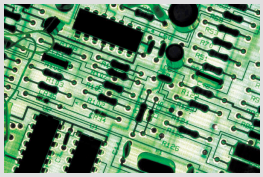
Overview

A. Hardware overview

CPUs, memory types, I/O, cryptographic co-processors, random number generators

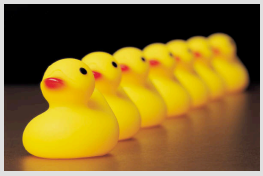


B. Communication modes and protocols

interaction model, contact-based vs. contact-less

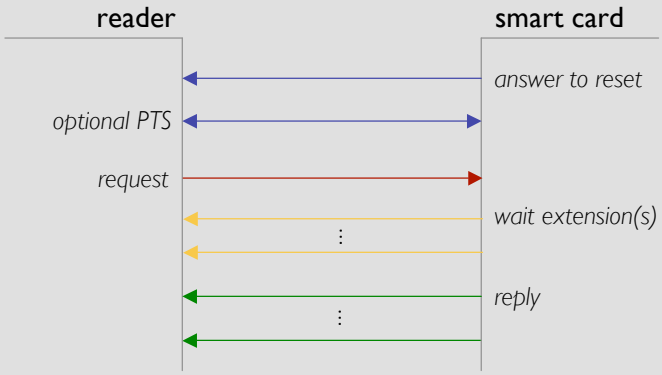
C. Classification of smart-card operating systems

static vs. dynamic, file-system-based vs. object-oriented



B. Communication: Interaction Model

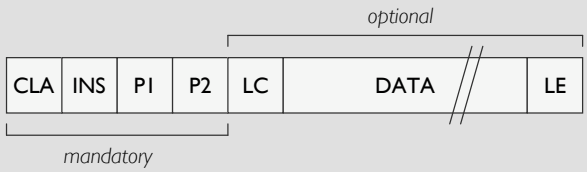
- strictly client/server
  1. *reset sequence when card is powered up*
  2. *smart card is passive (server), only replies to requests*
     - smart-card reader expects reply within some time frame
     - smart card may extend this time frame by sending "wait extensions"



B. Communication: Protocols

- Application protocol data units (APDUs)
  - *units of communication, up to 261 bytes*

Command APDU

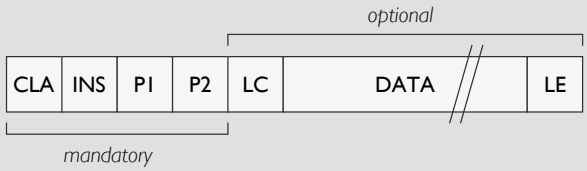


CLA	*application-specific class of instructions (cf. ISO 7816)*
INS	*specific instruction within the class defined by CLA*
P1/P2	*qualification of INS or input data*
LC	*number of bytes in DATA*
DATA	*command data*
LE	*maximum DATA bytes of the expected response*

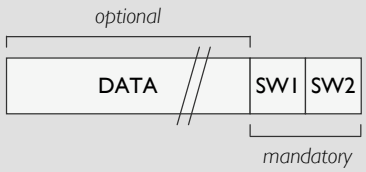
B. Communication: Protocols

- Application protocol data units (APDUs)
  - *units of communication, up to 261 bytes*

