Advanced Card Systems Ltd.
Card & Reader Technologies

ACOS6-SAM

Functional Specifications V1.07

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

Table of Contents

1.0. Introduction 4

1.1. Features 4
1.2. Technical Specifications 4
1.2.1. Electrical 4
1.2.2. EEPROM 4
1.2.3. Environmental 4

2.0. ACOS6-SAM 5

3.0. Card Management 7

3.1. Card Life Cycle States 7
3.1.1. Pre-Personalization State 7
3.1.2. Personalization State 7
3.1.3. User State 7
3.2. Card Header Block 8
3.3. Typical Development Steps of Card 8
3.4. Answer To Reset (ATR) 8
3.4.1. Customizing the ATR 8

4.0. File System 9

4.1. Hierarchical File System 9
4.2. File Header Data Structure 10
4.2.1. File Descriptor Byte (FDB) 10
4.2.2. Data Coded Byte (DCB) 10
4.2.3. File ID 10
4.2.4. File Size 10
4.2.5. Short File Identifier (SFI) 10
4.2.6. Life Cycle Status Integer (LCSI) 10
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 11
4.2.10. Parent Address 11
4.2.11. Checksum 11
4.2.12. Security Attribute Compact (SAC) 11
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

5.0. Security Features 13

5.1. File Security Attributes 13
5.1.1. Security Attribute Compact (SAC) 13
5.1.2. Security Attribute Expanded (SAE) 13
5.2. Security Environment 13
5.3. Mutual Authentication 14
5.4. Short Key External Authentication 14
5.5. Secure Messaging 14
5.5.1. Secure Messaging for Authenticity (SM-MAC) 14
5.5.2. Secure Messaging for Confidentiality (SM-ENC) 14
5.6. Key Injection 14
5.7. Anti-tearing Mechanism 14

6.0. Life Support Application 15

7.0. Contact Information 16

List of Figures

Figure 1 : ACOS6-SAM Set-up 5
Figure 2 : Card Life Cycle States 7
Figure 3 : File System Hierarchy 9
Figure 4 : File Life Cycle Status 11

List of Tables

Table 1 : Cycle Status Byte 10

1.0. Introduction

This document aims to provide detailed description of the features and functions of the ACS smart card operating system Version 6 – Security Access Module, also known as ACOS6-SAM, developed by Advanced Card Systems Ltd.

1.1. Features

ACOS6-SAM provides the following features:

- Full 32 KB of EEPROM for application data
- Compliance with ISO 7816 Parts 1, 2, 3, 4
  - Supports ISO 7816 Part 4 file structures: Transparent, Linear fixed, Linear Variable, Cyclic
- High-speed transmission rate 9.6 Kbps to 223.2 Kbps
- DES/3DES/3K3DES capability
- AES-128 support
- FIPS 140-2 compliant hardware based random number generator
- Key pair for mutual authentication
- Session key based on random numbers
- Secure Messaging function for confidential and authenticated data transfers
- Secure Access Module pairs with ACOS3, ACOS6, ACOS7, ACOS10 and MIFARE Ultralight® C, MIFARE® DESFire®, MIFARE® DESFire® EV1, MIFARE Plus® cards
- Stores and performs all key operations for mutual authentication, encrypted PIN submission, secure messaging, and e-Purse commands
- Multilevel secured access hierarchy
- Anti-tearing capability

1.2. Technical Specifications

The following are some technical properties of the ACOS6-SAM 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 KB
- EEPROM Endurance: 100,000 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

2.0. ACOS6-SAM

ACOS6-SAM is designed to be used as a general cryptogram computation module or as the security authentication module for ACOS3, ACOS6, ACOS7, ACOS10, MIFARE Ultralight C, MIFARE DESFire, MIFARE DESFire EV1 and MIFARE Plus client cards. The SAM card securely stores the cryptographic keys and use these keys to compute cryptograms for other applications or smart cards. Using this, the keys never leaves the SAM un-encrypted and the system security is greatly enhanced. The SAM can also perform the authentication procedure and purse MAC computation for the ACOS3/6/7/10 cards.

