PV204 Security technologies LABS

Introduction to smart cards

Petr Švenda svenda@fi.muni.cz
Faculty of Informatics, Masaryk University

CROCS
Centre for Research on
Cryptography and Security

The masterplan for this lab

- Secure channel and smartcard communication

1. Building Secure Channel protocol (together)
   - simple protocol → design attack → fix it → iterate
2. Communicate with smart card (GPPro tool)
   - ATR, basic info, CPLC
3. Communicate with card programmatically
   - Java java.smartcardio.* or C/C++ PC/SC API
   - CPLC data
   - Obtain list of supported instructions from unknown card

1. Building Secure Channel protocol

- Scenario: we like to transfer extrasupersensitive data between PC and smartcard
- Simple protocol → design attack → fix it → iterate
- Participate in discussion
  - (Steps and solution will be published after the labs in IS)

2. Communicate with smart card (GPPro)

- Contact PC/SC readers + cards
- GlobalPlatformPro tool
  - https://github.com/martinpaljak/GlobalPlatformPro/releases
  - Basic smart card commands, sending APDUs
  - Management of GlobalPlatform cards (JavaCard)
  - Type `gp --help` for all functionality
  - We will use basic functionality now, rest next week

gp --info

- Obtain information about smart card
  - gp --info
  - Obtain ATR (Answer To Reset)
  - Parse using https://smartcard-atr.appspot.com/parse?ATR=xxx

- Who is probable manufacturer of card?
- What is probable OS environment for this card?
- Is it JavaCard?
- What is card’s circuit serial number?
- When was card produced?

gp --apdu APDU_in_hexa --debug

- Send APDU command from command line
- Try gp --info --debug
  - Can you spot APDU command to obtain CPLC info?
- Send get CPLC APDU separately
  - gp --apdu 80CA9F7F --debug

- Can you relate card’s response data and gp --info?
- What is response status word?

3. Communicate with card programmatically

- SimpleAPDU project (IS, NetBeans)
  - Uses Java’s javax.smartcardio.* API
  - CardMngr.java – utility functions for card communication
- Obtain list of available readers
  - List readers = TerminalFactory.getDefault().terminals().list();
- Connect to card
  - CardTerminal.isCardPresent(), CardTerminal.connect(“*”);
- Obtain ATR: Card.getATR().getBytes()
- Send APDU:
  - ResponseAPDU resp = CardChannel.transmit(apdu)

3. Communicate with card programmatically

- Try to send get CPLC command
  - Pre-prepared in GetCPLCData() method
  - Necessary to set proper APDU
- Parse response buffer

- Can you relate card’s response data and gp --info?
- What is value of response status word?

Supported commands

- Card responds to some APDU commands
  - Generic ones (e.g., get CPLC data)
  - Custom ones (what card's owner wants)
  - Usually CLA/INS/P1 only (P2 sometimes)
- How to get list of commands supported by a card?
  - Look into documentation / standard (e.g., SIM commands)
  - Try to probe card (limited number of possible commands)
    - Be careful – many failed attempts may block your card!

Obtain list of supported commands

- Write code that will try all combination if CLA/INS
- Observe response codes
- Make list of CLA/INS which returns interesting code
- Analyse with curiosity!