Hardware Security

“embedded security”
smartcards & RFID

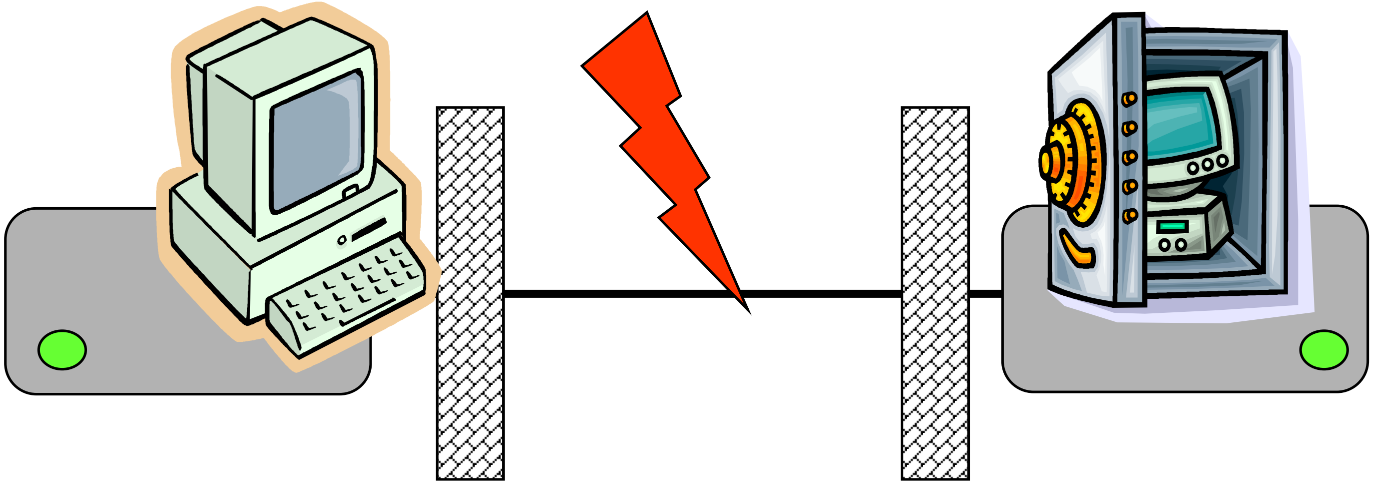
Erik Poll

Digital Security

Radboud University Nijmegen

Security-sensitive hardware: examples

Naive attacker model

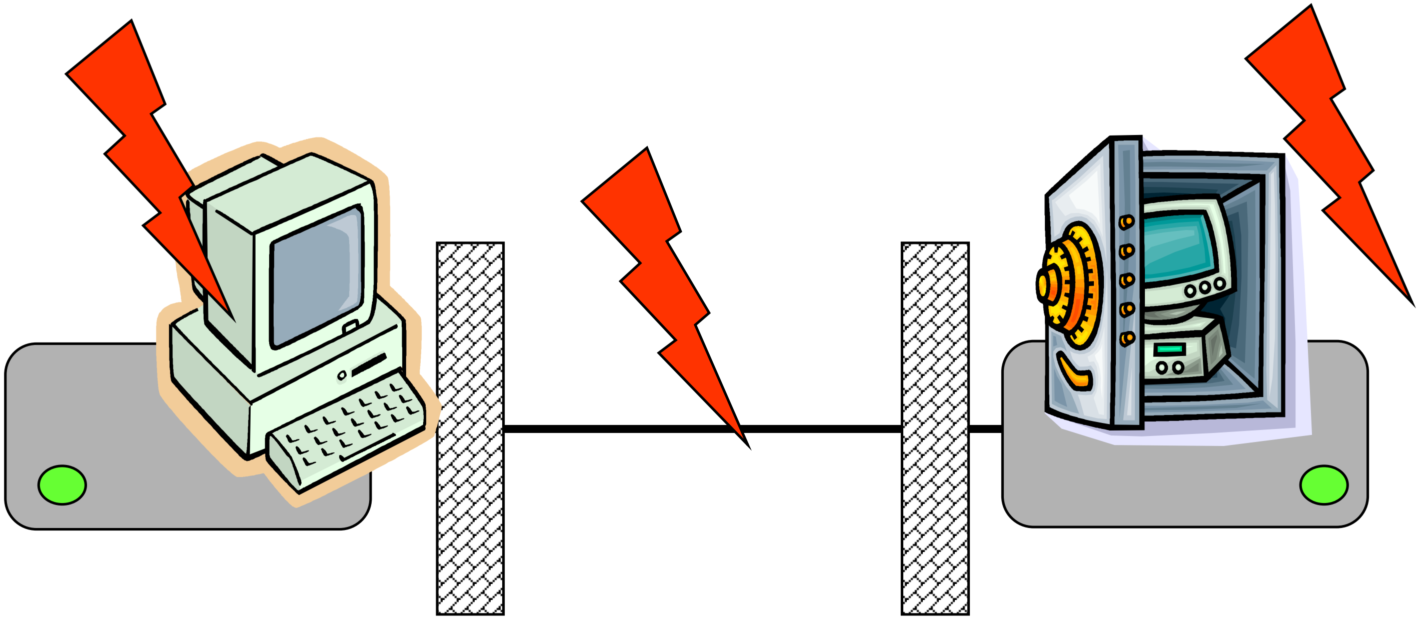


Attacks on *communication channel* between parties

Protection using cryptographic protocols (eg SSL/TLS)

The mathematician's view

Improved attacker model

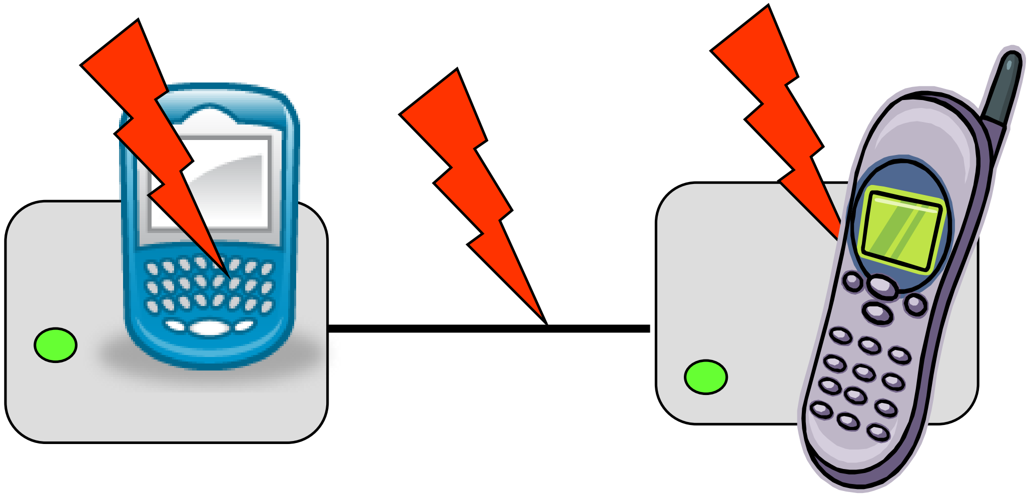


Attacks on *communication channel* or on *software in end-points*

Also takes into account *the software engineer’s view*

Better still: also take into account *sloppy users* at the end-points

Embedded security attacker model

