Advanced Card Systems Ltd.
Card & Reader Technologies

ACOS10 (Contact)

Functional Specifications V2.04

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

Table of Contents

1.0. Introduction 4

1.1. History of Modification for ACOS10 Contact 4
1.2. Symbols and Abbreviations 4

2.0. Technical Specifications 6

2.1. Electrical 6
2.2. Environmental 6
2.3. Communication Protocols 6
2.4. Memory 6
2.5. Cryptographic Capabilities 6
2.6. File Security 6
2.7. Answer to Reset (ATR) 6
2.8. Compliance to Standards 7

3.0. Card Management 8

3.1. Card Life Cycle States 8
3.1.1. Pre-Personalization State 8
3.1.2. Personalization State 8
3.1.3. User State 8
3.2. Card Header Block 8
3.3. Typical Steps in Card Development 9
3.4. Answer To Reset (ATR) 9
3.4.1. Customizing the ATR 9

4.0. File System 10

4.1. Hierarchical File System 10
4.2. File Header Data Structure 11
4.2.1. File Descriptor Byte (FDB) 11
4.2.2. Data Coded Byte (DCB) 11
4.2.3. File ID 11
4.2.4. File Size 11
4.2.5. Short File Identifier (SFI) 11
4.2.6. Life Cycle Status Integer (LCSI) 11
4.2.7. Security Attribute Compact Length (SAC Len) 11
4.2.8. Security Attribute Expanded Length (SAE Len) 11
4.2.9. DF Name Length/First Cyclic Record 12
4.2.10. Parent Address 12
4.2.11. Checksum 12
4.2.12. Security Attribute Compact (SAC) 12
4.2.13. Security Attribute Expanded (SAE) 12
4.2.14. SE File ID (for DF only) 12
4.2.15. FCI File ID (for DF only) 12
4.2.16. DF Name (for DF only) 12
4.3. Internal Security Files 12
4.3.1. PIN Data Structure 12
4.3.2. Key Data Structure 13
4.3.3. Security Environment File 13

5.0. Transaction Application 14

5.1. Account File 14
5.2. Transaction 14
5.2.1. Load transaction 14
5.2.2. Unload transaction 14
5.2.3. Purchase/Cash withdraw transaction 14
5.2.4. Update overdraw limit transaction 14

6.0. Security Features 15

6.1. File Security Attributes 15
6.1.1. Compact (SAC) 15
6.1.2. Security Attribute Expanded (SAE) 15
6.2. Security Environment 15
6.3. External Authentication 15
6.4. Secure Messaging 15
6.5. Mutual Authentication 15
6.6. Key Injection 16
6.7. Anti-tearing 16

7.0. Life Support Application 17

8.0. Contact Information 18

List of Figures

Figure 1 : Card Life Cycle States 8
Figure 2 : File System Hierarchy 10

List of Tables

Table 1 : History of Modification for ACOS10 Contact 4
Table 2 : Symbols and Abbreviations 5
Table 3 : ATR Protocol Bytes 7
Table 4 : ATR Historical Bytes 7
Table 5 : Life Cycle Status Byte 11

1.0. Introduction

The purpose of this document is to describe in detail the features and functions of the ACOS10 Contact Card, a versatile smart card operating system developed by Advanced Card System Ltd.

1.1. History of Modification for ACOS10 Contact

Date	Changes
2010/07/09	**ACOS10 revision 6.00** • Support for PBOC 2.0 e-Deposit and e-Purse payment applications
2016/03/18	**ACOS10 revision 6.10** • Updated to new IC platform. All functionality remains the same.

Table 1: History of Modification for ACOS10 Contact

1.2. Symbols and Abbreviations

Term	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
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
EF	Elementary File
EF1	PIN File
EF2	Key File
ENC (P, K)	Encryption of data P with key K using DES or 3DES
FCP	File Control Parameters
FDB	File Descriptor Byte
LCSI	Life Cycle Status Integer
LSb	Least Significant Bit
LSB	Least Significant Byte
MAC	Message Authentication Code

Term	Description
MF	Master File
MSb	Most Significant Bit
MSB	Most Significant Byte
PBOC	Peoples Bank of China
RFU	Reserved for Future Use
RMAC	Retail MAC
SAC	Security Attribute – Compact
SAE	Security Attribute – Expanded
SAM	Security Authentication Module
SCB	Security Condition Byte
SCDO	Security Condition Data Object
SE	Security Environment
SFI	Short File Identifier
SM-MAC	Secure Messaging with MAC
SM-ENC	Secure Messaging with Encryption
TLV	Tag-Length-Value
UQB	Usage Qualifier Byte
||	Concatenation

