Advanced Card Systems Ltd.
Card & Reader Technologies

# ACOS7 MOC Card

Functional Specifications

*Subject to change without prior notice*

info@acs.com.hk
**www.acs.com.hk**

# Table of Contents

# Figures

# Tables

# 1.0. Introduction

The purpose of this document is to describe in detail the features and functions of the ACS Smart Card Operating System Version 7 (ACOS7) developed by Advanced Card Systems Ltd.

## 1.1. Features

ACOS7 provides the following features:

- Compliance with ISO 7816 Parts 1, 2, 3, 4
- For contact interface, the switchable baud rate from 9600 to 115200 bps is supported
- Compliance with ISO 14443 Parts 1, 2, 3, 4
- For contactless interface, it is fully compatible with ISO 14443 A
- For contactless interface, the T=CL protocol according ISO 14443-4 is supported
- For contactless interface, it has data transfer support for:
  - 106 kbps, 212 kbps, 424 kbps and 848 kbps
- Full 8 KB of EEPROM for application data
- ISO 7816 Part 4 file structures support: Transparent, Linear fixed, Linear Variable, Cyclic.
- DES / Triple DES capability
- Hardware based random number generator compliant to FIPS 140-2
- Secure Messaging function ensuring data transfers are confidential and authenticated
- PBOC e-Purse and e-Deposit payment application support
- Compliance with Ministry of Construction (MoC) Standard
- Compliance with the technical requirements for chip operating system of CPU card in construction case
- Multi-level secured access hierarchy
- Anti-tearing function support

## 1.2. Technical Specifications

The following are some technical properties of the ACOS7 card:

### 1.2.1. Electrical

- Operating voltage: 5 V DC+/-10% (Class A) and 3 V DC +/-10% (Class B)
- Maximum supply current: < 10 mA
- ESD protection: ≤ 4 KV

### 1.2.2. EEPROM

- Capacity: 8 KB (8,192 bytes)
- EEPROM endurance: 100K erase/write cycles
- Data retention: 10 years

### 1.2.3. Environmental

- Operating temperature: -25 °C to 85 °C
- Storage temperature: -40 °C to 100 °C

## 1.3. Symbols and Abbreviations

| | |
|---|---|
| 3DES | Triple DES |
| AID | Application / Account Identifier |
| AMB | Access Mode Byte |
| AMDO | Access Mode Data Object |
| APDU | Application Protocol Data Unit |
| ATC | Account Transaction Counter |
| ATR | Answer To Reset |
| CHV | Card Holder Verify |
| COMPL | Bit-wise Complement |
| COS | Card Operating System |
| DEC (C, K) | Decryption of data C with key K using DES or 3DES |
| DES | Data Encryption Standard |
| DF | Dedicated File |
| ED | Electronic Deposit |
| ENC (P, K) | Encryption of data P with key K using DES or 3DES |
| EF | Elementary File |
| EF1 | PIN File |
| EF2 | KEY File |
| FCI | File Control Information |
| FCP | File Control Parameters |
| FDB | File Descriptor Byte |
| LCSI | Life Cycle Status Integer |
| GSESPK | Session key of Grey Lock |
| ID | Identifier |
| INS | Instruction Byte of Command Message |
| LCSI | Life Cycle Status Integer |
| LEN | Length |
| LSb | Least Significant Bit |
| LSB | Least Significant Byte |
| MAC | Message Authentication Code |
| MF | Master File |
| MOC | Ministry of Construction |
| MRL | Maximum Record Length |
| MSb | Most Significant Bit |
| MSB | Most Significant Byte |
| NA | No Application |
| NOR | Number of Record |
| PBOC | Peoples Bank of China |
| PIN | Personal Identification Number |
| PSE | Payment System Environment |

| | |
|---|---|
| RFU | Reserved for Future Use |
| RMAC | Retail MAC |
| SAM | Security Authentication Module |
| SC | Security Condition |
| SCB | Security Condition Byte |
| SFI | Short File Identifier |
| SM-MAC | Secure Messaging with MAC |
| SM-ENC | Secure Messaging with Encryption |
| SW1 | Status Word One |
| SW2 | Status Word Two |
| TAC | Transaction Authorization Cryptogram |
| TC | Transaction Counter |
| TLV | Tag-Length-Value |
| TTI | Transaction Type Indicator |
| UQB | Usage Qualifier Byte |
| \|\| | Concatenation |

# 2.0. Card Management

This section outlines the card level features and management functions of the ACOS7 smart card.