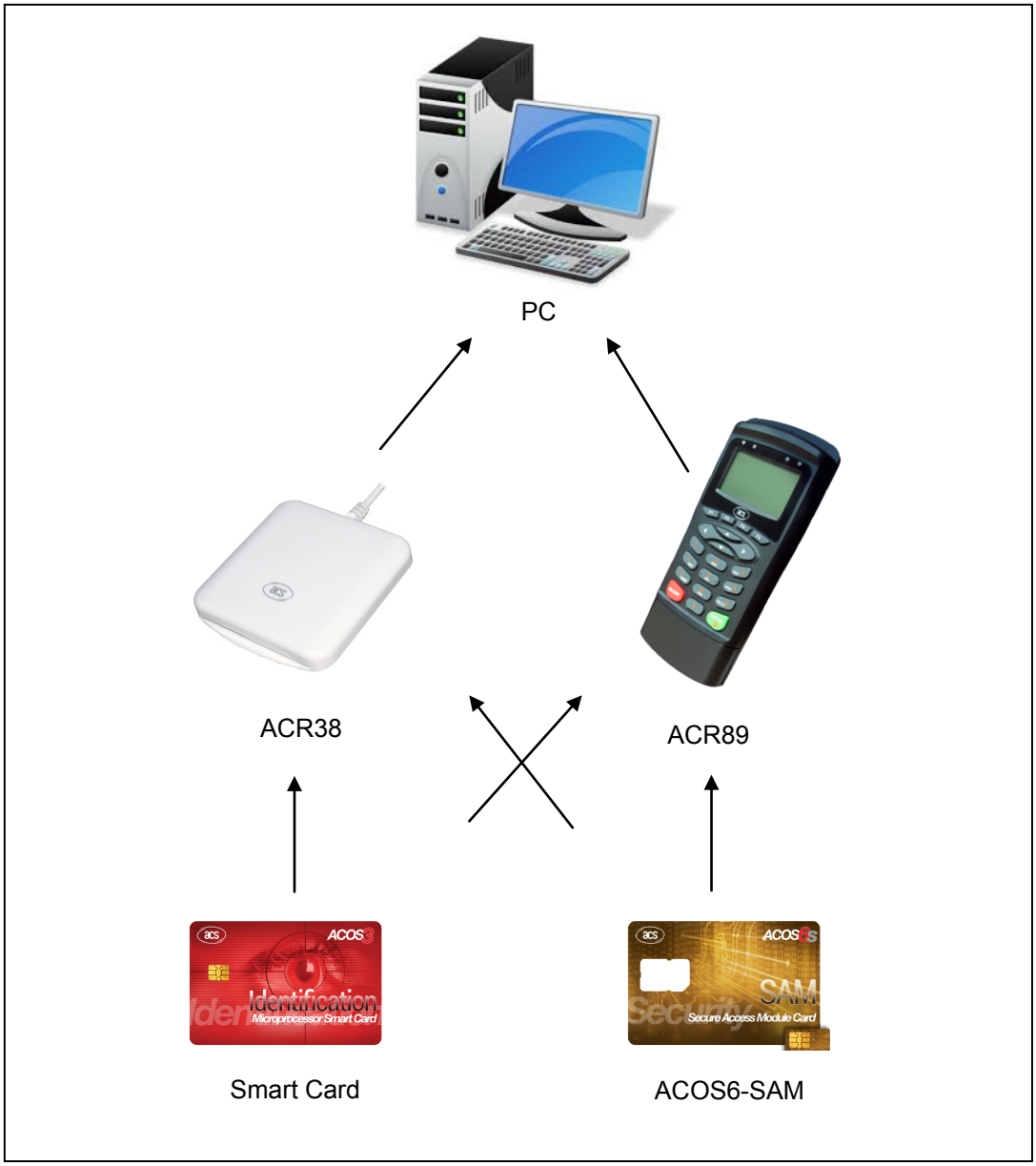

Figure 1: ACOS6-SAM Set-up

The ACOS6-SAM can be deployed in any application for these purposes:

- To store and secure the application's DES/3DES master keys.
- To generate and derive application keys based on a set of master keys.
- To perform cryptographic functions with client smart cards.

