Interface Control Document

for the

FORTEZZA
CRYPTO CARD

(PRODUCTION VERSION)

Revision P1.5

December 22, 1994

Prepared By:

NSA X21

Table of Contents

List of Figures

List of Tables

1.0. Introduction

1.1. Purpose

The purpose of this document is to provide the technical level of detail needed for third party vendors to understand the application of the FORTEZZA Crypto Card to their requirements and subsequently design operating system drivers and/or hardware adapters to use the resources provided by the FORTEZZA Crypto Card.

1.2. Card Overview

The FORTEZZA Crypto Card, hereafter called “The Card”, is a cryptographic module which implements the Digital Signature Algorithm (DSA), the NIST Secure Hash Algorithm (SHA-1), the Key Exchange Algorithm (KEA), and SKIPJACK. The Card complies with the PCMCIA specification Standard Release 2.1. The Card supports 41 individual commands which can be used to support cryptographic based authentication and encryption applications.

1.3. Related Documents

PCMCIA, PC Card Standard, Release 2.1, July 1993, Personal Computer Memory Card International Association.

Intel Corporation, Exchangeable Card Architecture (ExCA) Specification, Release 1.10, June 1992.

PCMCIA, Socket Services Standard, Release 2.1, July 1993

PCMCIA, Card Services Specification, Release 2.1, July 1993

PCMCIA, Recommended Extensions, Release 1.00, November 1992

FORTEZZA Cryptographic Interface Programmers Guide, Revision P1.4, 18 August 1994

MOSAIC Certificate Labeling Specification, Version P1.0

MOSAIC Key Management Concept, V332, Revision 2.5, February 1994

1.4. Terms and Abbreviations

API -	Application Programmers Interface.
Card, the -	The FORTEZZA Crypto Card.
Certificate -	A 2048 byte packet of information containing KEA and/or DSA information about a user or a generic data field.
Certificate Index -	The ordinal value used to access certificates on the card. The Certificate Index is used to bind a Certificate Label, Certificate

	and a set of Private Components together. The Certificate Index zero (0) is reserved for use by the SSO.
Certificate Label -	The ASCII/ANSI string that is a human readable alias for a certificate. The Certificate Label is always 32 bytes long for FORTEZZA. The Certificate Label must be formatted in accordance with the "FORTEZZA Certificate Labeling Format Specification".
CIS -	Card Information Structure.
DSA -	Digital Signature Algorithm.
DSA-Yb -	Digital Signature Algorithm Public Component of Recipient (128 bytes).
E-Mail -	Electronic Mail.
Even Word -	A value or address where the 2 LSB are 00. Examples of 32-bit even word boundary addresses are:0000 0000h, 0000 0004h, 0000 0008h, etc. All pointers on the card require the use of 32-bit even word boundaries addressing.
g parameter -	A DSA parameter (between 64-128 bytes) defined by the SSO. For FORTEZZA, g is a 128 byte parameter.
g size -	Size of the g parameter.
Hash -	An algorithm to digest any amount of data to a fixed size.
IV-	Initialization Vector used in the encryption/decryption process.
K value -	A 160-bit value used in the digital signature generation (same size as q).
K_s -	The User's Storage Key Variable (80 bits). Stored in Register 0 after a successful user check pin phrase.
KEA -	Key Exchange Algorithm for Electronic Public/Private Key Exchange.
KEA-Yb -	Recipient's Public Component used in key exchanges (128 bytes).
Key Register Index -	Index parameter to specify use of a temporary key storage register valid values are 0 - 9.
Key Registers -	A set of temporary storage registers for storage of encryption keys.
LAW -	Local Authority Workstation. Responsible for generating and signing user certificates.
Long -	32-bit big endian value.
Manufacturer Default PIN -	The SSO PIN phrase that must be entered to logon to the Card when it is first received from the manufacturer.
MEK -	Message Encryption Key generated by the Card's random number generator.
p parameter-	The prime modulus (64 - 128 bytes) used in the key exchange and DSA. For FORTEZZA, the prime modulus, p, is always 128 bytes.
PCMCIA -	Personal Computer Memory Card International Association.

