



# Authentication Concepts and Protocols Using Smartcards

**Danny De Cock**

Danny.DeCock@esat.kuleuven.be

Katholieke Universiteit Leuven/Dept. Elektrotechniek (ESAT)

Computer Security and Industrial Cryptography (COSIC)

Kasteelpark Arenberg 10

B-3001 Heverlee

Belgium

These slides can be downloaded at <http://godot.be> → recently presented slides

# Outline

- Problems with current smartcards
- Requirements
- 3 Options
- Selecting the applications
- Specifying the solution
- Future steps
- Questions?

# Problems with Current Smartcards

- Typical application:
  - Smartcard uses its secret or private key
    - E.g., secure signature creation device, information decryption, zero-knowledge proof...
- ***Trust & What You Sees What You Sign:***
  - Can the user trust his/her smartcard?
  - Can the user trust his/her smartcard executes as he/she intended?
  - Is the reader trustworthy?
  - Is the software using the smartcard trustworthy?
  - Can the user verify what the card has done?
    - Phishing, Trojan horses
- ***Interoperability:***
  - Can the user use his/her card abroad or in other systems?
  - Can I develop an application that will support several smartcards?

# Solving these Problems Requires...

## ■ Basic:

- More on-card computing power
  - Multi-threaded
- More on-card storage capacity
  - Interoperable means many trustees & configurations
- Online connectivity
  - Verifying other party, reader, user
- Sound certification

## ■ Interoperability:

- Standardized interfaces
  - APIs for the device and its services
- Standardized data formats
  - Semantic interoperability

# Three Reasonable Options

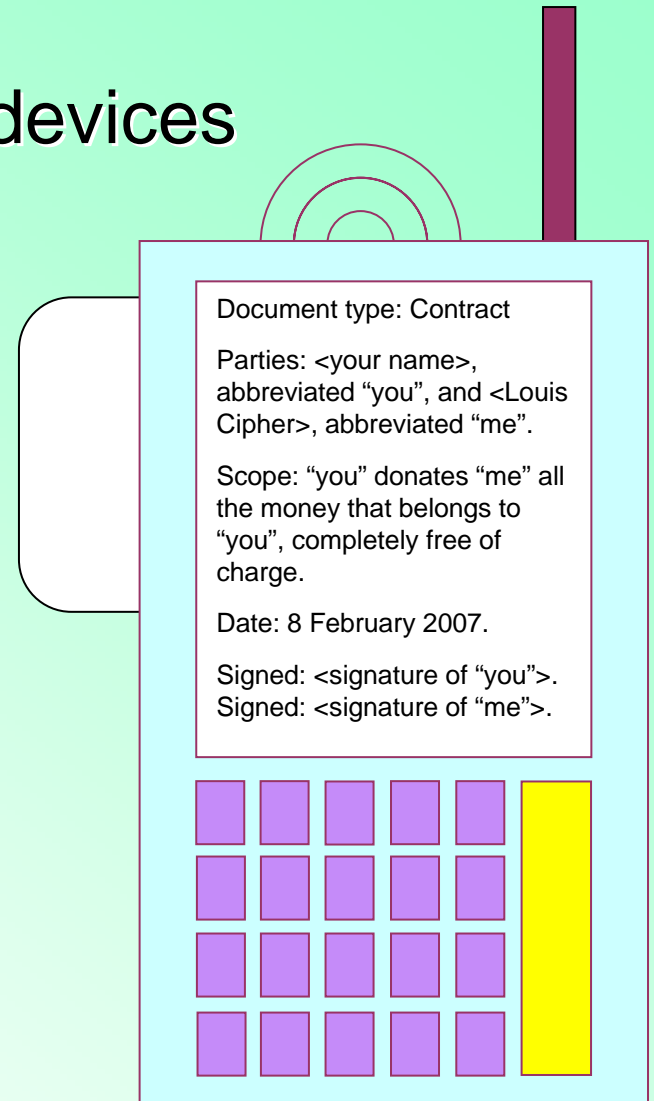
1. Smartcard verifies the reader and provides feedback to the cardholder
  - E.g., ePassport terminal authentication
  - ☹ Cheap, but does not solve the issues...
2. Smartcard → Smartcard Device
  - Smartcard device = smartcard + reader + user interface
  - 😊 Solves all issues, but...
  - ☹ Not cheap...
3. Poor man's solution
  - Increasing trust:
    - Using a smartcard reader proxy and/or a USB/serial port sniffer 😊
  - Increasing interoperability:
    - Standardizing data formats and service APIs

# Choosing a set of Applications

- A generic application does not exist!
  - No generic smartcard solution to fit all applications
  - Well standardized common primitives:
    - Data storage – store/retrieve/manage
    - Cryptographic operations – sign/verify, encrypt/decrypt, prove attribute
    - Access control – grant/deny
- Focus:
  - Private key operations

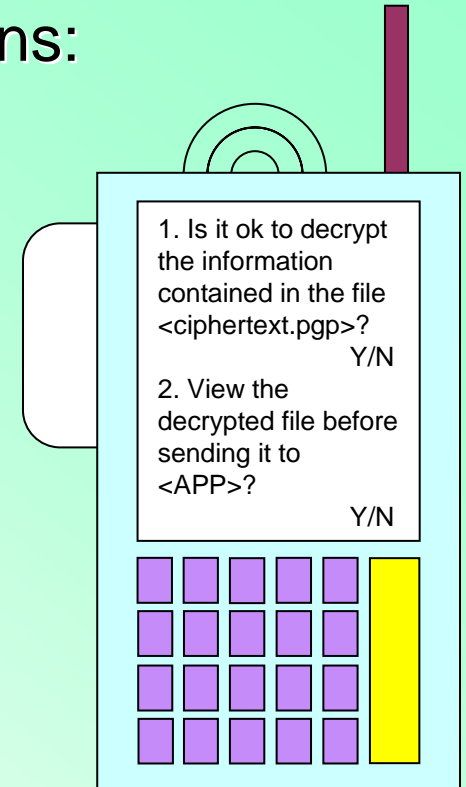
# Hardware Requirement

- Smartcards become smartcard devices
  - Tamper-evident crypto chip
  - Reasonable CPU + storage
  - Contact interface
  - Visual feedback
  - Keypad
  - Smartcard slot
  - Biometric sensor
  - Contactless interface
    - Radio, Infrared
  - Audible feedback
  - Large battery pack



# Using Smartcard Devices

- Smartcard Devices need high-level operations:
  - Digital signatures: Sign/Verify this data
    - Input: data to be signed
    - Internal processing:
      - Authentication of the user
      - Hashing, signing, encapsulation
    - Output: envelope with signed data
  - Data encryption: Decrypt this data
    - Input: encrypted data
    - Internal processing:
      - Authentication of the user
      - Decryption
    - Output: cleartext data
- Strong link with DRM technology!



(DRM: Digital rights management)

# Things to do Next...

- Agree on APIs to the Smartcard Device
  - 15 anniversary of standardizing money-related cards
- Be ready to deal with advances in cryptographic algorithms
  - Consider cryptographic algorithms as parameters
- Semantic interoperability – Data formats
  - Very challenging for border-crossing applications ☺
- Slowly migrate to the ideal hardware solution...

But: More political issues than technical...

# Questions?

- Email: [Danny.DeCock@esat.kuleuven.be](mailto:Danny.DeCock@esat.kuleuven.be)
- Web: <http://godot.be>
- Slides: <http://godot.be/slides>

# Example – Towards a Smartcard Device

