Advanced Card Systems Ltd.
Card & Reader Technologies

ACOS10 Contact Card

Functional Specifications V1.03

Subject to change without prior notice

Table of Contents

List of Figures

List of 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 10 (ACOS10) developed by Advanced Card System Ltd.

1.1. Features

ACOS10 provides the following features:

- Full 32 KB of EEPROM for application data
- Compliance with ISO 7816 Parts 1, 2, 3, 4
- High baud rate switchable from 9,600 to 223,200 bps
- Supports ISO 7816 Part 4 file structures: Transparent, Linear Fixed, Linear Variable, Cyclic
- DES/Triple DES capability
- Hardware based random number generator compliant to FIPS 140-2
- Secure Messaging ensures data transfers are confidential and authenticated
- PBOC e-Purse and e-Deposit Certified available for payment applications
- Multilevel secured access hierarchy
- Anti-tearing done on file headers and PIN commands

1.2. Technical Specifications

The following are some technical properties of the ACOS10 Contact 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: 32 Kbytes (32,768 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

Abbreviation	Description
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
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

Abbreviation	Description
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
SAC	Security Attribute – Compact
SAE	Security Attribute – Expanded
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

Table 1: Symbols and Abbreviations

2.0. Card Management

This section outlines the card level features and management functions.

2.1. Anti-tearing Mechanism

ACOS10 uses an ***anti-tearing*** mechanism in order to protect card from data corruption due to card tearing (i.e., card suddenly pulled out of reader during data update, or reader suffer mechanical failure during card data update). On card reset, ACOS10 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 Header Block

ACOS10 is a card operating system that has 32K EEPROM. In its initial state (where no file exists), user can access the card header block by using read/write binary with the indicated address.

2.3. Card Life Cycle States

ACOS10 has the following card states:

1. **Pre-Personalization State** – is the initial state of the card. The user is allowed to freely access the card header block (defined in the last section). The card header block can be referenced by its address using the READ BINARY or UPDATE BINARY command.

   User can personalize the card's Header Block as he wishes.

2. **Personalization State** – card goes into this state once the MF is successfully created and *Card Life Cycle Fuse* is not blown. User can no longer directly access the card's memory as in the previous state. User can create and test files created in the card as if in Operational Mode.

   User can perform tests under this state and may revert to the Pre-Personalization State by using the *Clear Card* command.

3. **User State** – Card goes into this state once the MF is successfully created and *Card Life Cycle Fuse* is blown. Alternatively, users can use the Activate Card command to go from the personalization state to user state.

   The card cannot revert back to previous states when *Card Life Cycle Fuse* is set and bit 5 of Special Function Flags (*Deactivate Card Enable Flag*) is not set. The *Clear Card* and *Deactivate Card* commands are no longer operational.