Command APDU



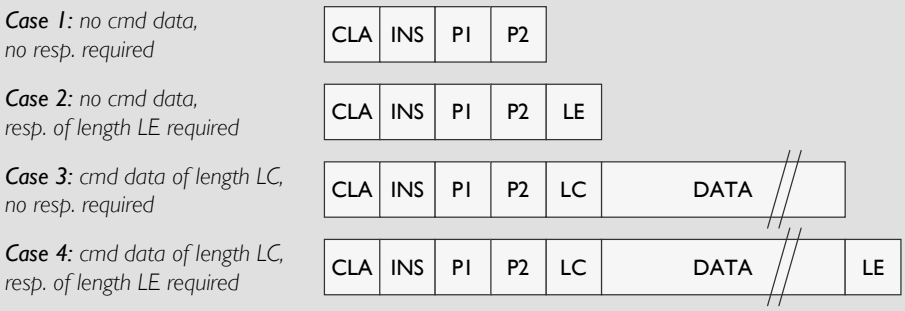
Response APDU



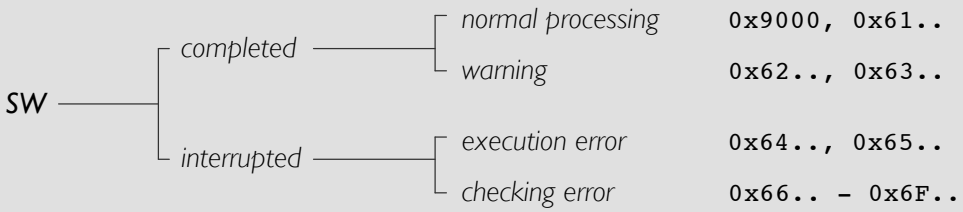
DATA	*response data*
SW1/SW2	*status word (cf. ISO 7816)*

B. Communication Protocols

- Possible command APDU structures

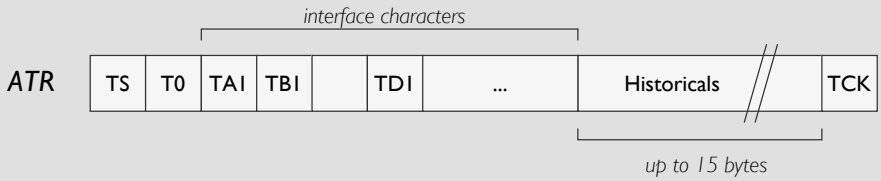


- Status word hierarchy



B: Communication: Protocols

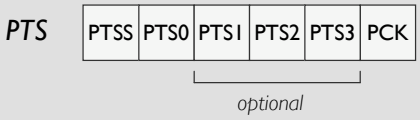
- Power-up sequence
  1. *start of power-up sequence: ATR (answer to reset)*
     - up to 33 bytes as defined in ISO 7816-3



TS	*mandatory initial character (bit signal and bit ordering conventions)*
T0	*indicator for presence of interface characters and historical characters*
TA1	*baud rates*
TB1	*programming voltages*
TD1	*protocols (e.g., T=0, T=1)*
Historicals	*card-specific information (e.g., type, model, and/or use)*
TCK	*optional check character*

B: Communication: Protocols

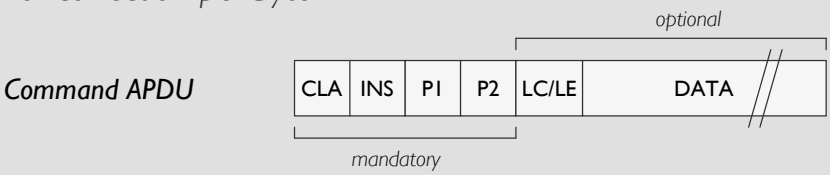
- Power-up sequence (cont'd)
  2. *optional: PTS (protocol type selection)*
     - must follow ATR
     - smart card accepts PTS by responding w/ identical PTS