## 2.1. Anti Tearing

The ACOS7 uses an *anti-tearing* mechanism in order to protect the card from data corruption due to card tearing which happens when the card is suddenly pulled out of reader during data update or when the reader suffers from mechanical failure during the card data update. On card reset, ACOS7 looks at the Anti-Tearing fields and does the necessary data recovery. In such case, the COS will return the saved data to its original address in the EEPROM.

## 2.2. Card Life Cycle States

ACOS7 has the following card states:

1. **Pre-Personalization State** – This is the initial state of the card.
2. **Personalization State** – The card goes into this state once the Master File (MF) is successfully created. During this state, the user can create and test the different files created in the card.

   It is also important to note that the user can perform tests under this state and may revert to the Pre-Personalization State by using the *Clear Card* command found in **Section 6.23 Clear Card**.
3. **User State** – After creating the desired file structure of the card, the user can then send the *Activate Card* command found in **Section 6.12 Activate Card** so that the card will go to the User State.

   After successfully running the *Activate Card* command, the *Clear Card* command will be disabled and the card can no longer go back to the previous states.

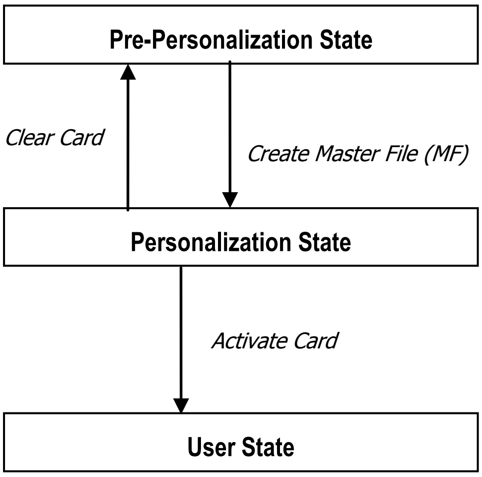

**Figure 1:** Card Life Cycle States

### 2.2.1. Typical Development Steps of Card:

1. During the Personalization State, the user creates his card file structure, starting with the Master File (MF). The Dedicated File (DF) and different types of Elementary Files (EF) are then created. Furthermore, the card's security design is tested in this state. If design flaws are found, the user can always return to the Pre-Personalization State using the *Clear Card* command.
2. Once the card's file and security design are final and have been thoroughly tested, the user can then send the *Activate Card* command to disable the *Clear Card* command.
3. The card then goes into the User State and can no longer go back to previous states.

## 2.3. Answer to Reset (Contact Card)

After a hardware reset (e.g. power up) is performed, the card transmits an Answer-to-Reset (ATR) in compliance with ISO 7816 Part 3. Furthermore, it is important to note that the ACOS7 supports the T=0 protocol.

**Note:** For full description of the ATR, kindly refer to ISO 7816 Part 3.

## 2.4. Answer to Select (Contactless Card)

After receiving a Request for Answer to Select (RATS) command from the application, the card transmits an Answer to Select (ATS) in compliance with ISO 14443 Part 4.

**Note:** For full description of the ATS, kindly refer to ISO 14443 Part 4.

# 3.0. File System

This section explores the file system of the ACOS7 smart card.

## 3.1. Hierarchical File System

The ACOS7 is fully compliant to ISO 7816 Part 4 file system and structure. The file system is similar to that of the modern computer operating system. The root of the file system is the Master File (of MF). Each application or group of data files in the card can be contained in a directory called a Dedicated File (DF). The Elementary Files (EF) can be stored in the MF or the DF.