Table 2: Symbols and Abbreviations

2.0. Technical Specifications

The following are some technical properties of the ACOS10 Contact Card:

2.1. Electrical

- Operating at 5 V DC+/-10% (Class A) and 3 V DC +/-10% (Class B)
- Maximum Supply Current: <10 mA
- ESD Protection: ≤ 4 KV

2.2. Environmental

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

2.3. Communication Protocols

- T=0 with baud up to 223.2 Kbps

2.4. Memory

- Capacity: 32 KB
- EEPROM Endurance: 100,000 erase/write cycles
- Data Retention: 10 years

2.5. Cryptographic Capabilities

- DES, 2K3DES (ECB)

2.6. File Security

- FIPS 140-2 compliant hardware based RNG
- Secure Messaging ensures that data transfers are confidential and authenticated
- E-purse and e-deposit for payment applications
- Multilevel secured access hierarchy
- Anti-tearing done on file headers and PIN commands

2.7. Answer to Reset (ATR)

After a hardware reset (i.e. 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.

Parameter	ATR	Description
TS	3Bh	Direct Convention.
T0	BEh	TA1, TB1, TD1 follows with 14 historical characters.
TA1	95h	Capable of high-speed communication.
TB1	00h	No programming voltage required.
TD1	00h	No further interface bytes follow.

Parameter	ATR	Description
14 Historical Characters		

Table 3: ATR Protocol Bytes

The 14 historical characters are composed of the following:

Historical Characters	ATR	Description
T1	41h	Indicates ACOS Card
T2	10h	Major version
T3	10h	Minor version
T4	00h	
T5	00h	
T6	00h	
T7	00h	
T8	00h	
T9	00h	
T10	00h	
T11	00h	
T12	0x	Card Life Cycle State indicator: Bit1: 1=Perso (or pre-perso) State; 0=User State
T13	90h	Not used. Compatible with ACOS2.
T14	00h	

Table 4: ATR Historical Bytes

The ATR may be completely changed using the ATR file. See **Customizing the ATR** for more information.

2.8. Compliance to Standards

- Compliance with ISO 7816 Parts 1, 2, 3, 4

3.0. Card Management

This section outlines the card level features and management functions.

3.1. Card Life Cycle States

ACOS10 Contact Card has the following card states:

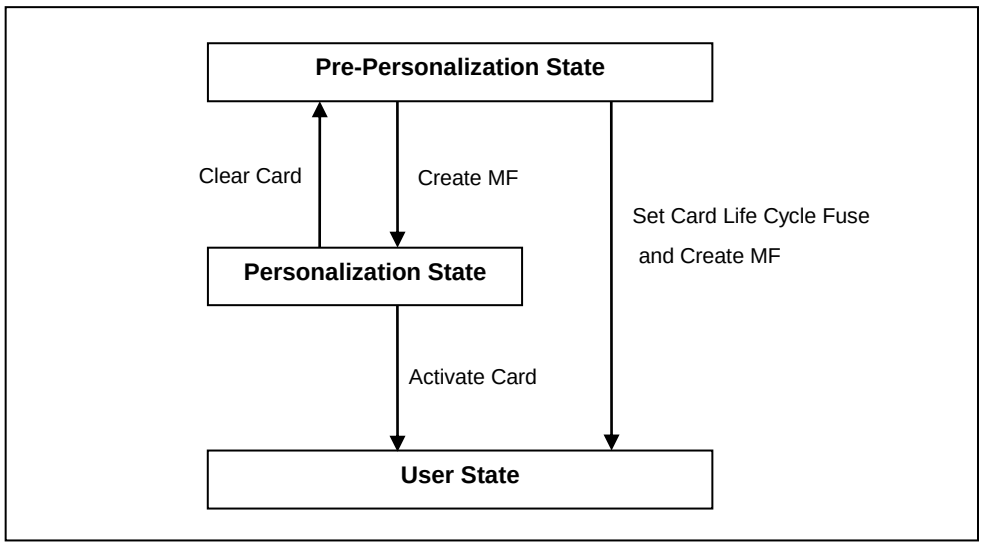

Figure 1: Card Life Cycle States

3.1.1. Pre-Personalization State

This 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. Card remains in this state as long as: (1) MF is not created; and (2) the *Card Life Cycle Fuse (address EEC7)* of the *Card Header Block* is FFh.

