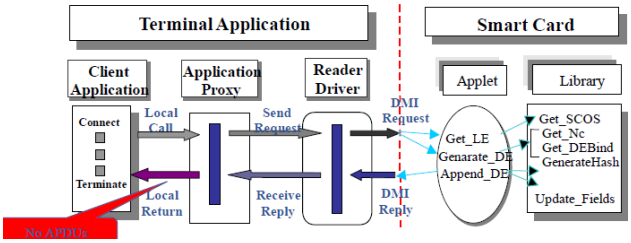
Java Card 2.1 Language Subset

Supported	Not Supported
• Packages • Interfaces • Dynamic object creation • Virtual Methods • Exceptions • Boolean, byte, short • Objects • Single dimensional arrays • Flow control statements	• "Integer" (?), Float, double, long, char, string, Multi dimensional Arrays • Multiple "Threads" • Dynamic class loading • Security Manager • Cloning • "Garbage collection" (?)

2.2.1 improved again (additional platform, crypto, object deletion, logical channels support, more funcs)

Java Card RMI in Action

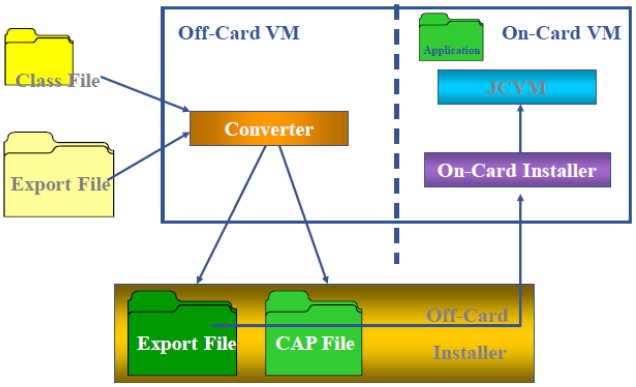
- Remote Method Invocation (RMI) by Gemplus, based on Direct Method Invocation (DMI).
- GemXpresso Java card specific since 1998.



- Wants more adoption (easier)
- Hide complex, more transparent

Java Card Security

- Java Card Security
  - Security Policy is enforced by the JCVM.
  - Objects are owned by the applet that created them.
  - Objects can be shared with other applets by using specially designed sharing methods.
- Java Cards have obtained EAL4+,EAL5+ Common Criteria Security Evaluation Level.
- Java card protection profile…
  - "…a modular set of security requirements designed specifically for the characteristics of the Java Card platform."



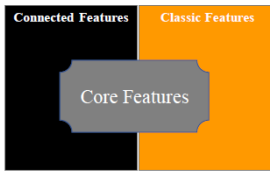
Summary of Benefits for Java Card Ver. 2.x.x

- **Interoperable**
  - Write-once-run-anywhere (…is it true?).
- **Secure**
  - Through the inherent security of the Java programming language
- **Multi-Application scope**
- **Dynamic**
  - New applications can be installed/deleted/securely after a card has been issued.
- **Open**
  - Off-the-shelf Java development tools,
  - Object oriented programming.
- **Compatible with Existing Standards**
  - ISO7816, EMV, ETSI, GlobalPlatform.

JAVA Card 3

Why = lots of advances, new apps (parallel+web), smart card = networked, multiple interface (nfc, contactless, usb), smart card programming = closer to normal java

- Released 31 March 2008
- Two separate editions
- Classic
  - Resource constraint devices
  - Based on the evolution of the Java Card 2.2.2
- Connected
  - High-end smart cards
  - Network oriented features
    - Web applications
  - Enhanced execution environment (JCVM)



Both editions:
Share key (Core) security features

- Improvements over the JC API 2.2.2
- Support for
  - Contact and Contactless APDUs
- Rely on the traditionally divided JCVM
  - Off-card and on-card processing of smart card applications
- Further cryptographic functionality
  - 4096-bit RSA, NSA Suite B = ECC, AES, etc.

- Improved connectivity
  - Act as a secure network node
    - Web server with Java Servlet APIs
- Parallel communications
  - ISO 7816, contact-less, USB, NFC
- Multi-Threading
- New JCVM
  - More efficient and smaller in size (remains to be seen when implemented)
  - Load Java class files without pre-processing
    - On-card byte code verification
  - Automatic garbage collection
- Deployment for both traditional and "Web" style applications

- JCVM Ver. 3
  - Connected Limited Device Configuration Specification Ver. 1.1 or the Java Micro Edition
- Intended for 32-bit CPUs
- Improvements over transaction atomicity and firewall

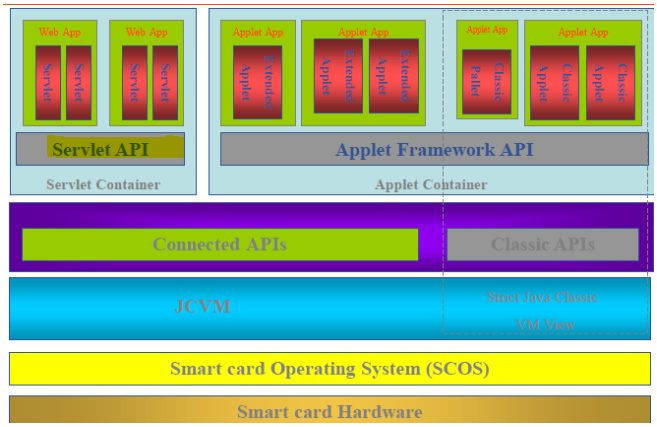
Traditional SC	High-End SC
8/16-bit CPU ~2kb RAM 48-64kb or ROM 8-32 kb EEPROM Serial I/O 9.6-30kb/sec Half Duplex	32-bit CPU 24-kb RAM >256-kb ROM >128kb EEPROM High Speed Interfaces E.g. USB More I/O channels → e.g. ISO 14443 Full Duplex

Java Card 3: The Four Core Categories of Supported Functionality

JC 2.1 Unsupported Functionality	JC Ver 3 Supported Functionality
"Integer" (?), Float, double, long, char, string, Multi dimensional Arrays Multiple "Threads" Dynamic class loading Security Manager Cloning "Garbage collection" (?)	Multi Threading On card bytecode verification Automatic garbage collection – More complexity due to networked nature More Utility Classes – Char, Long, String – Multi-dimensional Arrays – Boolean, Integer and more – String Manipulation classes – Networking Classes → Connector, connection – Date and time utility classes → calendar, Date, Timezone

Java Card 3 Web App

- Can send HTTP/HTTPS, or be a WEB Server



Classical computer security attacks apply to smart cards
- Trojan horses
- Buffers overflows
- Bugs exploits
- Java security problems

What is GlobalPlatform?

An "independent" organisation that:

- Maintains the Global Platform specifications
- Facilitate supporting systems and software
- Develop conventions for cross-industry application loading
- Complement other smart card standards
- Generate sufficient revenues to cover expenses
- Promote usage and adoption

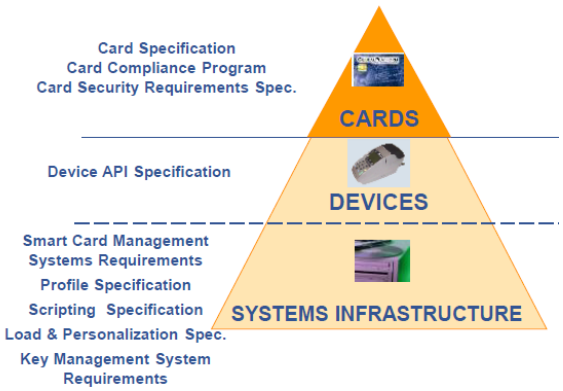
GlobalPlatform(GP)
Ver 2.0.1

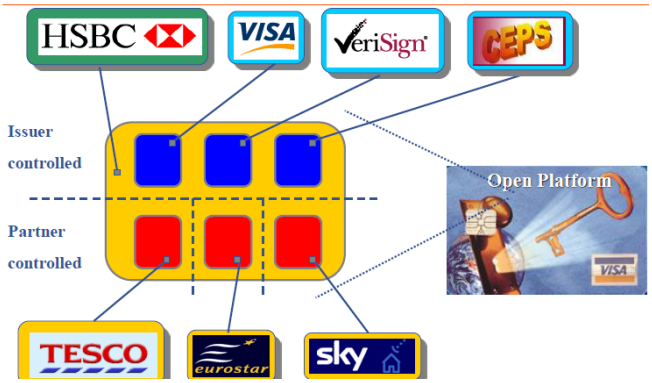
A lot of members,

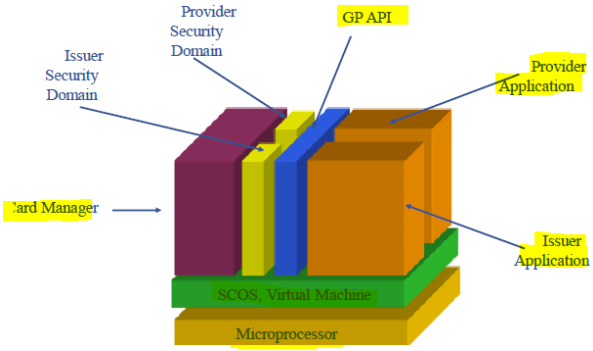
Standards = set of open/global/interoperable FW

- Facilitate partnerships with other industries
- Maximize protection of investment (new value proposition)
- Give choice of suppliers

Other components, (SCManufacturingSys, Terminals)



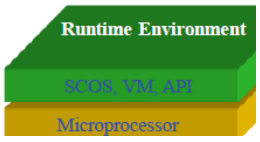




Smart card Micro-processor

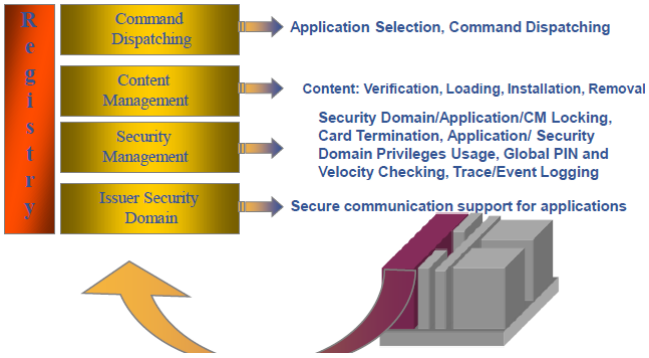
The Runtime Environment (RTE)

- Smart Card Operating System (SCOS)
- Virtual Machine (VM)
- API (services for applications)



Card Manager Responsibilities

- Communicate with off-card entities
- Initialise the card in a secure manner
- Manage all card changes: applet loads, installs, deletions
- Perform verification of Delegated Management functions
- Manage the application and card life cycle states
- Perform optional velocity checks on the global PIN
- Provide cryptographic support for the issuer's applications
- Manage card global data, useful for card management systems
- Owner of the Card Registry



Security Management

- Card Manager is at the centre of the
  - Global PIN Management
  - Application Locking
  - Card Locking
  - Card Termination
  - Operational Velocity Checking
  - Tracing and Event Logging

Security Domains

- Secure mechanism for adding applications to the card with Issuer's permission
- A mechanism for Issuers to assign some privileges to Application Providers
  - Personalization of applications
  - Runtime cryptography for applications
  - Delegated Management

Clear separation of what can be done

GlobalPlatform 2.1 Summary of Changes

- Application Reception Data from Security Domains
- Application Extradition
- Retrieval of Executable Module Information
- More Card Recognition Data
- New Secure Channel Protocols
- Card Manager and Issuer Security Domain Separation
- CVM (Additional Methods)
- Windows for Smart Card (WfSC) Portability

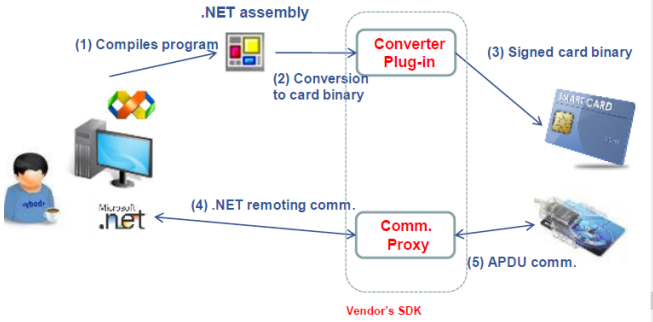
Allows partners to let them handle themselves = protection of investments

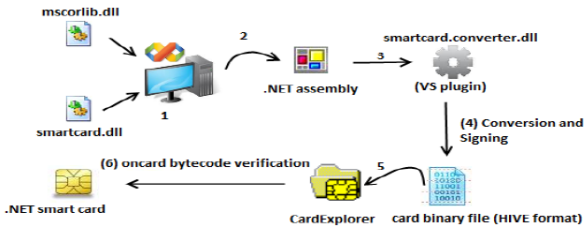
WfSC

- The WfSC Concept and how it came into existence.
- You have seen it…forget it!
  - It does not exist any more.

.NET Card Significance

- Smart card based corporate badges
  - Remote Access Control (VPN, Cert Based Auth.)
  - Microsoft: Over 300.000 badges delivered since 2005
- DoD
  - US: Combination of .NET and JavaCards
  - UK: .NET smart card for remote access to Microsoft Internet Security and Acceleration server (ISA) 2006 and (Intelligent Application Gateway) IAG 2007





- MSIL Code converted to the ".NET Smart Card Framework's Card Resident Binary Format"
- Card binaries need to be digitally signed (integrity & authenticity)
- Gemalto plug-in converts and signs applications automatically

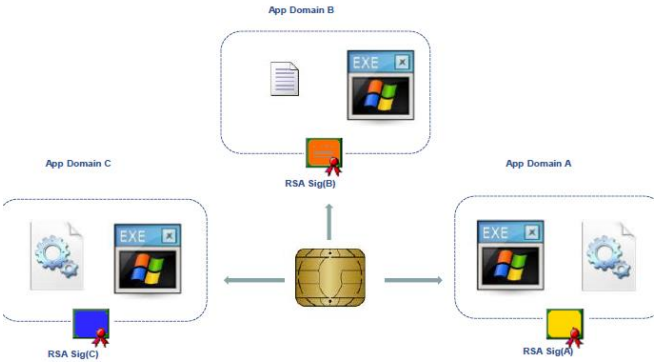
Gemalto .NET Virtual Machine

- Security Model
  - Role Based and Evidence based Access Control
- Assembly Strong Name
  - Digital Singature binding assemblies
- Application Code Security
  - Digital Signatures
- Application Isolation
  - Application boundaries through "Application Domains"
- Application Data Security
  - .NET objects inside assemblies
- Code Verification
  - Gemalto .NET on-card IL-code verifier

.Net Functionality

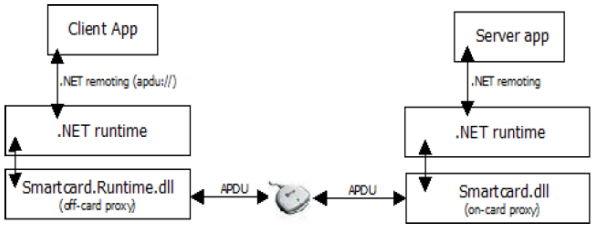
Supported	Not Supported
• Upload file format four times smaller than .NET assembly • .NET smart card framework is adapted to smart card memory model (persistent memory) • A set of APIS – PIN management, transaction management, garbage collection	• **Floating points** • **Multi dimensional arrays** • **Reflection (?)** – Get information about the type of an object at execution time – Dynamically creating objects – "Observing and modifying program execution at runtime" • **Multi Threading** • **Asynchronous Calls** • **Server -side remoting only** – **Card can't initiate a conection to the reader**

.NET smart card security model



.NET Card APDU "Isolation"/RMI

- No need to design APDUS
- Proxy Interface based on .NET remoting technology
  - Call on-card services using TCP or HTTP



*Source: "Vulnerability Analysis of the Gemalto .NET Smart card Operating System", Behrang Fouladi Azarnaminy, 2012 MSc in Information Security Projects

How secure is .NET card?

- Has EAL5+ certified Infinion chip
- .NET card OS was designed to achieve EAL4+
  - It actually achieved EAL5+, but for the chip…
  - No information about the .NET architecture
- No published vulnerabilities so far

What is a BasicCard?

- A smart card that is programmable in Basic language
- BasicCards are manufactured by ZeitControl
- They also provide APIs for Java and .NET languages
- Converted to ZeitControls' (ZC) P-code that the BasicCard will then execute

A Virtual Machine for the execution of ZeitControl's P-Code
A directory-based, DOS like file system
IEEE-compatible floating-point arithmetic

- 256-4800bytes of RAM
  - Run-time data
  - P-Code stack
- 2-72Kbytes of EEPROM
  - User's permanent data
  - File system
  - User's basic code (the application)

200-400 lines of source code = efficient

Available Crypto (RSA, DES, AES, SHA..)

- Compiler
  - Create's P-code
- ZC-Basic Multiple Debuggers
  - Run multiple Terminal and BasicCard programs simulaneously

Transaction ManagerTransaction Atomicity

- All BasicCards contain an automatic EEPROM Transaction Manager,
  - "Ensures that file operations, and changes to EEPROM data items, occur as a single transaction:"
- It works like this:
  1. "The card prepares a Transaction Log in EEPROM, which contains the write operations to be performed.
  2. The card writes to a single flag-byte in EEPROM, which activates the Transaction Log.
  3. The card performs the write operations on the Transaction Log.
  4. The card clears the flag-byte, to deactivate the Transaction Log.
  5. Then if Step 3 is interrupted, the flag-byte will remain set; so that the next time the card is powered it known to complete steps 3 and 4 before continuing".

Source: www.beckerelelectronics.com/basic

BasicCard RMI = hides the APDU again

Main DIFF = PRICE

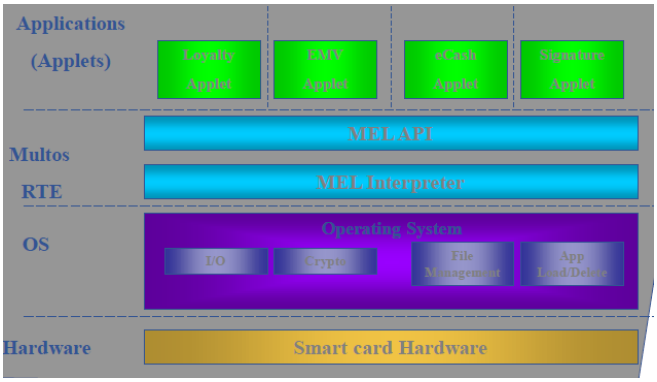
Same App =

- Price: Complete BasicCard Toolkit $109
- Storage Requirements
  - Multos → 1 kByte RAM, 16 kByte ROM, and 16 kByte EEPROM (? TO BE CONFIRMED ?)
  - BasicCard → 256 bytes RAM, 8 kByte ROM, and 1 kByte EEPROM

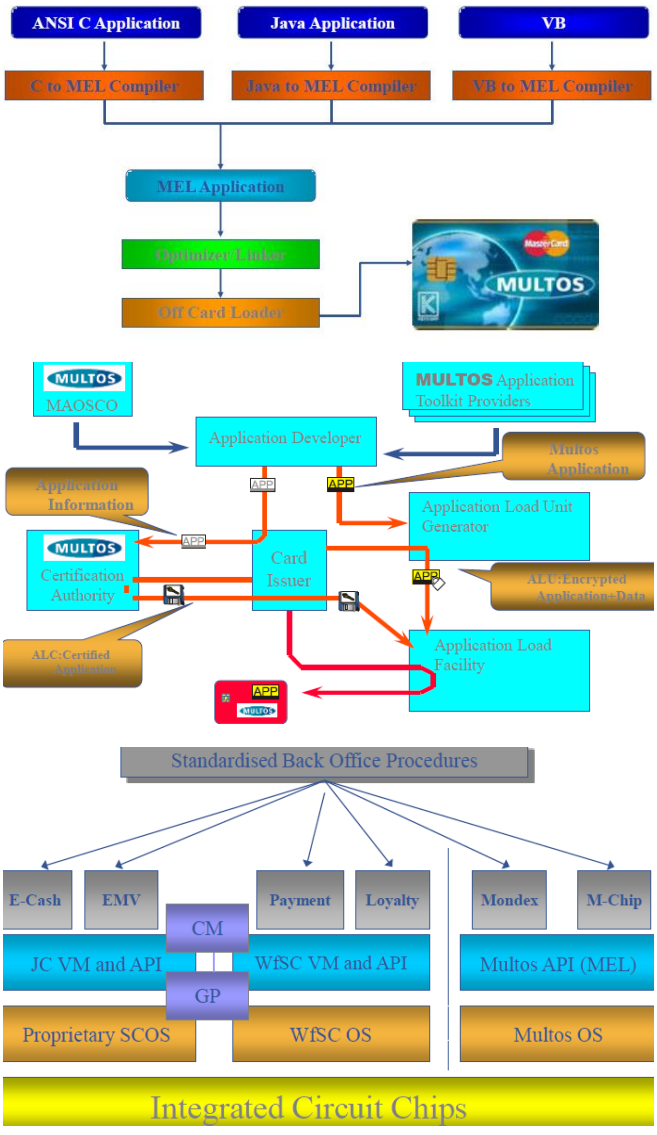
MAOSCO

- Drive adoption of MULTOS as industry standard
- Manage dev
- Provide MULTOS licensing + approval
- Matchmaker SUPPLY/DEMAND

MULTOS Architecture



Any language



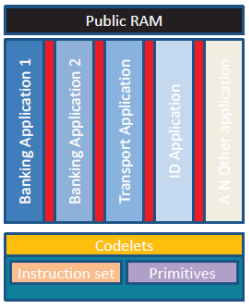
Multos = very simple, powerful, good business model

Every time it signs, it can earn money

Criteria	GP	Multos	WfSC	JavaCard
Concept	Platform	SCOS	SCOS	VM & API
Virtual Machine Technology	Interpreter	Interpreter	Interpreter	Interpreter
Byte Code	No predefined byte code	MEL	TBC	Java byte Code
Performance	N/A	2-3 times slower than native	TBC	2-3 times slower than native
VM & SCOS Extensions	Yes, subject to implementation	Yes, subject to ITSEC implementation	YES	Yes, subject to implementation
Security Level	EAL4+(France), EAL5+(Germany)	ITSEC 6 Achieved	Microsoft Approval	EAL4+, EAL5+
Based	Chip, SCOS, JVM, GP	Chip and SCOS	Chip, SCOS	Chip, SCOS, JVM
Crypto Engine	Yes (Optional)	YES	No	Yes (Optional)
Post Issuance Application Loading	Card Manager Security Domains, etc.	Application Load & Delete Certificates	Yes, protocol & security not defined	Yes, protocol & security not defined
Memory Requirements	~8K	~4K VM, ~7K Executive	TBC	Java Card VM: ~16K Java Card API:~8K
Libraries	~8K for crypto	~15K for crypto	TBC	--
Programming Languages	No predefined language	MEL, C, even Java or VB	VB (Subset)	Java (Subset)
Extensible APIs	Yes (Third Party)	Yes, only after evaluation	Microsoft Verified code extensions	Yes, Third party
Processors	OS depended	Optimised for 8-bit processors	Designed for 8-bit processors	Optimised for 32-bit but available on 8-bit
Portability	Portable across any Java Card or WfSC	Fully portable across Multos implementations	Portable across WfSC implementations	Portable across 2.1 compliant Java Cards
Specification Open/Closed	Open	"Open"	"Closed"	Open
Controlled/ Maintained	Global Platform	MAOSCO	Microsoft	SUN
Current Version	GP 2.1	Multos 5	WfSC 1.1	Java Card 2.2
Model	Issuer Centric & delegated management	Issuer Centric	Not Specified	Not Specified
Licence Required	No	Yes	YES	Yes
From	N/A	MAOSCO	Microsoft Approval	SUN
What for	N/A	Multos OS & VM	Implementation	VM Implementation
Who Pays	N/A	SCOS Implementer	SCOS Implementer	SCOS Implementer
Cost Evaluation	Depends on the level, country, deadline, etc.	~$600,000 (Varies)	Depends on the level, country, deadline, etc.	Will add ~40%-150% to the cost of the chip
Licence	N/A	£250,000+ £50,000 per annum	TBC	$400,000+ $70,000 Per annum
Card	~$3 PK <$2 non PK	~$3-$5 PK Card	~$2-$8	~$2-$7
Application Load/Delete Royalties (Issuer)	No	~~£0.025~~	Depends on the model	No
Application Loading Cost (Issuer)	~$0.30-$0.60, manufacturer Specific	~$0.30-$0.60, manufacturer Specific	~$0.30-$0.60, manufacturer Specific	~$0.30-$0.60, manufacturer Specific
Card Management Systems	Several available	Several available	Several available	Several available
Enhancing Acceptance	Liaise with ETSI (not official member yet)	Member of ETSI	TBC	Member of ETSI
Personalisation and Back Office Systems	> 30 Vendors at special prices	~ 17 vendors	TBC	TBC
SIM Cards	>200 million cards	Multos SIMS since early 2001	N/A	>200 million Java Cards (GSM 03.48 [GP cards] and GSM 03.29[JC SIM card])
Multi Sourced	YES	"YES"	YES	YES
Development Tools	Same as standard Java	MDS, Hitachi, QuickSmart, etc.	VB/C++ Programming Environments	SUN, Symantec, Cyberflex, Gemplus, etc.