Furthermore, the ACOS7 allows arbitrary depth DF tree structure, which means that the DFs can be nested. Please see Figure 2 an example of the ACOS7 File System Hierarchy:

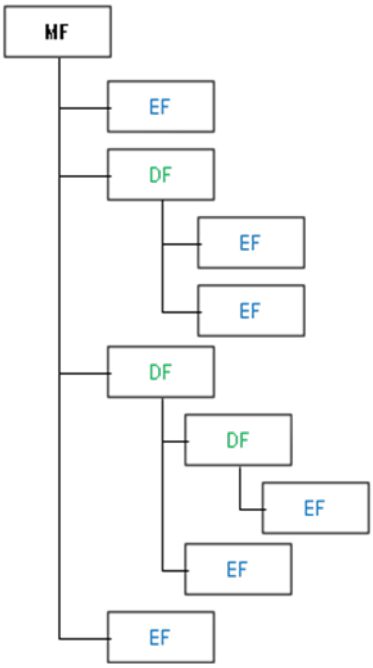

**Figure 2:** Example of Hierarchy of File System

## 3.2. File Header Data Structure

ACOS7 organizes the user EEPROM area by files. Every file has a File Header, which is a block of data that describes the file's properties.

### 3.2.1. Master File

The Master File has the following file header data structure:

| File Header | No. of Bytes | Description |
|---|---|---|
| File Descriptor Byte (FDB) | 1 | This field indicates the file type:<br>Master File: `3F` |
| File ID | 2 | Note that the MF has a constant File ID which is `3F 00`. |
| FCI SFI | 1 | This field is the Short File ID of the FCI (File Control Information). |
| Issuer FCI SFI | 1 | This field is the Short File ID of the Issuer's FCI (File Control Information). |
| Access Condition | 1 | This byte contains the access condition under Current DF. |
| MF Name | 5-16 | For the MF, this field is the Long Name. The MF can be selected through its long name, which can be up to 16 bytes. |

### 3.2.2. Dedicated File

The Dedicated File has the following file header data structure:

| File Header | No. of Bytes | Description |
|---|---|---|
| File Descriptor Byte (FDB) | 1 | This field indicates the file type:<br>Dedicated File: `38` |
| File ID | 2 | This field uniquely identifies a file under the MF.<br>The user can assign any file ID to the DF to uniquely identify it. |
| FCI SFI | 1 | This field is the Short File ID (SFI) of the FCI (File Control Information). |
| Issuer FCI SFI | 1 | This field is the Short File ID of the Issuer's FCI (File Control Information). |
| Access Condition | 1 | This byte contains the access condition under Current DF. |
| DF Name | 5-16 | For the DF, this field is the Long Name. The DF can be selected through its long name, which can be up to 16 bytes. |

### 3.2.3. Elementary File: Transparent /Binary File

***Transparent*** or ***Binary File*** defines the data that is managed as a stream of bytes, which are addressed by an offset coming from the start of file.

### 3.2.4. Elementary File: Linear Fixed File

***Linear Fixed File*** is the data grouped into records, which is a block of bytes with a pre-defined size. Likewise, data fields that are related are grouped into one record.

### 3.2.5. Elementary File: Linear Variable File

***Linear Variable File*** is similar with *Linear Fixed Variable*, except that each record in *Linear Variable File* has variable sizes.

### 3.2.6. Elementary File: Cyclic File

***Cyclic File*** is similar with *Linear Fixed Variable*, but it logically has no “last record.” An application views this file as having no limit, but in reality, the oldest record is overwritten with the newest record in the file.

### 3.2.7. Elementary File: CAPP File

The CAPP File is specifically used for CAPP Purchase. After creating the CAPP File, you need to append the data to the CAPP record to it using the Append Record or Update command.

### 3.2.8. Elementary File: PIN File

The PIN File is used for Access Control by using the *Verify PIN* command.