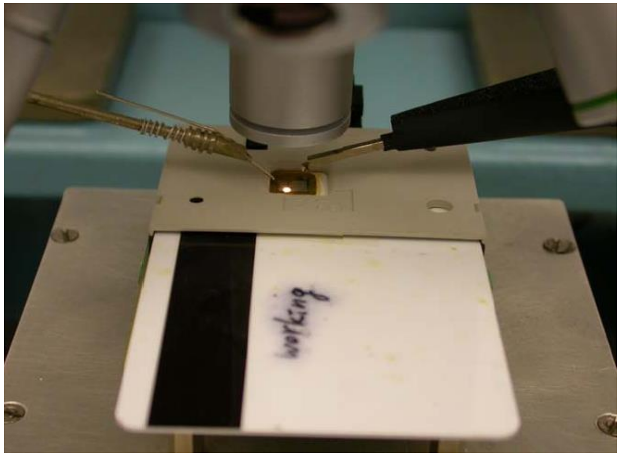


Attacks on *communication channel*, *software at the end-points*, or *hardware* in the end-points

Also takes into account *the hardware engineer’s view*

Specific threat for security hardware

Attacker can get
physical access
and carry out
physical attacks
on the endpoints



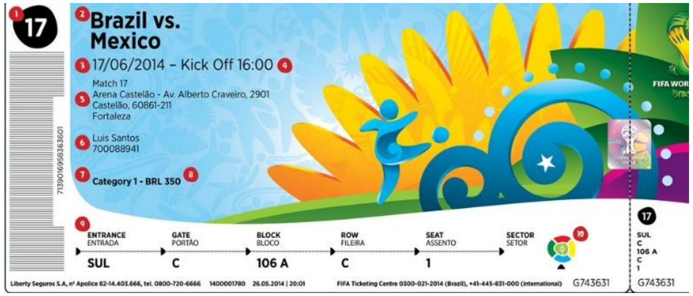
Smartcards

and RFID

Example smartcard & RFID uses

- bank cards
- SIMs in mobile phone
- public transport
- identity documents
  - modern passports and national ID cards contain (contactless) chip
- access cards
  - to control access to buildings, computer networks, laptops,...
- pay TV

Differences? Commonalities?