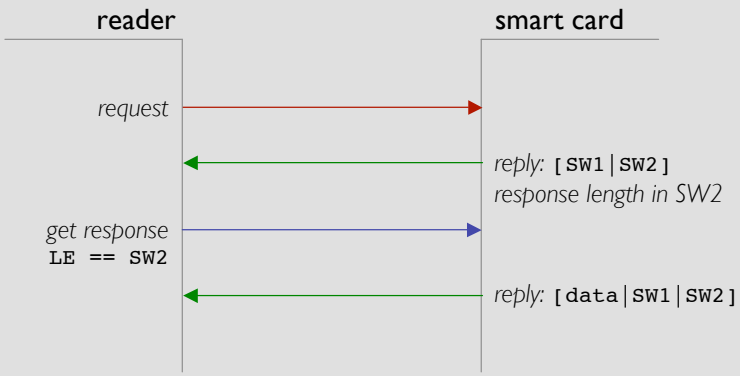


PTSS	*initial character that identifies the PTS, must be* `0xff`
PTS0	*format character: protocol selection, which optional characters are available*
PTS1/2	*additional parameters (e.g., guard time)*
PTS3	*RFU*
PCK	*XOR checksum of all PTS0-PTS3 (as present)*

B. Communication: Protocols

- ISO 7816, T=0
  - *character-oriented*
  - *error correction per byte*

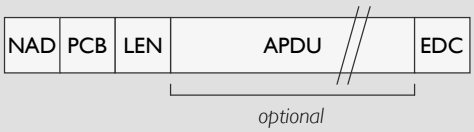




B. Communication: Protocols

- ISO 7816, T=1
  - *block-oriented (data link layer of ISO OSI reference model)*
  - *complex 3-level error correction scheme*

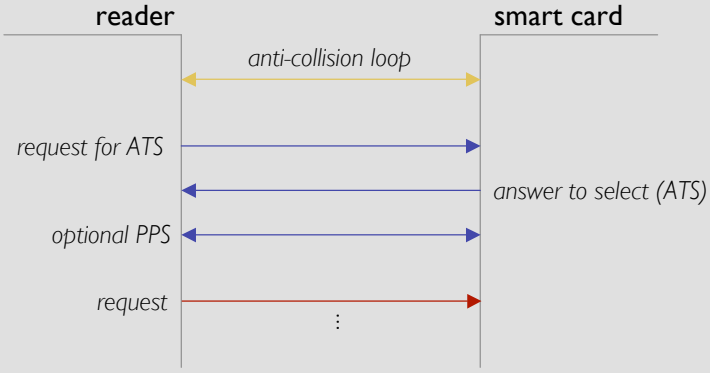
T=1 Block



NAD	*node address (source & destination address, 3 bits each)*
PCB	*protocol control byte (incl. 1 bit sequence counter)*
LEN	*APDU length:* `0x00 - 0xfe`
APDU	*request/response APDU (plain)*
EDC	*error correction code (LRC or CRC)*

B. Communication: Protocols

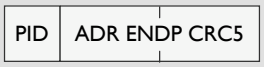
- ISO 14443, T=CL
  - *types A and B (different modulation)*
  - *anti-collision*
    - up to 14 smart cards communicating with 1 reader
  - *protocol: T=CL (very similar to T=1)*



B. Communication: Protocols

- USB 1.1 low-speed
  - *block-oriented (data link layer of ISO OSI reference model)*
  - *"fully" host-controlled (host polls)*
    - all communication is started by the host sending an USB token
    - if a response is required, the host polls the smart card until the response data becomes available (no wait extensions)

Token

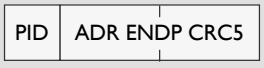


PID	*packet identifier (e.g., type: in/out/data + flow control)*
ADR	*device address [7 bits]*
ENDP	*device endpoint (only one for smart cards) [4 bits]*
CRC5	*error correction checksum [5 bits]*

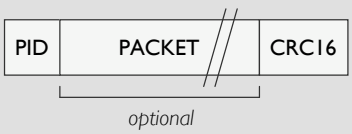
B. Communication: Protocols

- USB 1.1 low-speed
  - *block-oriented (data link layer of ISO OSI reference model)*
  - *"fully" host-controlled (host polls)*
    - all communication is started by the host sending an USB token
    - if a response is required, the host polls the smart card until the response data becomes available (no wait extensions)