MULTOS App Dev

- Secure, Multi-App OS
- Traditionally used in secure microcontrollers
- Typically now = smartcards
- Also now = secure general purpose microcontrollers
- GSM not in the list (but possible)

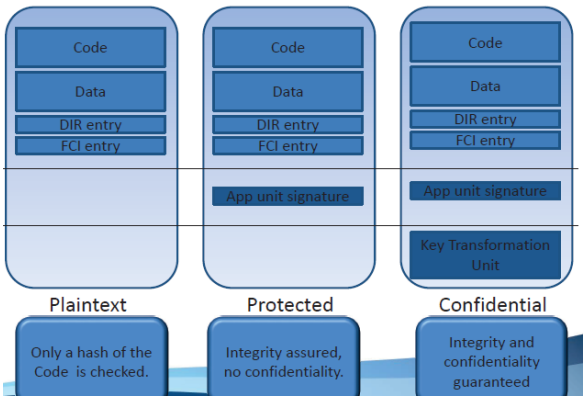


Application Abstract/Virtual Machines

Differences to Javacard

- Uses GP(Global Platform) for content mgmt.
  - Secure channel uses sym crypto
  - MULTOS uses asym crypto (certs)
- Requires key exchanges between multiple parties
  - MULTOS CA (KMA) handles all perso keys
- Javacard requires off-card app safety validation
  - Multos = fully “multi-party-secure”
- Java on-card security mgmt. by “context
  - Applets in same context can access data obj in same contexts
  - MULTOS app separated by FW, can only communicate via shared RAM under controlled conditions
- Java Perso usually done **online** (needs connection for every card)
  - MULTOS usually done offline

ALU Protection



Key Transformation Unit

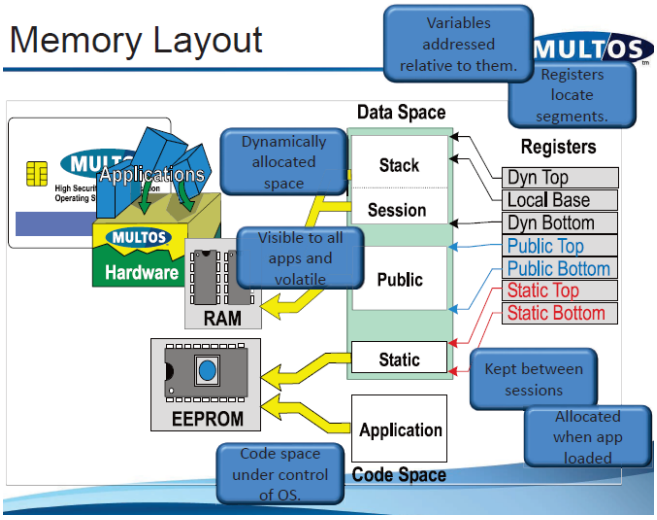
- Its not actually the whole ALU that is encrypted by the card public key during perso, just the KTU
- Contains 1 or more encrypted area descriptors
  - Start offset of encrypted area
  - Length of encrypted area
  - Random symmetric key used to encipher area
- The KTU is then encrypted by the target card’s public key.
  - Only that card can then read and load the ALU

Load Process Summary

- Create required space
- Load ALU
- Load ALC
  - Do checks
    Issuer ID, signatures, sizes, AID etc.
  - Create application if OK
  - If load fails, decrement retry counter

MULTOS VM

Memory Layout



MEL Instructions

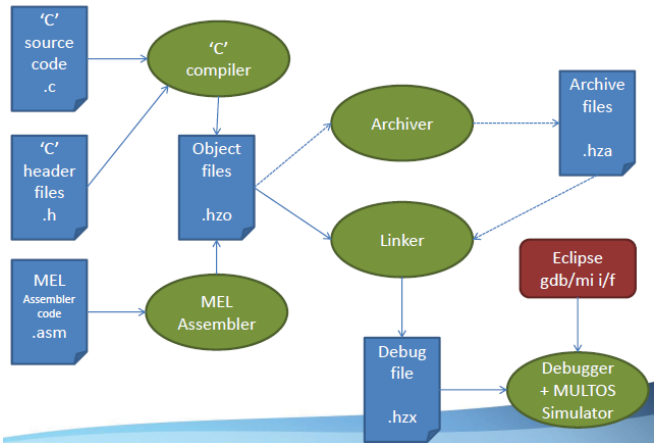
- 39 Instructions – 3 main groups
- Simple maths (Add, Subtract, Increment, Decrement etc)
- Logical (Comparison, condition register tests etc)
- Programme flow control (Call, Branch, Jump, Return, Primitive calls etc)

Primitives

- Much more complex functionality
  - E.g. Cryptography, memory management, mathematics
- Have up to 3 fixed parameters
- Use the stack for variable parameters and results
- Called using the PRIM instruction
  - But usually wrapped by the C-API (multos.h).

Main Components

- Toolchain plugin for Eclipse C/C++ Developer Kit
- Compiler / Assembler / Linker
  - hcl, hcc, has and hld
- Simulator with debugger interface for Eclipse
  - hsim and mdb
- Standard 'C' libraries and header files
- Some helper macros for things like crypto



Typical APDU Processing

- Check CLA byte valid
- Check INS byte valid
  - CheckCase for particular INS
- Execute related code
- Set any returned public data
- Set status word
- Exit

IN = >

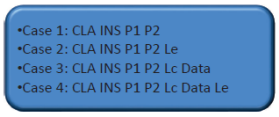
CLA = command
INS = Instruction
P1 = parameter 1
P2 = parameter 2

OUT = >

La = length
returned data
SW1/SW2 = status word

CheckCase

- Tells MULTOS how to interpret the command data in public
- Validates the command data in public
- "Case" refers to the ISO 7816-4 command cases

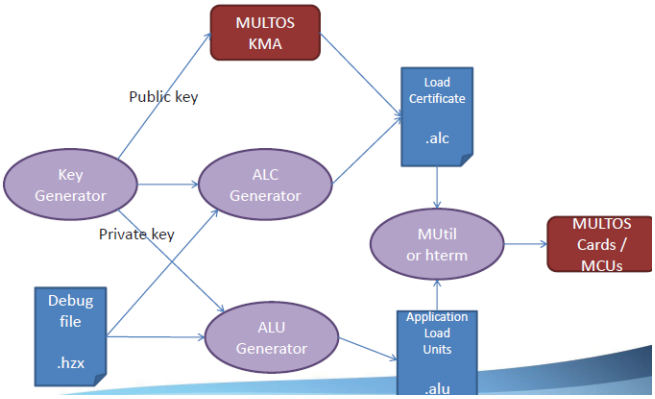


Performance

- Put variables that will be constantly changing into RAM.
  - Local variables (on stack)
  - Session memory
  - Remember RAM is limited, so don't go mad!
- Some standard 'C' library functions are slow
  - Coded to run on even the oldest of chips – use lots of MEL instructions
  - Check to see if there is a primitive available instead – much fewer instructions, call native code
  - E.g. shifting and rotating by variable number of bits
  - The C-API (multos.h) supports all the latest features

Other command line tools

- Key generator tool - hkeygen
- ALU generator - halugen
- ALC generator for test cards - melcertgen
- Object File lister (disassembler) - hls
- Binary file dump tool - meldump
- Terminal emulator - hterm
- MUtil – actually a GUI tool (separate download)



Load debug file to card

- Debug files end .hzx
  - Contain info needed for debugging
  - Plus the code, data, etc
- To load to a card with application defined attributes from CMD prompt

  c:\> hterm –cardtype MI-M3 -clean –load *myapp*.hzx

Create "Release" application

- Release files end .alu
  - Contain the code and static data
  - Contain the directory and permissions information
  - Debug information removed
- ALUs can be loaded using MUtil
  - See MUM.pdf for details
- Convert hzx to (unprotected) alu from CMD prompt

  c:\> halugen *myapp*.hzx

Summary

- MULTOS app loading is different, using asymmetric methods instead of symmetric ones but its not difficult.
- MULTOS app development in 'C' using Eclipse is very simple
  - Develop using MULTOS libraries and C-API
  - Debug using the integrated simulator
  - Deploy to developer cards (if you have them) or live "community" cards for final testing
- The developer tools hide the loading process

Smart Cards	Embedded Security
	
› Smart card payment › Electronic passports and ID documents › SIM cards for mobile communication › Transport ticketing	› Mobile device security and payment › Information and communications technology (ICT) security › Industrial and automotive security › IoT connected device security

Fake Passports & ID cards - some 2018 incidents

March 2018 : **Greece** Criminal network supplying identity and travel documents dismantled. More than 900 passports and identity cards seized. 620 found to be recorded on European databases as lost or stolen.

August 2018: **UK** man who made £84,000 selling fake passports handed 28-month prison sentence

September 2018 **Fake ID gang arrested in China with more than a million forged documents**
•Police detain 62 after raid uncovers 'mountain' of half-finished counterfeit credentials
•Identity cards, passports, birth certificates, driving licences, documents of all kinds were couriered to customers across 20 provinces

September 2018 **UK National Crime Agency seized thousands of forged documents**

December 18, 2018 : **Dubai** Over 1,000 fake passports seized at Dubai airports in2018

What is identity theft?

= the illegal use of one person's personal information by another to commit fraud or other crimes..
..in the real world and in the virtual world! (www.answers.com)

Identity theft can involve various security threats to ID documents:

- counterfeiting a complete travel document
- Photo substitution
- deletion / alteration of text in the visual or machine readable zone of the data page
- construction of a fraudulent document, or parts thereof, using materials from legitimate documents
- removal and substitution of entire pages or visas
- theft of genuine documents blanks
- new electronic attacks like "Skimming", "Hacking" and "Phishing"
- copying of chip content

Why do countries issue secure ID cards and passports?

- Personal identity documents confirm the identity of individual citizens, thus proving their legitimate residency within their homeland
- In the modern world traveling in other countries is only possible with a passport or National ID card (e.g. Schengen Area)
- Government offices, credit institutes, and e-businesses ask for ID cards as a means of unequivocally identifying the holders (e.g. opening a bank account, drawing government benefits etc.)
- Other documents, such as driver's licenses or social security cards may not be designed for 3rd party identification purposes
- Reduce crime by secure identification by linkage to biometrics
- Offline data storage of individual biometric information considering privacy and data protection

Important standards for ID cards

› ISO/IEC 7816	Contact based smart cards
› ISO/IEC 14443	Contactless smart cards
› ISO/ IEC 7810	Physical characteristics of ID cards
› ISO/ IEC 10373-3	Test methods of ID cards
› ISO/ IEC 18013	Drivers License
› Java Card Specifications	(if applicable)
› Global Platform Specifications	(if applicable)
› ICAO* Doc 9303 part 5	Specifications for TD1 Size Machine Readable Official Travel Documents (MROTDs) (Technical Spec endorsed by ISO Standards 7501)
› EU Directive CEN TC 224 WG 15	(European Citizen Card)

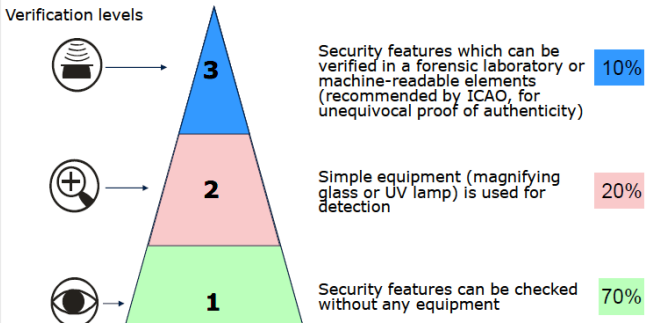
ICAO Standard

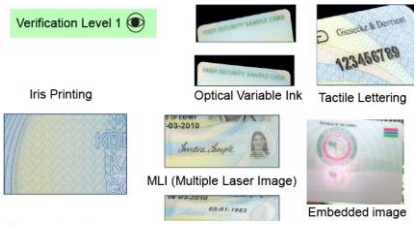
- Passport
- National ID cards
- Alien Registration Card (BRP)
- Registered Traveller Card

Specification	PVC PolyVinylChloride	PC PolyCarbonate	PC - Polyester Compound
Bending (ISO 10373) (approx.)	> 1.000	> 8000	> 8.000
Torsion (ISO 10373) (approx.)	> 1.000	> 10.000	> 10.000
Temperature resistance (approx.)	50°C	140 °C	110 °C
Lightfastness	low	High	High
UV-stability	low	High	High
Estimated Lifetime [years]	3 - 4	10	7-10
Material costs approx. (n x PVC)	1	6	4

› For national eID and eDL in Europe mostly polycarbonate cards are used.

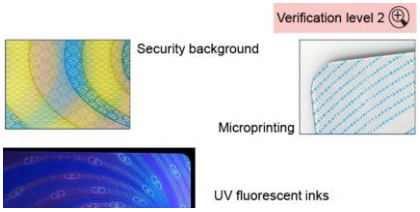
The three verification levels of security features (Industry standard)

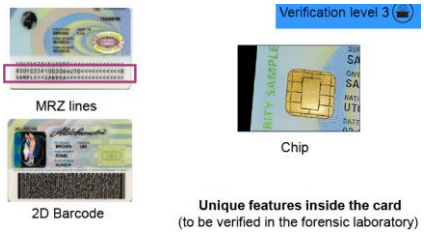




- Finding counterfeits on the street is hard!
- Police/Border officers have a few minutes
- They are under pressure to stay alert

- Security features have to be easy to check
- Security features have to be hard to forge





Unique features inside the card
(to be verified in the forensic laboratory)

Additional security of smartcard chip

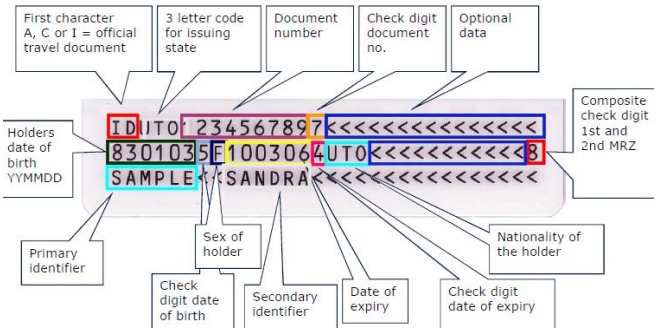
Data printed on the card is also stored in the chip

This data is digitally signed preventing alteration

Improvement to "optical" methods

Clear "Card is valid" feedback

No special knowledge necessary
no question "how should it look?"



Parameter	Contactless	Contact
Interfaces	Contactless proximity interface (ISO/IEC14443)	Contact card interface acc. to ISO/IEC7816
Supported services	All relevant governmental and citizen services	Governmental and citizen services excl. ICAO
Supported security	++ (certification up to CC EAL6+)	++ (certification up to CC EAL6+)
Robustness of cards	++ (All parts fully encapsulated)	O (Limited by contacts and construction)
Lifetime of cards	++	Depending on usage
Cost per card	++	+
Transmission speed	Up to 848 Kbit/s (8,6MByte VHBR)	223 Kbit/s (3,571 Mhz; F/D=16)
Design / optical card security	++	Limited by position of contact field

Proprietary (e.g.STARCOS®, MTCOS*)
Compliant to ISO 7816 part 1-4
Certified to CC EAL4+ High

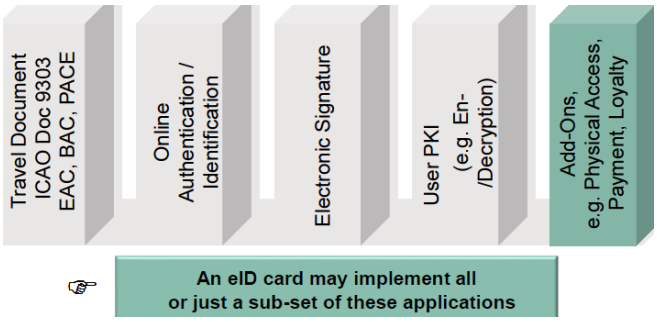
"Filebased, proprietary"

Faster

Java Card™ *compliant to:
- Oracle Java Card Spec.
- Global Platform specification
- ISO 7816 1-3
- Certified to CC EAL4+ High

"OPEN"
Fast

There are advantages for either kind of OS.
Both OS are very secure and flexible.
The choice is a philosophical question.
Outside Europe there is a trend towards Java Cards.



☞ **An eID card may implement all or just a sub-set of these applications**

The German ID card serves as a Travel and Visual identification document and offers authentication for the virtual world (introduced 2010)

Data	Application
■ ID-Security Document	■ Visual Identification
■ Biometric Features	■ Travel function like e-Passport
■ Citizen certificate ■ Name, Address ■ Date of birth etc.	■ Identification (national) ■ Authentication on the internet
■ Optional: Digital signature	■ Legally Binding Transactions

Biometrics in ID cards

Integration of Biometrics into Smart Cards

- Contactless / Contact-based
- Native / Java Card operating systems
- Biometric methods: face / finger / iris
  - one, two or ten fingers,
- On-card matching or just storage

Fingerprint Templates according to ISO-19794-2

The generated fingerprint template format can be fully compliant to the ISO-19794-2 compact format.

Minutiae based template 180 bytes

ICAO compliant image: 10 KBytes

NeID implementions in Europe – Key findings & trends I

Huge diversity of national implementations

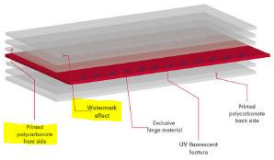
- Chip Operating Systems: JavaCard vs. Native
- Interfaces: Contact based, C'less, Dual Interface or Hybrid
- Card functionality: Main functionality resides in the card vs. the card is simply a token to back-end systems
- Variety of applications
  - eID/eAuthentication
  - eSignature / eHealth
  - ePass
- Legal aspects & privacy protection rules are very different from country to country

- Online authentication & eSignature for eGovernment services
- Number of cards with support of contactless interface increasing
  Main reason: ICAO ePass functionality as part of NeID cards
- Biometric data becomes an essential part of the NeID – if the legal privacy protection rules of the country allow it
- Strong demand for Common Criteria security certified solutions
- Increase of memory capacity: today > 128 kBytes
- Standard card lifetime today: 10 years
- Interoperability of NeIDs in Europe considered as the key issue
- EU-funded interoperability programs, e.g. STORK, PEPPOL

Country	Roll-Out	Inter-face	Memory (kBytes)	Bio-metrics	Life-time (years)	Costs (EUR)	Functions / Services
Finland* (FINEID)	1999/ 2000	Contact	n.k.	-	5	48.-	eID/eAuth, eSig, (eHealth)
Estonia (eCard)	2002	Contact	32	-	5	~16.-	eID/eAuth, eSig
Belgium (BELPIC)	2003	Contact	32	Face	5	10.-	eID/eAuth, eSig, eHealth
Austria (eCard) **	2004	Contact	32 52 (2010)	-	7	10.-	eID/eAuth, eSig
Sweden*	2005	Hybrid	32 (c'less) 32 (contact)	Face, FP	5	~ 40.-	ePass (BAC) eID/eAuth, eSig
Italy (CIE)*	2006	Contact	32	Face, FP	10	25.-	eID/eAuth
The Netherlands (G2)	2009	C'less	80	Face, FP	10	TBC	ePass (EAC 1.11)
Italy (CNS)**	2006	Contact DI (2009)	32	-	n.k.	20.-	eAuth, eSign, eHealth
Germany	2009	C'less	128	Face, FP	10	27	ePass (PACE), eID eSign

*not compulsary **citizen service card

Datapage construction



ePassport Elements

The **ICAO recommends/specifies (Doc 9303):**

- Contactless micro-processor chip
  - In conformity with ISO/IEC 14443, type A or B
  - Minimum 32 KBytes memory. Typical 80KB
  - Short-range antenna (max. 10 cm)
- Interoperability with documents readers worldwide
  - Image files for biometric data instead of templates
- Life span of up to 10 years (5 years advisable)
- Standardized data structure on the chip ("LDS" = Logical Data Structure)
- Multi-level PKI (digital signatures, encryption, PKD)

ICAO differentiates between mandatory and optional data:

1. Machine Readable Lines

P<UTOROBERTSON<<DAVID<<<<<<<<<<<<<<<<<<<
G1001872<3UTO6812034M1501283123456789<<<<<78

2. "Primary Identifier":
   - Facial Image (~15kB)

3. "Secondary Identifier(s)":
   - Fingerprints and/or (~10kB each)
   - Iris scans (~30kB)

4. Additional Text Data
   e.g. admin data of the issuing country or biographic data

LDS (Logical Data Structure)

- Is specified in ICAO **DOC 9303**
- Is supported by several ISO/IEC standards
  - file system according to **ISO/IEC 7816**
  - contactless transmission standard **ISO/IEC 14443**
  - biometric standard **ISO/IEC 19794**
- The current version is LDS 1.7 (**mandatory**), LDS1.8 (**optional but ASAP**)
- Various data from the printed data page is stored in different Data Groups (DG)
- Standardization body: **ISO/IEC JCT1/SC17/WG3/TF5**

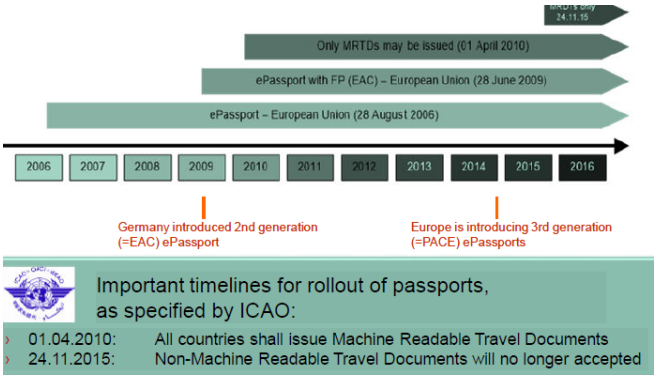
Security mechanisms for e-passports/e-ID cards - Overview

mandatory

- **Basic Security**
  - Passive Authentication (**PA**)
    Integrity of data

optional

- **Improved Security and Privacy**
  - Basic Access Control (**BAC**) -> **PACE** recommended
    Checks physical access to the document
    Prevents skimming and eavesdropping — mandatory in
  - Active Authentication (**AA**)
    Prevents chip cloning
  - Extended Access Control (**EAC**)
  - Checks entitlement of terminal
  - Protects sensitive data, e.g. biometrics through verification of a Terminal Inspection System Certificate
  - consists of CA (chip authentication) and TA (terminal authentication); — mandatory in
  - strong session keys by CA



Important timelines for rollout of passports, as specified by ICAO:

- 01.04.2010: All countries shall issue Machine Readable Travel Documents
- 24.11.2015: Non-Machine Readable Travel Documents will no longer accepted

How secure? FORGEABLE?

- This obviously is NOT a successful attack. It can be compared with copying the data page of a passport on a traditional copy machine!
- The public readable data can be read out; access condition: readable

- Concerning e-passport security the document considered in its entirety
  - Chip cannot be removed from booklet without tamper evidence
  - Chip in passport is an ADDITIONAL security feature within a dozen of traditional security features in a passport
- Application of ICAO security features (e.g. active authentication) prevents copying of data: Software operating systems are required to be Common Criteria EAL4+ certified
- Electronic authentication of the country signing certificate prevents change of data and enforces legitimacy (countries should join the ICAO PKD)

Authenticity & Integrity	Confidentiality	Originality
Is the chip from the rightful authority? Have the data been manipulated?	Can the communication be eavesdropped and is this the rightful reader?	Has the chip been cloned or substituted?
PA - Passive Authentication	Access Control BAC/ EAC/PACE	**AA** - Active Authentication
Mandatory	Optional	Optional

Procedure of Passive Authentication

1. Read the Security Object from the MRTD chip.
2. Retrieve the corresponding Document Signer Certificate and the trusted Country Signing CA Certificate.
3. Verify the Document Signer Certificate and the signature of the Security Object.
4. Compute hash values of read data groups and compare them to the hash values in the Security Object.

Passive Authentication enables an inspection system to detect manipulated dat. groups, but it does not prevent cloning of MRTD chips, i.e. copying the complete data stored on one MRTD chip to another MRTD chip.