17

Brazil vs. Mexico

17/06/2014 – Kick Off 16:00

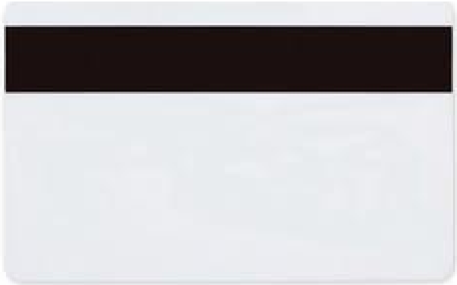
Match 17
Arena Castelão - Av. Alberto Craveiro, 2901
Castelão, 60861-211
Fortaleza

Luis Santos
700088941

Category 1 - BRL 350

ENTRANCE ENTRADA	GATE PORTÃO	BLOCK BLOCO	ROW FILEIRA	SEAT ASSENTO	SECTOR SETOR
SUL	C	106 A	C	1	

G743631



Differences & Commonalities

They all store some data

- for reading and/or writing

Some have some authenticity features

- to tell a real ticket from a fake one

Maybe some also have processing power?

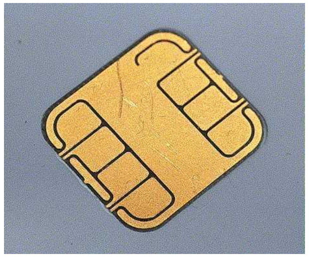
Authenticity: remember the old things replaced by new electronic alternatives

Differences? Commonalities?

Smartcard vs other computers

- No fundamental difference !
    - smartcard does not only offer data storage but also processing power
- Smartcard is *restricted* in its possibilities
    - *How, for example?*
- Smartcard can offer *security* that laptop cannot
    - *What, for example?*
        - *eg you cannot remove the hard drive*

Smartcard technicalities

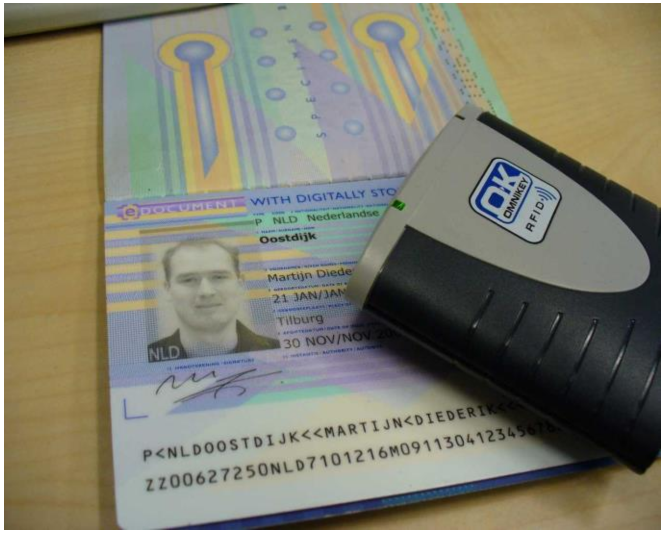


Form factors for smartcards

- traditional credit-card sized plastic card

- mobile phone SIM
  - cut-down in size

- contactless cards
  - aka *proximity card* or *RFID transponder/tag*
  - also possible: dual interface



- USB token

What is a smartcard?

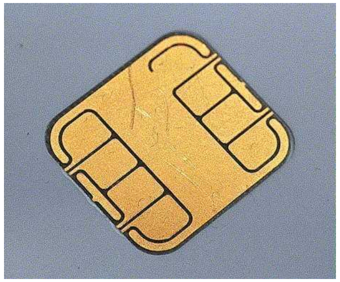
- Tamper-resistant computer, embedded in piece of plastic, with limited resources

  aka chip card or integrated circuit card (ICC)
- capable of *"'securely"*
  - storing data
  - processing data

  This is what makes a smartcard *smart*;

  stupid cards can store but not process.

  But processing capabilities vary a lot!



- Specificed in ISO7816 standard

What does “securely” mean?