Personality -	Certificates assigned to an individual (e.g. SSO, Self and Department).
PIN Phrase -	Personal Identification Phrase used to log onto the card.
PMSP -	Preliminary Message Security Protocol.
psize parameter -	The size of the p, the prime modulus (64-128 bytes).
q parameter -	The prime divisor. For FORTEZZA, q is always set to a 20 byte value.
qsize parameter -	Denotes the size of the prime divisor in bits.
r -	One of the two parameters defining the digital signature (s is the other). For FORTEZZA, r is always 160 bits.
Ra -	A 1024-bit random number generated for the public/private key exchange.
Rb -	A 1024-bit random number received from the other party involved in the key exchange.
Root -	The Root generates and signs all LAW certificates.
Root Certificate -	The certificate used to validate certificates from the LAW and other users.
s -	One of the two parameters defining the digital signature (r is the other). For FORTEZZA, s is always 160 bits.
Signature -	A value used to authenticate that the data came from a specific author and has not been modified. The signature is composed of two parts: r and s. The r & s are always 160 bits each, making the Signature 320 bits, for FORTEZZA.
SSO -	Site Security Officer.
TEK -	Token Encryption Key used with by the KEA for the public/private key exchange.
Tuple -	An information format defined by the PCMCIA specification.
User Personality -	Same as Personality.
Wrap -	Encrypt one key with a different key.
X -	User's secret component in the DSA or KEA exchange. For FORTEZZA, X is always 20 bytes.
Y -	User's public component in the DSA or KEA exchange.
Yb -	Recipient's public component in the DSA or KEA exchange. For FORTEZZA, Y is always 128 bytes.
Zeroize -	A process that clears the User and SSO PIN phrases, and other memory on the Card, as required.
Zeroize Default Pin -	The SSO PIN phrase that must be entered to logon to the Card once it has been zeroized.

2.0. FORTEZZA Crypto Card PCMCIA System Architecture

This specification describes the physical and logical interface to the Card. This includes the command set used by the Card. This document is intended for developers that are integrating directly with the Card.

The Card is a cryptographic module implemented on a Type I or Type II PCMCIA-compliant card. The Card interfaces via the standard 68-pin PCMCIA connector using a shared-memory interface to pass commands and data between the Card and the host adapter/driver. This generic interface facilitates the integration of the Card into different platforms and operating systems. However, since the Card is PCMCIA compliant it will operate with off-the-shelf socket services and card services defined under the PCMCIA specification.

Figure 1.0 illustrates the application of the Card in a system configuration. The following defines the functions of each identified layer:

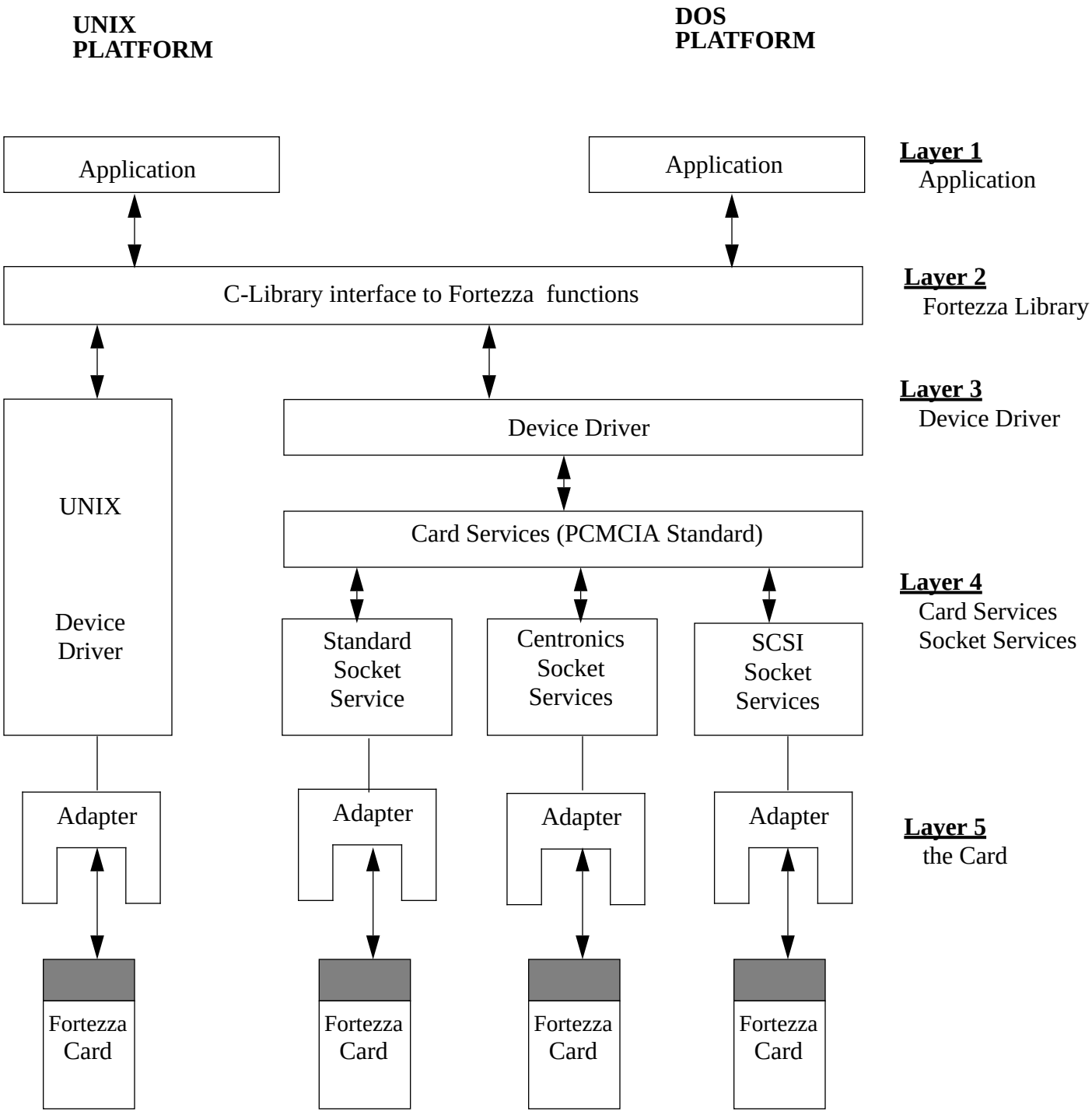
Layer 1 - Application -
This is an application that uses the services of the Card. For example, an E-Mail application could secure messages with the signature and encryption services of the Card.

