Advanced Card Systems Ltd.
Card & Reader Technologies

Native Chip Operating System

CARD & READER TECHNOLOGIES

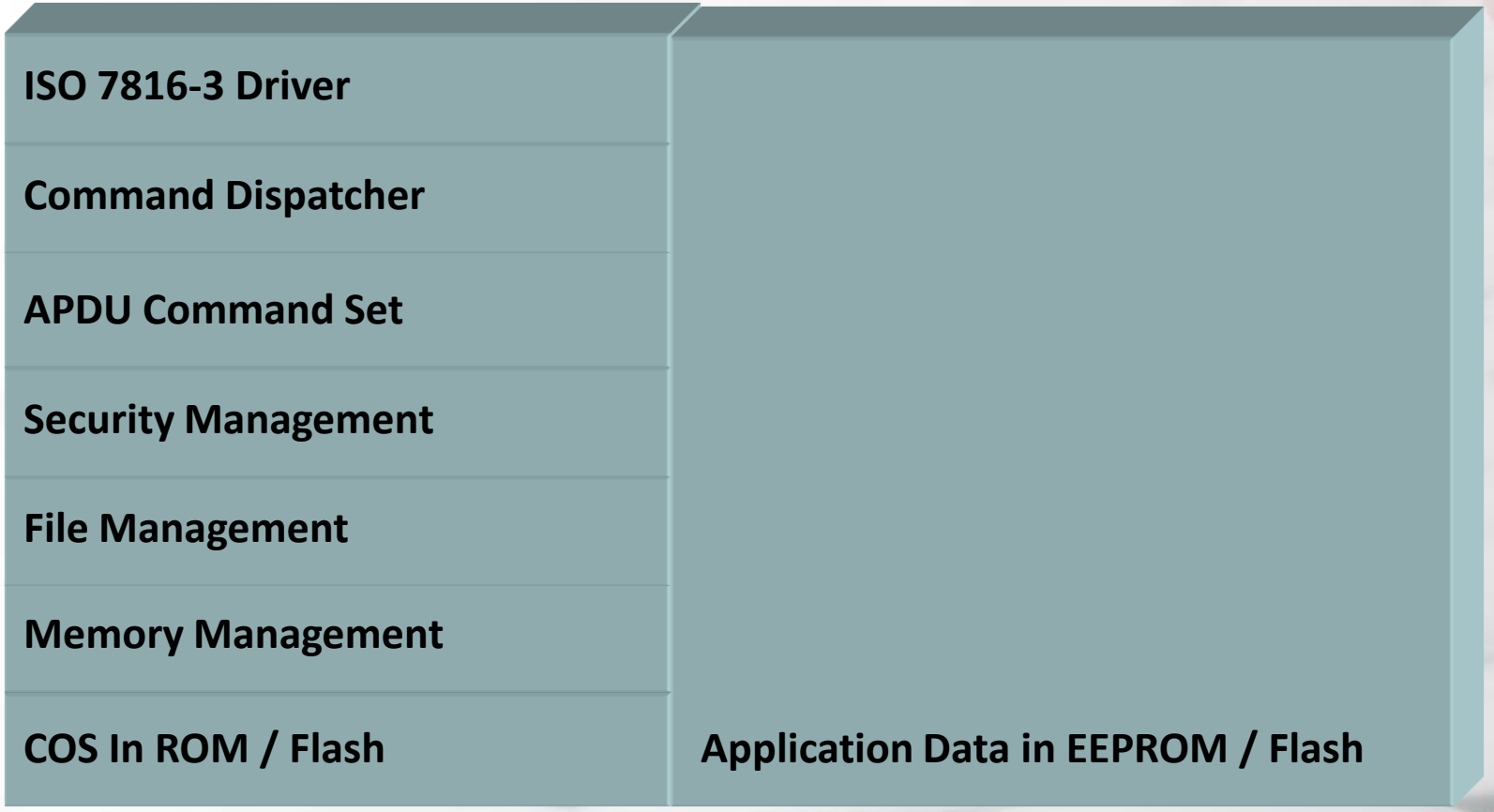
Contents

- Background
- Native COS Internal Architecture
- Memory & File Management
- Security Management
- Command Dispatcher
- Command Set
- Card Life Cycle

Background

- Native COS is how COS has been developed since the 1980's.
- It is still being used today due to its cost-effectiveness and performance (e.g. GSM SIM, EMV, etc.).
- All CPU contact cards conform to ISO 7816-1,2,3, while all CPU contactless cards conform to ISO 14443-1,2,3,4.
- A smart card terminal communicates with smart cards via APDU commands, and does not know or care whether the card is a native COS, a Java card or a MultOS, etc.

