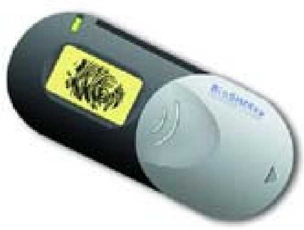
advanced card systems ltd

ADT60

Reference Manual



BioSIMKey

Fingerprint Scanner & Plug-in Smart Card Reader

Advanced Card Systems Ltd.

Room 302, 3/F., Shun Fat Industrial building,

Tel: 852-2796 7873 Fax: 852-2796 1286

Website: www.acs.com.hk E-mail : info@acs.com.hk

TABLE OF CONTENTS

1. INTRODUCTION

The ACS BioSIMKey (ADT60) is a device which combines a fingerprint scanner and a smart card reader/writer. The fingerprint scanner (TouchChip) makes use of the Active Captive-Sensing Technology from STMicroelectronics. The reader/writer part enables the communication between a computer (for example, a PC) and a smart card.

Since fingerprints cannot be lost, duplicated, stolen or forgotten, the TouchChip product is regarded as providing a more reliable and convenient solution than traditional security devices. With BioSIMKey, security is improved further by storing the fingerprint templates inside a SIM card instead of the PC. This not only provides a more secure environment but it also enhances portability and eliminates privacy concerns. It also gives the user the flexibility of being able to carry their fingerprint template with them, assured with the knowledge that no one else can use their smart card should it become lost or stolen.

In the present state of the smart card industry, different types of smart cards use different commands and different communication protocols. In most cases, this situation prevents the direct communication between a smart card and a computer. The BioSIMKey Reader/Writer establishes a uniform interface between the computer and the smart card for a wide variety of cards. By taking care of the card-specific particulars, it releases the computer software programmer from getting involved with the technical details of the smart card operation, which in many cases are not relevant for the implementation of a smart card system.

The BioSIMKey is connected to the computer through a USB interface. The device accepts a command from the computer, carries out the specified function on the TouchChip and the smart card and returns the requested data or status information back to the computer.

2. FEATURES

- Ideal for laptop users – can be carried in your pocket
- Enhanced security by requiring a fingerprint instead of PIN or password
- Active Capacitive-Sensing Technology ensures highest quality images
- ISO7816-1/2/3 compatible smart card interface
- Supports CPU-based cards with T=0 and/or T=1 protocol
- Support PPS (Protocol and Parameters Selection) with 9600 – 115200 bps in reading and writing into smart cards
- Full speed USB interface

3. FINGERPRINT SCANNER

BioSIMKey is built around the TouchChip Silicon fingerprint sensor. It is a fast, reliable and inexpensive fingerprint peripheral, which can be used to authenticate users of computers and all kinds of information technology devices.

The TouchChip device is suitable for applications such as desktop access control, network security, Internet-based applications and commercial verification and identification systems. It contains all the necessary biometric components: fingerprint sensing, image optimization and matching. Application Program Interface (API) is also provided for easy integration into applications and to save a lot of development time. (Please refer to Programmer's Guide for more information on API.)

TouchChip active capacitive sensing provides a much higher immunity to parasitic effects leading to a higher signal-to-noise ratio and the ability to capture a wider range of fingerprints than competing technologies, such as passive capacitive sensing.

Typically there are two processes involved in a biometric application:

Enrollment:
Before the identity of an individual can be verified via his/her fingerprints, it is necessary to capture one or several fingerprint samples. This process is called enrollment. The samples are referred to as fingerprint templates and can be stored on a broad range of media such as computer storage devices or smartcards.

Verification:
The verification process requires a user to verify his identity by placing his finger on the fingerprint scanner sensor. The live fingerprint is compared with a stored template using a matching algorithm in order to determine whether they represent the same set of fingerprint. The matching result is then made available to the computer.

When using the fingerprint device, the security level is mainly governed by two parameters:

False Acceptance Rate (FAR):
FAR is the probability that a false sample matches with the original template previously extracted from the subject's fingerprint images during enrollment.

False Rejection Rate (FRR):
FRR is the rate at which the system incorrectly rejects a legitimate attempt to verify.

4. SMART CARD READER

4.1 Supported Card Types

The BioSIMKey can operate MCU card with T=0 and T=1 protocol. The table presented in Appendix A explains which card type selection value must be specified for the various card types supported by the reader.