- Software and data on card cannot be “messed with”
- The smartcard can implement access control to restrict access to data or functionality, eg
  - deny possibility to read or write some data
  - only allowing it after entering password or PIN code
  - only allowing it after performing some security protocol
- The smartcard can implement cryptographic checks to ensure confidentiality or integrity, eg
  - encrypt / sign data it provides
  - decrypt / check signatures on data it receives

Security properties

- integrity of data and software on the card
- confidentiality of data and software on the card
- authenticity
  - resistance to copying/cloning
- tamper-resistance
  - NB *not* tamper-proof
- tamper-evidence
  - some attacks leave evidence, but not all attacks

Functionality: 3 types of smartcards

1. *stupid card* just reports some data

   Card reports (unique) serial number when activated

2. *stupid smartcard* aka memory card

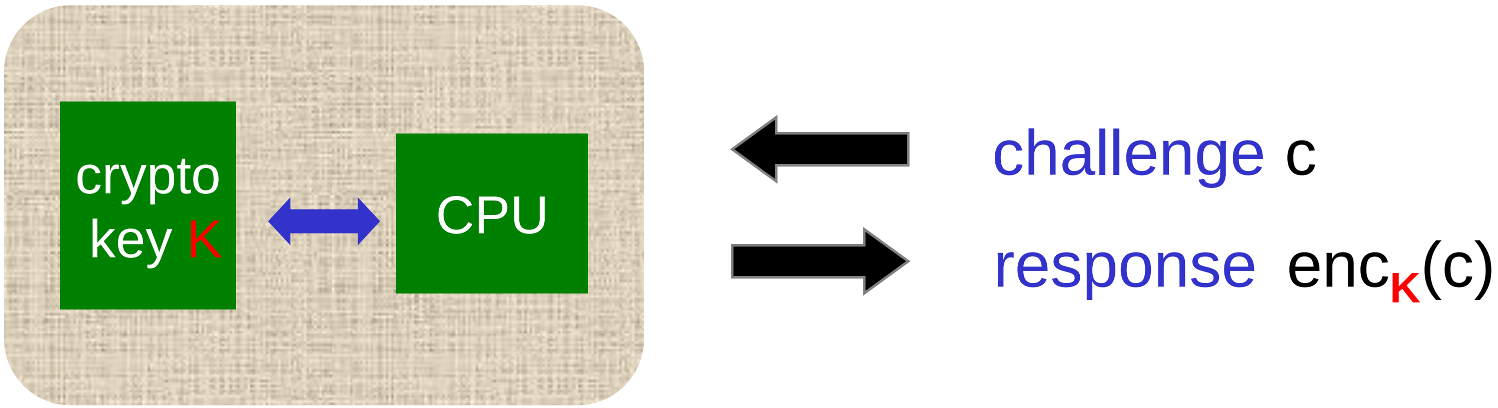
   Provides configurable file system with access control by means of *PIN code/passwords* or *cryptography* or even simpler: *irreversible writes*

3. *smart smartcard* aka microprocessor card

   Provides programmable CPU that can implement any functionality

   Eg complicated security protocols

Typical use of smartcard for <u>authentication</u>



crypto key K

CPU

challenge c

response $enc_K(c)$

- Card proves it knows the secret key, without revealing it
- The key K *never* leaves the card
- The card issuer does not have to trust the network, the terminal, or the card holder

Smartcard hardware

- CPU (usually 8 or 16 bit, but now also 32 bit)
- possibly also
  - crypto co-processor
  - random number generator
- two types memory
  - volatile RAM and
  - persistent ROM & EEPROM