It is important to note that one DF can contain one PIN File. However, you can create different PIN records under this PIN file and each record can be identified by the PIN identifier byte which will be discussed in the next section.

### 3.2.9. Elementary File: Grey Lock File

The Grey Lock File is a specific file used for MOC commands. The purpose of the Grey Lock File is for e-Purse Transactions, such as saving MAC, Transaction Log files and others related to the different e-Purse transactions.

### 3.2.10. Elementary File: Key File

The Key File is used for Access Control and is needed for various authentication commands.

It is important to note that one DF can only contain one Key File. However, you can create different Key records under this Key file. Each of these records can be identified by the Key Purpose and Key Index which will be discussed in the next section.

### 3.2.11. Elementary File: Electronic Deposit File

The Electronic Deposit (ED) File is specifically used for e-Deposit transactions.

### 3.2.12. Elementary File: Electronic Purse File

The Electronic Purse (EP) File is specifically used for e-Purse transactions.

### 3.2.13. Elementary File: Transaction Log File

The Transaction Log File is specifically used to store e-Deposit and e-Purse transactions.

# 4.0. Security

The different file commands are restricted by the Card Operating System (COS) depending on the target file’s security Access Conditions (AC). These conditions are based on PINs and Keys being maintained by the system. Card Commands are allowed if certain PINs or KEYs are submitted or authenticated.

Global PINs are PINs residing in a PIN EF, directly under the MF. Likewise, Local Keys are KEYs residing in a KEY EF, under the currently selected DF. There can be a maximum of 14 Global PINs, 14 Local PINs, 14 Global Keys, and 14 Local Keys at a given time.

## 4.1. File Security Attributes

Each MF and DF has a one-byte Access Condition (AC) byte for creating file access conditions. On the other hand, each EF file has a three-byte Access Condition for read and update access condition.

## 4.2. Secure Messaging

There are two Secure Messaging (SM) modes available for ACOS7, namely:

1. Secure Messaging with MAC (SM-MAC) – This ensures the authenticity of command.
2. Secure Messaging with Data Encryption and MAC (SM-ENC) – This ensures the confidentiality of command.

The table below summarizes the difference of the SM-MAC and SM-ENC:

| **SM-MAC** | |
|---|---|
| **Command** | **Key for SM** |
| Card Block<br>Application Block<br>Application Unblock<br>Update Record<br>Update Binary<br>Read Record<br>Read Binary | DAMK |
| PIN Unblock | DPUK |
| Reload PIN | DRPK |

| **SM-ENC** | |
|---|---|
| **Command** | **Key for SM** |
| Update Record<br>Update Binary | DAMK |
| PIN Unblock | DPUK |

**Table 1:** Command Support for Secure Messaging

## 4.3. Mutual Authentication

***Mutual Authentication*** is a process in which both card and card-accepting device verify that the respective unit is genuine. A ***Session Key*** is the result of a successful execution of mutual authentication and the *Session Key* is only valid during a “session”. A “session” is defined as the time after a successful execution of the mutual authentication procedure and a reset of the card or the execution of another mutual authentication procedure.

## 4.4. Key Injection

**Key Injection** can be used to securely load a key or a diversified key from an ACOS6-SAM card into a client ACOS7 card.  For the purpose of key injection, we shall refer to the ACOS6-SAM card with the key to inject as the “source SAM” and the ACOS7 card to receive the key the “target SAM”.

This function allows a ‘master and subordinate’ SAM relationship and the subordinate SAM can perform different specific operations.

The target SAM uses the *Set Key* command while the source SAM will use the *Get Key* command to perform key injection.

## 5.0. Life Support Application

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. ACS customer using or selling these products for use in such applications do so on their own risk and agree to fully indemnify ACS for any damages resulting from such improper use or sale.

# 6.0. Contact Information

For additional information please visit http://www.acs.com.hk

For sales inquiry please send e-mail to info@acs.com.hk