4.1.1 Microcontroller-based smart cards (asynchronous interface)

The BioSIMKey supports EEPROM microcontroller-based cards with internal programming voltage (VPP) generation and the following programming parameters transmitted in the ATR:

PI1 = 0 or 5
I = 25 or 50

The BioSIMKey performs the Protocol and Parameters Selection (PPS) procedure as specified *in ISO7816-3:1997*.

When the card ATR indicates the specific operation mode (TA_2 present; bit b5 of TA_2 must be 0) and that particular mode is not supported by the BioSIMKey, the reader will reset the card and set it to negotiable mode. If the card cannot be set to negotiable mode, the reader will reject the card.

When the card ATR indicates the negotiable mode (TA_2 not present) and communication parameters other than the default parameters, the BioSIMKey will execute the PPS and try to use the communication parameters that the card suggested in its ATR. If the card does not accept the PPS, the reader will use the default parameters (F=372, D=1).

For the meaning of the aforementioned parameters, please refer to *ISO7816, part 3*.

4.2 Smart Card Interface

The interface between the BioSIMKey and the inserted smart card follows the specifications of *ISO7816-3* with certain restrictions or enhancements to increase the practical functionality of the BioSIMKey.

4.2.1 Smart Card Power Supply VCC (C1)

The current consumption of the inserted card must not be higher than **50mA**.

4.2.2 Programming Voltage VPP (C6)

According to ISO 7816-3, the smart card contact C6 (VPP) supplies the programming voltage to the smart card. Since all common smart cards in the market are EEPROM based and do not require the provision of an external programming voltage, the contact C6 (VPP) has been implemented as a normal control signal in the BioSIMKey. The electrical specifications of this contact are identical to those of the signal RST (at contact C2).

4.2.3 Card Type Selection

The controlling PC has to always select the card type through the proper command sent to the BioSIMKey prior to activating the inserted card.

For MCU-based cards the reader allows the user to select the preferred protocol, T=0 or T=1. However, this selection is only accepted and carried out by the reader through the PPS when the card inserted in the reader supports both protocol types. Whenever an MCU-based card supports only one protocol type, T=0 or T=1, the reader automatically uses that protocol type, regardless of the protocol type selected by the application.

4.2.4 Interface for Microcontroller-based Cards

For microcontroller-based smart cards only the contacts C1 (VCC), C2 (RST), C3 (CLK), C5 (GND) and C7 (I/O) are used. A frequency of 4 MHz is applied to the CLK signal (C3).

4.2.5 Card Tearing Protection

The BioSIMKey provides a mechanism to protect the inserted card when it is suddenly withdrawn while it is powered up. The power supply to the card and the signal lines between the BioSIMKey and the card are immediately deactivated when the card is being removed. As a general rule, however, to avoid any electrical damage, **a card should only be removed from the reader while it is powered down.**

NOTE - The BioSIMKey never does by itself switch on the power supply to the inserted card. This action must be explicitly done by the controlling computer through the proper command sent to the reader.

5. POWER SUPPLY

The BioSIMKey requires a voltage of 5V DC, 100mA, regulated, power supply. The BioSIMKey gets the power supply from the PC through the cable supplied along with the device.

Status LEDs

Red LED on the front of the reader indicate the activation status of the smart card interface:

Red LED Indicates power supply to the smart card is switched on, i.e., the smart card is activated.

6. USB INTERFACE

The BioSIMKey is connected to a computer through a USB following the USB standard.

i) Communication Parameters

The BioSIMKey is connected to a computer through USB as specified in the USB Specification 1.1. The BioSIMKey is working in full speed mode, i.e. 12 Mbps.

USB Interface Wiring

Pin	Signal	Function
1	**V_{BUS}**	+5V power supply for the reader
2	**D-**	Differential signal transmits data between BioSIMKey and PC.
3	**D+**	Differential signal transmits data between BioSIMKey and PC.
4	**GND**	Reference voltage level for power supply

NOTE - In order for the BioSIMKey to function properly through the USB interface, ACS PC/SC device driver has to be installed. Please refer to the *BioSIMKey Device Driver Installation Guide* for more detail.

7. PC-READER COMMUNICATION PROTOCOL

In normal operation, the BioSIMKey smart card reader acts as a slave with regard to the communication between a computer and the device. The communication is carried out in the form of successive command-response exchanges. The computer transmits a command to the reader and receives a response from the reader after the command has been executed. A new command can be transmitted to the BioSIMKey smartcard reader only after the response to the previous command has been received.