  EEPROM serves as the smartcard's hard disk
- *no power, no clock!*

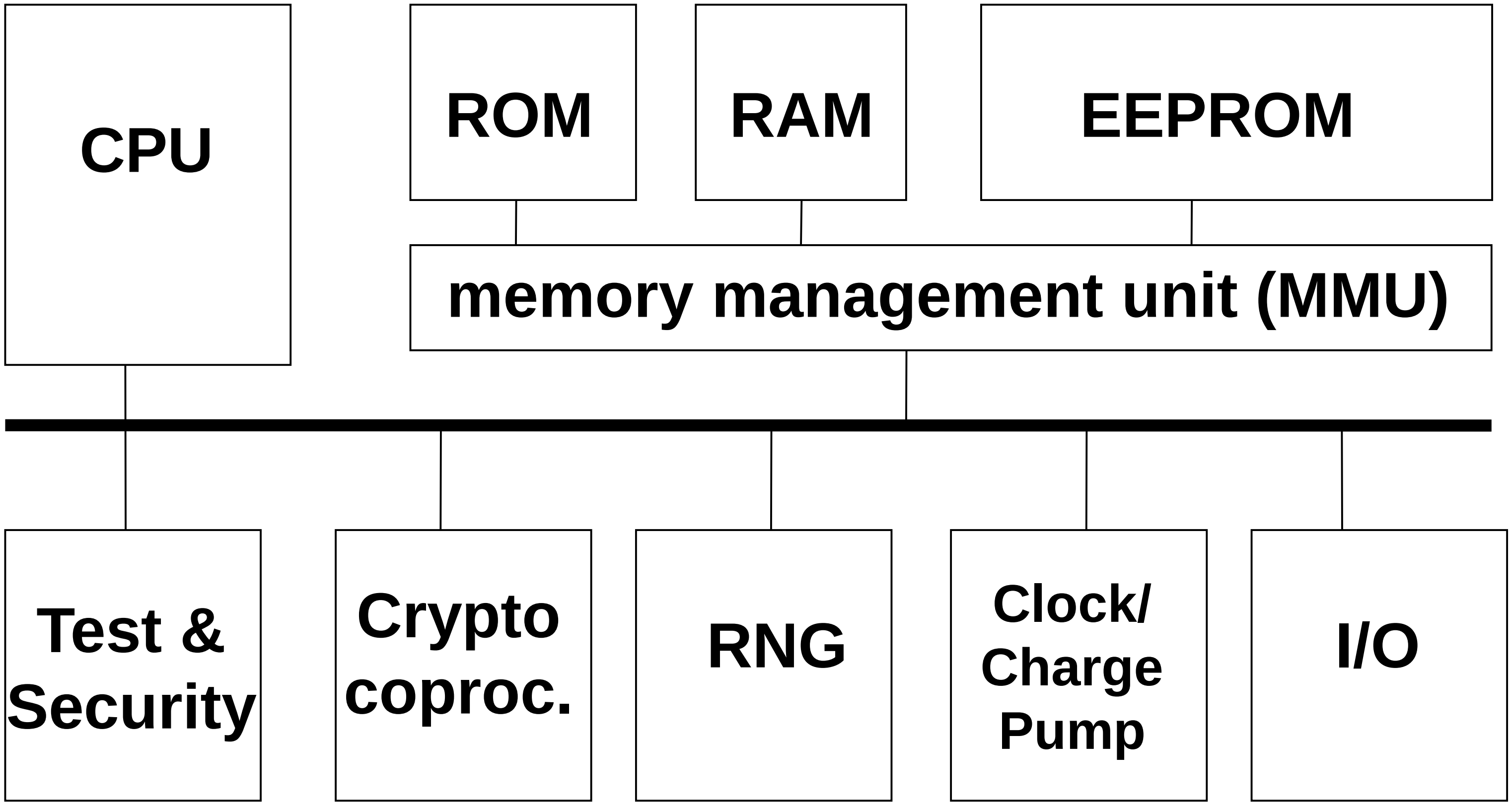
Smartcard hardware

A typical card has 512 bytes RAM, 16K ROM, 64K. EEPROM and operates at 13.5 MHz

Important reason for low capabilities: cost!

Also, keeping smartcard simple means we can have high confidence in software; you don’t want Windows 8 as operating system on a smartcard

Smart card chip



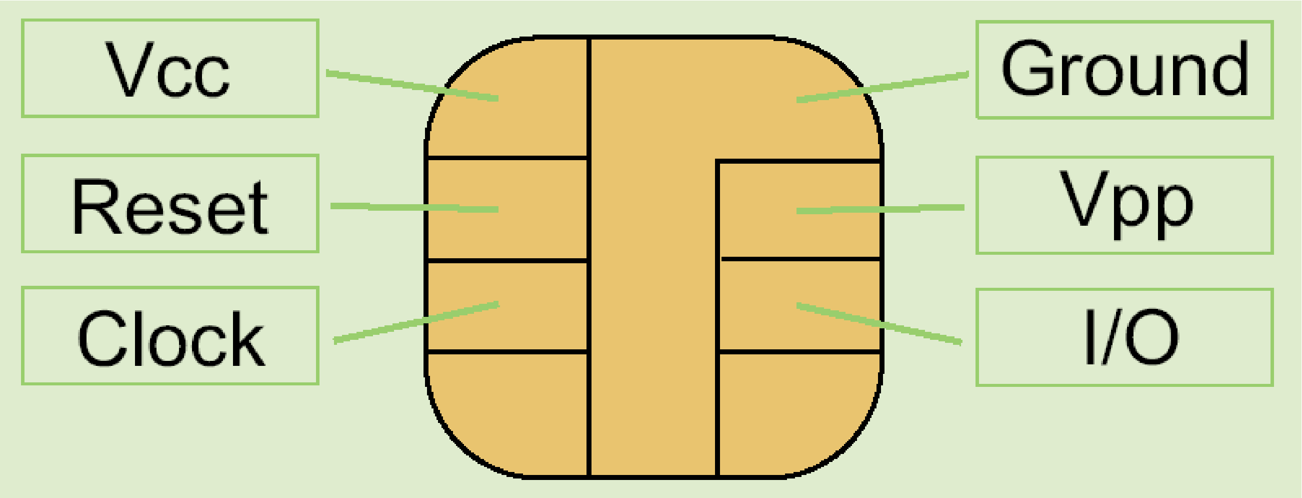
Test & Security

- Self-test hardware & software
  - checking if RAM & EEPROM work
  - checksums for ROM and static EEPROM
- Possible additional monitoring and response against attacks