- To use as a secured encryption module.

When used with ACOS3 or ACOS6 client smart cards, ACOS6-SAM can perform the following functions:

- Initialize the ACOS3/6 card with diversified keys based on the card's unique serial number.
- Perform mutual authentication process, and generate of session key.
- Perform secure messaging with ACOS3/6.
- Compute secured e-Purse commands.
- Perform secure key injection with ACOS6.

When used with ACOS7, ACOS10, ACOS10-PSAM smart cards, ACOS6-SAM can perform the following functions:

- Initialize the ACOS7/ACOS10/ACOS10-PSAM card with diversified keys based on the card's unique serial number.
- Perform mutual authentication process and generate session key.
- Perform secure key injection with ACOS7/ACOS10.

  Note: *ACOS6-SAM does not perform PBOC/MOC based purse commands. For that, please use ACOS10-PSAM which is a PBOC-compliant payment SAM.*

When used with MIFARE Ultralight C smart cards, ACOS6-SAM can perform the following functions:

- Initialize the UL-C client card with diversified keys based on the card's unique serial number.
- Perform mutual authentication process.

When used with MIFARE Plus, MIFARE DESFire/MIFARE DESFire EV1 smart cards, ACOS6-SAM can perform the following functions:

- Initialize the MIFARE Plus, MIFARE DESFire/MIFARE DESFire EV1 client card with diversified keys based on the card's unique serial number.
- To perform mutual authentication process.
- To perform secure messaging.
- To perform secure key injection.

The programming method of ACOS6-SAM is different from ACOS3 cards. It is designed to conform to ISO 7816 Part 4 file system and command set. To get the application developer up to speed, we have included a quick start guide and sample personalization. The following subsections describe the specific SAM functions.

3.0. Card Management

This section outlines the card level features and management functions.

3.1. Card Life Cycle States

ACOS6-SAM has the following card states:

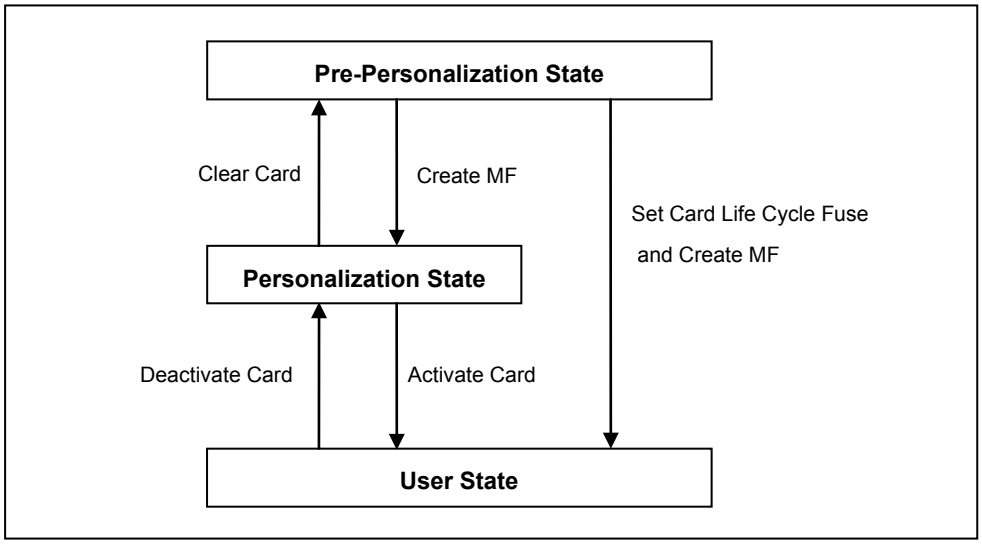

Figure 2: Card Life Cycle States

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.