There are only two cases where the reader transmits data without having received a command from the computer, namely, the Reset Message of the reader and the Card Status Message.

7.1 Command

7.1.1 Normal Command (Length < 255 bytes)

A command consists of four protocol bytes and a variable number of data bytes and has the following structure:

Byte	1	2	3	4 ... N+3 (0<N<255)	N+4
	Header	Instruction	Data length = N	Data	Checksum

Header Always 01_H to indicate the start of a command.

Instruction The instruction code of the command to be carried out by the BioSIMKey

Data Length Number of subsequent data bytes.(0 < N < 255)

Data Data contents of the command.
For a READ command, for example, the data bytes would specify the start address and the number of bytes to be read. For a WRITE command, the data bytes would specify the start address and the data to be written to the card.
The data bytes can represent values to be written to a card and/or command parameters such as an address, a counter, etc.

Checksum The checksum is computed by XORing all command bytes including header, instruction, data length and all data bytes.

The following example shows the structure of a command with instruction code = 91_H and three data bytes with the values 11_H, 22_H and 33_H, respectively:

byte	1	2	3	4	5	6	7
	01_H	91_H	03_H	11_H	22_H	33_H	93_H

7.1.2 Extended Command

A command consists of six protocol bytes and a variable number of data bytes and has the following structure:

byte	1	2	3	4 5	6 ... N+5 (N>0)	N+6
	Header	Instruction	Data Length = N		Data	Checksum
			FF_H	Data Length N		

Header Always 01_H to indicate the start of a command.

Instruction The instruction code of the command to be carried out by the BioSIMKey.

Data Length Number of subsequent data bytes, and is encoded in 3 bytes. The first byte is FF_H. The second byte and the third byte represent data length N.

Data Data contents of the command.

For a READ command, for example, the data bytes would specify the start address and the number of bytes to be read. For a WRITE command, the data bytes would specify the start address and the data to be written to the card.

The data bytes can represent values to be written to a card and/or command parameters such as an address, a counter, etc.

Checksum The checksum is computed by XORing all command bytes including header, instruction, data length and all data bytes.

7.2 Response

The response from the BioSIMKey to any command depends on whether the command has been received by the reader without error (e.g., checksum error).

7.2.1 No transmission error with normal response (Length < 255 bytes)

The response by the BioSIMKey to a correctly received command consists of three protocol bytes, two status bytes and a variable number of data bytes and has the following structure:

byte	1	2	3	4	5 ... N+4 (0<N<255)	N+5
	Header	SW1	SW2	Data length = N	Data	Checksum

Header Always 01_H to indicate the start of the response.

SW1 Indicates the command execution status:

90_H = command successfully executed

60_H = error in command data; command cannot be executed

67_H = error detected in command execution

FF_H = status message initiated by the reader

SW2 Further qualification of the command execution status.

A table listing the possible values of the status bytes SW1 and SW2 and the corresponding meaning is given in Appendix B.

Data Length Number of subsequent data bytes (0 < N < 255)

Data Data contents of the command.

For a *READ_DATA* command, for example, the data bytes would contain the contents of the memory addresses read from the card. The data bytes can represent values read from the card and/or status information.

Checksum The checksum is computed by XORing all response bytes including header, status bytes, data length and all data bytes.

The following example shows the structure of the response to a command which has successfully been executed and which returns three data bytes with the values 11_H, 22_H and 33_H, respectively:

byte	1	2	3	4	5	6	7	8
	01_H	90_H	00_H	03_H	11_H	22_H	33_H	92_H

7.2.2 No transmission error with extended response

The response by the BioSIMKey to a correctly received command consists of three protocol bytes, two status bytes and a variable number of data bytes and has the following structure:

byte	1	2	3	4	5	6	7 ... N+6 (N>0)	N+7
	Header	SW1	SW2	Data length = N			Data	Checksum
				FF_H	Data Length N			

Header Always 01_H to indicate the start of the response.

SW1 Indicates the command execution status:

90_H = command successfully executed

60_H = error in command data; command cannot be executed

67_H = error detected in command execution

FF_H = status message initiated by the reader

SW2 Further qualification of the command execution status.

A table listing the possible values of the status bytes SW1 and SW2 and the corresponding meaning is given in Appendix B.