DG	Content	Read / Write	Mandatory / Optional	Access Control
DG1	MRZ	R	m	BAC
DG2	Biometric: Face	R	m	BAC
DG3	Biometric: Finger	R	o	BAC + EAC
DG4	Biometric: Iris	R	o	BAC + EAC
...		R	o	BAC
DG14	SecurityInfos*	R	o	BAC
DG15	Active Authentication	R	o	BAC
DG16	...	R	o	BAC
SO_D	Security Object	R	m	BAC

BAC = Basic Authentication Checks

BAC checks that it has physical control to the travel document by requiring the MRZ to be read optically

BAC is based on symmetric cryptography

The Basic Access keys are generated from printed data of the MRZ with limited randomness

MRZ data field	Possibilities	Entropy
Date of Birth	365 × 100	15 bit
Date of Expiry	365 × 10	12 bit
Document number	10^9 for numeric 36^9 for alphanumeric	48 bit

☞ Theoretical entropy: 73 bit

☞ Reality: entropy ~50…60 bit for random alphanumeric and ~40 bit for sequential numeric document numbers

The EAC concept

Data protection by

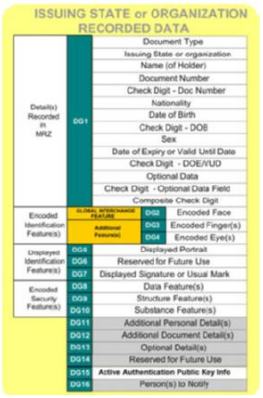
1. Mutual authentication
2. User Role and Right concept

Mutual authentication is based on

- Card Authentication
- Terminal Authentication

Authentication is based on **asymmetric cryptography** (NIST/Brainpool)

EAC **avoids** that the ID cards signs **unknown challenges** (e.g.time and location of traveller)

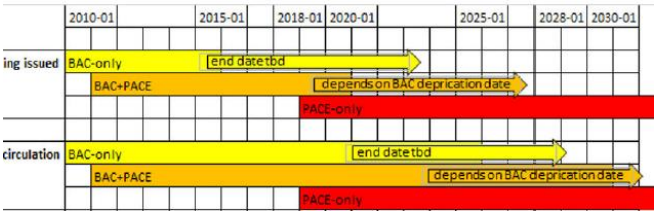


PACE protocol

New: PACE (Password Authenticated Connection Establishment)

PACE solves the possible security problems of BAC by exchanging strong passwords to prevent eavesdropping immediately after access control. For getting access to the chip, the same combination of document number, birthdate and expiry date is used as password. Additionally, the chip can accept one or more Card Access Numbers (CAN) as a password. Typically, a CAN is only 6 digits long and is printed on the document, like the Machine-Readable Zone

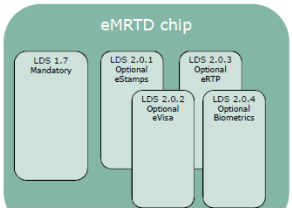
- Based on asymmetric cryptography (Diffie-Hellman)
- If document access key is broken, the session keys for no eavesdropped sessions can be obtained
- Short, key generation input can be used or MRZ can further be used!
- Two-stage retry counter spoils brute force attacks

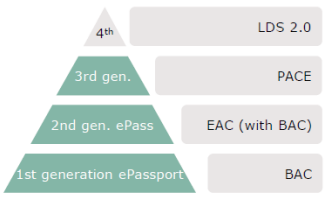


Future

eMRTD

- Technologies for eMRTD (**e**lectronic **M**achine **R**eadable **T**ravel **D**ocuments) are evolving from the digitization of the MRTD data page. LDS2.0 does not replace LDS1.7
- After introducing BAC / EAC and PACE ePassports, the 4th generation electronic passports will benefit from LDS 2.0 applications; "**Digitization of the contents of a book**"
- LDS2.0 functionality offers optional applications (AIDs) alongside the mandatory LDS 1.7
- Enable issuing Countries to authorize receiving Countries to write to eMRTD





LDS 2.0 – eVISA & Additional Biometrics Implications on data memory

Travel Records (e-Stamps)

- Equivalent to inked stamps in a Passport
- ~50 eStamps :1.5kByte/ea data + sign =**75kB**

Storing Visa information (e-Visa)

- Data + Signature ~ 6k byte
- 6kByte/e-Visa for 5 … 10 eVisa
- Memory demand: **30kByte** … **60kByte**

Additional Biometric information

- Optional Application that allows additional biometrics
- Iris, fingerprints, voice, …
- Storage principle determines memory demand: Template or high-resolution Picture
- Memory demand: Several 100 Bytes … up to 20kByte per data set

LDS 2.0
Performance requirements and standards

Performance Requirements of LDS 2.0

- Read out time of an LDS 2.0 ePassport should not significantly increase
  - … even if the information of an additional eVisa is read out
  - … even if several hundred eStamps are read out

Standards for LDS 2.0

- VHBR (**V**ery **H**igh **B**it **R**ates) and Larger Frame Sizes need to be supported for LDS 2.0 ePassports
- VHBR is standardized in ISO/IEC 14443
- Larger Frame Sizes standardized in ISO/IEC 14443

Mobile Identity?

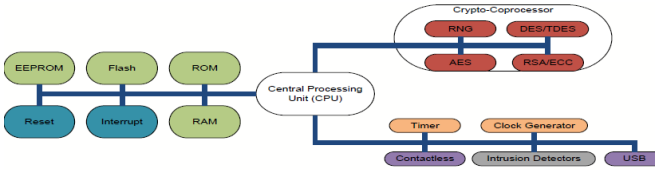
Where are the credentials?

Who has control?

Summary

- eID and ePassports use international standards.
- These standards form the basis of various other identity tokens such as eDL, Residence Permits, etc. But there many local/national requirements
- Backwards interoperability is a key driver for any design change
- Security of devices is a combination of physical and electronic elements, and is evolving with time.
- There are many hundreds of legally accepted designs, all of which change every few years.
- The demand for more security and more mobility is increasing, driven by the trend for digitization and the increasing movement of people.

Java Card Programming



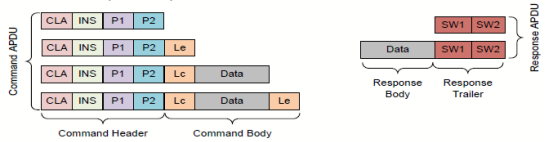
Acronym	Description
EEPROM	Electrically Erasable Programmable Read Only Memory
ROM	Read Only Memory
RAM	Random Access Memory
RNG	Random Number Generator
DES	Data Encryption Standard
TDES	Triple DES
AES	Advance Encryption Standard
RSA	Rivest, Shamir, and Aldeman (public key algorithms)
ECC	Elliptic Curve Cryptography (public key algorithms)

Constraints on Smart Card Programming.

- Limited Processing Capacity.
- Limited Storage Capacity.
- Restricted Communication Interface.
- Restricted User Interface.
- Restricted Execution Time Requirement on Applications.
- Limited Application Programming Interface (API) Support.

MASTER(Terminal) SLAVE(CARD)

Application Data Unit (APDU)



Command APDU		
Field Name	**Byte Length**	**Description**
CLA	1 byte	Instructions class – this indicates the class of command: proprietary or interindust
PI	1 byte	Instruction code – this indicates the instruction to be executed.
P2	1 byte	Parameter bytes.
Lc	0, 1, or 3 bytes	
Data	1 – 65,535 bytes	Data bytes.
Le	0, 1, or 3 bytes	Number of data bytes expected in response from a smart card.
Response APDU		
SW1	1 byte	Indicates the command processing outcome: successful or error.
SW2	1 byte	
Data	1 – 65,535 bytes	Data bytes.

Read Only Memory (ROM).

- Persistent data storage, write once and read multiple times.
- Data is written during the card manufacturing process.
- Comparatively large storage space then other types of memory.
- Usually stores Smart Card Operating System (SCOS) and/or runtime environment.

As an application developer, you might not have access to write to this memory. However, you can always read from it – using provided APIs

Electrically Erasable Programmable Read Only Memory (EEPROM)

- Persistent data storage, write and read multiple times.
- Storages applications code and data.
- Restricted by number of write cycles (approx. 100,000 to 500,000 write cycles).

Application is installed on EEPROM. Persistent variables are also stored in EEPROM. Please be careful with EEPROM usage with programming.

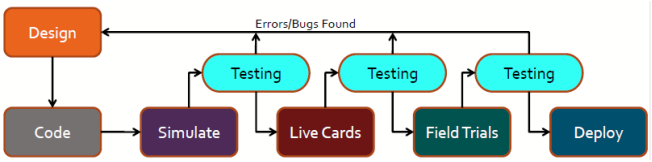
Random Access Memory (RAM)

- Data loss when card is powered off.
- Usually used during the application execution to store execution specific and temporary data.

Used during application execution. Non-Persistent or Transient variables are stored on this memory.

Smart Card Application Identifier (AID)

- Identifies uniquely an application on a smart card
- The AID is a 5-16 bytes identifier that consists to two components:
  - Registered Identifier (RID)
  - Proprietary Application Identifier Extension (PIX)
- The RID is 5 bytes long and compulsory, on the other had PIX can be 0-9 bytes long and it is optional.



Smart Card Application Programming Guidelines (2).

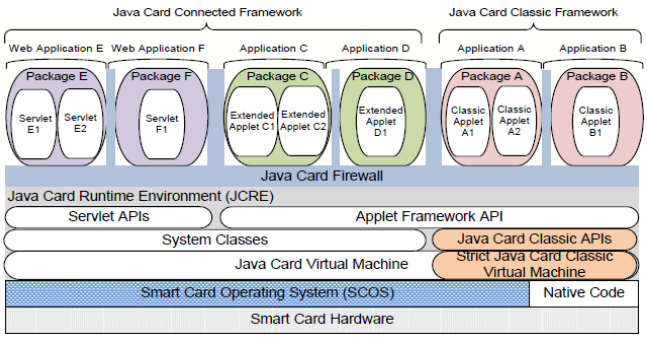
1. Understand the application requirements and targeted smart card's capability.
2. Adequately design the application for unexpected/unpredictable behavior of the application.
3. Design a clear and concise communication interface (the commands that an application will respond to if issued by a terminal).
4. Assign only the essential data structures that has to be stored on the EEPROM.
5. Consider life of individual data structures, even the data structure that are going to be stored in the RAM should be considered.
6. Reuse memory space (variables) within the application, especially data arrays if they do not create any security vulnerability.
7. Design a thorough and comprehensive testing plan for the application.

A multi-application smart card that supports a scale down version of Java language is referred as Java Card.

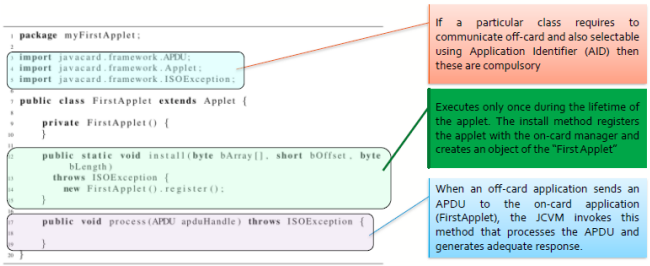
Java Card provides a secure, reliable and robust runtime environment to individual applications.

Provide to a some extend – hardware independence during the application programming.

Benefits
- Easy and rapid application development.
- Secure and robust application resource sharing.
- Secure application segregation – through firewall mechanism.
- Well- structured and defined Application Programming Interfaces (APIs).
- Free development and test environment.



Java Card v2.x and 3.x Classic	Java Card 3.x Connected
• Support boolean, byte, and short data types. • Single thread execution. • Single dimension arrays. • Exceptions. • ISO/IEC 7816-4 (APDU) and Remote Method Invocation (RMI) support. • Garbage collection (best effort). • Optional support for integer data type.	• Support all data types except float and double. • Multithreading. • Multi-dimensional arrays. • Support primitive wrapper classes (e.g. Boolean and Integer). • String manipulation classes (e.g. StringBuffer, StringBuilder, etc.). • Input/Output (I/O) classes (e.g. Reader, Writer, and Stream). • Support Java SE features like generics, metadata, assertions, and enhanced for loop, varargs, and static inputs, etc. • Compulsory on-card bytecode verification.



ISO7816 provides basic structure of smart card file system and associated commands to create and manage them.

ISO7816 does not provide the details on how the file system is implemented on a smart card.

Java Card 2.2.x and 3.x does not support ISO7816 file system.

Developer has the freedom/responsibility to design, implement and manage the data (file system and data structure) at the Java Applet level. BUT.....

- Developers also have to define adequate commands to manipulate the store data.
- Developers have to implement access control to data contents and additional security measures (e.g. cryptography).
- Java Card APIs support the implementation and management of data structures., but design decisions are on the sole discretion of the developers.

Java Card Objects (1)

- Objects are instantiation of a class. An object is the conceptual realization of the state and behavior defined by the associated class.
- State of a class is the internal data storage (e.g. literals, constants and variables).
- Behavior of a class is the implemented functionality.

Persistent Objects

- Created on EEPROM memory.
- State of a persistent object is preserved over the lifetime of an application.
- Its better to create few bigger persistent objects then number of small persistent objects.
- Persistent Objects are created using "new" keyword.

Transient Objects

- Object is created on RAM memory.
- State of a transient object is not preserved over the lifetime of an application.
- There are two types of transient objects: CLEAR_ON_DESELECT and CLEAR_ON_RESET.
- Transient Objects are created using factory methods, defined by Java Card.

Atomicity

- It ensures the integrity of the data items of a persistent object.
- Update of a data item is either completed successfully or else the value is restored to the original values (if encountered by an error during the update process).

Transactions

- Transactions allow updating of multiple data items across different objects to be atomic.
- If an error occurs, the values of multiple data items will be restored back to the original value.
- In Java Card, data items that require to part of a transaction should be defined between JCSystem.beginTransaction() and JCSystem.commitTransaction().

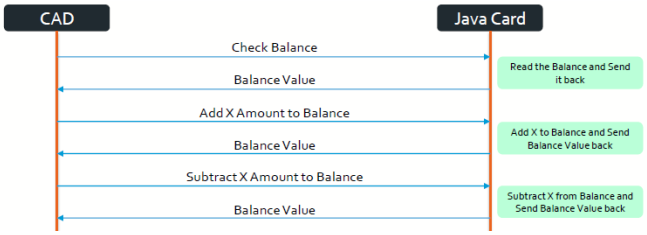
1. java.io
2. java.lang
3. java.rmi
4. javacard.framework
5. javacard.framework.service
6. javacard.security
7. javacardx.annotations
8. javacardx.apdu
9. javacardx.apdu.util
10. javacardx.biometry
11. javacardx.biometry1toN
12. javacardx.crypto
13. javacardx.external
14. javacardx.framework.math
15. javacardx.framework.string
16. javacardx.framework.tlv
17. javacardx.framework.util
18. javacardx.framework.util.intx
19. javacardx.security

Developers Responsibilities:

- Design and Implementation of commands that an off-card application can use to request different services from the respective application.
- Design and Implement handling of errors during the execution of the application (e.g. exception handling).
- Define and implement applets responsibilities and functionality.

Applet Responsibilities:

- Handling of I/O communication (e.g. APDUs).
- Data storage and management along with processing of the associated commands.
- Maintain and manage internal state machine.
- Management of shareable resources



Application Identifier (AID)

- `0xD0 0x00 0x00 0x00 0x62 0x02 0x01 0x0C 0x0E 0x0A`

Class Byte

- `0xC0`

Instruction bytes for individual commands/requests

INS Byte	Name	Description
0xE1	INS_Read	Read the Balance
0xE2	INS_INCREASE	Add the value X to the Balance
0xE3	INS_DECREASE	Subtract the value X from the Balance

Error Messages

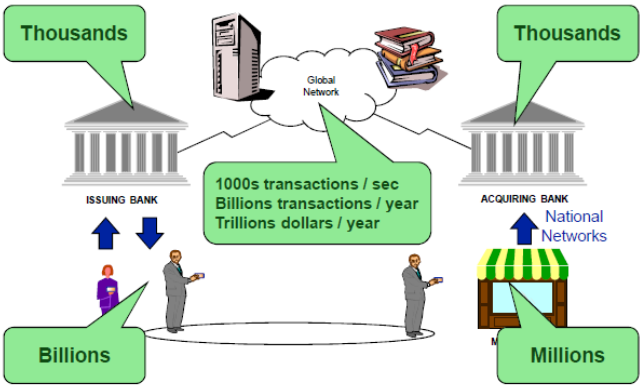
Error	Name	Description
0XF0C0	SW_CLANOTSUPPORTED	CLA byte in the command APDU is invalid.
0XF0C1	SW_INSNOTSUPPORTED	INS byte in the command APDU is invalid.
0XF0C2	SW_LCNOTSUPPORTED	Length of input data is invalid.
0XF0C3	SW_INCEXCEEDS	The counter value exceeds its limit (input +counter > limit).
0XF0C4	SW_DECEXCEEDS	The counter value become negative (counter – input < 0).
0XF0C5	SW_INVALID	Command structure is invalid.
0XF0C6	SW_UNDEFINED	Undefined error occurred during execution.

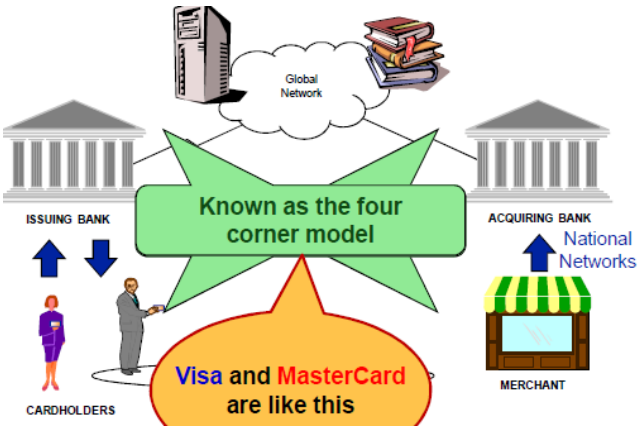
Conclusion

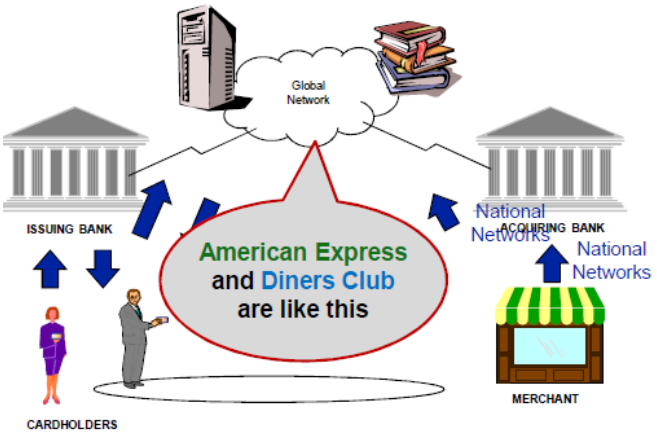
- Java card or embedded development is unique.
- As a programmer, it puts you in number of constraints that you have to abide by.
- Focus on design and specification of your application.
- Use online resources if you get errors or stuck in a programming problem.

Payment System Info Sec + Crypto

Early Days = physical security



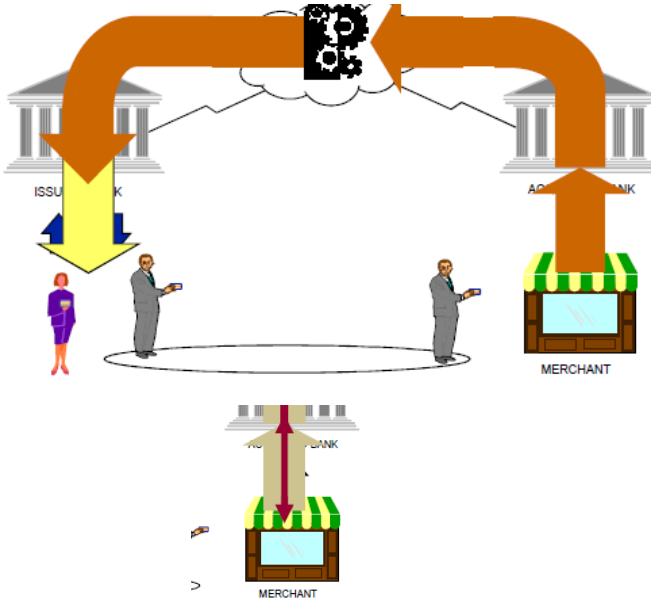




Credit, Debit, Charge, Pre-Paid?

- Transaction is the same
- A credit account accumulates the transactions and issues a statement on a monthly basis
- A debit account deducts the transaction amount straight from the current account
- Credit cards allow only a part of the sum to be paid – interest will be charged on the residue and added to next month's statement – "revolving credit"
- Charge cards are similar to credit, but the amount must be settled in full
- Pre-paid cards are like debit – against an separate account loaded with the initial funds. Some can be re-loaded

Clearing & Settlement



Authorisation =

Risk Management

- Payment Systems run Risk Management departments – not "Risk Elimination" departments
- Objective is risk mitigation and fraud control to curb the level of fraud, but balanced against the cost of doing so
- Typical target is to bring fraud down to below 0.01% for both Issuers and Acquirers
- One of the underlying (business) requirements is to do this in a cost-effective way, using devices, procedures and rules which are fit for purpose

Chips

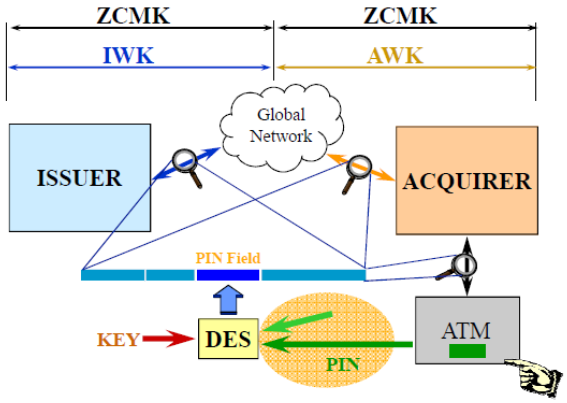
- Online Symmetric Key auth
- Offline Data Auth, offline PIN Check, Local Risk mgmt.

Uses ISO (every country have a national standards body)

- ISO 7810 – Identification Cards – physical characteristics
- ISO 7811 – Identification Cards – recording techniques
  - embossing, mag-stripe LoCo/HiCo
- ISO 7816 - Integrated circuit(s) cards with contacts
  - Primary chip card standard – 10 parts
- ISO 8583 – Bankcard Originated Messages
- ISO 14443 – Contactless – Proximity Cards
- ISO 9796-2 – Digital signature schemes giving message recovery

- **EMV** – was an acronym of Europay, MasterCard & Visa …
  - EMVCo is now Amex, Discover, JCB, MasterCard, UnionPay and Visa
  - Working together they produced an implementation specification for use of chip cards in the payment transaction environment
  - Based on ISO standards – in particular 7816 for the chip electro-mechanical and command structure and 9796, 9797, 10116 & 10118 for the crypto
- **3D-Secure** – Visa sourced specification for e-commerce
- **CAP** – MasterCard sourced specification for Token Authentication

PIN Protection



ISO 9564-1 Format 0

String 1 – 16 Hex digits

0 | 4 | 1 | 2 | 3 | 4 | F | F | F | F | F | F | F | F | F | F

Len | — PIN — | | — Pad — |

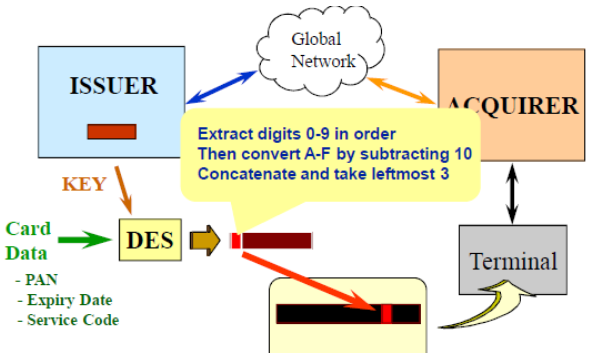
String 2 – 16 Hex digits

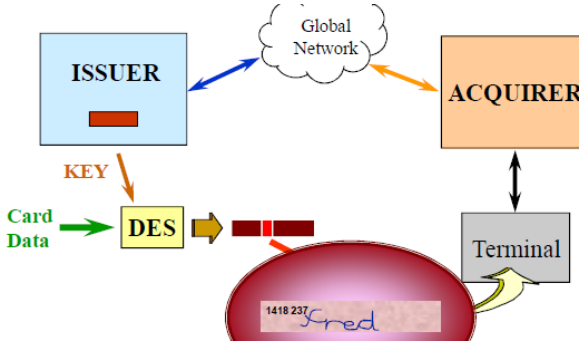
0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

| Zero fill | | — right 12 digits PAN — |

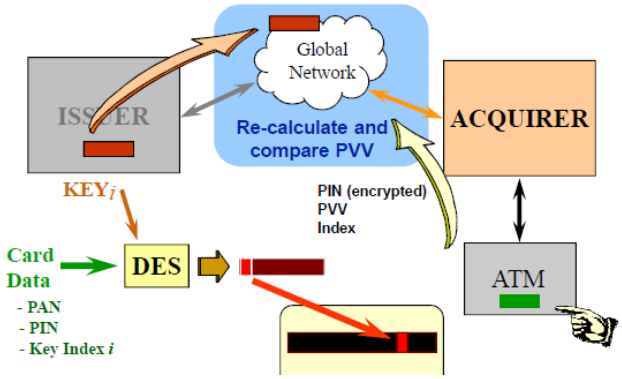
XOR

0 | 4 | 1 | 2 | 3 | 4 | E | D | C | B | A | 9 | 8 | 7 | 6 | F





PIN Verification Value (PVV)



CVV & PVV Size

- 3 digits (CVV & CVV2) is 1:1,000 guess
- 4 digits (PVV) is 1:10,000 guess
  = PIN guess for 4 digit PIN
- Online authorisation system limits attacker scope
- Issuers can detect attacks after a small number of attempts and can respond accordingly

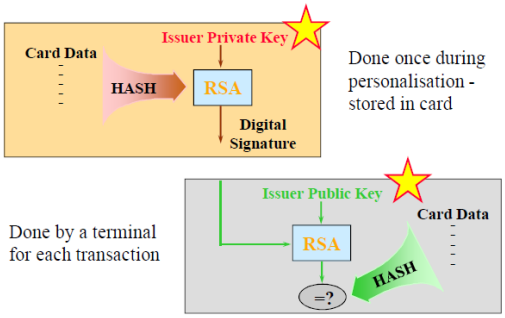
EMV Credit/Debit

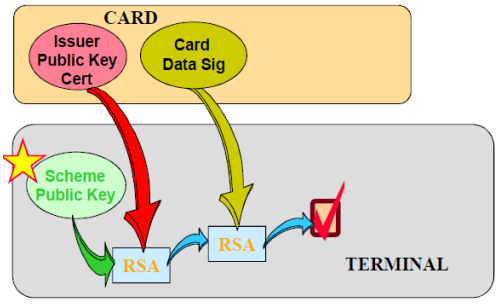
- Offline Data Authentication – SDA/DDA/CDA
- Card Authentication
- Issuer Authentication
- Transaction Certificate
- Management Functions (Script processing)

Offline Data Authentication

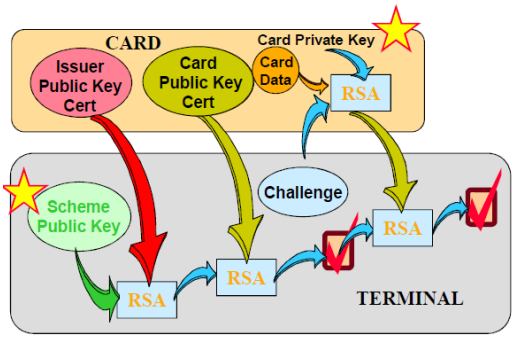
- Operates offline.
- SDA demonstrates to the terminal that the card data has not changed since it was personalised by the issuer.
- DDA demonstrates the above, plus that the card contains the unique private key linked through the certificate chain.
- CDA demonstrates both the above, plus that the on-line cryptogram has not been substituted by a "wedge" attack.
- Note: Since most cards now support public key crypto, SDA is little used today.