The 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.

The card cannot revert back to previous states when Card Life Cycle Fuse is set (00h) 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.

3.2. Card Header Block

ACOS6-SAM 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 Development Steps of Card

1. User personalizes the card's header block using UPDATE BINARY.
2. User then creates his card file structure, starting with MF. Dedicate Files (DF) and Elementary Files (EF) 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 (write 00h to address EEC7h).
4. Card goes into Operational Mode, when the MF is created again. User can re-construct the file system under this state. Card can no longer go back to previous states.

User may choose to set the enable DEACTIVATE CARD command in card header block. This allows step 3 and 4 to be replaced by the ACTIVATE CARD command. If the application developer wishes to clear this card, the DEACTIVATE CARD command can be used. To control the access to the DEACTIVATE CARD command, an extended security attribute can be set.

3.4. Answer To Reset (ATR)

After hardware reset (e.g., power up), the card transmits an Answer To Reset (ATR) in compliance with ISO 7816 Part 3. ACOS6-SAM supports the protocol type T=0 in direct convention.

For full descriptions of ATR options see ISO 7816 Part 3.

3.4.1. Customizing the ATR

ACOS6-SAM's ATR can have a customized 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.

4.0. File System

This section explores the file system of the ACOS6-SAM smart card.

4.1. Hierarchical File System

ACOS6-SAM 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)**.

The ACOS6-SAM allows arbitrary depth DF tree structure. That is, the DFs can be nested. Please see the figure below:

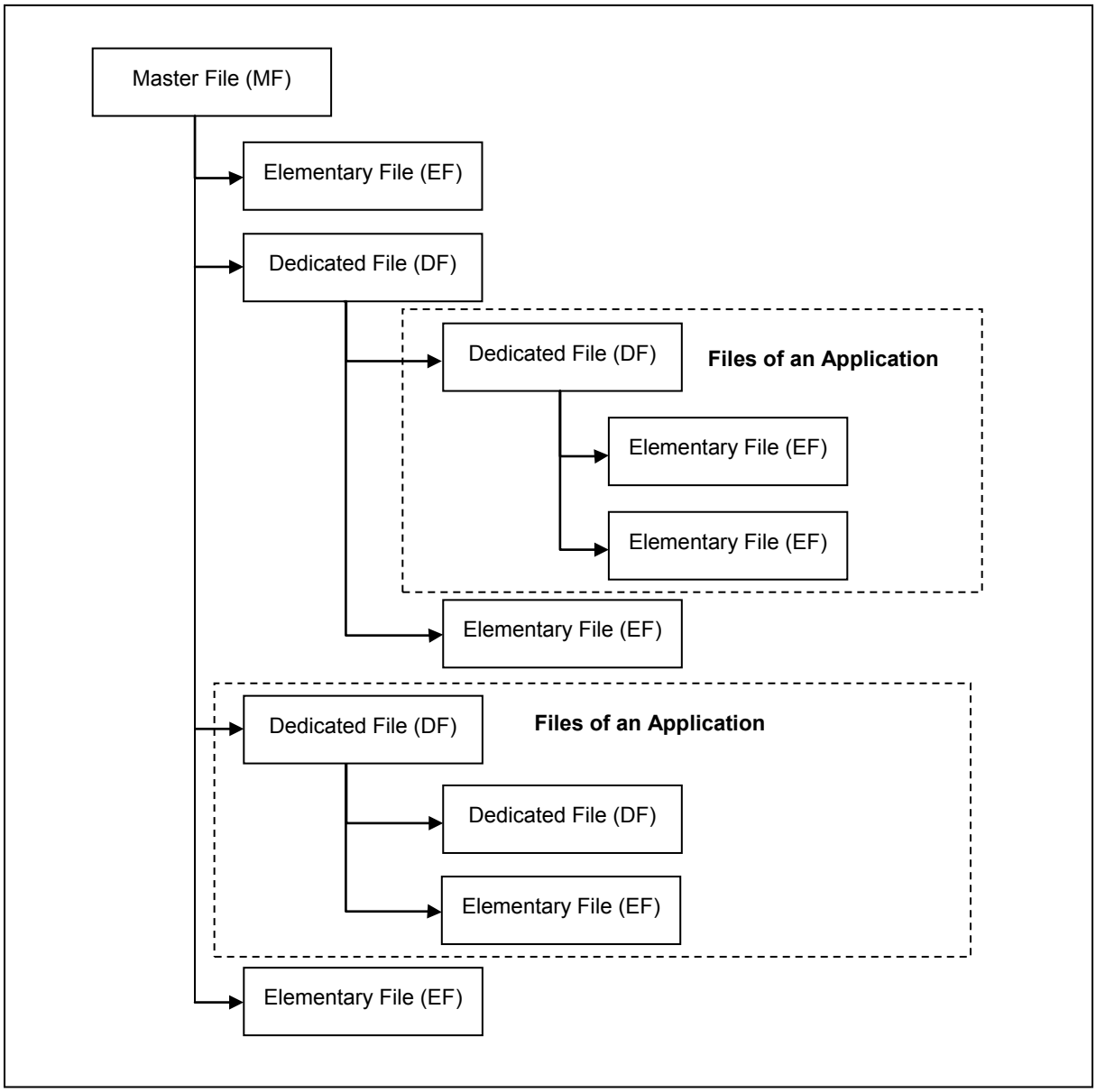

Figure 3: File System Hierarchy

4.2. File Header Data Structure

ACOS6-SAM 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)

The size of the File Header block varies depending on the file type.

4.2.2. Data Coded Byte (DCB)

ACOS6-SAM 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)

Short File Identifier is a five-bit value that represents the file's Short ID. ACOS6-SAM allows file referencing through SFI. The last 5 bits of the File ID does not necessarily have to match this SFI. Two files may have the same SFI under a DF. In such case, ACOS6-SAM 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:

b7	b6	b5	b4	b3	b2	b1	b0	Hex	Meaning
0	0	0	0	0	0	0	1	01	Creation state
0	0	0	0	0	0	1	1	03	Initialization state
0	0	0	0	0	1	-	1	05 or 07	Operational state (activated)
0	0	0	0	0	1	-	0	04 or 06	Operational state (deactivated)
0	0	0	0	1	1	-	-	0C to 0F	Termination state