Data Length Number of subsequent data bytes, and is encoded in 3 bytes. The first byte is FF_H. The second byte and the third byte represent data length N.

Data Data contents of the command.

For a *READ_DATA* command, for example, the data bytes would contain the contents of the memory addresses read from the card. The data bytes can represent values read from the card and/or status information.

Checksum The checksum is computed by XORing all response bytes including header, status bytes, data length and all data bytes.

7.2.3 Transmission error

If the receiving party of a command (i.e., the BioSIMKey) or a response (i.e., the computer) detects an error in the data length or the checksum of a command, it disregards the received data and sends a "NOT ACKNOWLEDGE" message to the transmitting party upon completion of the faulty transmission. The "NOT ACKNOWLEDGE" message consists of two bytes:

byte	1	2
	05_H	05_H

If the BioSIMKey responds with a 'NOT ACKNOWLEDGE' message to a command from the computer, the computer would normally transmit the command again.

If the computer detects a transmission error in a response from the BioSIMKey, it can send the 'NOT ACKNOWLEDGE' to the reader upon which the reader will transmit the most recent response again.

7.3 Reset Message

A reset of the reader occurs automatically whenever the reader is being powered up. A reset can also be actuated through the RS-232/USB interface.

In either case the reader transmits **one time** a Reset Message, which has the same structure as the normal response to a command and the following contents:

byte	1	2	3	4	5	6
	Header	**SW1**	**SW2**	**Data length**	**Data**	**Checksum**
	01_H	FF_H	00_H	01_H	BAUD=12_H	

BAUD Indicates the hardware baud rate setting (default baud rate), which is set to 9600 bps (this is only valid in the RS232 reader).

The reader does not expect an acknowledge signal from the computer. After transmitting the Reset Message the reader is waiting for the first command from the computer.

7.4 Card Status Message

When a card is being inserted into the reader or an inserted card is being removed from the reader while the reader is idle, i.e., not executing a command, the reader transmits a Card Status Message to notify the host computer of the change in the card insertion status.

In a system where these unsolicited messages from the reader to the computer are not desired, they can be disabled with the *SET_NOTIFICATION* command. Please note that the setting made with this command is volatile and will be lost with the next reader reset or power up. By default, the Card Status Message will be transmitted by the reader after a reset.

The Card Status Messages have the following structure and contents:

Card Status Message for Card Insertion

byte	1	2	3	4	5
	Header	**SW1**	**SW2**	**Data length**	**Checksum**
	01_H	FF_H	01_H	00_H	FF_H

Card Status Message for Card Removal

byte	1	2	3	4	5
	Header	**SW1**	**SW2**	**Data length**	**Checksum**
	01_H	FF_H	02_H	00_H	FC_H

A card status message is transmitted only **once** for every card insertion or removal event. The reader does not expect an acknowledge signal from the computer. After transmitting a status message, the reader waits for the next command from the computer.

NOTE - If the card is being removed from the reader **while a card command is being executed**, the reader will transmit a normal response to the computer with the response status bytes indicating the card removal during command execution (see *Appendix B: Response Status Codes*).

7.5 Transmission Protocol

The start of a command (to the reader) or a response (from the reader, including the Reset Message and Card Status Messages) is indicated by the respective party through the transmission of the single byte Start-of-Text (STX) character with the value 02_H.

The end of a command or response is indicated through the single byte End-of-Text (ETX) character with the value 03_H.

Within the command and response transmission only ASCII characters representing the hexadecimal (hex) digits 0...F are used. Each byte of a command or response is splitted into its upper and lower halfbyte (nibble). For each halfbyte is transmitted the ASCII character representing the respective hex digit value. For example, to transmit the data byte $3A_H$, two bytes are actually sent on the interface, namely, 33_H (ASCII code for '3') followed by 41_H (ASCII code for 'A'):

Data byte value	$3A_H$	
Transmitted values	33_H = '3'	41_H = 'A'

The following example shows the transmission of a command with instruction code $A2_H$ and one data byte with the value $3D_H$. The command has the following structure:

byte	1	2	3	4	5
	Header	**Instruction**	**Data length**	**Data**	**Checksum**
	01_H	$A2_H$	01_H	$3D_H$	$9F_H$

This command is transmitted on the serial interface in 12 bytes as follows:

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**'0'**	**'1'**	**'A'**	**'2'**	**'0'**	**'1'**	**'3'**	**'D'**	**'9'**	**'F'**	**ETX**
	02_H	30_H	31_H	41_H	32_H	30_H	31_H	33_H	44_H	39_H	46_H	03_H

For the representation of the hex halfbyte values as the corresponding ASCII characters in commands, the BioSIMKey accepts both upper case characters 'A' ... 'F' (41_H ... 46_H) and lower case characters 'a' ... 'f' (61_H ... 66_H):

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**'0'**	**'1'**	**'A'**	**'2'**	**'0'**	**'1'**	**'3'**	**'D'**	**'9'**	**'F'**	**ETX**
	02_H	30_H	31_H	41_H	32_H	30_H	31_H	33_H	44_H	39_H	46_H	03_H

... is equivalent to:

byte	1	2	3	4	5	6	7	8	9	10	11	12
	STX	**'0'**	**'1'**	**'a'**	**'2'**	**'0'**	**'1'**	**'3'**	**'d'**	**'9'**	**'f'**	**ETX**
	02_H	30_H	31_H	61_H	32_H	30_H	31_H	33_H	64_H	39_H	66_H	03_H

In its response messages, the BioSIMKey uses upper case characters 'A' ... 'F'.

8. SMARTCARD COMMANDS

The commands executed by the BioSIMKey smartcard reader can generally be divided into two categories, namely, Control Commands and Card Commands.

Control Commands control the internal operation of the BioSIMKey. They do not directly affect the card inserted in the reader and are therefore independent of the selected card type.

Card Commands are directed toward the card inserted in the BioSIMKey. The structure of these commands and the data transmitted in the commands and responses depend on the selected card type.

8.1 Control Commands

8.1.1 GET_ACR_STAT

This command returns relevant information about the particular BioSIMKey model and the current operating status, such as, the firmware revision number, the maximum data length of a command and response, the supported card types, and whether a card is inserted and powered up.

Command format

Instruction Code	Data length
01_H	00_H

Response data format

<table>
<tr><th colspan="10">INTERNAL</th><th>MAX_C</th><th>MAX_R</th><th colspan="2">C_TYPE</th><th>C_SEL</th><th>C_STAT</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

INTERNAL 10 bytes data for internal use only

MAX_C The maximum number of command data bytes.

MAX_R The maximum number of data bytes that can be requested to be transmitted in a response.

C_TYPE The card types supported by the BioSIMKey. This data field is a bitmap with each bit representing a particular card type. A bit set to '1' means the corresponding card type is supported by the reader and can be selected with the *SELECT_CARD_TYPE* command. The bit assignment is as follows:

byte	1								2							
card type	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See Appendix A for the correspondence between these bits and the respective card types.

C_SEL The currently selected card type as specified in a previous *SELECT_CARD_TYPE* command. A value of 00_H means that no card type has been selected.

C_STAT Indicates whether a card is physically inserted in the reader and whether the card is powered up:

00_H : no card inserted

01_H : card inserted, not powered up

03_H : card powered up

8.1.2 SET_PROTOCOL

This command is used to control the line speed of the communication channel between BioSIMKey reader and host device. The line speed of the communication is controlled by two factors, namely, the Delay Factor and the Baud Rate.

Command format

Instruction Code	Data length	Data
		DELAY N
03_H	01_H	

to change only the Delay Factor (for RS232 reader only), or

Instruction Code	Data length	Data	
		DELAY N	BAUD RATE
03_H	02_H		

to change the Delay Factor and the Baud Rate (for RS232 reader only).

DELAY Determines the time delay inserted by the BioSIMKey between two consecutive bytes sent in order to adapt to slower host system speeds. The time delay is given by N * 0.1msec, with N ranging from 0 ... 255 (00 - FF_H). The default value is N = 0 (delay changes only valid on RS232 reader).

BAUD RATE Selects the baud rate (bps) of the serial interface between reader and host system. The default hardware baud rate is 9600 bps. (baud rate changes only valid on RS232 reader).

BAUD RATE	Serial baud rate (bps)
12_H	9600
11_H	19200
10_H	38400
03_H	14400
02_H	28800
01_H	57600
00_H	115200

Response data format

No response data

The new protocol becomes effective by the completion of the *SET_PROTOCOL* command, immediately **after the BioSIMKey has sent out the response string to the** *SET_PROTOCOL* **command**.