- Test functionality has to be disabled before use!

  by writing to EEPROM, blowing fuses, or physically removing hardware

Contact cards (ISO 7816)



External power supply and external clock

- Originally 5 V, now also 3V or 1.8V
- Vpp - higher voltage for writing EEPROM - no longer used as it introduces a serious security weakness

Multi-application & post-issuance

Old-fashioned smartcards contain one program, that can never be changed

Modern smartcard platforms

- are multi-application, ie allow multiple, independent programs (aka applets) to be installed on one card
- allow post-issuance download: applications to be added (or removed) after the card has been issued to the card holder

This is tightly controlled, by *digital signatures*

Examples of such platforms: JavaCard and MULTOS

Application management using the GlobalPlatform standard

Multi-application cards

- Multi-application vision: everyone carrying *one* card, with all their smartcard applications
- This is not going to happen. Problems:
  - trust

    banks won't allow untrusted programs of others on their cards; or allow their programs to be seen by others
  - marketing

    who gets to put their logo on the plastic?
- Still, multi-application is useful for development & card management by a single vendor
  - eg used to add services to SIMs that are out in the field

The terminal problem!

The fundamental security problem with smartcards

there is no trusted I/O between user and card

- no display
- no keyboard

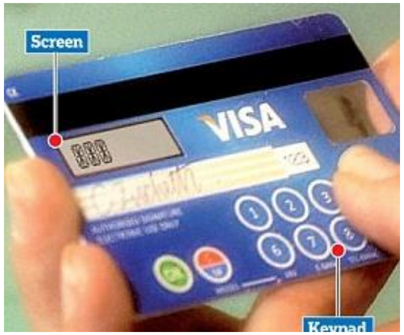
Why is this a problem?

Is this a problem for card holder or card issuer?

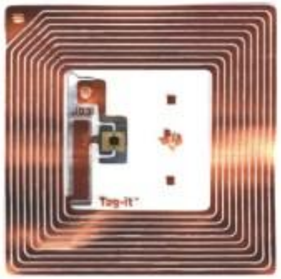
Solutions:

- Card with built-in display & keyboard
- Alternative: give people a reader

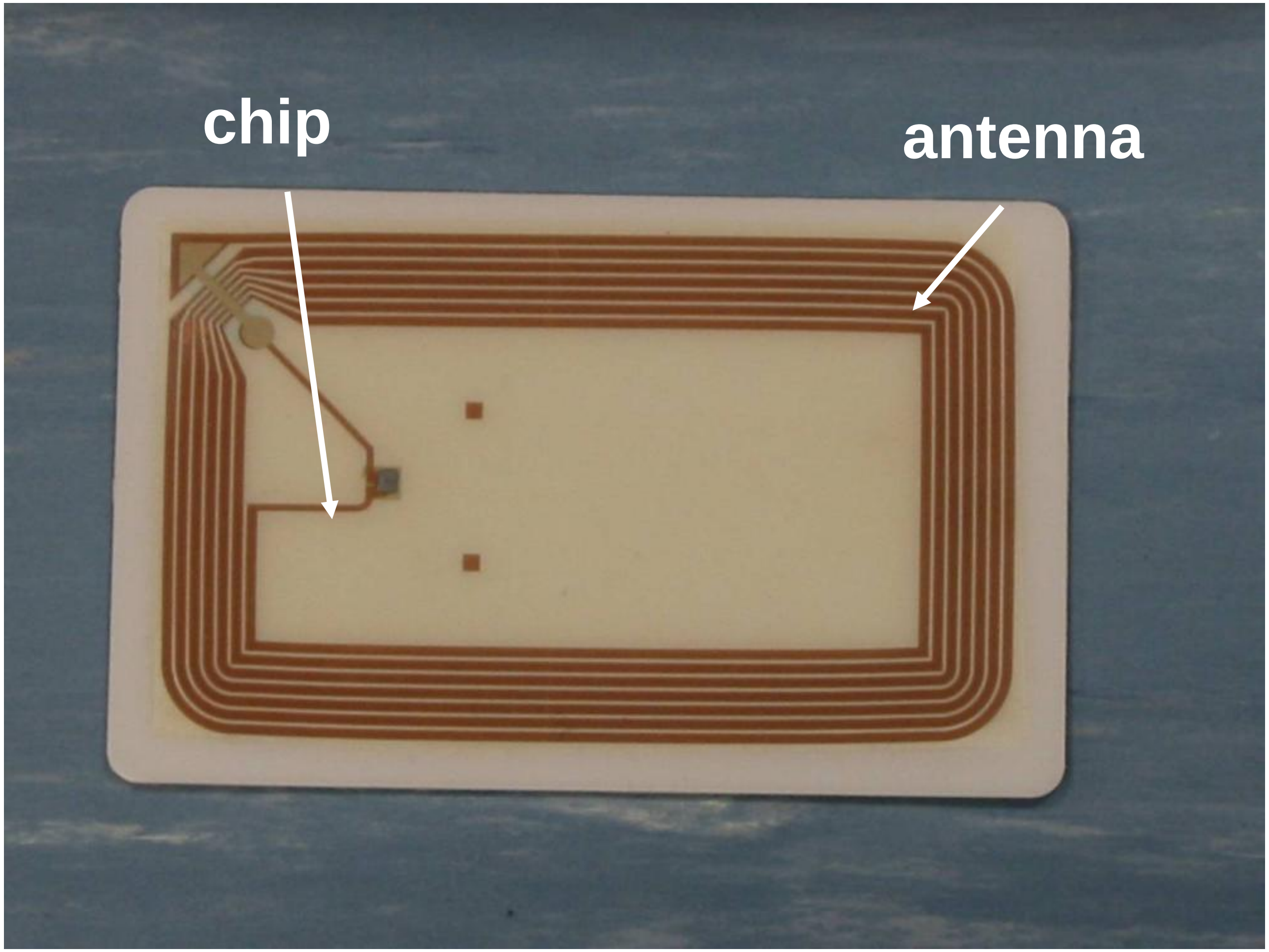
Why not use laptop or PC?