Static Data Authentication (SDA)

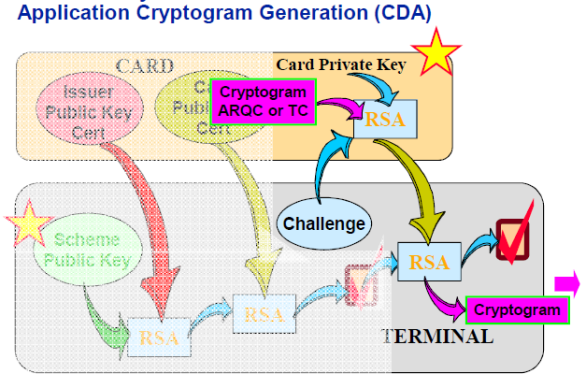




Dynamic Data Authentication (DDA)



Combined Dynamic Data Authentication and Application Cryptogram Generation (CDA)

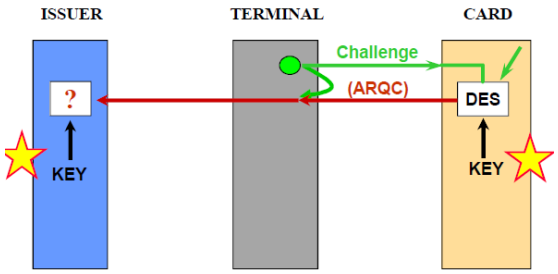


Offline PIN Check

- Plaintext
  - Straightforward - PIN block sent "in clear" in VERIFY command
- Encrypted
  - PIN block encrypted using card Public key and sent in VERIFY command
  - Card decrypts with private key and verifies
  - Key can be for PIN encryption only with additional certificate chain

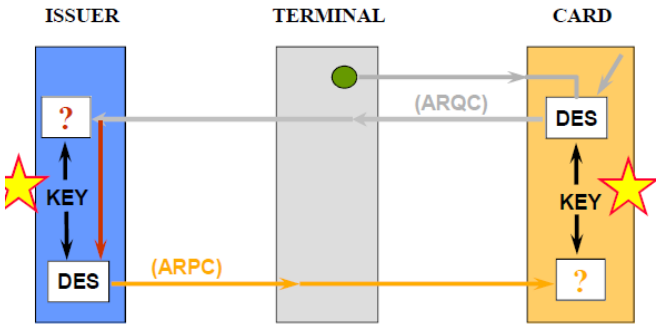
Card Authentication

- Operates online
- Demonstrates to Issuer that card contains genuine key loaded during personalisation



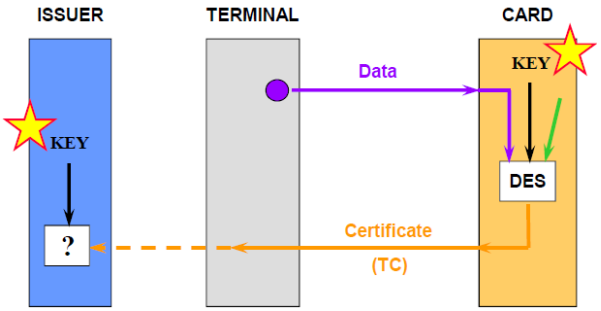
Issuer Authentication

- Operates online
- Demonstrates to the card that the connection is with the genuine Issuer who personalised the card



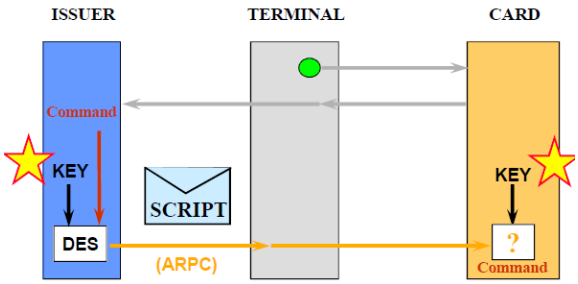
Transaction Certificate

- Performed on complete transaction
- Sent with Data Capture record



Script Processing

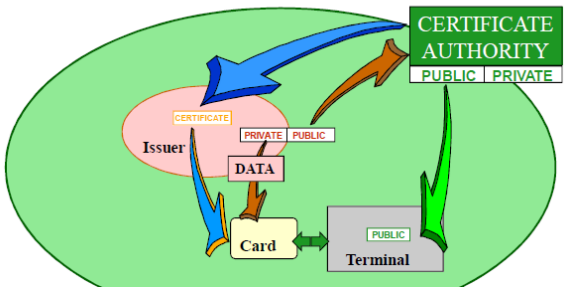
- Performed after transaction completed
- Possible script functions
  - PIN change
  - PIN unblocking
  - Update of card data elements
  - Application blocking/unblocking
  - Card blocking
- Transmitted using Secure Messaging



Key Management

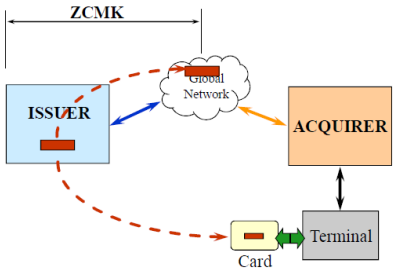
- Two classes of Keys
  - RSA keys (public)
  - DES keys (secret)

RSA Keys



- Payment System Public Keys
  - Several lengths (768, 896, 1024, 1152, 1408 &1984)
  - Distributed globally
  - Longer key lengths required over time
- Issuer Keys
  - Signed by Payment System Certification Authority
  - Card Data and Certificates signed by Issuer Private Key
- Card Keys
  - Possible key pair per card (DDA/CDA)

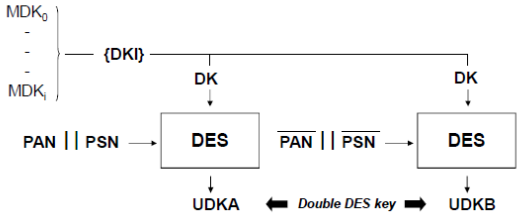
DES Keys



- Keys must be double length
- Unique DES keys per card
  - Usually derived (from PAN)
  - Different for cryptograms and secure messaging
- If required for STIP or dispute resolution, exchange with Payment System as per today's issuer working keys

Double length for 3DES

Derived & Session Keys



EMV recommends the use of card session keys.
Typically the Application Transaction Counter (ATC) is used with the UDK for a new session key per transaction. Security counters limit key abuse.

Script Keys

- Script processing requires two further keys
  - MAC key for secure messaging
  - Encryption key for data content – required for PIN change command

Key Personalisation

- Each card needs its keys personalised securely during production – maybe 1,000s of cards per hour.
- DES keys are quick and derivation process is suited to localised HSMs in production bureau.
- RSA keys are not "derivable". Each needs two randomly generated prime numbers that have to be checked for primality.
- Can be pre-produced by issuer (secure transfer required) or created locally (needs processing resource). Some cards can create their own key pair - code only used once.

What Now?

A Next Generation of EMV:

- Take advantage of the increase in card capability to shift issuer functions out of the terminal and into the card.
- Update offline cryptography to ECC with key sizes that are likely good for all time
- Use a Secure Channel to provide confidentiality and protect against wedge/shim attacks.
- Provide support for payment related services such as ticketing, coupons and loyalty.
- Online cryptograms have little need to change, but are expected to migrate to AES.

When:

- Start in next 1-2 years allowing long migration before RSA keys are likely to run out of steam ~ 2030.

CNP = Card Not Present

- Most of what has been discussed up to now assumes that the Card (and Cardholder) are physically present at a Merchant/Acquirer terminal
- Another class of transactions exist where such an assumption is not valid – such transactions are termed Cardholder Not Present (CNP transactions)
- CNP includes in particular e-Commerce transactions, but also Mail Order/Telephone Order (MO/TO)

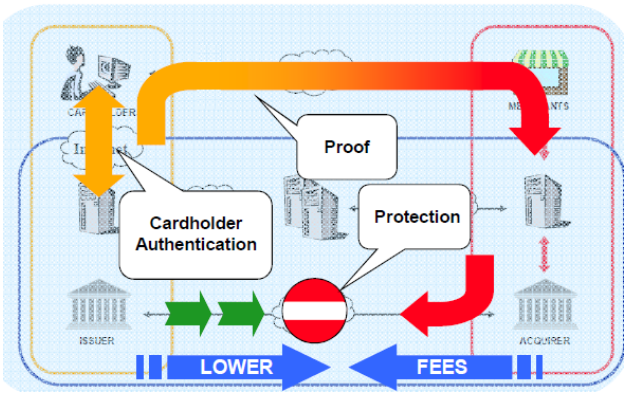
What is the difference?

- Under the rules, in general, merchants and Acquirers get more protection in Cardholder Present transactions than they do in CNP transactions
- More protection ⇨ less risk ⇨ lower costs
- It is also the case that, less protection ⇨ more risk ⇨ higher costs
- "Chargebacks" are the main cause of higher costs
- These are provisions that allow an Issuer to reverse a transaction and send it back to the acquirer/merchant if certain rules were not followed.
- Most occur when cardholders dispute an item on their bill.

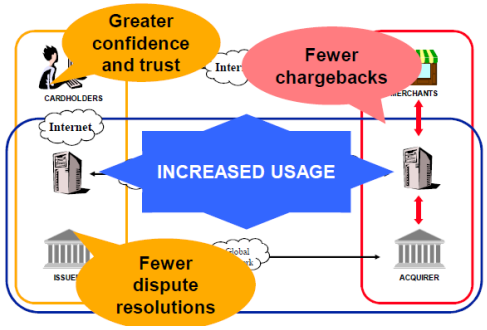
3D-Secure

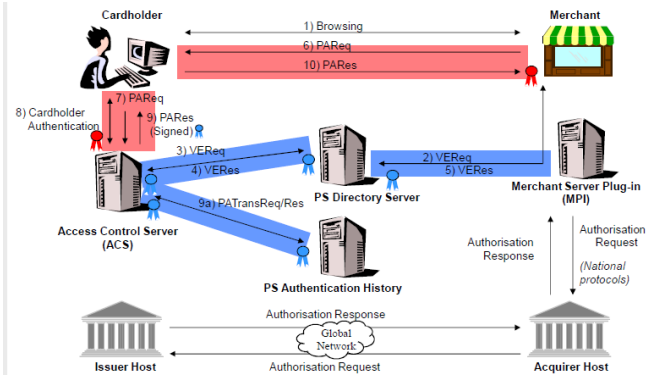
- **3-D Secure** as an EMVCo specification came about in 2001 as a protocol to improve the security of Internet transactions and allow the payment systems to re-balance the rules.
- **3-D Secure** version 2 was published by EMVCo at the end of 2017. This aims to improve the cardholder experience by supporting "frictionless" transactions where the cardholders do not have to actively authenticate themselves.

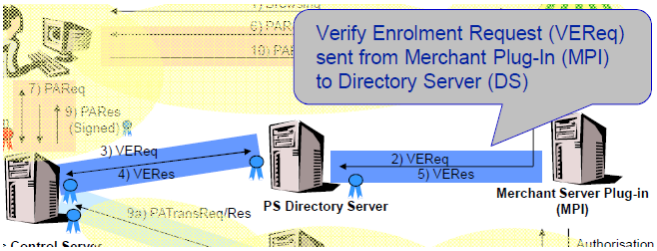
Objective - *basics*

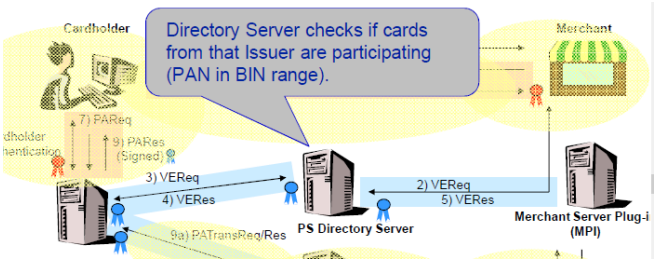


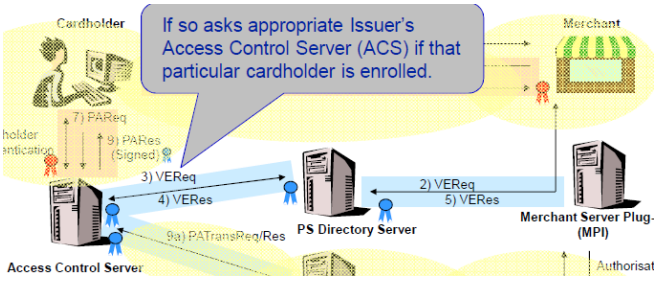
Objective - *benefits*

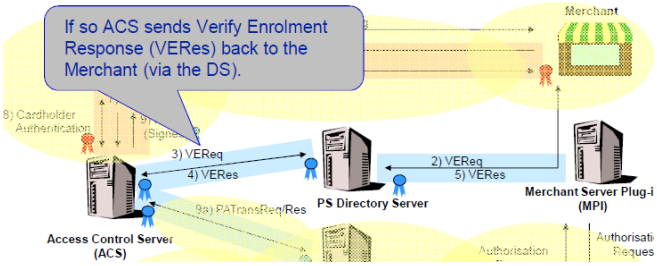




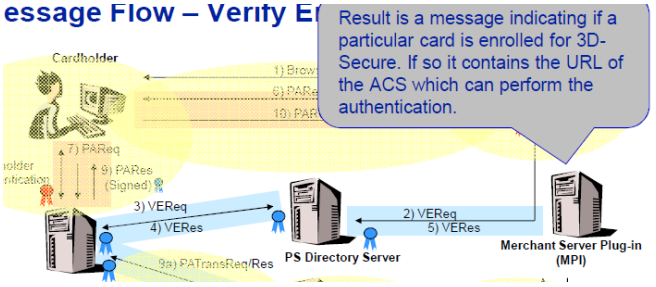


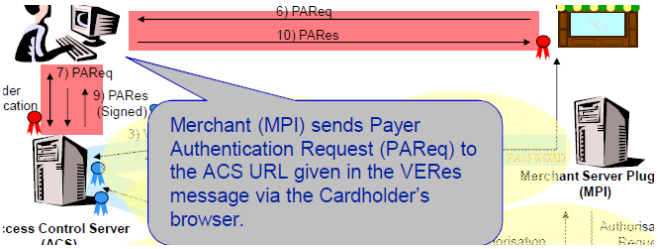


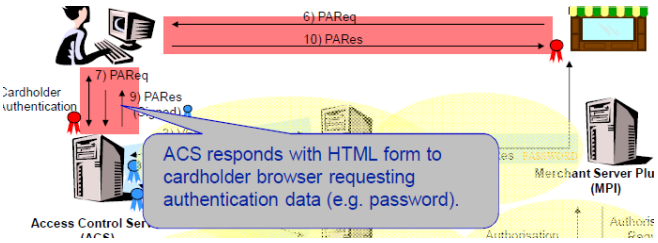


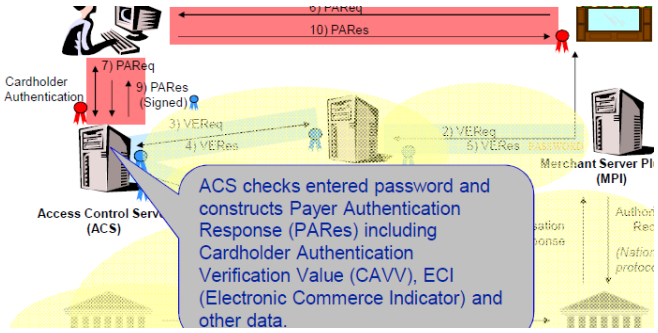


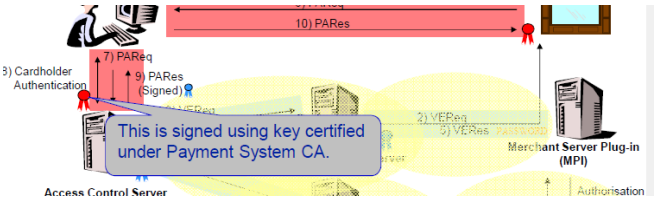
essage Flow – Verify E

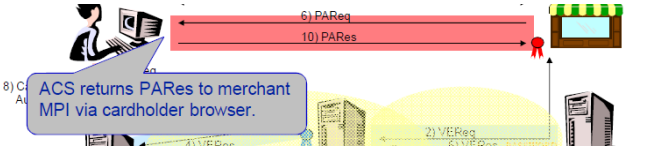


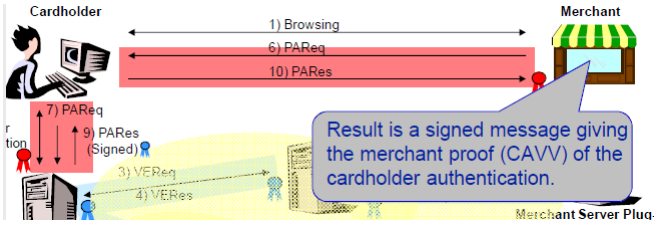












Summary of Evidence

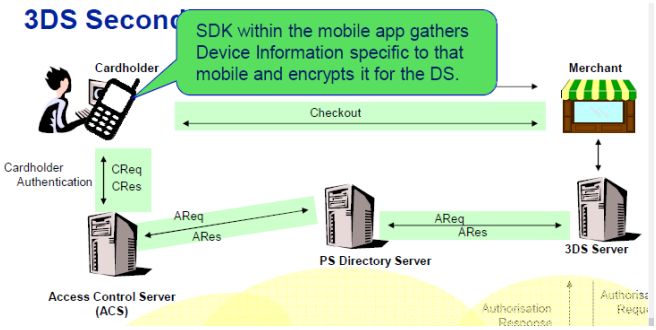
- Issuer – obtains cardholder authentication data from cardholder (password, token, etc.).
- Issuer – submits copy of the PARes (and other data) to the authentication history server (step 9a).
- Merchant – obtains a signed message (PARes) from issuer which dictates the action to be undertaken (proceed or not with transaction).
- Merchant – submits transaction for authorisation with the knowledge that the issuer has accepted the liability of this transaction with regards to cardholder identification. Once authorised, there is a liability shift from acquirer/merchant to issuer (similar to cardholder present situation).

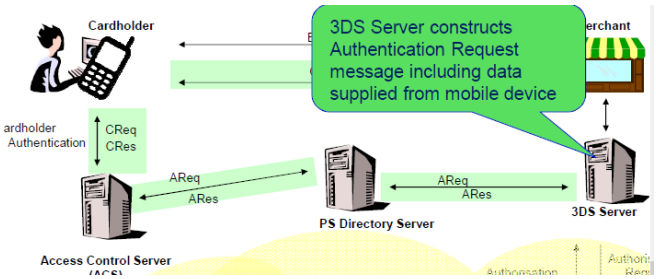
A Version 2 of 3-D Secure:

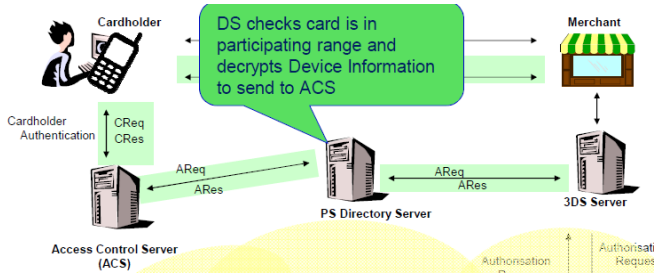
- Allows for new format devices – in particular, smart phones.
- Provides more device and environmental information, allowing the ACS to authenticate transactions without needing the cardholder to actually enter a password or passcode.

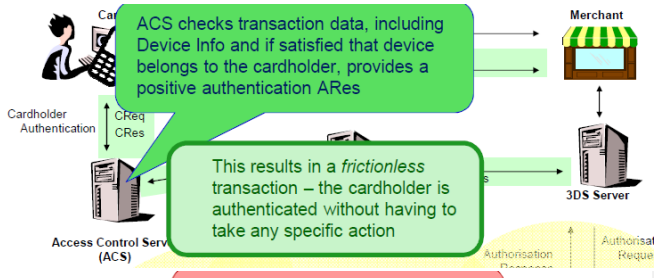
When:

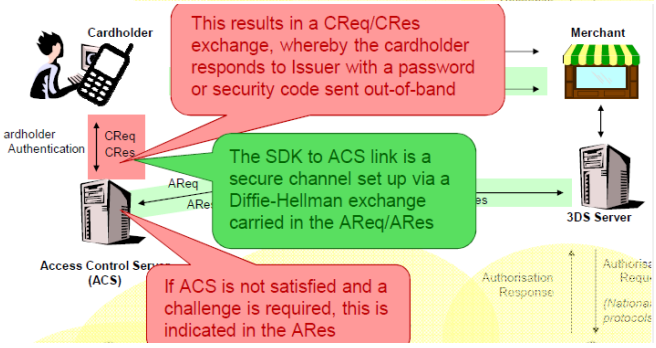
- Draft specifications published late 2016.
- 3-D Secure Version 2 published late 2017.











Contactless

Cardholder message:

- "Wave and Pay for transactions below ~~£20~~ £30 where you see the contactless indicator . . ."
- ". . . from time to time you may be asked to revert to contact Chip and PIN for your security"

Behind-the-scenes:

- Offline DDA/CDA style authentication
- 'pre-authorised' offline contactless counter in chip
- Reset every time cardholder performs a chip and PIN transaction
- If counter runs down, chip forces contact transaction
- Issuer risk limited to counter maximum

In conjunction with transport . . . 2

Credit/Debit

Simply accepts contactless payment card:

- Badge in/out each trip.
- Back-office system sorts it out and submits one transaction at the end of the day.
- Fares capped at price of "travel card".

Indicators and Specs

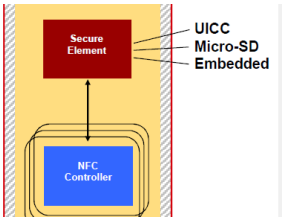
Contactless indicator on cards

Point of Sale indicator agreed between Visa, MasterCard and American Express

EMV **Contactless Communication Protocol** and **Entry Point** specs allow common acceptance points

Going Mobile

- Mobile use based on Near Field Communication (NFC) technology.
- Initially allows handsets to work like contactless cards
- EMVCo agreed User Interface will allow common configuration of application choice and priority



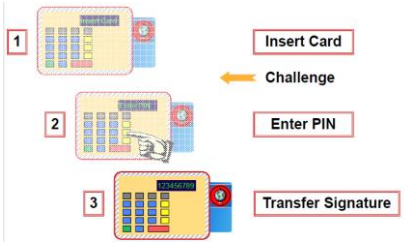
More Mobile

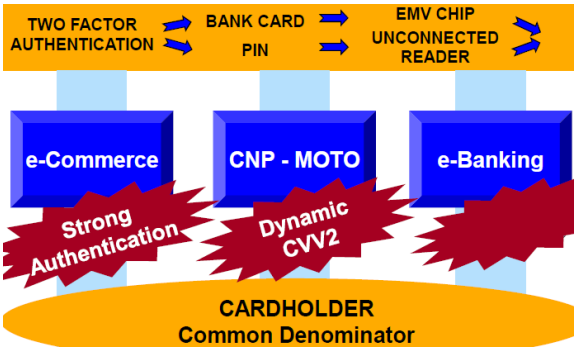
- Mobile handsets as mobile wallets
  - User interface allows consumer to control and select from a range of payment applications
  - Multiple brands supported in same wallet
  - Facilitates web purchasing using the mobile
- Mobile POS
  - "Add-on" readers & PIN pads allow small merchants to use their mobiles for card transactions
  - Network connection is mobile data link
  - User has contract with "agent" that acts as link to acquirer
- Mobile Apps for e-commerce
  - Users download apps for favourite merchants. Payment options and authentication largely happen transparently

Passcode Authentication (otherwise known as Token Authentication or Token Based Authentication)

- Two factor authentication
  - Something you own – EMV chip card
  - Something you know – PIN
  - Stronger than static passwords
  - Good against phishing
- Can include other features
  - Use of random challenge
  - Input of amount and currency
  - Signing of external data
- Cardholder uses banking/payment card and small unconnected handheld reader to produce a "token" or "signature".
- Based on EMV cryptogram.
- One-Time Passcode - different for each transaction.