8.1.3 SELECT_CARD_TYPE

This command sets the required card type. The firmware in the BioSIMKey adjusts the communication protocol between reader and the inserted card according to the selected card type.

Command format

Instruction Code	Data length	Data
		TYPE
02_H	01_H	

TYPE See Appendix A for the value to be specified in this command for a particular card to be used.

Response data format

No response data

8.1.4 RESET

This section describes the *RESET* command only for the case when no card type is selected or when the card type 00_H is selected. For all other cases, please refer to the specific section described for each individual card type.

Command format

Instruction Code	Data length
80_H	00_H

Response data format

ATR			

ATR The answer-to-reset string returned by the card.

The return status code for this command is 90 00_H when the inserted card is a T=0 card and 90 01_H when the inserted card is a T=1 card, otherwise the status code is 60 20_H.

8.1.5 SET_NOTIFICATION

This command disables / enables the Card Status Messages transmitted by the reader to notify the host computer of the insertion or removal of a card.

Command format

Instruction Code	Data length	Data
		NOTIFY
06_H	01_H	

NOTIFY Specifies whether the Card Status Message shall be transmitted to notify the host computer of card insertion / removal

01_H : transmit Card Status Message

02_H : do not transmit Card Status Message

Response data format

No response data

8.1.6 SET_PPS_MODE

This command selects the PPS mode to be used

Command format

Instruction Code	Data length	Data
		PPS_Mode
07_H	01_H	

NOTIFY Specifies whether the Card Status Message shall be transmitted to notify the host computer of card insertion / removal

00_H : baud rate to/from the card is from 9600 bps to 115200 bps (default)

01_H : baud rate to/from the card is at 9600 bps only

Response data format

No response data

8.2 Card Commands (MCU-based Card)

The available commands and the parameters specified in the card commands as well as the data transmitted in the response from the BioSIMKey depend on the selected card type.

8.2.1 RESET

This command powers up the card inserted in the card reader and performs a card reset. If the card is powered up when the command is being issued, only a reset of the card is carried out. The power supply to the card is not switched off.

Command format

Instruction Code	Data length
80_H	00_H

Response data format

ATR					

ATR Answer-To-Reset as transmitted by the card according to ISO7816-3.

NOTE - The ATR is only returned in the BioSIMKey response if the communication protocol of the card is compatible with the reader, i.e., if the card can be processed by the BioSIMKey. Otherwise, the BioSIMKey returns an error status and deactivates the smart card interface.

8.2.2 POWER_OFF

This command powers off the card inserted in the card reader.

Command format

Instruction Code	Data length
81_H	00_H

Response data format

No response data

8.2.3 EXCHANGE_APDU

To exchange an APDU (Application Protocol Data Unit) command/response pair between the MCU card inserted in the BioSIMKey and the host computer.

Command format

Instruction Code	Data length	Data									
	LEN	CLA	INS	P1	P2	Lc	BYTE 1	...2	...	BYTE N	Le
$A0_H$											

LEN Length of APDU command data, N, + 6 ($0 < N \leq$ MAX_R)

CLA APDU instruction class byte

INS APDU instruction

P1 APDU parameter byte 1

P2 APDU parameter byte 2

Lc APDU command data length

BYTE x APDU command data

Le Expected APDU response data length (Le = 0 menas no data is expected from the card)

NOTE - With the T=0 communication protocol it is not possible to transmit data to the card and from the card in a single command-response pair. Hence, only either Lc or Le can be greater than 0 in an *EXCHANGE_APDU* command when a T=0 card is in the reader. If both parameters have a value greater than 0, the BioSIMKey does not execute the command and returns an error status.

Response data format

BYTE 1	...	...	BYTE N	SW1	SW2

BYTE x Response data from card (if any)

SW1, SW2 Status code returned by the card.

8.2.4 EXCHANGE_T1_FRAME

To exchange an APDU (Application Protocol Data Unit) command/response pair between the MCU card inserted in the BioSIMKey and the host computer using T1 protocol.

Command format

Instruction Code	Data length	Data
	LEN	T1 BLOCK FRAME
$A1_H$		

LEN Length of APDU command data, N

DATA T1 Block frame to be sent to the card

Response data format

BYTE 1	...	...	BYTE N

BYTE x Response T1 Block from card (if any)

APPENDIX A: SUPPORTED CARD TYPES

The following table summarizes which values must be specified in the *SET_CARD_TYPE* command for a particular card type to be used, and how the bits in the response to the *GET_ACR_STAT* command correspond with the respective card types.