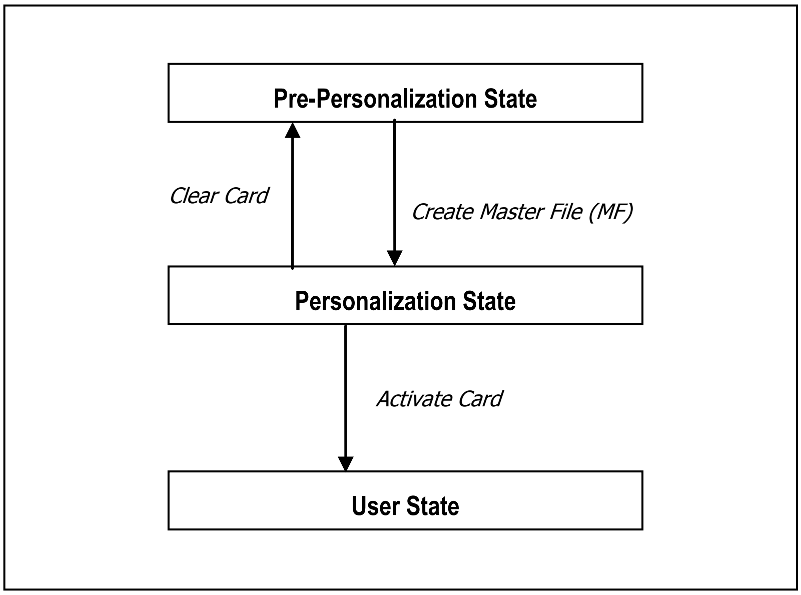

Figure 1: Card Life Cycle States

2.3.1. Typical Steps in Card Development

1. User personalizes the card's header block using UPDATE BINARY.
2. User then creates his card file structure, starting with MF. DF's and EF's are created and the card's security design is tested at this state. If design flaws are found, user can always return to state 1 using the *Clear Card* command.
3. Once the card's file and security design is final and tested, perform *Clear Card* command and blow the *Card Life Cycle Fuse* using the *Update Binary* command.
4. Card goes into Operational Mode, when the MF is created again. User can then re-construct his file system under this state. Card can no longer go back to previous states.

2.4. Answer-To-Reset

After a hardware reset (e.g. power up), the card transmits an Answer-to-Reset (ATR) in compliance with ISO 7816 Part 3, ACOS10 supports the protocol type T=0 in direct convention. The protocol function is not implemented.

The following is the default ATR. For full descriptions of ATR options see ISO 7816 Part 3.

2.4.1. Customizing the ATR

ACOS10's ATR can customize the transmission speed or have specific identification information in the card. The new ATR must be compliant to ISO 7816 Part 3, otherwise the card may become unresponsive and non-recoverable at the next power-up or card reset. Therefore, it is only recommended to change T0 (lower nibble), TA1 and historical bytes.

3.0. File System

This section explores the file system of the ACOS10 Smart Card.

3.1. Hierarchical File System

ACOS10 is fully compliant to ISO 7816 Part 4 file system and structure. The file system is very similar to that of the modern computer operating system. The root of the file 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). Each DF or MF can store data in Elementary Files (EF).

The ACOS10 allows arbitrary depth DF tree structure. That is, the DFs can be nested. Please see figure below.

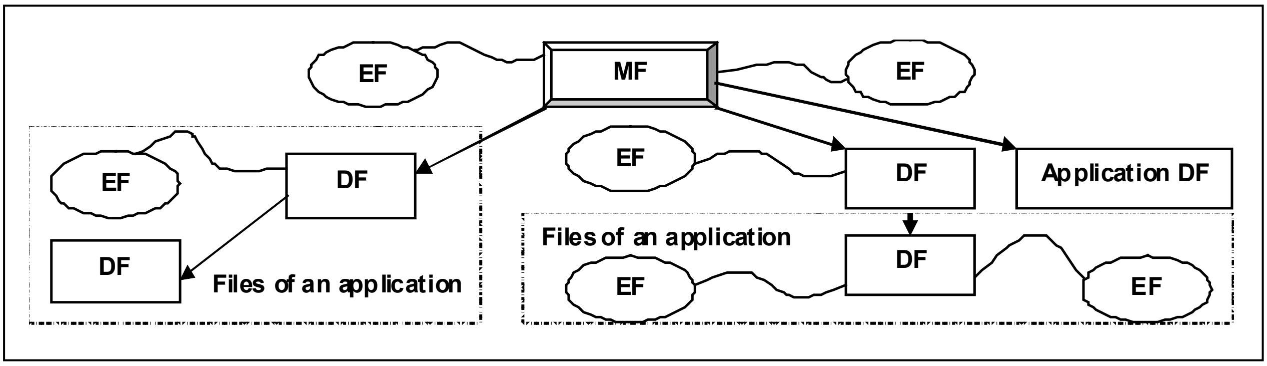

Figure 2: Example of Hierarchy of DFs

3.2. File Header Data Structure

ACOS10 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. Knowledge of the file header block will help the application developer accurately plan for the usage of the EEPROM space.