  OR
- Challenge/Response – includes random challenge.

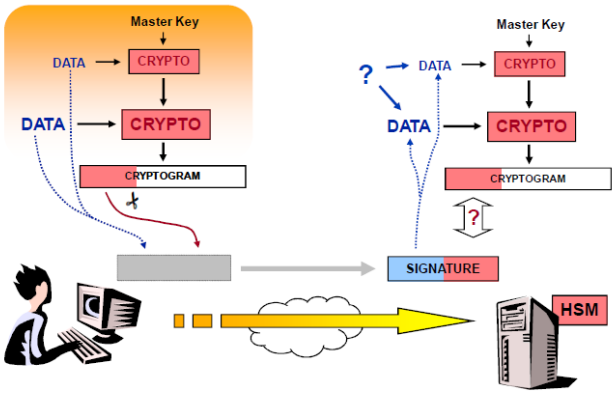




Objectives

- One solution applicable to all three "spaces"
- Customer friendly
  - Common cardholder experience
  - Ubiquitous readers
- Minimal impact
  - To verification services - including ACS, host and stand-in
  - To card stocks - re-use existing implementations by personalisation of additional co-existent authentication application

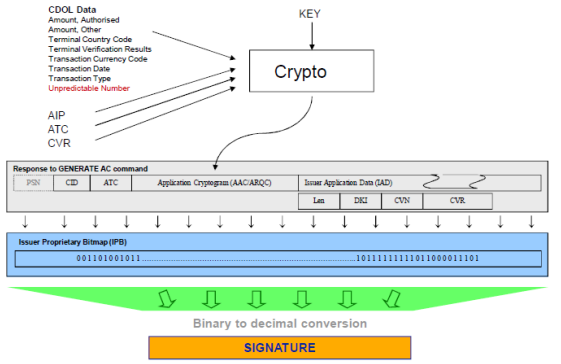
Data Transfer



Data Issues

- Data not known by the verification service needs to be transferred in the signature.
- Alternatively the verification service may be able to extrapolate from other information or try multiple values.
- Issuer systems typically hold databases with per card information, but this is generally not the case for stand-in services.
- Using the same card across several services (e.g. e-commerce and e-banking) can give rise to some interesting system architecture issues.

Signature Construction



Standard full function credit/debit card with on the back ……
a built-in PIN pad and display for authentication

Global Payment Systems – use modern technology and cryptography …

… that is fit for purpose …

… balances fraud v business …

SIM Cards

Why Use a Smartcard in Telecoms?

- The SIM is a smart card issued by a Mobile Network Operator to grant access to service.
- The USIM is an evolution of SIM, aligned with the ISO 7816 specifications.
- The SIM or USIM is plugged into a handset as required by the standards, and provides:
  - Identification and authentication of the user account on the network
  - Securing the communication over the wireless network.
  - Storing user information
  - Enabling operators to provide customised services for users

Why Use a Standardised SIM/USIM?

Secure

In the users hand

Managed

- The (U)SIM needs to be highly interoperable with a large range of handsets (Mobile Phones).
- The handsets need to highly interoperable with any type of SIM and know how to use the SIM files and procedures to configure the user's service.
- The Mobile Operator may not be allowed to restrict the types of handsets connected to its network.

The SIM/USIM and Security Protection

- Analogue networks (no SIM) were experiencing up to 15% cloning fraud (£ Multi-Millions).
- They also allowed calls to be eavesdropped.
- In 1988, C-Net system introduced first ever Smartcard for Telecoms use.
- In 1990, ETSI defined a tamper-resistant Smartcard (SIM) to identify the user and services in GSM (2G system). Since then cloning fraud has been negligible in most networks.
- The SIM also generates cipher keys used in GSM to protect privacy of communication.
- In 1999, 3GPP defined the USIM for 3G systems that improved on GSM capabilities.

Future = maybe eSIM

Original = ETSI, pass to 3GPP, all standards free

Standardised Features - Other Features

- Roaming list
- SMSC number
- SMS Store
- Last Dialled numbers
- Access Control Class
- GPRS Authentication and encryption files

see GSM 11.11 and TS 31 102 for a full list

SMSC number = sms center

The SIM now exist in 5 Form factors that reflect the way it is used:

ID-1 or half ID-1

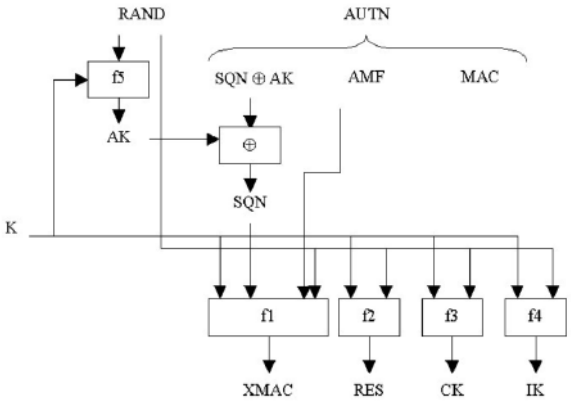
Plug-in UICC (mini SIM/2FF) the most common form factor

mini-UICC (micro SIM/3FF)

nano SIM/4FF

MFF1 and **MFF2**
MFF1 connector, MFF2 solder

ırdised Features – 3G Auth



MAC signed by your key
SQN prevents replay
CK = cipher key
RES = result
IK = integrity key
AK = anonymity key

GSM(2G) + 3G = only framework, no standard algorithm of encryption

3G gives an example (Default) algo, so everyone uses, 2G don't have

2G v 3G Authentication Comparison

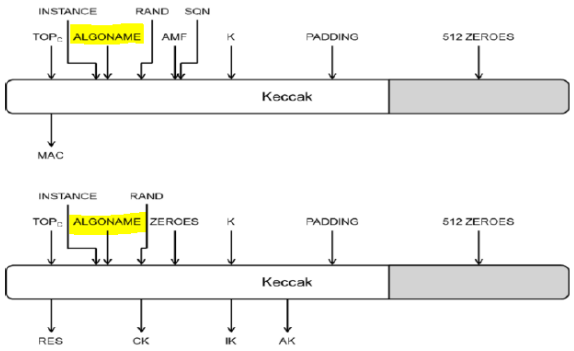
GSM			UMTS		
Description	bits	alg	Description	bits	alg
Ki Subscriber authentication key	128		K Subscriber authentication key	128	
RAND random challenge	128		RAND random challenge	128	
XRES expected result	32	A3	XRES expected result	32-128	f2
Kc cipher key	64max	A8	CK cipher key	128	f3
			IK integrity key	128	f4
			AK anonymity key	48	f5
			SQN sequence number	48	
			AMF	16	
			MAC	64	f1
			AUTN	128	
Example algorithm COMP128-1			Example algorithm Milenage		

Kc might not even have 64 bit, might use parity

1 algo good = widely tested, interoperability, switch between networks easily.
1 algo not good = everyone uses same, if compromised, everyone suffers.

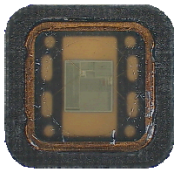
So u want equally good, but can customize.

New Algorithm - TUAK



Is the (U)SIM Really Attack-Resistant?

- Tamper resistance is always a question of hardware and software as well as a question of time (and money)
- In banking cards, security and attack resistance is evaluated to common criteria standards, whereas (U)SIMs are often tested less formally.
- Security systems are designed for a finite lifetime
  - A chip design of the 80s (1.5 µm) of the last century is prone to electron microscope attacks
  - A software implementation of the 90s will not necessarily protect algorithm and keys against being broken these days



- Some known attacks on legacy solutions
  - The attack on COMP 128-1, the authentication algorithm
  - The attack against the SIM OTA security of some implementations of 2013

Comp128-1 is an example algo, but not part of standards in GSM (ppl are still using it)

- 1998 Comp 128-1 (A3/A8) successfully attacked
  - Black box attack against the GSM-MoU example algorithm
    - Chosen plaintext-ciphertext attack
    - Attack against the algorithm of just one card, not against the system itself
  - Does not utilise any hardware or software property of the SIM
  - Some major operators did not use Comp128-1 from outset

2013 OTA attack in blackhat las vegas

- *What was claimed…*
  - *… This talk ends this myth of unbreakable SIM cards … and illustrates that the cards - like any other computing system – are plagued by implementation and configuration bugs."*
- Or, as mentioned elsewhere:
  - *"Note, in most cases this is not a 'bug' causing the problem. It is poor security choices being made by network operators."*

The 2013 OTA Attack – Aims/Processes

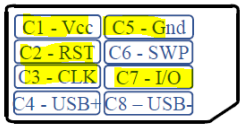
- The aim was to break the OTA security of the SIM to be able to download malicious applications to a Java SIM, then maybe
  - trace the device and the user,
  - send SMSs to premium numbers,
  - divert incoming and outgoing calls,
  - silently include third parties on a call and
  - ..and/or probe for platform weaknesses
- The attacker sends a rogue binary SMS to the phone saying "here is an OTA application for you". The binary SMS includes a signature field, and the attacker doesn't know the OTA key at this point so the signature will be incorrect.
- A UICC reacts in one of four ways (the attacker needs (d) to extract the key);
  a. it may reject the incoming message and not send a response;
  - the way recommended by the 3GPP specification since 2012;
  b. it may send an unsigned response saying "your signature was invalid";
  c. it may send what looks like a signed response saying "your signature was invalid", but with all zeroes in the signature field;
  d. it may send a genuinely signed response saying "your signature was invalid"
  - discarded in the relevant UICC specifications in late 2013.

The Attack - Prerequisites

- The SIM must uses Single DES for OTA security
  - Single DES was deprecated in ETSI specifications since 2008
- The SIM must respond to a fake download request with a specific MAC containing known data
  - Not a recommended response mode since 2012
- The network allows the forwarding of binary messages
  - Blocked in (probably most) networks
  - Possession of the card or a (cheap) fake base station are alternatives
- The OTA implementation does not use a different key for the enciphering of the OTA application or does not encipher it at all
  - Either approach would be bad practice

…If you could find a scenario where all these requirements are met, it would be because of bad choices/practice by the MNO rather than any bug or fault with the SIM platform…

Start-up and File Access Physical / Electrical Connection



Handset is Master UICC/SIM is Slave	Handset sets the supply voltage used	Handset knows UICC or SIM from ATR
Most handsets do not connect to C6, C4 or C8	SWP interface is used for contactless transactions	The handset can stop the CLK at any time to save power

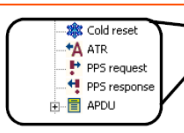
Start-up and file access: SIM

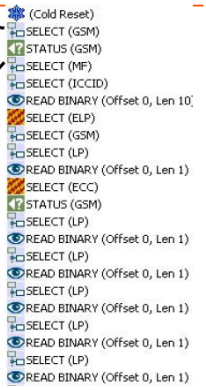
Flow described in GSM 11.11 ch. 11.2.1 SIM initialisation.

Class byte = 'A0'

Direct selection of files from the MF (Master File).

RunGSM command used to authenticate to the network.





Command APDU						
Header (required)				Body (optional)		
CLA	INS	P1	P2	Lc	Data Field	Le

ICCID = unique hardware id for the chip

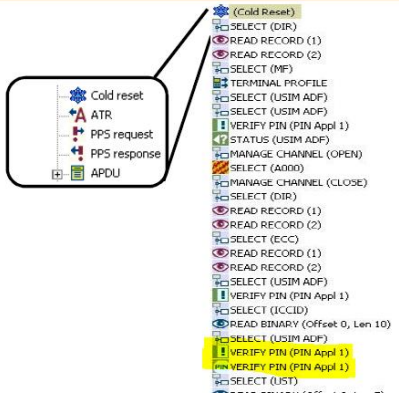
Start-up and file access: USIM/UICC

Flow described in 3GPP TS 31.102 chapter 11.2.1 SIM initialisation.

Class byte = '00'

Indirect selection of files from the MF (by selecting ADF USIM).

Authenticate command used to authenticate to the network.



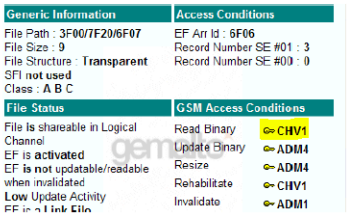
GSM no SQN, random

CLA	INS	P1	P2	Lc	Data
Sent to the SIM card					
Data (optional)		SW1	SW2		
Sent from the SIM card					

- ✓ RUNGSM APDU CMD: A0 88 00 00 10 <RAND>
- ✓ RUNGSM APDU RSP: 11 6D 12 9F FF 2F 5A 9E 07 00 0D FE
  - 4 bytes of SRES (11 6D 12 9F)
  - 8 bytes of Kc (FF 2F 5A 9E 07 00 0D FE)
  - 2 bytes of SW (90 00)

APDUs, PINs & File Types

- PINs are used to secure file access in the (U)SIM.
- Each Pin can have a PUK (see 3GPP TS 11.11).
- The USIM has 'Global' and 'Local' PINs.
- The SIM only has 'Local' PINs.
- Keys are used to encrypt and verify secure messages (see 3GPP TS 03.48) and are typically used for 'Over the Air' updates of files and applications..



CHV = Card Holder Verification

File Types

Linear Fixed	Transparent	Cyclic
•Many records, all are the same length.	•A single block of data.	•Many records, all are the same length.
•READ RECORD command reads.	•READ BINARY command reads.	•READ RECORD command reads.
•UPDATE RECORD command writes.	•UPDATE BINARY command writes.	•UPDATE RECORD command writes.
•Can only be resized by deleting file and creating a new one correctly sized.	•Can be read / written from any offset (P1,P2)	•Can only be resized by deleting file and creating a new one correctly sized.
•Supports absolute record, relative record and SEEK.	•Can only be resized by deleting file and creating a new one correctly sized.	•Supports relative record only.
•Last Record does not wrap to first record.	•Used for most files	•Last Record wraps to first record.
•Used mainly by the Phonebook.		•Used for Last Number Dialled

	SIM	UICC / (U)SIM
Class Byte used……	Class = "A0"	Class = "00"
Root & App Directory……	MF (3F00) 7F20	MF (3F00) ADF (7FFF)
Support Multiple Channels……	No	Yes
Authentication Command…….	RunGSM	Authenticate
Can be used for GSM access…..	Yes	Yes
Can be used for 3G access….....	Yes	Yes
Support SIM Toolkit……..	Yes	Yes
Specified in releases……..	Ph.1 to Rel.4	Rel.99 to Rel.7
Standards development……...	Frozen	On-going

SIM Value Added Services

- The (U)SIM adds:-
  - Secure storage.
  - Mobility between devices.
  - User Identification.
  - An environment to add applications.
- The two main ways this was achieved were:
  - SIM Toolkit (3GPP TS 31.111)
  - JSR 177

SIM Toolkit

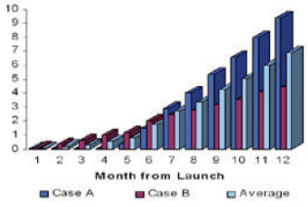
- Applications run on the (U)SIM and give the handset instructions to carryout simple tasks.

User Interface	Network interface	Handset Interface	Miscellaneous
•DISPLAY TEXT •GET INPUT •SELECT ITEM •Display Idle Mode Text •GET INKEY	•SETUP CALL •SEND SHORT MESSAGE •SEND USSD •ENVELOPE(SMS-PP Download)	•PROVIDE LOCAL INFORMATION •POLLING INTERVAL •POLLING OFF •Timers •MORE TIME	•TERMINAL PROFILE • CALL CONTROL

USSD = UDP version of SMS

Call Control = can use to divert calls

- Typically used to deliver special handset menus, banking services and new handset detection.
- Mainly 'text' based – this can be an issue on high end phones.
- However they generate lots of revenue!
- Example Vodafone DateTrak Services
- 10% of users tried it
- Users spent $25-30 each



DateTrak = dating service (tinder)

SIM Toolkit:Advanced Features

Event Triggering	Triggers SIM applications based on events such as:- incoming call, call dropped, Location update, Handset changed.
Bearer Independent Protocol	Allows the SIM to use new types of bearer such as:- Packet (GPRS, 3G), WLAN, IrDA, Bluetooth, USB
Call Control	Allows complex redirection and barring services for calls and SMS.
Launch Browser	Allows SIM Toolkit Launch a WAP browser to a specific address using specific parameters.

Typical SIM Toolkit App

- Handset Trigger (new phone trigger)
- Network Selection
- Handset Locking (lock sim to handset)
- Global Roaming
- Phonebook Backup
- Information Services (sms)
- Digital Rights Mgmt (lock mobile apps to sim)

SIM Toolkit Implemented on Java SIM

What is Java SIM?

- *Java Card with pre-installed SIM applet and SIM-related APIs*
- *Same functionally as native GSM SIM*
- *SIM Toolkit applications exist as Java Card applets*

Why do Mobile Network Operators need it?

- *Dual/multiple sourcing*
- *Open platforms*
- *Costs reduced - commodity products*
  - *New entrants encourage competition*
  - *Opens up 3rd party development opportunity*
- *Java is oriented about security!*

The Java Card Runtime Environment (JCRE)

- *Comprises Java Card VM and classes within the Java Card Framework*
- *Each applet on the card has unique AID assigned*
- *An applet becomes active when a SELECT (AID) APDU is received*
- *The applet's "select()" method called at that time*
- *All future APDU commands are sent to this now active applet*
  - *The "process()" method called each time an APDU is received*
  - *The applet can access and process the command APDU then return a status*
- *If a SELECT (AID) APDU is received for a **different** applet, the current applet is suspended*
- *The applet's "deselect()" method is called at that time*

The Java SIM framework has two extra packages:

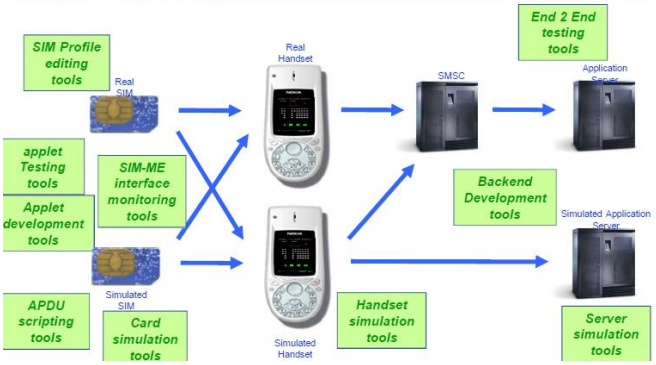
- ***sim.toolkit***
  - *Core package – manages SIM toolkit-related activity*
  - *ProactiveHandler, ProactiveResponseHandler, ToolkitRegistry, EnvelopeHandler, EditHandler, ViewHandler classes*
- ***sim.access***
  - *Access to the GSM SIM file system*
  - *SIMSystem and SIMView classes*
  - *Methods for reading and writing to individual files*
  - *Does not compromise integrity*

JSR 177 – SATSA (Security And Trust Services API)

- A set of Java methods (classes) that allow Java MIDlets on the handset to call routines on the (U)SIM.
- Not well supported by handsets.
- Possibility of being Media rich.
- (U)SIM routines can be called by name or by issuing APDUs.
- Typically use the (U)SIM for cryptographic functions such as PKI Signing.

Testing:

Use of Tools



Improved Technologies

- Higher memories, Higher speeds.
- Integration with PC's (USB interface).
- Integration with RFID/NFC.
- Lower Voltage classes (1.2 Volt).
- Addition of contactless interface.

...But, just because something is standardised does not mean it is supported by all phones…

Emerging/Growing Markets

- Chinese and Asia markets still growing fast.
- 3GPP2 (America) has specified a USIM like entity called the (R)UIM.
- Use of (U)SIMs in PC's to allow billing & authentication using existing Telecoms billing systems.

...But, developments around embedded SIM could mean a decline in traditional SIM cards …

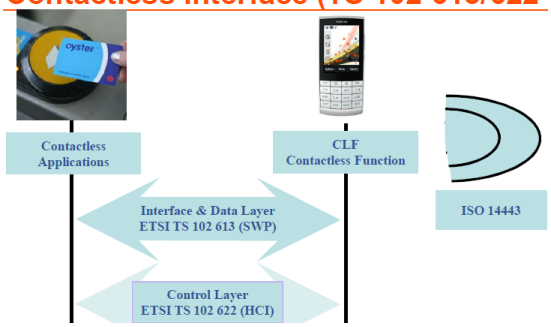
New UICC Features (ETSI TS 102.412)

High Speed Interface	This adds a 8Mb USB channel to eventually replace the ISO interface. Will have IP layer.
Contactless Interface	Allows the UICC to act as a contactless smartcard via the terminal.
Secure UICC – Terminal Interface	Secures the terminal to UICC interface so secure transactions can take place such as DRM.
Confidential Applications	Allows the creation of firewalled third party areas. These third party areas can be used and managed safely by third parties
Web Server on a card	Specifies how the SIM card can act as a WAP server.

USB Interface

USB IC interface (Pins C1, C5, C4 & C8)			
Smartcard Class	**Ethernet Class**	**Mass Storage Class**	**Other USB Classes**
•Used for APDU access. •Mirrors the ISO interface. •Allows all current specifications to be re-used.	•Allows direct IP communication with the UICC. •Will allow us to re-use internet protocols and mechanisms. •Smartcard Webserver. •sFTP.	•Allows high memory to be integrated easily. •Richer configuration enabled. •Transfer of content and rights between handsets.	•Many other USB classes may be relevant but have not been investigated yet.

Contactless Interface (TS 102 613/622



Secure Channel

- Applications run on the (U)SIM and give the handset instructions to carryout simple tasks.

APDU application to application	APDU platform to platform	Application to Application IP (TLS)	Platform to Platform IP (IPsec)
•Specific APDU communication between applets on the handset and the UICC secured.	•Whole APDU interface secured	•Specific IP communication between applets on the handset and the UICC secured.	•Whole IP interface secured

User Interface (MMI)

- challenges
  - different screen resolutions
  - different implementations of JVMs or web browsers
  - different interpretation of APIs
  - different start-up time and performance
- options
  - SIM Tool Kit (STK)
  - Smart Card Web Server (SCWS)
  - Java 2 Platform Micro-Edition (J2ME)

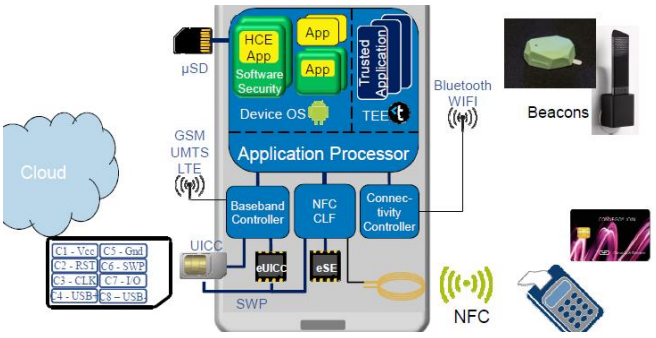
SIM Tool Kit

- pros
  - proven and tested technology
  - interoperable between different mobile terminals
  - OTA provision of user interface
  - terminal-independent as Java Card Applet and UI in SIM
- cons
  - only text based user interface
  - user experience is different between mobile terminals
  - patchy phone support for advanced features

Smart Card Web Server

- pros
  - uses built-in browser of mobile device to render data
  - for simple web server content quite high interoperability
  - OTA provision of user interface
- cons
  - new technology in UICC
  - difficult to define the exact look and feel of the interface
  - different versions of web pages for different browsers and screen
  - not widely supported by phones

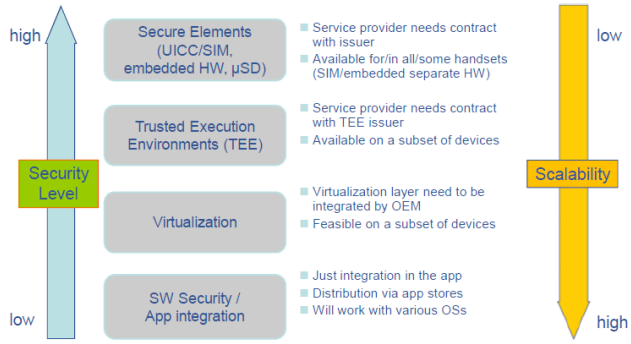
Smart Phone



Managing Trust - the Core Elements

- HW Security Elements
  - The UICC (providing the home for the Subscriber Identity Module, the SIM)
    - Dedicated piece of HW provided by the MNO
    - Used for authentication of the subscription and for providing the ciphering key for the enciphering of the radio interface
    - Home to (security requiring) applications and keys
    - Specified by ETSI (and 3GPP)
  - Embedded Secure Element (eSE)
    - Similar to the UICC provided by the OEM in its device
    - Providing security for a variety of services
- Trusted Execution Environment
  - Special SW in the "Trustzone" of the controller
    - APIs, including Admin API, specified by GlobalPlatform
    - Proprietary as well as open source TEE solutions
- Host Card Emulation (HCE)
  - Routing mechanism by Google to enable the development of NFC applications in SW
    - Example: emulation of a payment smart card in the device SW or the cloud
- Software Security
  - SW protection at the application level
  - Code obfuscation, White Box Crypto, Device Fingerprinting

Scalability v Execution Environment



Personalisation of an eUICC (SIM)

- Today: HW, SW and security data out of one and the same source
  - The SIM manufacturers develops OS including security optimised algorithms
  - The SIM manufacturer generates personalisation data (serial numbers, keys, MNO credentials, …), loads them into the chip in its premises, together with an MNO specific profile, and sends SIMs and data to the MNO
- Tomorrow: Split system
  - OEMs (device vendors) are provided with chips containing OS and algorithms (of the SIM manufacturer)
  - Subscription Manager (SIM manufacturer) generates personalisation data (serial numbers, keys, MNO credentials. …) and sends these data to MNO and OEM
  - OEM loads data on its premises in a secure way into the UICCs

Two questions:

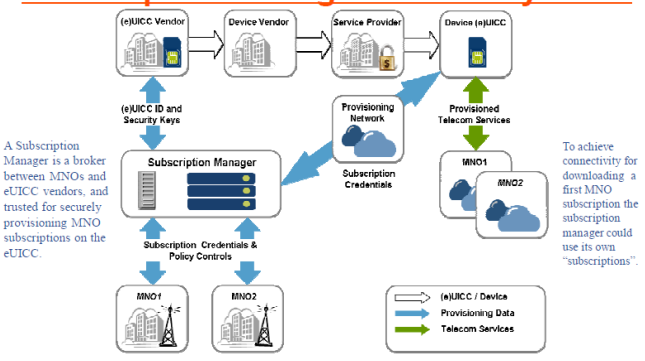
- How to handle the case that the device is not specifically produced for a specific MNO (in particular in the case of M2M devices)?
- How to change the subscription of a device in the field?