RFID



Inside a metrorio card



RFID

- RFID = Radio-Frequency IDentification
- RFID devices are called tags or transponders
- “smartcard chip with an antenna”
- Often not so smart: RFID tag are often stupid

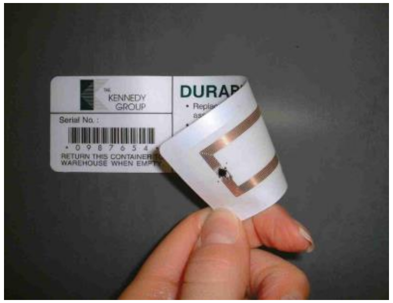
  Many tags only support data transfer from tag to reader

  Powerful RFID tags are also called contactless smartcards

Many types of RFID tags

- with different read ranges & capabilities, operating at different frequencies
- Many just transmit a fixed code when activated:

  - Animal identification RFID tags
  - Item management - RFID bar codes (Global TAG)
  - Container identification - with battery for large range
  - Anti-theft systems - one bit of information



- More advanced cards include

  proximity cards (ISO14443)
  - read range less than 10 cm
  - eg contactless bank cards, and e-passport, and public standards
  - advanced programmable cards talk ISO7816,

    simpler cards talk proprietary protocols, eg MIFARE

Pros & cons of contactless over contact?

advantages

- ease of use
- no wear & tear of contacts on card and terminal
  - less maintenance
  - less susceptible to vandalism

disadvantages

- easier to eavesdrop on communication
- communication possible without owner's consent
  - for replay, relay, or man-in-the-middle attacks (more on that later)
- RFID tags often have more limited capabilities to provide security

passive vs active attacks on RFID

passive attacks

- eavesdropping on communication between passport & reader
- possible from several meters

active attacks

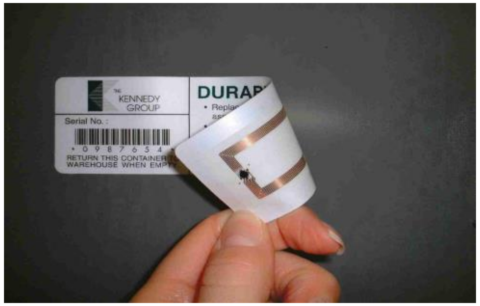
- unauthorised access to tag without owner's knowledge
- possible up to ≈25 cm
  - activating RFID tag requires powerful field!
- aka virtual pickpocketing
- variant: relay attack

Privacy

RFID introduces obvious privacy risks

Note: unlike a traditional barcode, an RFID barcode can provide unique ID *for each individual product*

ISBN 0-596-00167-3
9 780596 001674



Anti-collision protocol &

- Needed for terminal to select one card to talk to, if several cards are in the field of a reader
- For this, cards send out some random number for the reader to identify them. This number can be
- Usually this number is fixed for a card, which can then lead to privacy concerns