Native COS Architecture



Advanced Card Systems Ltd.
Card & Reader Technologies

Memory & File Management

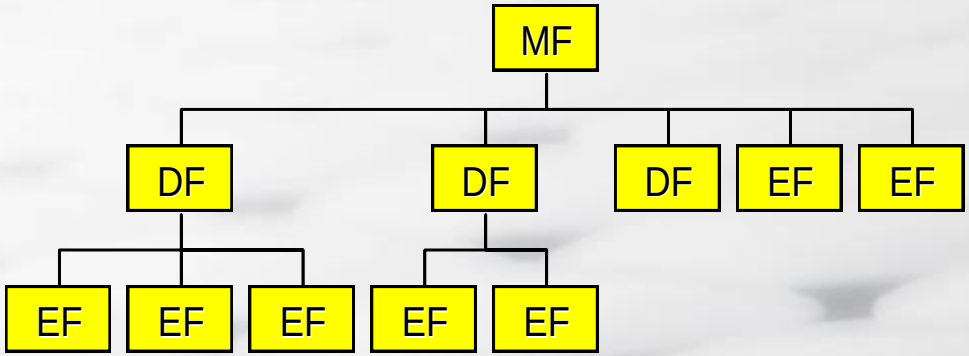
CARD & READER TECHNOLOGIES

Native COS Memory

- The size of a native COS refers to the size of the EEPROM (or flash).
- The EEPROM contains purely application data, system data, secret keys and secret code; it does not contain any executable programming code.
- The size excludes the COS, usually in ROM and sometimes in flash.
- The EEPROM is a contiguous block of memory that becomes files using memory and file management

Card File Architecture

- The card is organized into files.
- MF (Master File) is the root of the file structure. It can be seen as a main directory.
- DF (Dedicated File) is a file which contains other files. It can be seen as a directory, where each DF will behave like an independent card.
- EF (Elementary File) is a file containing data. It has various file types namely, transparent, fixed record, variable records, cyclic and internal file (e.g. key files, purse files).