Two New Scenarios

Needed: Provisioning of subscriptions over-the-air or over-the-wire
after production, outside of factory

Needed: New ecosystem with **dynamic subscription management**
(provisioning and changing of subscriptions and profiles)

Subscription Management Ecosystem



Mobile Connect Proposal

GSMA Mobile Identity program - a new approach by the GSMA

- MNOs hold the world's largest reservoir of IDs
  - The subscribers are verified by official documents, …
    - in most countries
- MNOs could become trusted identity "brokers" and provide secure services to various end-users
  - Accessing personal data securely on portals
    - Mobile identity storage
      health records, secure cloud, loyalty programs, …
  - Banking and financial services
  - Enhanced access to eGovernment services
    - Pensions, social security payments
    - Birth/life events registration
  - Signing documents on the go ready for archiving to meet regulatory compliance
  - Proving one's age online
  - Mobile voting
  - Unlocking secure premises

Attacks on Smart Cards

- Main types of attacks
- Overview of issues on systems utilizing smart card tech
- Rumours/hypes about smart cards security
    - Is it weakest link?
- Importance of overall system design
- Really have to break the card?
- Examine a number of use cases

Given the resources, time and money, all can break

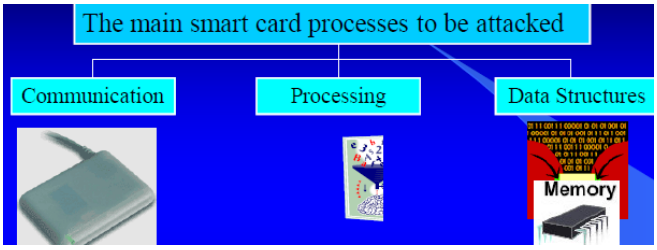
Security goals = absence of risk
real world infosec goals = provide relying parties with rational confidence that certain undesirable outcomes are unlikely
Smart card security goal = controlling the risks
Smart card security = maintaining CIA

- Smart card hardware tamper resistance
- Smart card OS logical security
- Org and overall system security

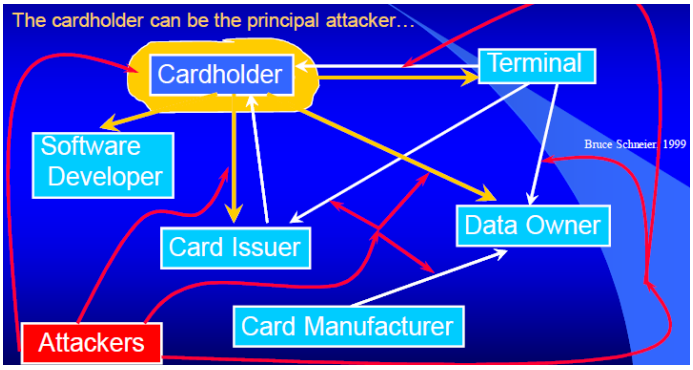
Aim = Prevent, Detect , Recover

Smart cards = single element in system

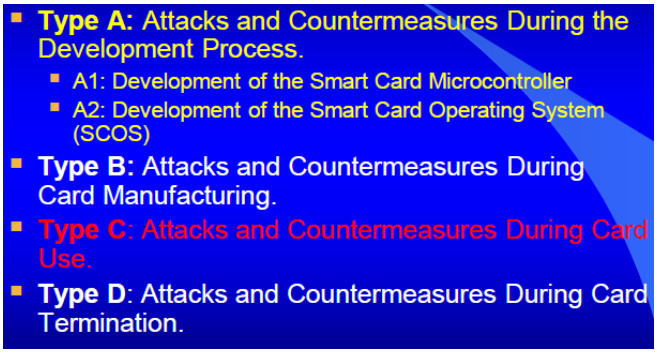
Attack points



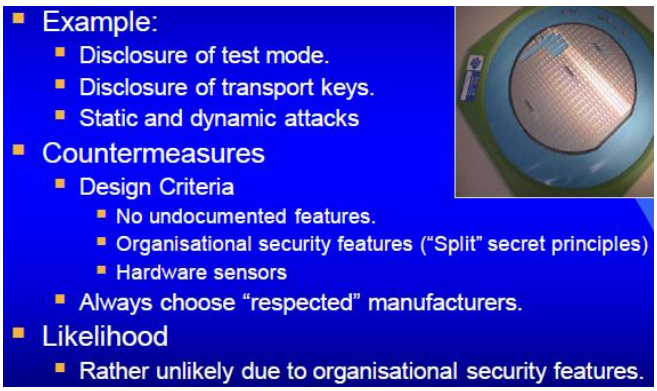
Attacker's goal = violate any design assumption and mount an attack in any above steps



Smart Card Threats

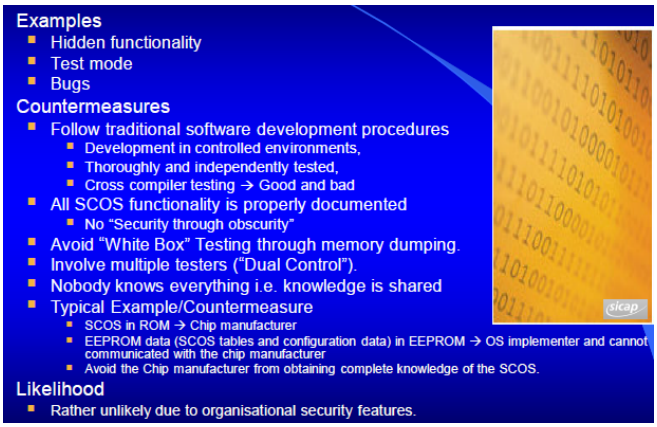


A1: Modify the hardware of smartcard microcontroller during manufacturing



- Add hidden chips

A2: Attacks during the dev of Smart Card OS



B: Attacks and countermeasures During Card Manufacturing

D: Attacks and Countermeasure during Card Termination

Threat

- Re-enable card usage after card termination.
  - Un-block cards
  - Change expiration date

Countermeasures and likelihood

- Card manufacturing:
  - Unlikely due to organisational features.
- During card use:
  - Unlikely due to card hardware and software protection mechanisms.

C: Attacks and countermeasure during Card misuse

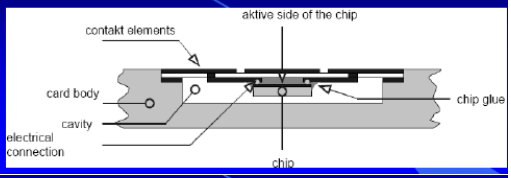
- **Social Attacks**
- Hardware Attacks
  - Physical/Invasive
    - Obtaining Access to the Silicon (i.e. the microprocessor)
      - Chip Probing and Bus Reading
      - Reverse Engineering
      - Chip Rewriting Attacks
      - Drilling
  - Side Channel
    - Timing Attacks
    - Fault Induction Attacks
      - Glitch Attacks
      - Differential Fault Analysis (DFA)
    - Direct Memory Reading
    - Power Analysis
      - Simple Power Analysis (SPA)
      - Differential Power Analysis (DPA)
    - Radiation Electromagnetic Radiation
- Logical Attacks
  - Java card Platform Attacks

Social Attacks

- **Threat**
  - A "classical" one.
  - Obtain information directly from "within".
    - It often "involved"…

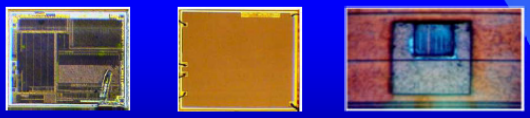
- **Countermeasures:**
  - "Thorough" organisational processes.
  - Split secret principles
- **Significance:**
  - Could be devastating…

Physical Attacks Begin With… Access to the Silicon (Decapsulation)



Countermeasures

- Passivation Layer → prevents oxidation
  - A sensor circuit can determine via resistance or capacity measuring whether this passivation layer is still present.
- Protective Layers i.e. shields
  - Conducting metallic layers, one on top of the other



- Significance
  - Card is destroyed → but could be reassembled?
  - If no global secrets → no big threat.
  - Specialised and often very expensive equipment (Focus Ion Beam, Laser Cutters, etc.)

Light Attack

Optical Fault Induction Attack

- Suggested around 2002 (Anderson & Skorobogatov)
- Strong light source → "unprotected" microprocessor
  - Photoelectric effect leads to unusual-chip-behavior
- Low cost version of the attack
  - Regular flashlight placed in a conventional light micro-scope.
  - Flash a limited area of the RAM of a microcontroller (PIC16F84).
  - Makes possible to selectively set certain bits in the RAM of this microcontroller to the value 0 or 1.

Smart Card Probing (Microscope & needles)

- **Threat**
  - Monitor data buses, for…
    - Keys, sensitive data, PIN numbers, applications
    - Obtain the complete running program
  - Physical access to the microprocessor.
    - E.g. Emission attacks.

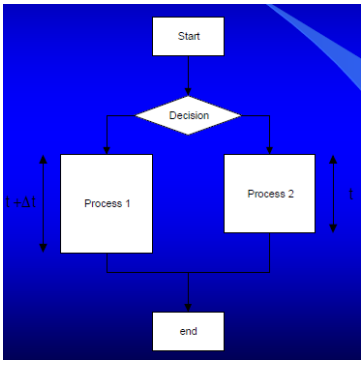
Countermeasures

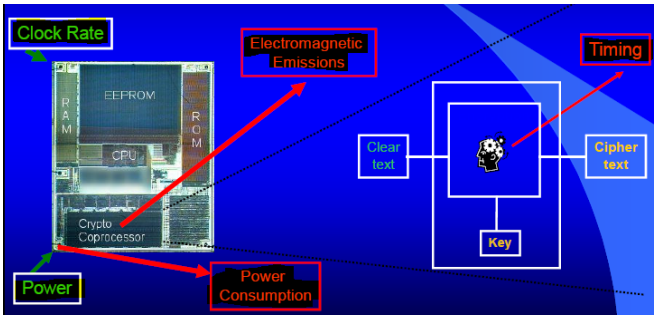
- An active shield → metal mesh where data passes continuously on lines.
  - Upon modification → chip does not operate anymore.
- Obfuscated logic and buried buses in multiple layers
- Encrypted buses
  - The scrambling is carried out by scramblers, which are directly located on the memory.
- Individual chip bus line scrambling is more effective
  - Too expensive

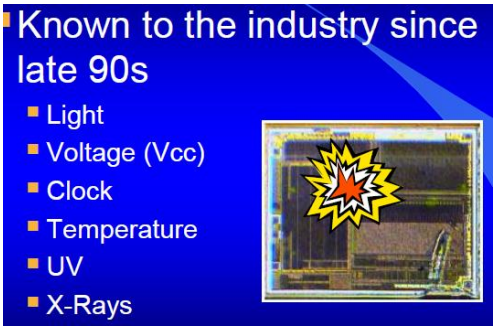
- **Significance**
  - Very difficult to perform for sub-micronics technology smaller than 0.35 am.
  - – Requires chip dismantling
  - – All "respectable" smart card manufacturers offer some of the aforementioned countermeasures

SIDE CHANNEL

Timing Attacks







Fault Attacks

One evening you ask him to drink some Vodka (even better Raki) so that he mistakes a cent for a dollar or the other way around.
Can you tell what is in the vault? (*)
- If vault = 13 grams →
  - (10 = 2 dollars x 5grams) + (3 = as one cent counted as dollar)
- If vault = 17 grams →
  - (12 = 4 cents x 3 grams) + (5 = as one dollar counted as cent)

Glitch Attack

- High or low frequency can induce errors in the processing.
  - "a finely tuned clock glitch is able to completely change a CPU's execution behaviour including the omitting of instructions during the executions of programs".

```
40    ** PIN Verification Example
50    If Retry_Counter = 3 then goto 800
100   Input "Insert PIN"; User_PIN$;
150   If User_PIN$ = Stored_PIN$ then goto 200 else goto 350
200   REM ********** Correct PIN Checking **********
250   Retry_Counter = 0;
300   gosub CONTINUE_PAYMENT;
350   REM ********** Incorect PIN Checking**********
400   Print "Incorect PIN";
450   Retry_Counter=Retry_Counter+2;
500   Goto 40
800   Gosub BLOCK_CARD;
```

Change clock frequency to introduce a logical "jump"

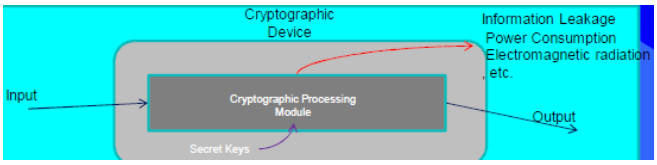
Counter Measures

- Voltage Detectors
- Frequency Detectors
- Physical Integrity Sensors
- "Extreme"?
  - Temperature Detectors
  - UV Detectors

Fault Attacks Significance

- Well known to the "respectful" manufacturers
- "Good" smart card manufacturers → Secure smart cards
  - Attacks/anomalies will be detected or create no effect
- The attack requires
  - Equipment
  - Extremely accurate timing
- Could be very effective
- Relative low/underestimated significance

Power Analysis

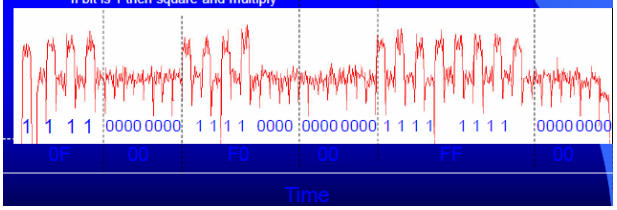


- Smart card microprocessor power consumption may leak information about processing data
  - Microprocessor logic gates consume power
  - Equipment…

Simple Power Analysis

- Threat
  - Deduce what is happening in the microprocessor, e.g. my measuring power consumption
- On "unprotected" cards
  - Power Consumption = *f (Data & Processing)*, for example
    - Program A, Data A = Power Consumption A
    - Program A, Data B = Power Consumption B
- Making it smart card specific
  - Identify an RSA calculation or a DES operation

A well known implementation of RSA is based on "square and multiply"
- $s = m^d \bmod n$
  - S = RSA signature
  - n = (p * q) = large modulus typically 1024 bits, with p & q large primes
  - m message, e.g. say 1023 bits
  - d private exponent such that : $e * d \equiv 1 \bmod (p-1)(q-1)$, where e = public exponent
  - The attacker aims at retrieving d
- Exponent bits are scanned from left (MSB) to right (LSB)
  - If bit is 0 then square
  - If bit is 1 then square and multiply



Countermeasures

- Hardware
  - Add "Noise" → obfuscate execution cycle
  - Provide a balanced power consumption, i.e. independed of data
    - More power consumption → less attractive for certain applications
- Software
  - Desynchronise instructions
  - Random calculations

Significance on "unprotected" microprocessors

- Can be an exceedingly efficient attack
- However…
  - Specific to microprocessor technology
  - Specific to algorithm (e.g. RSA "square and multiply") implementations.

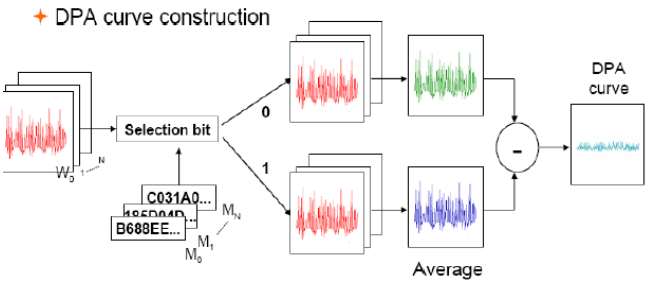
Differential Power Analysis

- Attack Requirements
  - Precise measurement of the power consumption
  - Knowledge of algorithm used
  - Set of plain and cipher texts
  - Large number of measurements
- Exception !
  - Although knowledge of design of target system reduces the complexity of mounting DPA
  - But in practice this can be inferred from timing information

- Two Phase Attack
  - Data Collection (Online with Smart cards)
  - Data Analysis (Off-line on a PC)
- Strategy
  - Make lots of measurements (Data Collection)
  - Divide them into two or more different sets
  - Compute the difference between these two sets
  - Normalize the difference (reveal the keys secret)

- N plaintexts are fed to the device and N cipher texts are generated.
- Power consumption curves of each execution are also generated.

DPA operator & curve



- Summary
  - An efficient type of attack
  - Does not require expensive hardware
  - Statistics
  - Cryptography for the attack
  - Programming skills
  - Experience in instrumentation to build up an automatic measurement system
  - Electronic Skills to improve the results
- Countermeasures
  - Asynchronous Circuits
  - Software random delays
  - Smart card industry is providing enough countermeasures

Software Attacks

Java Card Version 3 (Released in 2008)

- TCP/IP stack
- Embedded web server
- USB Interfaces
- …
- …

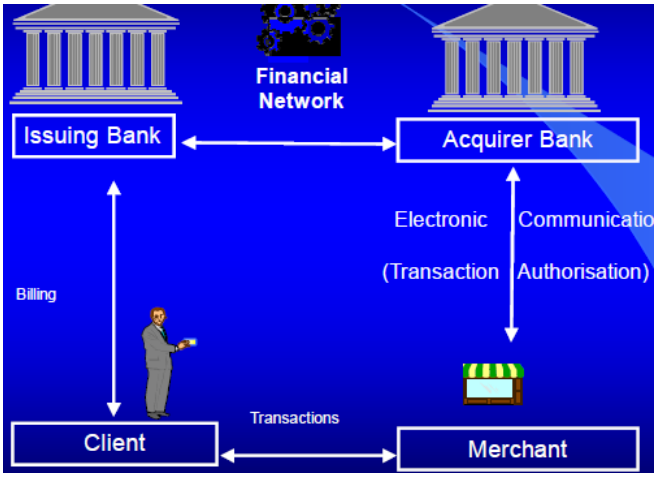
Do classical computer security attacks apply to smart cards?

- Trojan horses
- Buffers overflows
- Bugs exploits
- Java security problems

Use cases

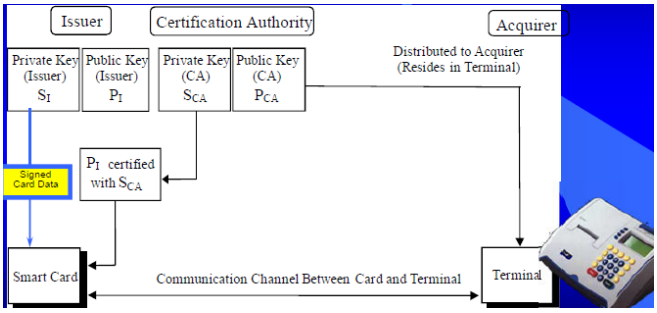
Banking

- Mag stripe card

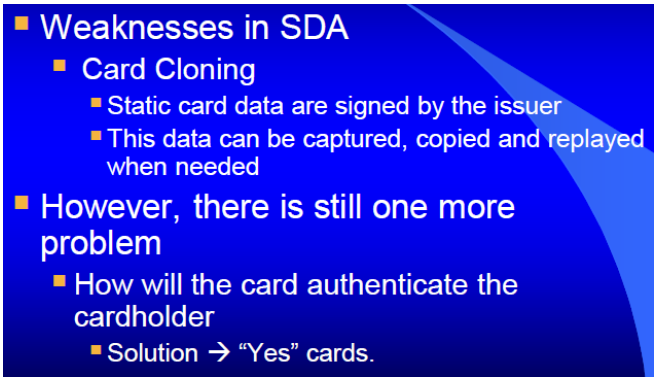


- EMV
  - Joint work on the Europay-MasterCard-Visa (EMV) Integrated Circuit Card Specification for Payment Systems.
  - Motivation → Enable terminals to accept smart cards
- Card Authentication
  - Confirm that the card is legitimate
    - Static Data Authentication (SDA)
    - Dynamic Data Authentication (DDA)
    - Combined Data Authentication (CDA)
- Card and Issuer Authentication
- Cardholder Verification
  - Through the card PIN
- Transaction Authorisation
  - Provided that the necessary funds are available in the card's account

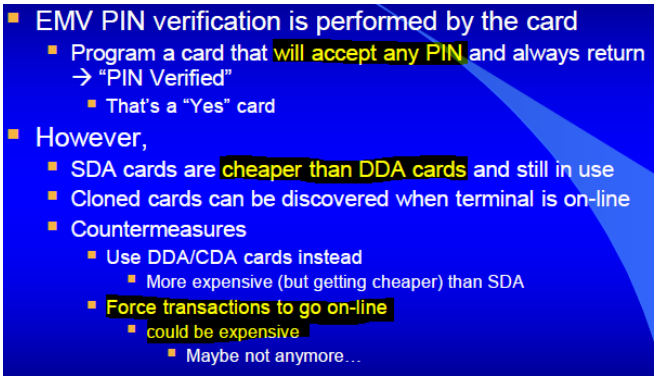
SDA



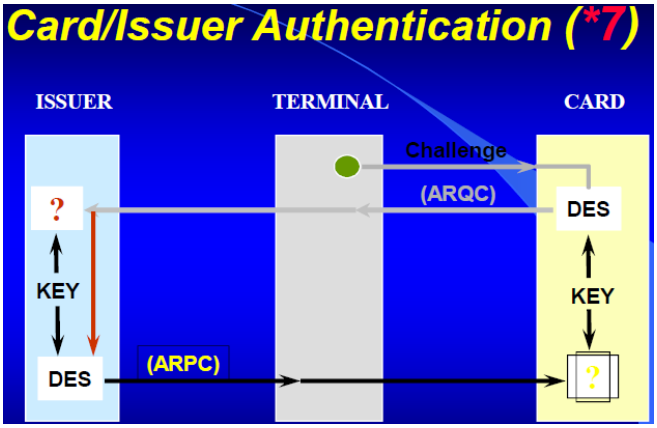
- **Cards can be cloned! (EXAM RELATED)**



YES CARD!