NFC

NFC = Near Field Communication

Contactless communication in mobile phones compatible with RFID (ISO14443)

Phone can be in 3 modes:

- active : phone acts as reader for RFID tag
- passive : phone acts as tag fror RFID reader
- peer-to-peer : phone talks to another phone

NFC hardware inside the phone

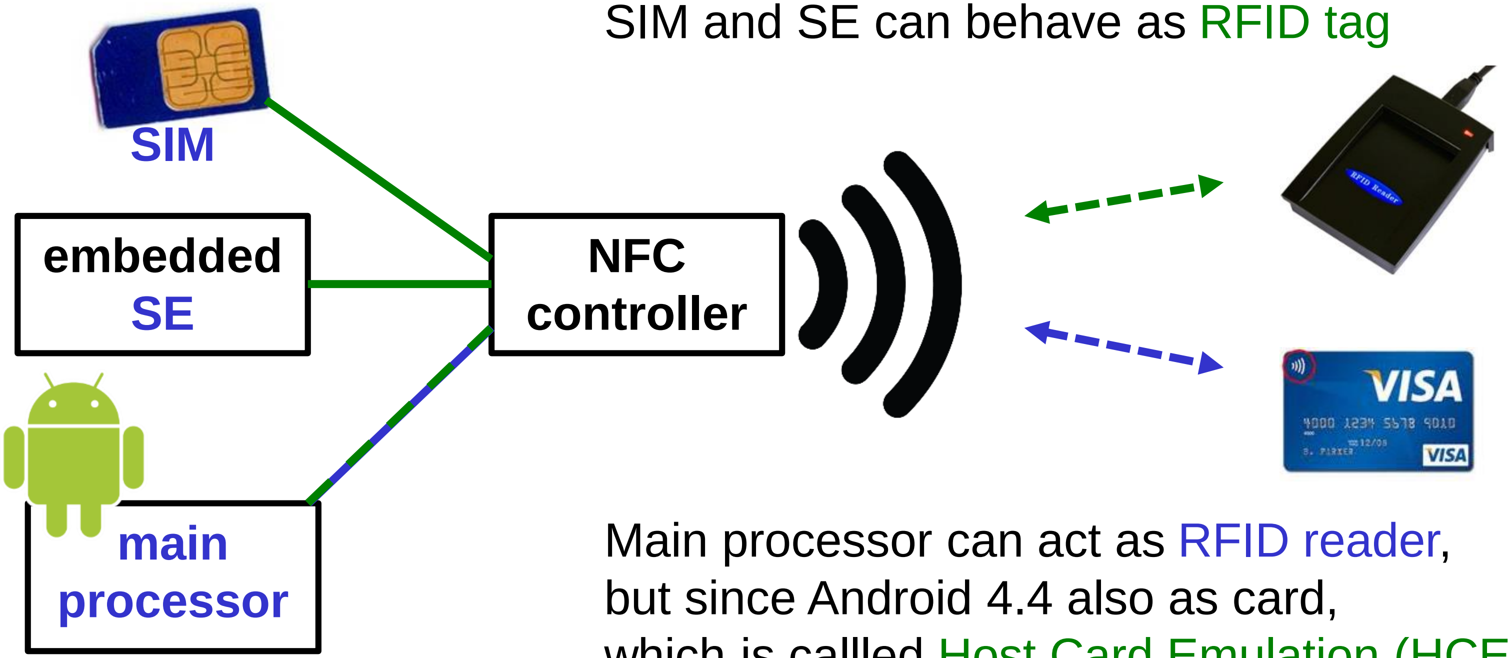
NFC communication may be handled by 3 different processors in the phone

1. the SIM card
2. an embedded Secure Element (SE)
3. the phone’s main CPU (which eg runs Android)

Strictly speaking, the SIM is also an SE, but one that can removed.

Processors in phone for NFC

SIM and SE can behave as RFID tag



Main processor can act as RFID reader,
but since Android 4.4 also as card,
which is callled Host Card Emulation (HCE)

BlackBerry already supported HCE

NFC use cases ?

- Reading tags in eg posters.

  But QR codes work fine for that?

- Mobile phone payments

business complications & security

- Different companies control the SIM and the SE, namely the telco (eg Vodaphone) and the handset manufacturer (eg Google or Samsung).
  This complicates use of NFC, as any use of SIM or SE will require their approval.

- Main CPU offers less security guarantees than SIM and SE.
  - *You can root your phone, but not the SIM or SE*
- Partial remedy: use of TEE (Trusted Execution Environment) for more secure environment in main OS
  - eg ARM TrustZone

Reading tags with your own NFC phone?

- NXP TagInfo for Android