Token



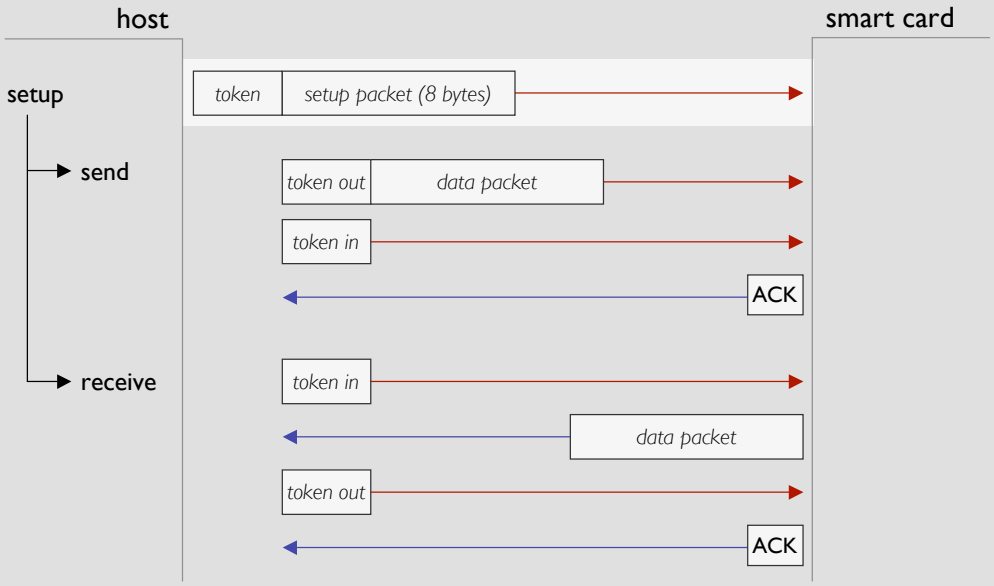
Data Packet



PID	*packet identifier (e.g., type: in/out/data)*
PACKET	*0-8 bytes*
CRC16	*error correction checksum*