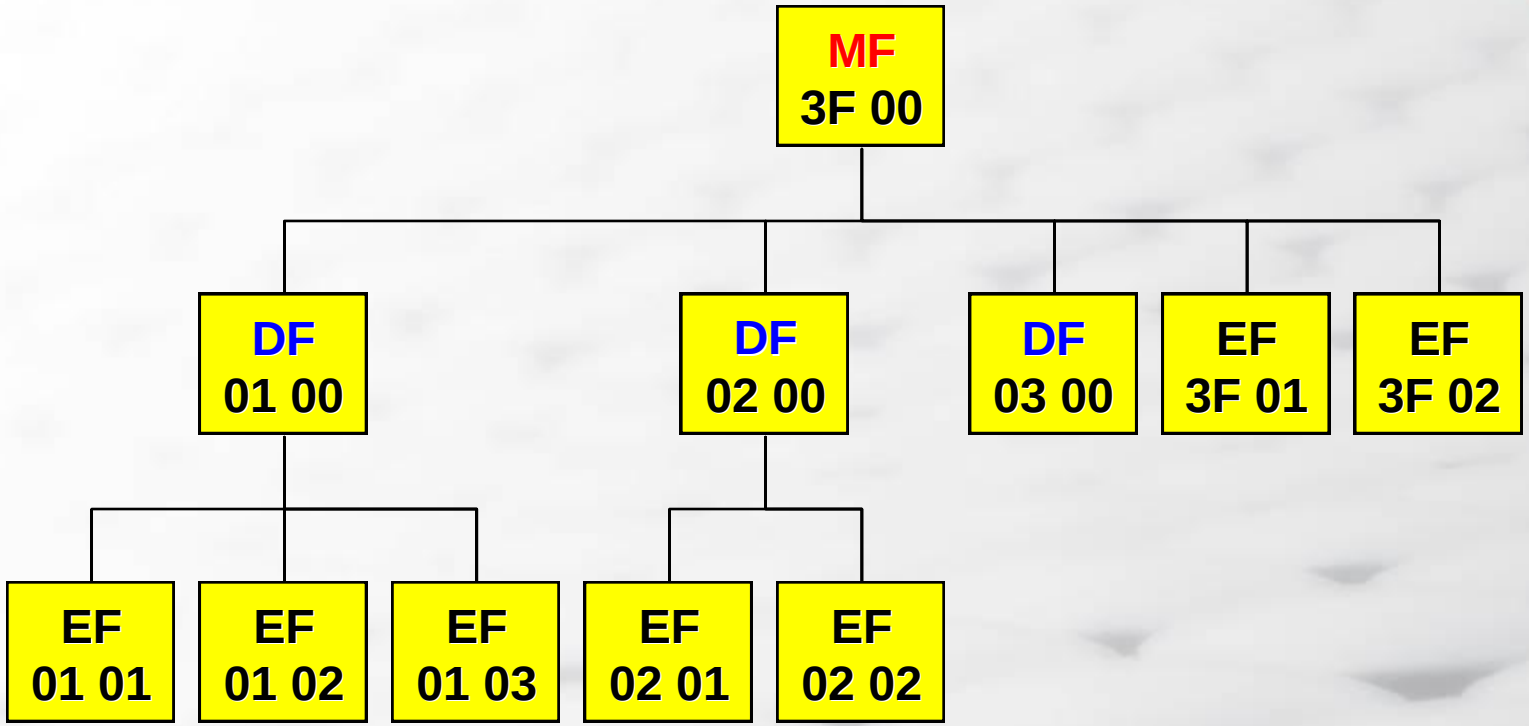


In the 7816-4, a DF may even contain another DF.

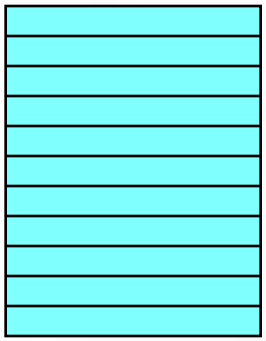
Files (MF, DF, EF)

- A file has 2 bytes of file ID.
- A file has a header and a body.
- The header describes the file (eg. ID, file type, size, access control, status, etc.).
- ISO 7816-4 specifies that the Identifier of the MF is to be 3F 00.

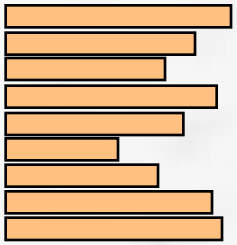
Example of File Identifiers



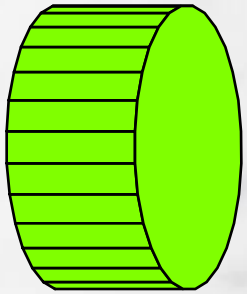
Elementary File Structures



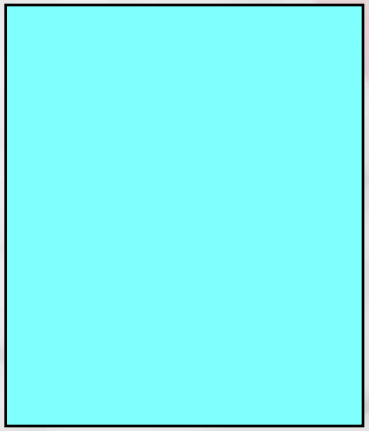
Linear fixed



Linear variable



Cyclic



Transparent

- ISO 7816-4 defines four different types of files.
- The file types can have other COS internal files such as secret code, keys, counters and purse files.

File Access Control

- **EF access can be:**
  - Plain read, write, update
  - MAC-ed read, write update
  - Ciphered read, write update
  - Initially accessible for personalization, but after which, is locked
- **An access condition is assigned to each possible access**
- **It is COS-checked to ensure that the access condition has been achieved before the access is granted.**
- **Access is made via APDU commands (e.g. Read / Write / Update Binary, Read / Write / Update Record)**

Advanced Card Systems Ltd.
Card & Reader Technologies

Security Management

CARD & READER TECHNOLOGIES

Secret Codes/Keys

- **Keys are used to protect file access in read / write / update.**
- **Keys can also protect sensitive actions:**
  - Creation / management of files relating to the card security
  - One key for one purpose
- **Keys are kept in internal EF's for usage by the COS.**
- **Each key, besides its key content, has a key descriptor to describe its behavior.** (e.g. pre-usage conditions, post-usage conditions, access conditions, and capabilities like encrypt, decrypt, verify, credit and debit.)
- **Successful key operation changes the COS state, thus enabling APDU commands to perform what was previously not allowed.**

Authorization Register & Authorization Mask

- AR is a variable that is initialized to zero upon reset.
- With a Select DF command, it can be masked with an Authorization Mask (AM) to implement global and local authorization.
- AM is one of the DF descriptor parameters.

Secret Codes

- A Secret Code is like a password, presented in plain or ciphered.
- Ciphered presentation is done using a session key.
- A session key is established using non-replayable data such as random numbers or counters, with one coming from the card and the other from the external environment, and a common reference key.
- The key is unique for each and every card, achieved through a key diversification algorithm of a card unique data and a master key.