3.1.2. Personalization State

The card goes into this state once the MF is successfully created and *Card Life Cycle Fuse* is not blown (still FFh). 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.

3.1.3. 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.

3.2. Card Header Block

ACOS10 is a card operating system that has 32 KB 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.

3.3. 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.

3.4. Answer To Reset (ATR)

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.

3.4.1. 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.

3.4.1.1. Customized ATR for Microsoft Windows Usage

For Windows 7 and above operating systems: Windows automatically attempts to download the smart card's minidriver whenever a smart card is inserted into the smart card reader. Since ACOS10 is not intended to conform to Windows default usage, such smart card minidriver is not necessary. However, if the ACOS10 is inserted into a Windows system, Windows may search online for the driver and may give a warning that the "device driver was not successfully installed" for the smart card. There are two ways to solve this issue:

1. Disable smart card plug and play and certificate propagation in Windows.
2. Change the ATR so Windows will recognize the ACOS10 smart card to use ACS's Unified Null Driver.

For the first solution, please follow instructions in this Microsoft support link to disable smart card plug and play. This may have to be done for every computer that will be used in this system. http://support.microsoft.com/kb/976832.

For the second solution, ACS has developed a Unified Null driver for ACOS line of smart cards. The Unified Null driver will satisfy the Windows requirement to have a minidriver for the card, hence the warning from Windows every time the card is inserted will now be removed. The Unified Null Driver can be downloaded automatically from Windows Update if Automatic Updates are turned ON. In order for Windows to recognize the ACOS10 smart card to use the Unified Null driver, the ATR must be customized. In the case of ACOS10, the ATR should be:

BF XX 00 00 41 43 53 5F 41 43 4F 53 31 30 5F 4E 44 90 00h.

4.0. File System

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

4.1. 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 (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)**.

Furthermore, the ACOS10 allows for an arbitrary depth DF tree structure, which means that the DFs can be nested. Below is an example of the ACOS10 File System Hierarchy:

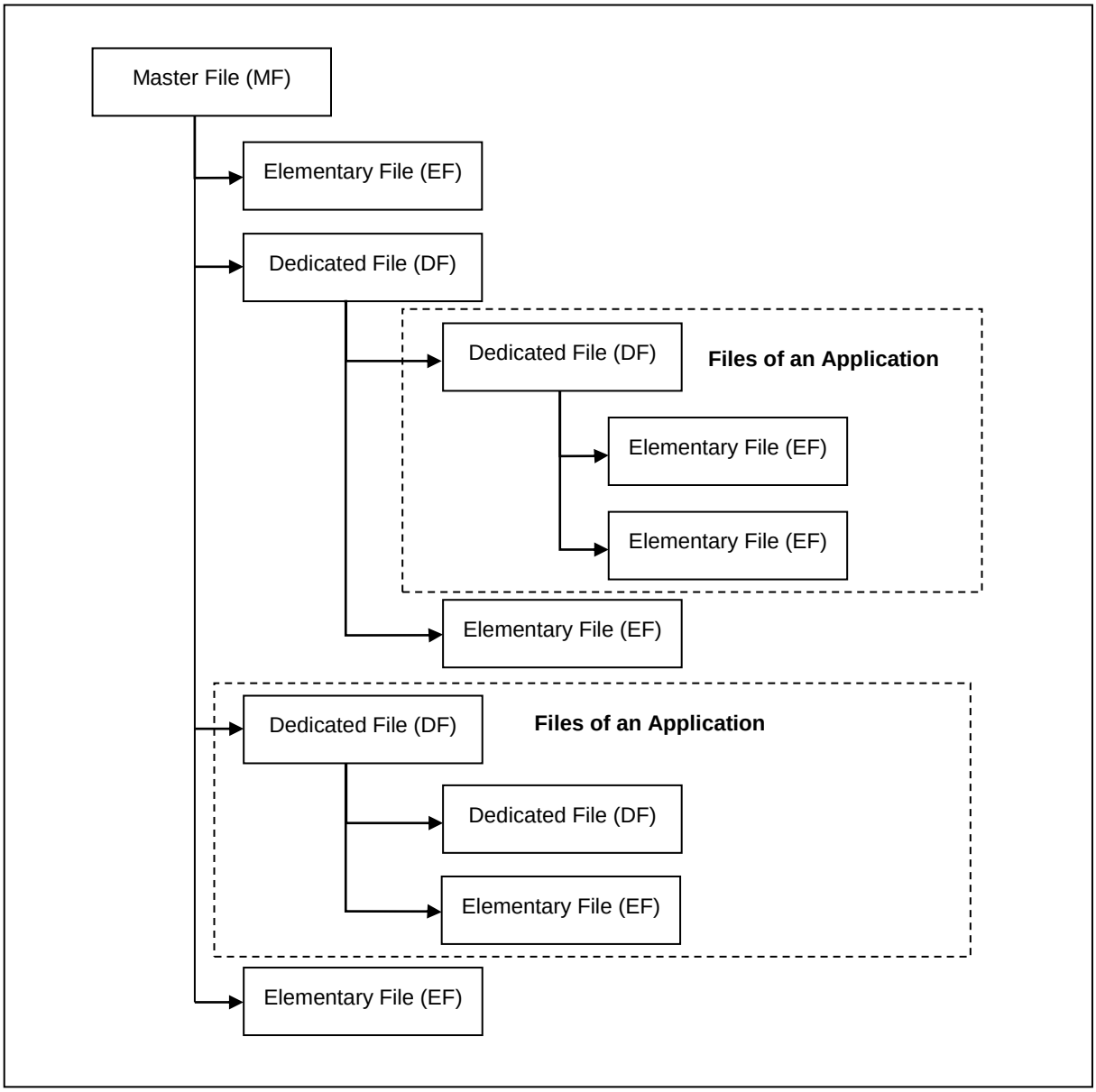

Figure 2: File System Hierarchy

4.2. File Header Data Structure

The 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.

4.2.1. File Descriptor Byte (FDB)

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

4.2.2. Data Coded Byte (DCB)

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

4.2.3. 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.

4.2.4. 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.

4.2.5. Short File Identifier (SFI)

This is a 5-bit value that represents the file's Short ID. The 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.

4.2.6. 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	1	-	-	-	08 – 0F	Termination state

Table 5: 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 Terminated State, all commands to the file will not be allowed by the COS.

4.2.7. Security Attribute Compact Length (SAC Len)

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

4.2.8. Security Attribute Expanded Length (SAE Len)

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

4.2.9. 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.

4.2.10. Parent Address

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

4.2.11. 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.

4.2.12. 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.

4.2.13. 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.

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

4.2.14. 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.

4.2.15. 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.

4.2.16. 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.

4.3. Internal Security Files

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

4.3.1. PIN Data Structure

The PIN is used for VERIFY command. This is used for Card Holder Verification.

4.3.2. Key Data Structure

Keys are used for authentication commands.

4.3.3. Security Environment File

A Security Environment (SE) File is an Internal Linear Variable EF that stores SE definitions. Every DF has a designated SE FILE, whose file ID is indicated in the DF's header block.

5.0. Transaction Application

5.1. Account File

The Account file can be an Electronic Purse (EP)/Electronic Deposit (ED) File. The Transaction log file with fixed record length is equal to 23 and has maximum 20 records.

Each DF can contain only one Account file and the card can only have a maximum of four Account files.

5.2. Transaction

5.2.1. Load transaction

The Load transaction is used to credit the EP or ED balance within the ICC with an authorized amount from a bank account. This transaction is always PIN protected.

5.2.2. Unload transaction

The Unload transaction is used to debit the ED balance within the ICC with an authorized amount which is transferred to a bank account. This transaction is always PIN protected and supported only by an ED sub-application.

5.2.3. Purchase/Cash withdraw transaction

The EP/ED Purchase transaction is performed offline at a POS terminal only. The ED Cash withdraw transaction is performed offline at a bank terminal (ATM). The transaction of ED sub-application is always PIN protected.

5.2.4. Update overdraw limit transaction

The Update Overdraw Limit transaction is used to change the value of the Overdraw Limit authorized by the issuing bank. This transaction is PIN-protected and performed online at a bank terminal.

6.0. Security Features

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 PINs 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.

6.1. 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).

6.1.1. 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.

6.1.2. Security Attribute 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).

6.2. 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.

6.3. External Authentication

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

6.4. 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 (SM-ENC) – This ensures the confidentiality of command.

6.5. 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.

6.6. 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.

6.7. Anti-tearing

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.

7.0. 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.

8.0. Contact Information

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

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