Cyber-mouse card type code	Card Type
00_H	Auto-select T=0 or T=1 communication protocol
$0C_H$	MCU-based cards with T=0 communication protocol
$0D_H$	MCU-based cards with T=1 communication protocol

APPENDIX B: RESPONSE STATUS CODES

The following table summarizes the possible status code bytes SW1, SW2 returned by the BioSIMKey:

SW1	SW2	Status
90	00	OK – command successfully executed
90	01	OK – using T=1 protocol (only in response to the RESET command)
90	10	OK – synchronous protocol is used (only in response to the RESET command). The exact card type should be selected by using the SELECT_CARD_TYPE command.
60	01	No card type selected
60	02	No card in reader
60	03	Wrong card type specified
60	04	Card not powered up; This status code is also returned in a response if the card was temporarily removed during a card access.
60	05	Invalid Instruction Code
60	20	Card failure
60	22	Short circuit at card connector
62	01	Secret code verify failed
67	01	Command incompatible with card type
67	02	Card address error
67	03	Data length error
67	04	Invalid length of response (with READ command)
67	05	Secret code locked
67	12	APDU command aborted (only MCU-based card using T=1 protocol); the command abortion may be caused by a card internal failure.

APPENDIX C: TECHNICAL SPECIFICATIONS

Device

BioSIMKey Fingerprint Scanner & Smart Card Reader/Writer

Power supply

Supply voltage .... Regulated 5V DC
Supply current.. < 100mA (without smart card)

Universal Serial Bus Interface

Type .... USB, four lines: +5V, GND, D+ and D-
Connector .... supplied together with the reader
Speed.. 1.5 Mbps

Fingerprint Scanner Interface

Power supply .... USB powered
Power consumption .... Max 100 mA @ 5.5V
Active sensor size.................................. 12.8 x 18 mm
Array size .... 256 x 360 pixels
Array pitch .... 50 microns
Image resolution .... 508 DPI
ESD tolerant .... +/- 15kV

Smart Card Interface

Standard .... ISO 7816 1/2/3, T=0 and T=1
Supply current.. max. 50mA
smart card read / write speed .... 9600 – 115200 bps
Short circuit protection .... +5V / GND on all pins

The presence of the smart card power supply voltage is indicated through a red LED on the reader

CLK frequency .... 4 MHz
Card connector .... sliding contacts (8 contacts)
Card insertion cycles.............................. min. 100,000

Case

Dimensions .... 66.5mm (L) x 30mm (B) x 16mm (H)
Color .... Black

Operating Conditions

Temperature .... 0 - 40° C
Operation humidity................................. 5% - 95%

APPENDIX D: RECOMMENDDED DEVICE CLEANING PROCEDURES

D.1 Introduction

Key elements of image quality are the consistency within the actual image and the background of the image. Software algorithms are more accurate and generally faster when the image quality is consistent and the background has not changed dramatically. Dirty residue, oils, or other material on the surface of the TouchChip may obscure the image, leaving parts of the image unrecognizable, or creating false features within the image. Regular use of the TouchChip may leave residue or other foreign materials on the surface. Performance degradation in terms of False Match and False Non-Match are indicative of such problems.

It is recommended that the sensor be visually inspected and periodically cleaned as described in section D.2. It is also recommended that before each touch, the sensor be cleaned as described in section D.3.

D.2 Periodic Cleaning

Dampen a lint-free cloth or cotton swab with alcohol or acetone. Gently rub the cloth across the sensor surface in a left and right direction. Move slowly down the sensor to cover the entire surface area. Repeat this process 4 times. Visually observe that no residual solution remains on the sensor. After performing the periodic cleaning operation, a surface conditioning is suggested to obtain the maximum performance from the TouchChip sensor. Dampen a lint-free cloth with fragrance-free moisturizing lotion, and gently rub the cloth across the sensor. Make sure that all the lotion will be removed as completion of the cleaning process.

Acid-based fluids, and abrasive materials are not recommended for cleaning the TouchChip.

D.3 User Cleaning

Before each authentication, it is recommended that the user simply wipe the sensor with her/his finger, and then position the finger for the authentication. With this action we assure that residue from previous usage will be removed hence giving the best surface conditioning.