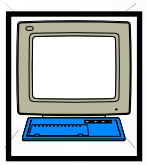
Keys

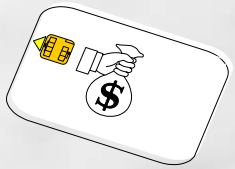
- **Keys are used by triple DES or AES cryptographic functions.**
- **3DES keys may be 128 bits or 192 bits.**
- **AES keys may be 128 bits, 192 bits or 256 bits.**
- **Keys are to be kept in the internal EF.**
- **Example of keys in a smart card:**
  - Internal Authentication Key
  - External Authentication Key
  - Mutual Authentication Key
  - Signature Key
  - Credit Key
  - Debit Key
  - And more…

Secure Messaging

- Ensures that what is sent to the card is from the authorized source and has not been tampered
- Ensures that the response from the card is indeed coming from the card and has not been tampered
- Achieved using a CMAC and encryption if confidentiality is required
- Secure messaging is shown by a CLA (class) byte set at 04 or 84, and can be used for ISO and Proprietary commands

Example:
Command that sends data to the card





04/84	INS	P1	P2	Lc+3	Data+CRYCKS2-0	Le=3

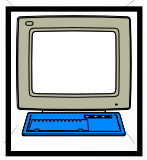
The application calculates the **CRYCKS** and sends the 3 least significant bytes to the card with the command APDU.

The card calculates the **CRYCKS**, checks its integrity and returns the 3 most significant bytes.

CRYCKS7-5	90 00

CRYCKS = The Cryptographic Checksum of the command APDU obtained by a 3DES computation

Example:
Command that retrieves data from the card



04/84	INS	P1	P2	Empty	Empty	Le+3

The card calculates the **CRYCKS** and sends the 3 most significant bytes to the host with the response APDU.

Data	CRYCKS7-5	90 00

CRYCKS = The Cryptographic Checksum of the command + response APDU obtained by using DES

Command Dispatcher

Command Dispatching:

```
public void process( APDU apdu )   {
   switch(apduBuffer[ISO7816.OFFSET_INS])
      {  case INS_BIN_READ:
         case INS_BIN_UPDATE:
            ProcessFileCommand(apdu);
            break;
         case INS_SET_STATUS:
            ProcessSetStatus(apdu);
            break;
         case INS_VERIFY_PIN:
            VerifyPIN(apdu);
            break;
         case INS_PUT_KEYS:
            PutKeys(apdu);
            break;
               ……
               default:
            ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
      }
}
```

APDU Command Set

- The APDU Command Set can be classified into:
  - Proprietary administrative commands
  - ISO-7816 part 4 commands
  - Applications-related commands:
    - Secure portable file
    - E-purse
    - PKI

Administrative Command Set

- Manufacturer Initialization
- Creation of files
  - MF
  - DF
  - EF
- Deletion of files (usually last created is first deleted)
- Freezing / locking of access conditions
  - Passing of control from main to sub-issuer
  - Changing of unsecured access to secured access
  - Locking of access
- Changing the status of the card life cycle

Important Secure Portable File: ISO 7816 Part 4 Commands

- Verify
- Disable Verification
- **Unblock**
- Get Challenge
- Internal Authentication
- External Authentication
- **Establish Session Key**
- Select File
- Read Binary / Record
- Write Binary / Record
- Update Binary / Record
- Erase Binary / Record
- Get Response

E-Purse Command

- Read Balance with balance MAC
- Debit returning debit certificate counter-signing terminal certificate, logging transaction in an atomic process
- Debit signature
- Incremental debit
- Credit with credit certificate, logging transaction in an atomic process
- Credit signature

PKI Command

- Generate Asymmetric Key Pair
- Asymmetric Public Encrypt
- Asymmetric Private Encrypt
- Asymmetric Public Decrypt
- Asymmetric Private Decrypt
- Symmetric Encrypt
- Symmetric Decrypt

Card Related Application Design Using Native COS

- Know your application requirements (e.g. portable file, purse requirement, PKI requirement).
- Choose the right COS that meets your requirements.
- Fully understand the COS.
- Design the card mapping & key management.
- Design the SAMs APDU command set complementing the COS security.
- Design SAMs card mapping.
- Design the SAM-Application-Card APDU transaction flow for all subsystems.
- Provide test cards and test SAMs to each subsystem vendor, so that each subsystem vendor knows how to use the card and the SAM.

Questions?