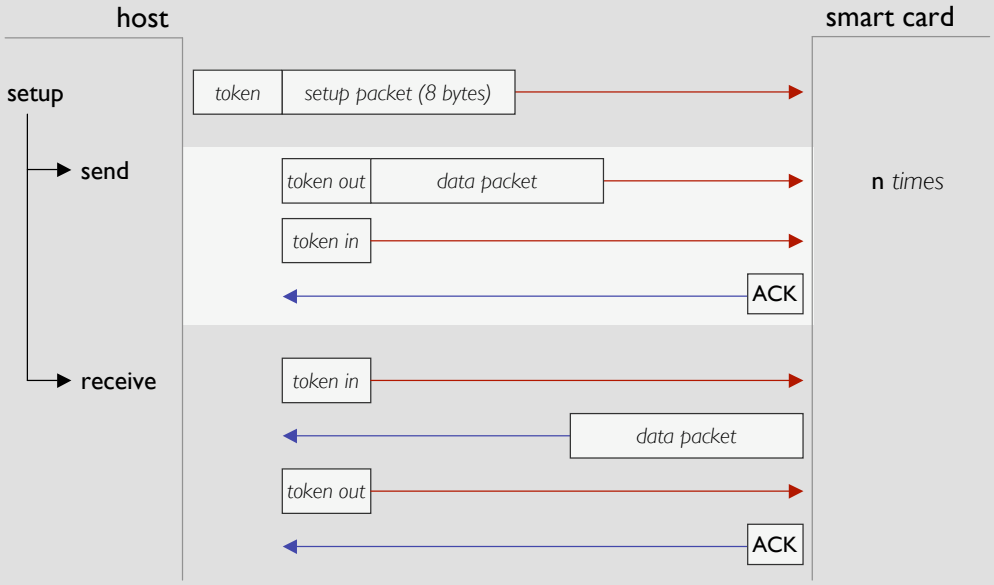
B. Communication: Protocols

- USB 1.1 low-speed
  - *send/receive sequences*



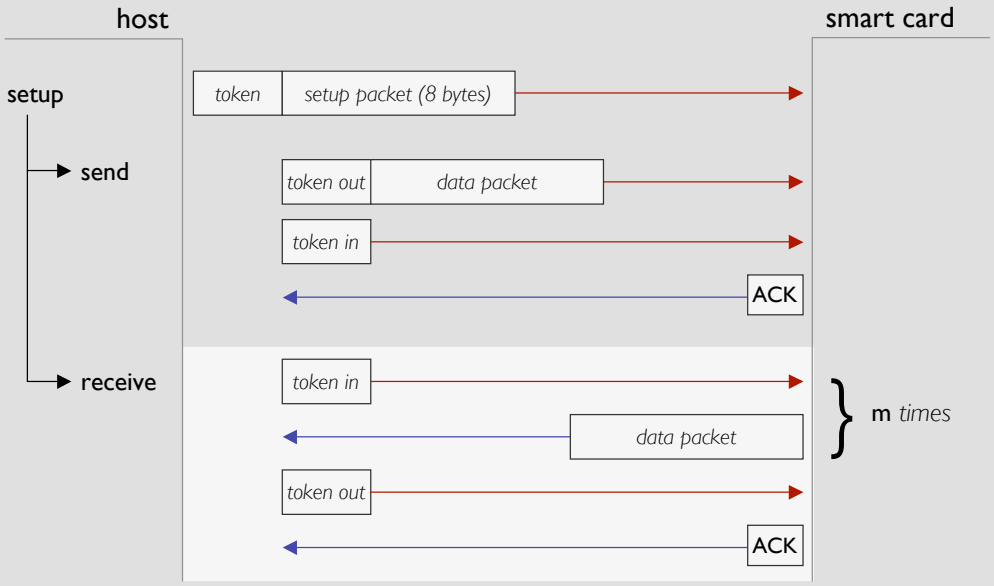
B. Communication: Protocols

- USB 1.1 low-speed
  - *send/receive sequences*



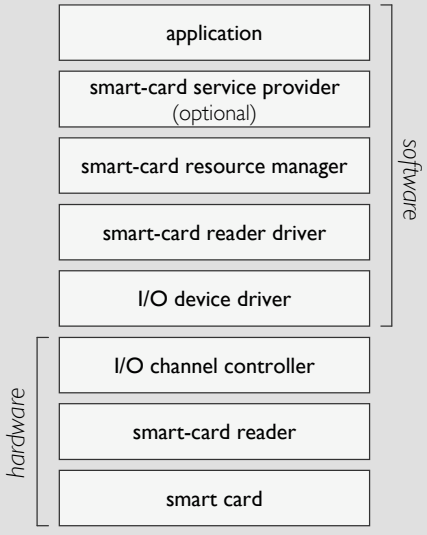
B. Communication: Protocols

- USB 1.1 low-speed
  - *send/receive sequences*



B. Communication: Off-Card

- PC/SC
  - *"Interoperability Specification for ICCs and Personal Computer Systems" (PCSC)*
  - *hides the specifics of smart-card readers from applications*
    - smart-card service provider
      standard interface to smart cards, provided by the card manufacturer
    - smart-card resource manager
      identify/locate smart cards, simultaneous access (shared, exclusive)
    - smart-card reader driver
      translate requests from the resource manager to the reader driver
  - *de-facto standard*



DEMO

New Year's Competition

- We are looking for the strongest QuadLink applet
  - *JavaCard applet, maximum size 4 KB*
  - *no EEPROM usage (only RAM)*
  - *based on the UI framework of DeepLink*
- Deadline: January 8th
- Live Competition on January 8th
  - *applets get 1 point for each match win*
  - *for matches with no direct winner, the applet wins that needed less time*
  - *prices for the three top-winning applets*