3.2.1. File Descriptor Byte (FDB)

This field indicates the file's type. The size of the File Header block varies depending on the file type.

3.2.2. Data Coded Byte (DCB)

ACOS10 does not use this field. It is part of the header to comply with ISO 7816 Part 4.

3.2.3. File ID

This is a 16-bit field that uniquely identifies a file in the MF or a DF. Each file under a DF (or MF) must be unique.

3.2.4. File Size

This is a 16-bit field that specifies the size of the file. It does not include the size of the file header. For record-based EF's, the first byte indicates the maximum record length (MRL), while the second indicates the number of records (NOR). For non record-based EF (Transparent EF), the first byte represents the high byte of the file size and the second is the low-order byte. For DF's, this field is not used.

3.2.5. Short File Identifier (SFI)

This is a 5-bit value that represents the file's Short ID. ACOS10 allows file referencing through SFI. The last five bits of the File ID does not necessarily have to match this SFI. Two (2) files may have the same SFI under a DF. In such case, ACOS10 will select the one created first.

3.2.6. Life Cycle Status Integer (LCSI)

This byte indicates the life status of the file, as defined in ISO 7816 part 4. It can have the following values:

b8	b7	b6	b5	b4	b3	b2	b1	Hex	Meaning
0	0	0	0	0	0	0	1	01	Creation state
0	0	0	0	0	1	0	1	05	Operational state (activated)
0	0	0	0	0	1	0	0	04	Operational state (deactivated)
0	0	0	0	1	-	-	-	08 – 0F	Termination state

Table 2: Life Cycle Status Byte

- In Creation/Initialization states, all commands to the file will be allowed by the COS.
- In Activated state, commands to the file are allowed only if the file's security conditions are met.
- In Deactivated state, most commands to the file are not allowed by the COS.
- In Terminated State, all commands to the file will not be allowed by the COS.

3.2.7. Security Attribute Compact Length (SAC Len)

This byte indicates the length of the SAC structure that is included in the file header below.

3.2.8. Security Attribute Expanded Length (SAE Len)

This byte indicates the length of the SAE structure that is included in the file header below.

3.2.9. DF Name Length/First Cyclic Record

If the file is a DF, this field indicates the length of the DF's Name.

If the file is a Cyclic EF, this field holds the index of the last-altered record.

Otherwise, this field is not used.

3.2.10. Parent Address

Two (2) bytes indicating the physical EEPROM address of the file's parent DF.

3.2.11. Checksum

To maintain data integrity in the file header, a checksum is used by the COS. It is computed by XOR-ing all the preceding bytes in the header. Commands to a file will not be allowed if the file is found to have a wrong checksum.

3.2.12. Security Attribute Compact (SAC)

This is a data structure that represents security conditions for certain file actions. The data is coded in an "AM-SC" template as defined in ISO-7816. The maximum size of this field is 8 bytes. See **Section 5.1.1** for more information.

3.2.13. Security Attribute Expanded (SAE)

This is a data structure that represents security conditions for certain card actions. The data is coded differently from SAC, and is also defined in ISO 7816. The maximum size of this field is 32 bytes. See **Section 5.1.2** for more information.

For DF files, additional fields are included in the file header.

3.2.14. SE File ID (for DF only)

For DF, this field is made up of 2 bytes containing the File ID of one of its children. That file is known as the Security Environment File, which is processed internally by the COS.

3.2.15. FCI File ID (for DF only)

For DF, this field is made up of 2 bytes containing the SFIs of FCI File and Issuer FCI File of its children.

3.2.16. DF Name (for DF only)

For DF, this field is the file's Long Name. Files can be selected through its long name - which can be up to 16 bytes.

3.3. Internal Security Files

The behavior of the COS will depend on the contents of the security-related internal files. When internal files are activated, its READ condition should be set to NEVER. Typically, a DF should have: (1) a Internal Linear Variable File (FDB = 0C) to hold PIN codes for verification, (2) a Internal Linear Variable File to hold KEY codes for authentication, and (3) an SE file to hold security conditions.