Table 1: Cycle Status Byte

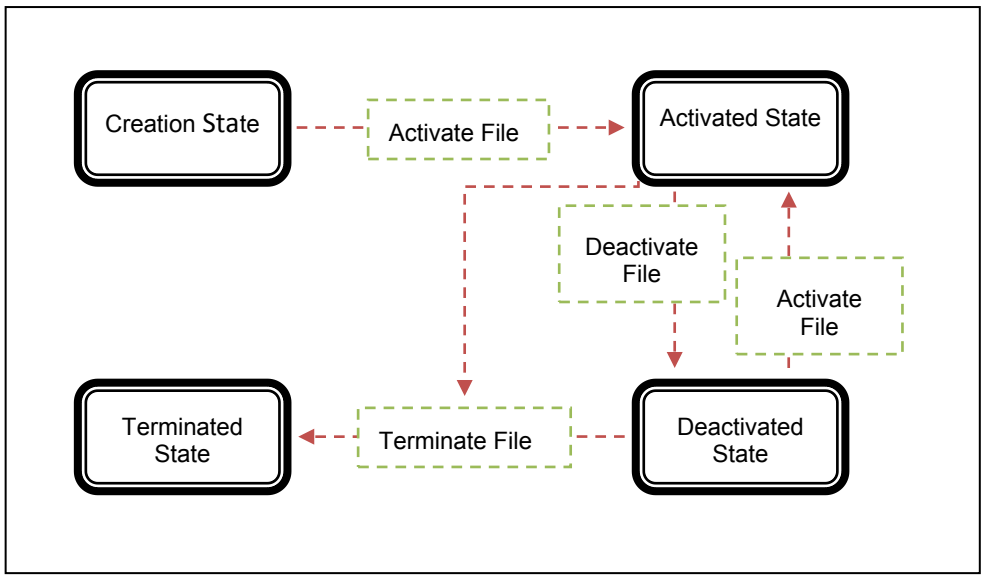

Figure 4: File Life Cycle Status

- 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.

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. See **ACOS6-SAM Reference Manual** for more information.

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. See **ACOS6-SAM Reference Manual** for more information.

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

4.2.14. SE File ID (for DF only)

For a DF, this field consists of two 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 a DF, this field consists of two bytes containing the File ID of one of its children. That file is known as the *File Control Information File*, which is processed internally by the COS.

4.2.16. DF Name (for DF only)

For a 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 internal files are activated, the READ condition must be set to NEVER. Typically, a DF should have: (1) a Key File to hold PIN codes (referred as EF1) for verification, (2) a Key File to hold KEY codes (referred as EF2) for authentication, and (3) an SE file to hold security conditions.

A Key File is an Internal Linear Variable file. It may contain (1) PIN data structure or (2) KEY data structure.

Each unsuccessful attempt will decrement $CNT_{Remaining}$. A successful submission of the PIN number will reset the $CNT_{Remaining}$ to the $CNT_{Allowed}$. If the lower nibble reaches zero, then the PIN is locked and further PIN submission is not possible.

5.0. Security Features

This chapter illustrates the access rights and security capabilities of the ACOS6-SAM card along with its environment and usage. They are:

- File Security Attributes
- Security Environment
- Mutual Authentication
- Short Key External Authentication
- Secure Messaging
- Key Injection
- Anti-tearing Mechanism

Furthermore, 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. Security Attribute 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. 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).

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. The Tag is 80h with the length of 01h.

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. 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.

5.4. Short Key External Authentication

Short Key External Authentication uses a card challenge and terminal response method to gain authorization to the card. This allows for shorter external authentication or one-time-password that is more optimal for human input.

5.5. Secure Messaging

5.5.1. Secure Messaging for Authenticity (SM-MAC)

ACOS6-SAM supports two types of Secure Messaging - *Secure Messaging for Authenticity (SM-MAC)* and *Secure Messaging for Confidentiality (SM-ENC)*. SM for Authentication allows data and command that is transferred into the card and vice versa to be authenticated. This ensured the command is not modified or replayed. Data blocks sent from the sender to the recipient are appended with 4 bytes of MAC. The receiver then verifies the MAC before proceeding with the operation. Before performing SM, both parties must first have a session key by performing mutual authentication. See **ACOS6-SAM Reference Manual** for more information on this.

5.5.2. Secure Messaging for Confidentiality (SM-ENC)

ACOS6 Version 4.02 and above supports ISO secure messaging (SM). Secure messaging ensures data transmitted between the card and terminal/server is secured and not susceptible to eavesdropping, replay attack and unauthorized modifications. Almost all the command can also use secure messaging initiated by the terminal.

5.6. Key Injection

Key Injection can be used to securely load a key or diversified key from an ACOS6-SAM card into a target ACOS6-SAM or client ACOS6 card. For the purpose of key injection, we shall refer to the ACOS6-SAM with the key to inject the "*source SAM*" and the ACOS6/ACOS6-SAM 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 SAM cards uses the Set Key command and the source SAM will use the Get Key command to perform key injection.

Note: *The key injection feature is available for ACOS6-SAM revision 4.02 and ACOS6 revision 3.02 onwards.*

5.7. Anti-tearing Mechanism

ACOS6-SAM uses an anti-tearing mechanism in order to protect 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 card data update. On card reset, ACOS6-SAM 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.

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.