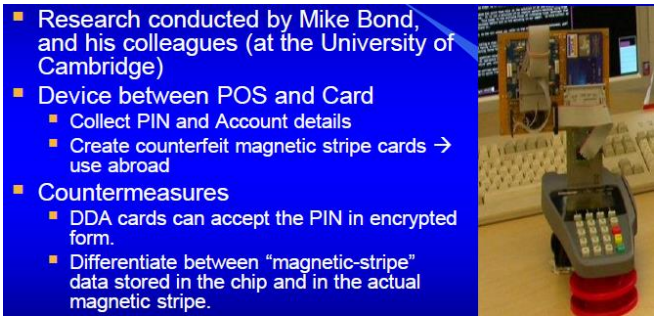


Specifications failed

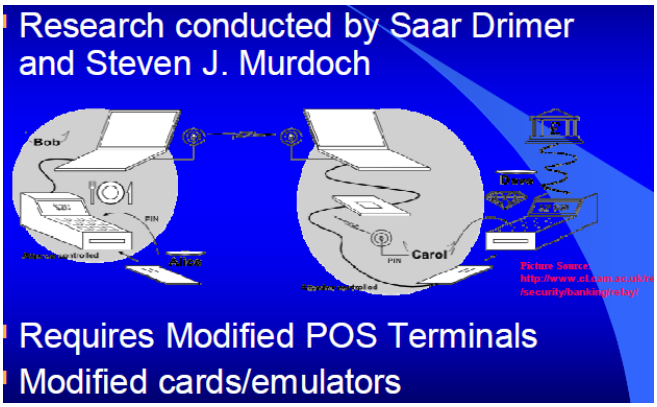


Card is tamper resistant and wont be able to extract DES Key

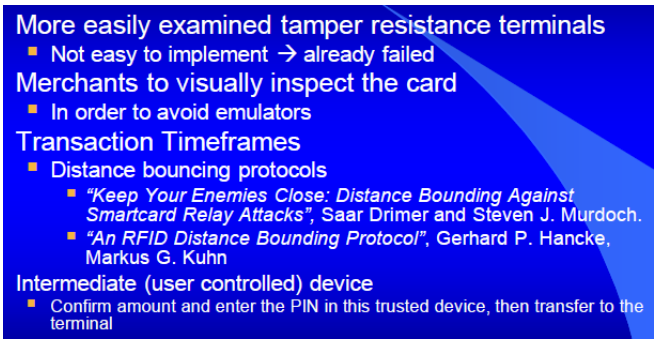
EMV Interceptor



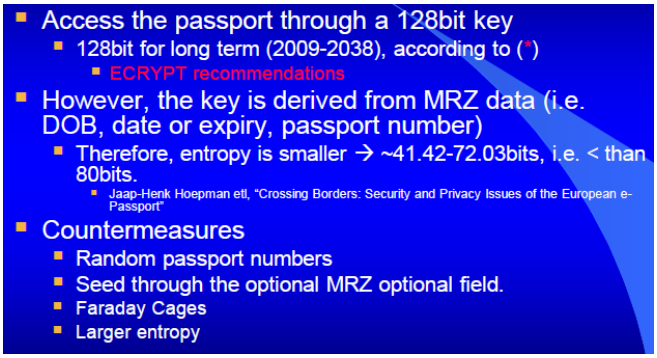
Relay Attacks



Countermeasures

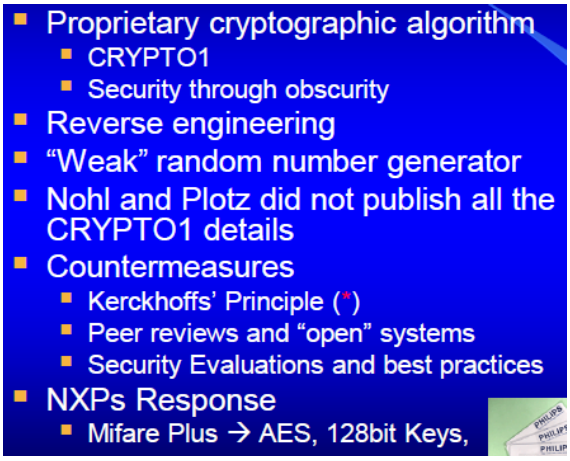


EPassports

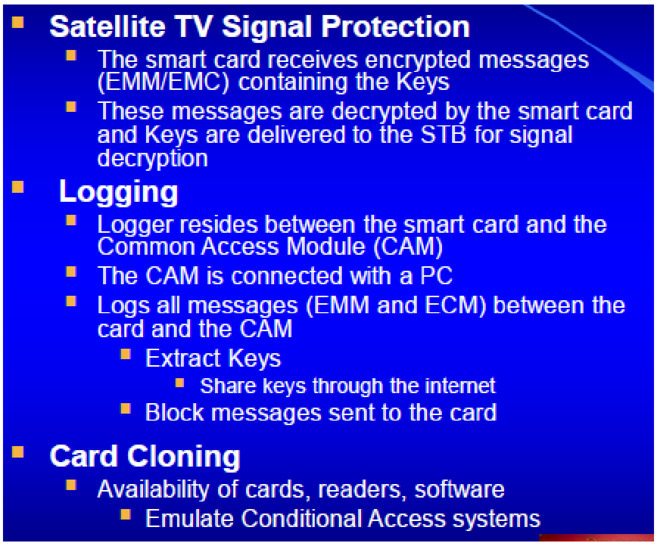


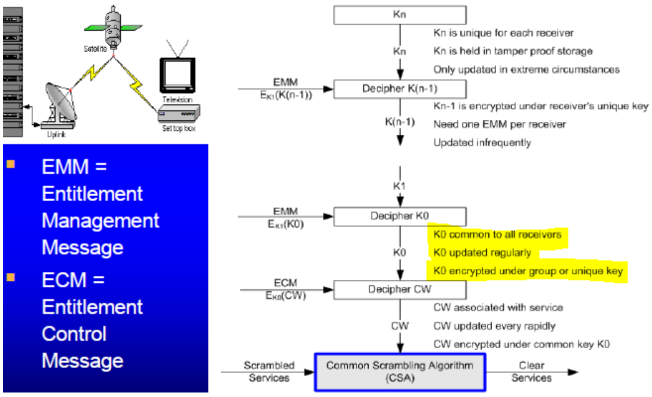
Mifare Cards

Security

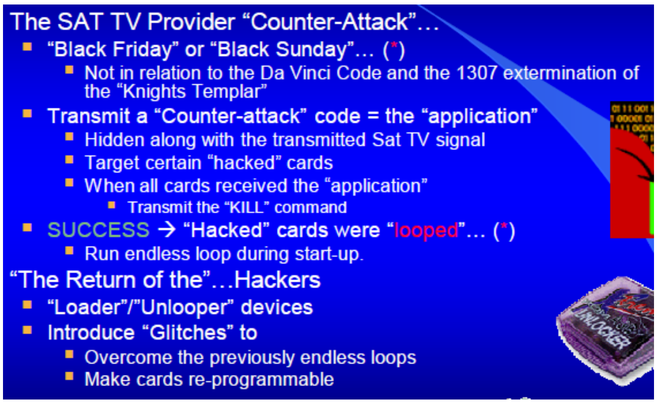


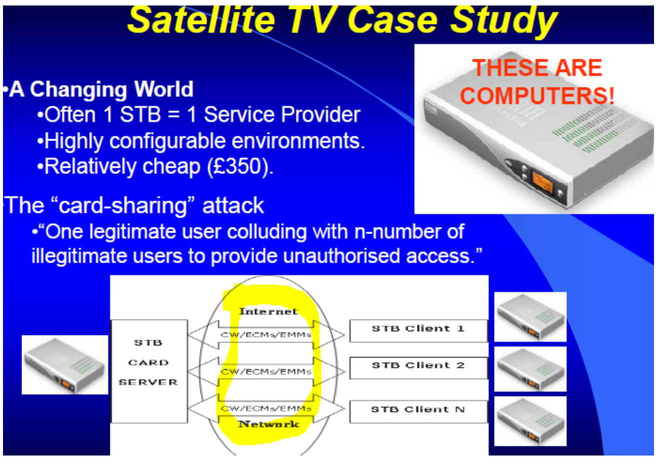
Logging and Card Cloning



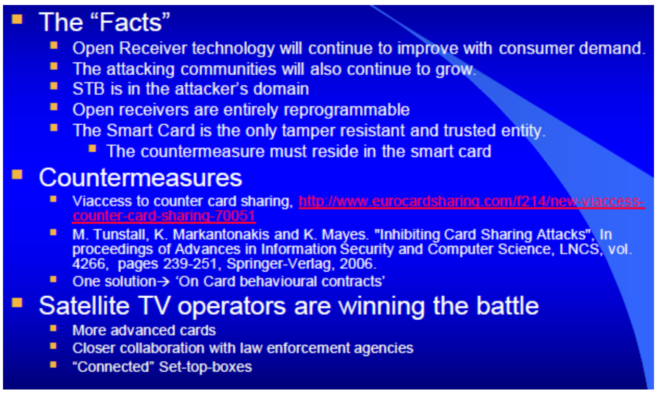


Counter by SAT TV providers





Conclusions



Game Console Security

When: Late 1980's

Why: A home computer Amstrad CPC 6128

- I wanted to protect my own computer programs
- A Floppy Diskette copy protection mechanism introduced disks with "bad" sectors
  - These where difficult to reproduce by "standard" OS floppy disk calls
  - Standard copy programs assumed a uniform sector format
  - Floppy disk controllers through standard OS calls relied on "complete" sectors and not on direct bit access
  - Protection was in software
    - Special bit-by-bit copy programs could duplicate these disks by using reference libraries of known protection methods
    - "Protected programs would be completely stripped of the copy protection system, and transferred onto a standard format disk"
    - Protection method could be disassembled and disabled

- Game console manufacturers have attempted to fight game piracy since they came into existense

- Initially there were "catridges"
  - They were , initially, "difficult" to copy
- Playstation
  - "Mod" a chip on the motherboard to enable the console to play pirate content
- Xbox…
  - Was similarly cracked

Xbox360

- The first generation of Xbox was a piracy destination.
- Xbox 360 [~2005] was designed to be "Piracy Proof with strong built in measures to defeat piracy"
  - "Legitimate" DVDs with… "Bad" sectors
    - Hang-on…. are we going back to the 80's?

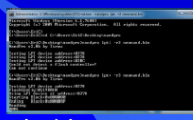
- However; four months after it was released it was cracked by hackers.
- The exploit found it's way by modifying the console's firmware.
  - The modified firmware will report that any inserted disk is a "legitimate"
- The exploit was patched in latter updates; however that promoted hackers to find a better way to defeat the system.

Reset Glitch Hack (RGH)

- Microsoft made it mandatory to install the latest update that patches the previously identified firmware exploit.
- "Xbox 360 was designed to withstand software attacks"
  - However, what about another hardware attack?
- The attackers instead of defeating the anti-piracy mechanism; they went around it.

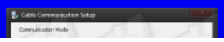
- To successfully hack Xbox the hacker will need to first preform a NAND dumb. (NAND has all the files needed to start up)
- To do so they needed to either solder a (USB SPI Programmer) or a (LPT cable) and connect it to the Xbox motherboard to preform the process.

- Next is to connect the end port to the computer to preform the NAND dumb; "free" programs can be used.
- After preforming NAND dumb next step is to program a CPLD.



- CPLD (Complex Programmable Logic Device) is a programmable logical device
  In simple terms; CPLD is a programmable chip which can be bought for as much as 15$.

- The CPLD is then connected to a computer, using a cable, where the chip is programmed.
- Many programs can be used to preform the operation and these will use values retrieved from the previous step.



- Next step is connecting the programmed CPLD to the Xbox 360's motherboard.

- After the modification is made; hackers can download any game image to their Xbox 360 and run it without needing to insert a disk
- Due to the counter the Xbox 360 bootime is delayed by approximately 90 – 120 seconds which gives space for the Reset Glitch Hack to run.
- Microsoft later introduced multiple versions of Xbox 360; all suffered from the same issue.
- Last version however named Corona was immune to this attack

Corona wasn't released to defeat the rest glitch hack; in fact it was developed in Microsoft labs prior to the discovery of the RSG.

The most notable difference in the Corona model lays in the motherboard; a chip called HANA was missing; likely to have been emerged with Southbridge to make one single chip.

The change wasn't meant to counter RSG; however it did so.

NFC

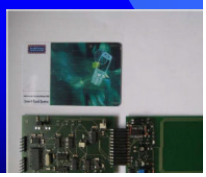
Secure Element Options

- Software
- Hardware Shared Security Software SIM
- Standalone Hardware Security SIM
  - Portable
    - SIM
  - Fixed
    - Embedded chip in the mobile phone
- MicroSD



Cloning and Skimming Attacks

- Not New…
  - (Heydt-Benjamin,2007) et all
  - E-passport clones → (Boggan, 2008)
    - Skimming replay and relaying on payment systems
    - Hardware platform and emulators, e.g. OPEN PCD (Graz, 2010)
      - Disadvantages
        - Form factor, time, skills knowledge, cost (~$750), connection with PC
      - Advantages
        - Longer distances → (Kirschenbaum and Wool 2006)
  - Proxmark3 (~$450)
    - Portable,
    - Read and emulate contactless cards
    - Lots of potential
  - Eavesdropping on contactless tokens
    - A few meters (Finke and Kelter 2006);
    - Up to 50cm (Hancke 2008)

NFC Skimming/Cloning

Research conducted by Lishoy Francis, et al. (Smart Card Centre)
Near Field Communications (NFC)

- adds contactless functionality to mobile devices (standardised, ISO 18092, ISO14443 A/B, FeliCa)
- Plenty of applications in ticketing, payments, access control, etc.
- Secure Element (SE) stores secret keys, establishes trust with Service Provider and the device, (Java Card, GP, Mifare Standard)

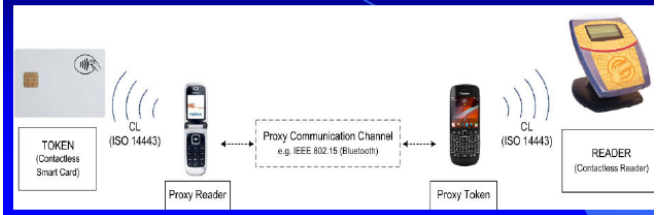
Configure an NFC enabled mobile phone as a skimming/emulator tool against valid contactless tokens.

- Skimming MIDlet is used to extract information from passive tokens (Systems/protocols using "static authentication')
- Developed a Java card applet with spoofed AID and install on 'unlocked" SE
- Possible because
  - The manufacturers allow the SE to be unlocked to assist developers
- The above steps can be easily repeated by non experts

Countermeasures

- Mandatory code signing for NFC communications API.
- User to approve SE activity
- Bind application with unique identifiers
- Use dynamic authentication instead of static data

NFC Relay Attack



- Two NFC phones
  - Nokia 6131 NFC = proxy reader
  - BlackBerry 9900 = proxy token
- Communication Channel
  - Bluetooth (over JSR 82 API)
    - 720 kilobits/sec, 10m-30m(?)
  - Other channels
    - GPRS (reliance on GSM coverage), audit trails of relayed data

Summary

- Acceptable form-factor
  - difficult to visually detect the attack.
- Software based solutions
  - easy to obtain.. easy to "mount"
- Countermeasures
  - Appear to be difficult to incorporate in existing protocols as major changes will be required

Further Threats…

- Counterfeit products [1]
  - Detection
    - Visual Inspection
    - Decapsulation
    - X-Ray inspection
- Hardware Trojans
  - Detection [2]
    - Formal Verification
    - Functional Testing
    - Optical Inspection
    - Side-Channels
    - Trojan Detection Circuitry




Conclusions

- "Smart cards" are here to stay…
  - Bank cards, SIM cards, E-Passports, ID cards, Transport, Satellite TV…
- Unprotected smart cards can be broken by advanced analysis techniques.
- Smart card security is a never ending battle!
- Perfect security does not exist!

- Adi Shamir
  - "…the general problem of protecting smart cards in a cost effective way against active probing and microsurgery attacks seems to be currently unsolvable"
    - It can be claimed that the above statement is partially true…
      - the capabilities of the attackers
      - where is your product is coming from?
      - business model
- "Smart card may not be the magic bullet for digital security. What they do offer is a combination of features not found on any other device that is as cost effective, small and portable."
- *Maya Angelou* (a poet/historian/author/actress/playwright/civil-rights activist,etc.)
  - "We don't understand electricity. We use it"
  - Similarly, hackers or users of "hacked" products
    - Do not really need to understand how attacks work they simply want (and they often manage) to be able to use them

- Smart cards are only a single element within a system.
  - They are only at the "top of the iceberg".
- Smart card system developers should:
  - Obtain smart cards from respectful manufacturers
  - Look for Common Criteria Security Evaluated Products
  - Remember smart cards are a single element in the complete picture
    - Protocols, On-line fraud systems, etc.
- Always seek Expert/In-depended advice
  - Avoid "Security Through Obscurity"…

Trusted Execution Environment

- Trusted
  - Someone or something **you rely upon** to not compromise your security
- Trustworthy
  - Someone or something **will not** compromise your security
- Trusted is about how you use something
- Trustworthy is about whether it is safe to use something
- **Trusted Execution Environments** are what you may choose to rely upon to execute sensitive tasks

TEE [Vasudevan et al.]

- Isolated Execution
  - TEE may be malicious
- Secure Storage
  - Integrity, Confidentiality, Freshness
- Remote Attestation
- Secure Provisioning
- Trusted Path

Example Applications

- Cryptography
  - Key storage (never leaves the TEE in the clear)
  - Key usage policy enforcement
- Password verification
  - Commonly used to unlock keys
- Digital Rights Management (DRM)
  - Typically involves cryptography
  - Also requires control over peripherals

TCB = smallest amount of code that you must trust in order to meet security requirements

Confidence in TCB with:

- **Static verification**
- **Code inspection**
- **Testing**
- **Formal Methods**

Trusted Platform Module (TPM)

- It was hoped that the TPM could create a TEE through the **Static Root of Trust Measurement** (SRTM)
  - Measure hash all software loaded since BIOS
  - OS would perform isolation
  - Attempted by Microsoft as Next-Generation Secure Computing Base (a.k.a. Palladium)
  - Shelved in 2004, but some aspects remain e.g. in Bitlocker disk encryption and Early Launch Anti-Malware (ELAM)
- Problem was that TCB became too large and too dynamic
- Two "identical" computers could have different hashes

Dynamic Root of Trust Measurement (DRTM)

- Rather than trust everything since BIOS, reset CPU and start measuring from that point on
- TPM v1.2 added dynamic registers (17–23)
  - Set to -1 on boot
  - Can be reset by OS to 0
- Register 17 is special
  - Only set by calling SKINIT (AMD-V) or SENTER (Intel TXT)
  - Disables DMA, interrupts, debugging
  - Measures and executes Secure Loader Block

Hypervisor TEE

- DRTM was intended to allow loading of a hypervisor
  - e.g. Xen or VMWare ESX
  - Hypervisor loads and isolates virtual machines
  - TPM can attest to hash of hypervisor
  - TPM sealed storage can be released only to hypervisor once it has been loaded properly
  - Some optimism that this would help for cloud computing
- Hypervisor is still a huge amount of code to validate (Xen contains a full copy of Linux; VMWare is of similar size)

Flicker TEE [McCune et al.]

- Flicker takes advantage of DRTM but for much smaller amounts of code (Pieces of Application Logic – PAL)
  1. Suspend OS
  2. Execute small amount of code on main CPU
  3. If necessary unseal storage and make changes
  4. Increment counter on TPM, storing this and data in sealed storage
  5. Restore OS
- Flicker runs at highest level of privilege so PAL is protected from OS but not other way around

Intel Identity Protection Technology [Carbin]

- Runs Java applet on separate CPU
  - Management Engine part of chipset so bound to physical hardware
  - ARC4 CPU
  - Applications currently available include
    - Key generation and storage (integrated with Windows Cryptographic API)
    - One time password generation (VASCO MYDIGIPASS.COM)
    - Secure PIN entry
      - Possible because chipset also manages video

ARM TrustZone

- CPU buses extended to a "33rd bit", signaling whether in secure mode
- Signal exposed outside of the CPU to allow secure peripherals and secure RAM
  - Potentially could have indicator for which mode CPU is in
- Open system and documented, but only allows one secure enclave

Trustonic

- TrustZone is not very useful by itself due to only allowing one enclave
- Gemalto developed the Trusted Foundations system
- G+D developed MobiCore
- Both split the one secure enclave into several, essentially through a smart card operating system
- Trustonic now developing based on MobiCore
- License fees required to implement code
- Samsung Knox is similar, but also introduces secure boot

Intel Software Guard Extensions (SGX)

- Tightly integrated with CPU
  - Modifies memory management
  - Enclaves protected from other code, and vice versa
- As enclaves are built, the code is measured in a similar way to the TPM
- Can be combined with IPT for trusted display
- Demonstration uses HDMI DRM for trusted path

Comparisons

- Economic
  - Lock-in
  - Who pays the license fees?
- Performance
  - What can you do in the TEE?
- Functionality and flexibility
  - How well connected is the TEE to the CPU and what can it do?
- Security
  - What attacks are feasible

Detour: Security Economics

- About 8 years ago, researchers started looking at the Economics behind security decisions
- Failures in security were not primarily a result of not enough cryptography, but due to failure of incentives
- Why should you protect your computer against malware when it will be a victim of a DDoS who pays the cost
  - c.f. why should you clean your sewage when it is the people downstream who bear the cost of pollution
- Supporting an economic model is now the primary goal of many security mechanisms
  - e.g. printer cartridge authentication

IT Economics is Different

- Network effects
  - Value of a network grows super-linearly to its size (Metcalfe's Law says n^2, Briscoe/Odlyzko/Tilly suggest $n \log n$); this drives monopolies, and is why we have just one Internet
- High fixed and low marginal costs
  - Competition drives price down to marginal costs of production; but in IT industries this is usually (near as makes no difference) zero; hence copyright, patents etc. needed to recover capital investment
- Switching costs determine value
  - Switching from an IT product or service is usually expensive; Shapiro-Varian theorem: net present value of a software company is the total switching costs to the nearest competitor

Economic Impact on TEE Design

- Time to market is critical
  - High fixed/low marginal costs, network effects, and switching costs give first mover a big advantage
  - Security often doesn't help here and often hinders
- Appealing to complementers also important
  - People buy your product because of other products you enable
- Locking someone into your platform is valuable
  - Your customers know this too
- Regulation is often needed but normally too slow

Economic Considerations

- Wide variety of business models in play, and these have as much to do with the design choices as security
- Open specification (TPM) vs closed specification (SGX)
- Platform specific (IPT) vs generic platform (TPM)
- License fees paid to
  - Chipset vendor (IPT)
  - Handset manufacturer (Samsung Knox)
  - OS vendor (Google NFC)
  - CPU core developer (Trustonic)

Performance

- TPM
  - TPM is really slow, and only has slow communications bus to CPU
  - Flicker TEE code runs on main CPU so can be as fast and has access to as much RAM as OS will spare
- Trustzone (Trustonic, Samsung Knox)
  - TEE code runs on main CPU so is as fast, but RAM may be limited
- Intel IPT
  - TEE code runs on separate CPU, which is moderately fast
- Intel SGX
  - TEE code runs on main CPU so is as fast, as much RAM as needed

Functionality

- TPM is isolated from CPU, so can only operate on what it is provided with (hence why Flicker and similar systems are needed)
- Trustzone only allows one compartment (hence why Trustonic and Knox are needed)
- IPT could be thought as similar to TPM (though used very differently)
  - Also has access to display (likely as a result of its ME heritage)
- SGX very tightly integrated into CPU
  - Has control over virtual memory management
  - Very fast context switching and high-speed communications

Flexibility

- TPM functionality baked into hardware
  - Designed to be flexible but what is there cannot be changed
- Flicker allows arbitrary code to be run, but it does not have access to OS or drivers
  - As a result only computation and very simple I/O possible
- Intel IPT, Trustonic, Knox can run arbitrary code
  - But it has to be licensed by Intel, Trustonic, Samsung first
- SGX allows anyone to run arbitrary code
  - But how will attestation key business model be managed?

Security: Side Channel Attacks

- If malicious code can share the same CPU as the TEE there is a risk of side-channel communication
  - If goal is to separate two pieces of malicious code then covert channel communication is also a risk (but normally now)
- Examples include hyperthreading vulnerability in Intel CPUs
- Trustzone (Trustonic, Knox), SGX shares same CPU between TEE and untrustworthy code
- Intel IPT shares CPU with Intel-managed but possibly compromisable code
- TPM runs only security-oriented code which is (hopefully) well written and tested

Security: Physical Attacks

- Physical attacks on TEE generally considered outside of threat model
  - Very hard to defend against, not a scalable attack
- Leak of attestation keys could be a problem
  - DRM application cares whether code runs on a CPU or emulator
  - Revocation can help with this
- TPM comes from smart card world so likely has some physical protection of keys (but this has been found flawed in some cases)
  - Interface to CPU not protected at all
- Trustzone keys are in unencrypted flash
- SGX/IPT keys are on CPU which should be non-trivial to extract

Security: Platform Binding

- TPM chip is not well bound to CPU (could be removed or replaced)
- Intel IPT is more tightly integrated with CPU so much more challenging to remove or monitor communications
- Trustzone and SGX is the CPU so should be infeasible to modify without some serious hardware investment
- Good platform binding allows new types of applications (c.f. smart cards)
- Important to distinguish between scalable attacks
  - Break once, run anywhere
  - Broken until revoked
  - One device at a time

Comparison with Smart Cards

- Performance
  - TPM similar
  - SGX, TrustZone, IPT much faster
- Flexibility
  - TPM less flexible
  - Trustzone, SGX better (IPT, Trustonic, KNOX similar)
- Security
  - TPM similar (IPT too?)
  - Trustzone, SGX, Trustonic, KNOX likely less secure

TEE Goals

	Isolated Execution	Secure Storage	Remote Attestation	Secure Provisioning	Trusted Path
TPM	Not really (too limited)	Yes (but very limited)	Yes	Yes	No (easily bypassed)
Flicker	Yes (but no drivers)	Yes (through TPM)	Yes (through TPM)	Yes (through TPM)	Limited
Trustonic/ KNOX	Yes (but restricted)	Yes	Yes	Yes	Somewhat
IPT	Yes (but restricted)	Yes	Yes	Yes	Somewhat
SGX	Yes	Yes	Yes	Yes	Probably

Yes = “claimed” rather than proven

TEE and Host Card Emulation (HCE)

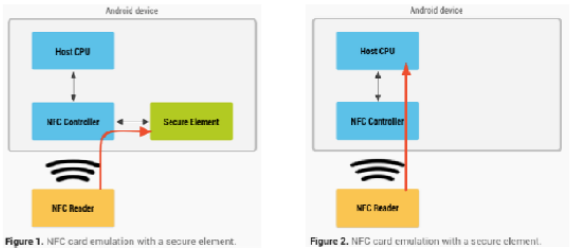


Figure 1. NFC card emulation with a secure element.