An Internal file may contain (1) PIN data structure or (2) KEY data structure.

4.0. PBOC Application

4.1. PBOC File

PBOC file can be an Electronic Purse (EP)/Electronic Deposit (ED) File. The transaction log file with fixed record length equals to 23 and has maximum 20 records. Each DF can contain only one PBOC file.

The access condition of *ED Get Balance* command is the same as Read Access condition of PBOC File. *EP Get Balance* command is free to access.

The PBOC Transaction is following the standards:

1. China Financial Integrated Circuit Card Specifications – Part 1 Electronic Purse/Electronic Deposit Card Specification
2. China Financial Integrated Circuit Card Specifications – Part 2 Electronic Purse/Electronic Deposit Application Specification

4.2. PBOC Transaction

For more details about PBOC Transaction, please refer to China Financial Integrated Circuit Card Specifications – Part 2 Electronic Purse/Electronic Deposit Application Specification.

5.0. Security

File commands are restricted by the COS depending on the target file's (or current DF's) security Access Conditions. These conditions are based on PINs and KEYs being maintained by the system. Card Commands are allowed if certain PIN's or KEY's are submitted or authenticated.

Global PIN's are PINs that reside in a PIN EF (EF1) directly under the MF. Likewise, local Keys are KEYs that reside in a KEY EF (EF2) under the currently selected DF. There can be a maximum of: 31 Global PINs, 31 Local PINs, 31 Global Keys, and 31 Local Keys at a given time.

5.1. File Security Attributes

Each file (MF, DF, or EF) has a set of security attributes set in its headers. There are two types of security attributes Security Attribute Compact (SAC) and Security Attribute Expanded (SAE).

5.1.1. Compact (SAC)

The SAC is a data structure that resides in each file. It indicates what file actions are allowed on the file, and what conditions need to be satisfied for each action.

- The SE record is found in the SE file - whose ID is specified in the current DF's header.

5.1.2. Expanded (SAE)

The SAE is a data structure that resides in each file. It tells the COS whether or not to allow file commands to proceed. SAE is more general compared to SAC. The format of SAE is an access mode data object (AMDO) followed by one or more security condition data objects (SCDO).

5.2. Security Environment

Security conditions are coded in an SE File. Every DF has a designated SE FILE, whose file ID is indicated in the DF's header block. Each SE record has the following format:

<SE ID Template> <SE Authentication Template>

SE ID Template: The SE ID Template is a mandatory data object whose value states the identifier that is referenced by the SC byte of the SAC and SAE.

SE Authentication Template: The Authentication Template (AT) defines the security condition that must be meant for this SE to be satisfied. The security conditions are either PIN or Key authentications.

5.3. External Authentication

External authentication uses a card challenge and terminal response method to gain authorization to the card.

5.4. Secure Messaging

There are two Secure Messaging (SM) modes available for ACOS10, 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 Application Block Application Unblock Update Record Update Binary Read Record Read Binary	DAMK
PIN Unblock	DPUK
Reload PIN	DRPK

SM-ENC	
Command	Key for SM
Update Record Update Binary	DAMK
PIN Unblock	DPUK

Table 3: Command Support for Secure Messaging

5.5. Mutual Authentication

Mutual Authentication is a process in which both the card and the card-accepting device verify that the respective entity is genuine. A *Session Key* is the result of a successful execution of mutual authentication. 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.

5.6. Key Injection

Key Injection can be used to securely load a key or diversified key from an ACOS6-SAM card into client ACOS10 card. For the purpose of key injection, we shall refer to the ACOS6-SAM with the key to inject the "*source SAM*" and the ACOS10 to receive the key the "*target SAM*".

This function allows for a master and subordinate SAM relationships and the subordinate SAMs can perform different specific operations.

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

6.0. 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 customers 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.

7.0. Contact Information

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

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