Figure 2. NFC card emulation with a secure element.

android 4.4 KitKat

Conclusions

- TEEs may offer Isolated Execution, Secure Storage, Remote Attestation, Secure Provisioning, Trusted Path
- Wide variety of products available which fulfill these goals to varying extent
- Design choices affect both what TEE applications can be supported and how secure they are
- Smart cards may offer better security, but lack of platform binding makes some applications infeasible to support
- Economics will have as much to do with design choices adopted then security
- Ownership of the platform is a key difference between different solutions
- The security of HCE will be critically dependent on the trustworthiness of the mobile device execution environment

Multi-app Smart Card Transition Case

Migration Req.

- **Business req**
- **System + architecture req**
- **Example of transition options**

Business

- Retain Issuer Control!
- Cut Fraud Levels!
- "Live" Program in…-1 to +3 months!
- Obtain a Competitive Advantage!
- Increase Cardholder Base
- Increase Revenue Streams
- Keep the Cost as low as possible
- Multi-application capabilities?
- Establish Partnerships
  - Identify potential "Killer" application combinations
  - Distribute responsibilities
- Increase Program Awareness
  - Internally within the issuing institution
  - Advertising Campaign (Web page design, etc)
- Secure Funding
- Define Program Roadmap

- Local regulations

Technical

- Backoffice needs connectivity (at least all ATMs, retail branch card readers)
- Timeline
- Skills/Expertise required
  - Outsource
  - Security mgmt.
  - Customer support
  - Retail staff training
- EOL
- Hardware (ATMs, Terminals, Card)

- Identify potential "Killer" application combinations
  - Is there a "killer" application? → No → Killer Combination? → Yes!
- Decide on Post Issuance capabilities
- Select a Multi-Application Smart card Platform
- Select the Program Vendors
  - Smart card, Terminal, Application Providers, Smart card Management System, Key Generation equipment, …
- Decide on Post Issuance capabilities
  - ATM, Internet, Kiosks, etc.
- Testing …
- Roll-Out

Marketing = convince customers

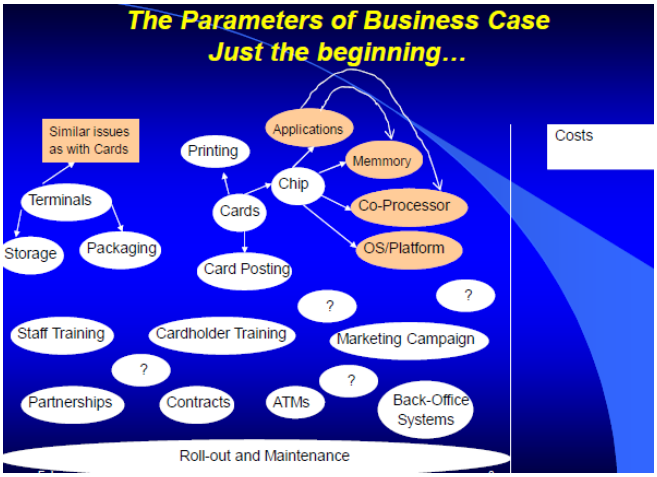
Example of Transition Options

Phase 1: Basic (single app) – chip & pin support

Phase 2: Dynamic (multiapp) – Chip & pin support

Convince merchants

- Merchants need to take the cost of fraud
- US fraud levels is low compared to rest of world.
- Interoperability with international banks (tourists)



80GBP for one card

- Backoffice, needs space, delivery

Can earn money via

- Partnerships
- Multi-app

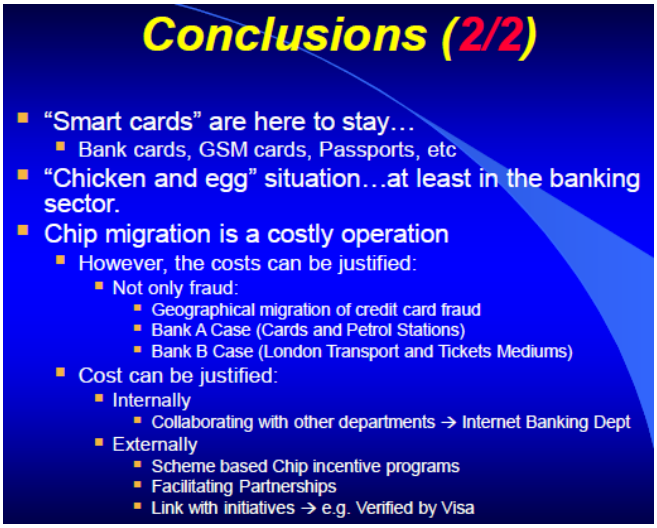
Convince internal

- Use other department (security, marketing) budget

The Pro's and Cons…
To the Merchant

- Possibility of reduced bank charges and charge-backs
- Insulates retailers from effects of fraud liability shift
- Increased sales conversion and improved efficiency due to authorization speed
- Less paperwork (Receipts)


Conclusions (1/2)
Selecting the winner is often a case of:
Existing relationships (with Mastercard/Visa and suppliers)
Not a major issue as often schemes do not object to use one or another preferred platform.
Issuer Expertise
Centralised versus decentralised Issuing
Geographical Areas
Preference on Platforms
Product offerings
Vendor offerings
Security versus Cost
Scheme (i.e. Visa, Mastercard) based Chip Incentive Programs (almost identical)


Conclusions (2/2)
"Smart cards" are here to stay…
Bank cards, GSM cards, Passports, etc
"Chicken and egg" situation…at least in the banking sector.
Chip migration is a costly operation
However, the costs can be justified:
Not only fraud:
Geographical migration of credit card fraud
Bank A Case (Cards and Petrol Stations)
Bank B Case (London Transport and Tickets Mediums)
Cost can be justified:
Internally
Collaborating with other departments → Internet Banking Dept
Externally
Scheme based Chip incentive programs
Facilitating Partnerships
Link with initiatives → e.g. Verified by Visa

IOT Security

- IoT definition
- Security challenges of IoT
- Security responsibilities
- Compare IoT initialisation and personalisation VS traditional smart card processes

IoT = Hype, Hope, Horror?

IoT security from others

- extreme sports of continual response to attack.
- 4 horsemen (Cloud, Big Data, Mobile Social)

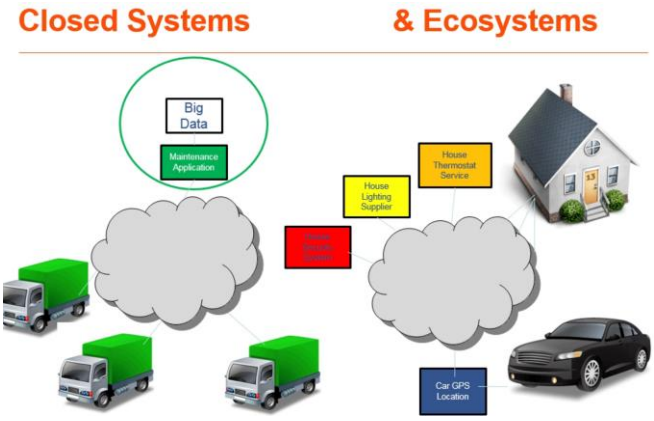
IoT = main thing is that there is internet (able to talk)

Also, internet of everything

- Mobile + wearable device (mobile/fitness/health)
- Automotive (cars)
- Smart Home/Smart Grid
- Industrial Control
- Extended Embedded Systems (logistics/maintenance)

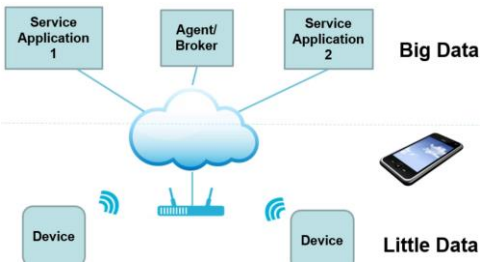
IoT = don't want devices to turn into surveillance devices

Mirai Botnet = Can be a problem to you even if you don't have any (e.g. CCTV into botnet)



Not just the device…

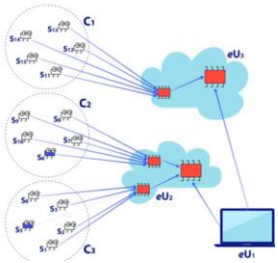
There's service app, agent/broker/ cloud app



Who is responsible for which part?

Primitives – NIST SP800-183

- #1 Sensor
- #2 Aggregator
- #3 Communication Channel
- #4 eUtility (external utility)
- #5 Decision Trigger



Smart Home = Device, Hubs/APIs, Agents

Ecosystem - Smart Home

Reprogrammed the fw chip V850 from internet

Closed System Case Study: Automotive

Sells the USB patch, that brings the responsibility to the car owner.

The Problem of Home IT Security

- **Cyber Security in the home has a bad track-record:**
  - Naïve user behaviours & poor PC management (e.g. Patching)
- **Out of sight – out of mind:**
  - ADSL routers, printers, cameras, other embedded systems – "What's Firmware"?
  - Industrial control systems – metamorphosis into IT systems
- **Vendors might not take ownership:**
  - Timely patch availability, auto-patching?
  - Awareness of privacy concerns?
  - Security expertise in consumer goods vendors? – ICS experiences not great!

IoT Vulns in 2014 = at least 25 vulns per device
Large scale analysis of security vuln of embedded firmware = 38 new vulns

OWASP Internet of Things Top 10 (2018)

1. Weak, Guessable or Hardcoded Passwords
2. Insecure/unnecessary Network Services
3. Insecure Ecosystem Interfaces (e.g. poor authentication)
4. Lack of secure update
5. Insecure or Outdated components
6. Insufficient Privacy Protection (poor use of storage etc)
7. Insecure Data Transfer and Storage
8. Lack of device management once deployed
9. Insecure default settings
10. Poor Physical Security

Risk Management Themes

Device Integrity	Service Integrity	No Trespassing	Agency	Big Data Management
Vendor owns and Manages	Cloud Service Integrity	Not a Noisy Neighbour	Transparency & Consent	Ethical Aggregation (Privacy)
Ethical Behaviour	Communications Integrity	Never "Hunt and Destroy"	Unbiased representation	Access through legal process/or none?
Secure Design/SSDLC	Secure Design /SDLC	Cross service security context	Data Brokerage	Securing Big Data

Why not normal internet security ?

- The Internet is not perfect, but it works. IoT is a bigger Internet, so we just need IPv6 to give us enough IDs ..right?
- No…….and for a few reasons why, including:
- Many of the things that are talked about in IoT are very resource limited, like tags/RFIDs etc.
  - They are unlikely to have IP protocol stacks and so will probably connect via a different protocol/channel to a kind of gateway device connected to the Internet
  - There are many different types of small devices and some of the dominant ones use proprietary non-public protocols
  - There are billions of such "legacy" devices that will around for many years
  - The business case for small tag devices is marginal so they already trade-off security for cost savings
  - Lack of motivation of manufacturers
  - Short life time of devices – no subscription models

Smart Cards, SIMs, e-passports etc. provide attack resistant trust anchors in systems:

- protect stored data and run sensitive algorithms
- design, implementation/production, issuance and secure management
- deploying initialised and personalised security devices with IDs, operational and management cryptographic keys plus sensitive credential data.

Achievable because:

- a small number of Issuers working within industry standards and frameworks, pay for, issue and control the devices…for some commercial/operational benefit

For IoT:

- who issues, benefits from, and controls the security anchors in IoT?
- Will all the devices conform to the same standard? -which?
- Who will manage the security?

Initialisation and Personalisation in IoT

- Most "things" are produced without knowing the eventual owners/users
- Any embedded security in a thing can be initialised for a given purpose and perhaps personalised to unique ID, but no true personal data (like name, bank account etc.) is available
- Post issue personalisation may be required, outside a secure environment
- Remote personalisation will require security management keys
- If the keys are published defaults then anyone could configure a new device, and perhaps maliciously
- If the keys are secret then who holds them? – and who is allowed to access them or indeed take control of them?
- Modules can be multi-application and required for a range of purposes, but is it practical to have many shared secret keys?
- PKI could be used, but resource limited things will struggle with asymmetric algorithm performance, and what about a CRL for 50billion+ things?

Privacy

- Data collected = can be extensive (closely related to user+behaviour)
- MAJOR concern
- Can be anonymised, but combing multiple source = fill in missing gaps
- Location + habits + health + purchasing + associates + home + employment + family = all inferred from collected multi-media big data
- Data may be traded (commercial benefit, or else thing may not be IoT enabled)
- Difficult to know if private data is collected/used = where/when/how/who?

Need:

Micro

- Security by Design
- Embedded security capability

Macro

- Management infrastructure *(N.B. Preserve device ownership)*
- Industry and Cross-Industry standards and supporting ecosystems – e.g. Smart Meters
- Cross vendor collaboration
- Motivation (from the Demand side?)

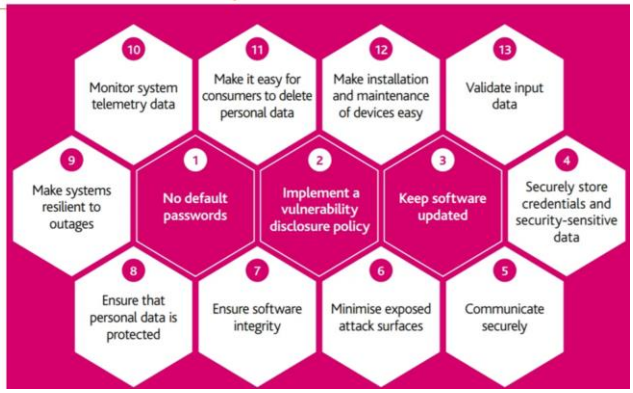
Motivation: Regulatory Intervention

Federal Trade Commission ("FTC") vs. D-Link 5/Jan/17

- *Defendants have failed to take reasonable steps to protect their routers and IP cameras from widely known and reasonably foreseeable risks of unauthorized access, including by failing to protect against flaws which the Open Web Application Security Project has ranked among the most critical and widespread web application vulnerabilities since at least 2007. Among other things:*
- *Defendants repeatedly have failed to take reasonable software testing and remediation measures …security flaws, such as "hard-coded" user credentials and other backdoors, …. failed to take reasonable steps to maintain the confidentiality of the private key….*

- Medical devices? Things which can impact energy infrastructure?.....

Code of Practice for Consumer IoT Security (UK: October 2018)



IoT Security Foundation- Solutions
www.iotsecurityfoundation.org

Needs to be more dynamic than traditional standards – think 'IETF'
Extend existing standards to incorporate IoT, particularly in sector verticals

Examples of other work:

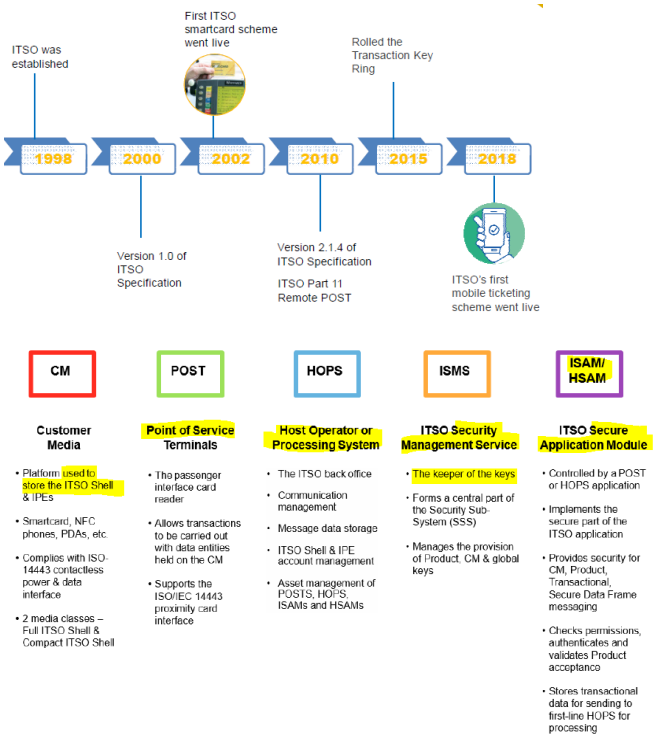
- Test Labs & Certification
- Smart Buildings
- Support for UK 'Secure by Design' - Code of Practice -> ETSI TS 103 645 V1.1.1

Interesting hub based approach (orange)

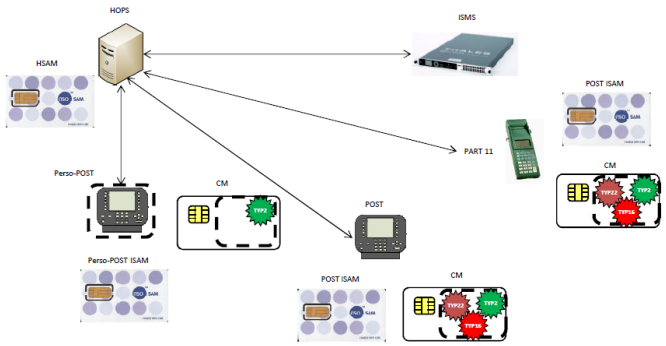
Fix the IoT = so security vuln = rare

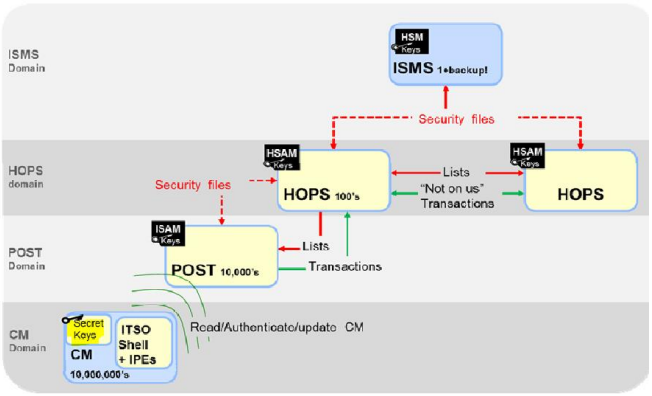
Integrated & Interoperable Smart Ticketing

- Transport operators use the Specification to build and deliver interoperable smart ticketing schemes for concessionary and commercial travel
- In theory, you could use just one smartcard as an 'electronic wallet' for tickets for your end-to-end journey
- By contrast, Oyster is a proprietary scheme restricted to London. However, ITSO smartcards and tickets can be read London



ITSO Scheme Components - Interactions

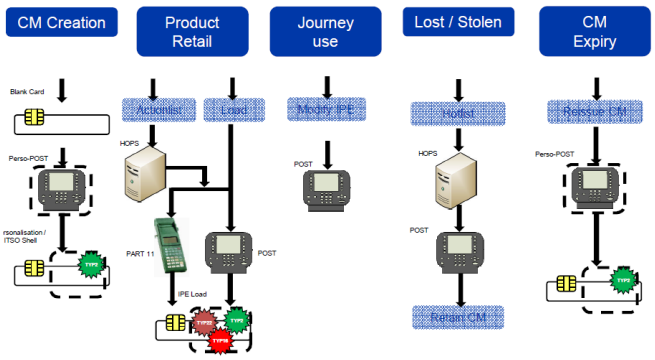




ISMS Activity

- The ISMS manages changes for the security environment and as such it is an agent for change
- To demonstrate the volumes and changes, here are the current statistics (end January 2019):
  - **114** different HOPS processing ITSO transactions in the UK
  - **840,459** security message throughput for the month
  - **87,211** active ISAMs in various POSTs
  - **511** active Customer Media Definitions
  - **1,386** Total products (IPEs) serviced from the ISMS

ITSO Scheme Components – CM Life



ITSO Security fundamentals

- Our goal is to maintain high levels of security.
- The core of this is the principle of Security by Design:
  - Penetration Testing and Vulnerability management
  - Security Hardening (secure defaults, least privilege principle, minimised attack surface)
  - Defence in depth (Firewalls, central logging, IPS, Anti-malware, strong policies & procedures, regular training)
  - Source code audits and secure coding practices
  - Keeping up to date with security research
  - NCSC CiSP member for early warning of emerging threats
  - Regular ITSO Security Committee meetings chaired by independent security and cryptography expert Fred Piper, Royal Holloway University London
- The ITSO scheme is largely based on symmetric cryptography, for which Triple DES is currently used
- Asymmetric crypto (RSA) is primarily used as a means of protecting symmetric keys in transport (i.e. ISMS to ISAM)
- Transactional data (i.e. POST to CM) is protected from manipulation via a cryptographic seal/MAC using Triple DES whilst maintaining ITSO's free read principle
- CM/POST communication is via secure session established by proving knowledge of the correct symmetric key
- In addition to the messaging security ITSO also uses TLS1.2 to protect the HOPS-HOPS traffic
- The security of the scheme is subject to regular independent assessment and evaluation

- The ISMS uses FIPS 140-2 certified Hardware Security Modules to store and generate keys
- HOPS and POSTs use the ITSO Secure Application Module (ISAM) to store keys and process cryptographic functions - originally certified to Common Criteria EAL4+
- 3rd Gen (H3) ISAM supports 3DES and AES encryption

ITSO Security: Risk Management v Operational

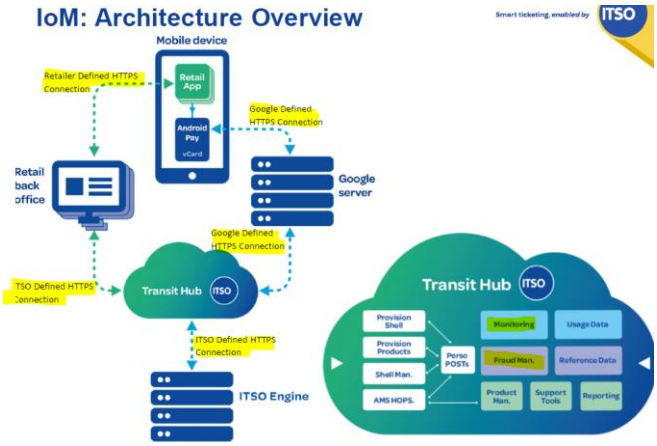
- The ITSO scheme requires resilient offline security because roaming devices (POST/CM/ISAM) are 'out of communication reach' for most of the time
- At the time security was designed for the scheme (around 20 years ago) the technology best supported symmetric crypto - asymmetric would have been slower and require increased key storage space
- Certain security seals use keys diversified with the media serial number, which prevents card cloning
- Every transaction has a unique reference generated at the time of capture, traceable back to the CM/POST
- The ITSO scheme is considered complex, but this provides members greater choice based on business requirements

ITSO on Mobile

- ITSO on Mobile introduces a new security model for ITSO
- Using mobile handsets vs a secure smartcard introduces some security risks, but also brings benefits such as Over the Air ticket delivery and integrated route planning
- Worked closely with members of RHUL ISG and Google Security Team to ensure security is not weaker than the original ITSO model, whilst maintaining ITSO compatibility
- Newly defined HCE based emulated smartcard (mCMD2) integrated into Google Pay (requires NFC and Android 4.4)
- New back end ticket fulfilment systems (ITSO Transit Hub & ITSO Engine)
- Newly defined IoM VPN based on hardened TLS1.2 config
- Extended ITSO Certification requirements on Security

ITSO on Mobile: Security enhancements

- To combat security weaknesses introduced by moving assets from a secure smartcard we have made multiple enhancements:
  - Modified key diversification function now provides limited life keys, by using date as an input to the function
  - Keys are only stored in handset memory, are user specific, short lived, and new key material delivered after every reboot
  - Integration of multiple Google security technologies such as device attestation to collect risk data and detect rooting
  - Real time fraud detection engine
  - Regularly enforced device 'check in' and re-attestation
  - Hardened VPN communications
  - Member HCE app solutions are subject to new security assessments and approval as part of ITSO Certification



ITSO Security: Future

- More frequent rolling of Global keys (e.g. Transaction and Directory Keys) to prevent compromise
- Implement new encryption using AES & ECC to replace 3DES and RSA currently in use
- Rollout of IoM VPN configuration to HOPS-HOPS VPN
- ITSO Specification updates to define and enforce up to date security practices
- Continue to provide guidance for operators/schemes in order to provide a reasonable balance between security and operational efficiency

ITSO = localised to culture, concession, yearly tickets

EXAM

Feedback on the processes

Some require critical thinking

List the factors influencing the price, instead of price itself

smart meter: we might have different form factor. Might be similar in challenges.

- Isolation, security, etc…