Layer 2 - FORTEZZA Crypto Card C Library -
The FORTEZZA Crypto Card Library provides the application with a platform-independent Application Programmers Interface (API) to the Card. This allows the application developer to use a set of ANSI C-compliant function calls to access the Card. It also insulates the application developer from the hardware specifics of the card and the data structures it uses. All function calls are made in the host system's data formats. The library handles all translations from the host to card data types. A separate document, the FORTEZZA Cryptologic Interface Programmers Guide, referenced in Chapter 1, provides information about the FORTEZZA API to application developers.

Layer 3 - Device Driver -
The device driver provides the library with a platform-independent interface to the Card. The library communicates with the device driver through the host's file system. Using ANSI standard file streams, the library performs open, close, read, write and seek commands to send and receive data from the card through the device driver. This is a character oriented device driver. The device driver may optionally separate out the Card and Socket Services layers.

Layer 4 - PCMCIA Card and Socket Services -
Card Services provide a single-component, hardware-independent interface for the device driver to the socket services on the host system. This removes the burden from the device driver of locating and managing the socket services and cards when multiple socket services, sockets and cards are present on the host system.

The Socket Services provides a hardware-independent interface for the device driver for a particular socket implementation. This allows the socket to reside on the motherboard, on an ISA/EISA card, S-Bus card or over serial, parallel or SCSI interfaces. It is up to the device driver



Fortezza PCMCIA ICD

- Physical Interface
- Logical Interface
- Command Set

FIGURE 1.0 Hardware/Software Relationships

to locate the Card from the one or more socket services that may reside on the host system.

Layer 5 - PCMCIA Card Implementation -
This is the physical PCMCIA card that contains the cryptographic functions and executes the commands requested by the application.

This specification will support users developing the hardware and software layers above the Card. This includes adapters for different busses, as well as device drivers for different operating systems. The remainder of this document will describe the technical operating details of the Card. This includes an overview of the Card's operating states, its hardware and software design, the shared memory interface and finally the specific command set.

2.1. Operating States

Before the Card can be accessed by the User, a Site Security Officer (SSO) must initialize the card with operating parameters, the User ID and User certificates. This must be done prior to any other operations by the User. All Cards are initialized at the factory with a default SSO PIN Phrase. After the first logon to the Card by the SSO, the SSO must change the SSO PIN Phrase from the default value. The Card is now ready to be initialized. During initialization, the Card will limit execution of commands to those required to complete the initialization process by the SSO. After initialization is complete, the SSO provides the initialized card to the User. The SSO can logon to the Card at any time after initialization to change any parameters. The Card allows the SSO ten consecutive failed logon attempts. The Card stores the logon attempt count internally. After ten consecutive failed SSO logon attempts, the Card will zeroize all key material and initialization values and return to the zeroized state. At this time, the SSO must use the Zeroize PIN Phrase to log back onto the Card. All Card parameters will need to be reinitialized.

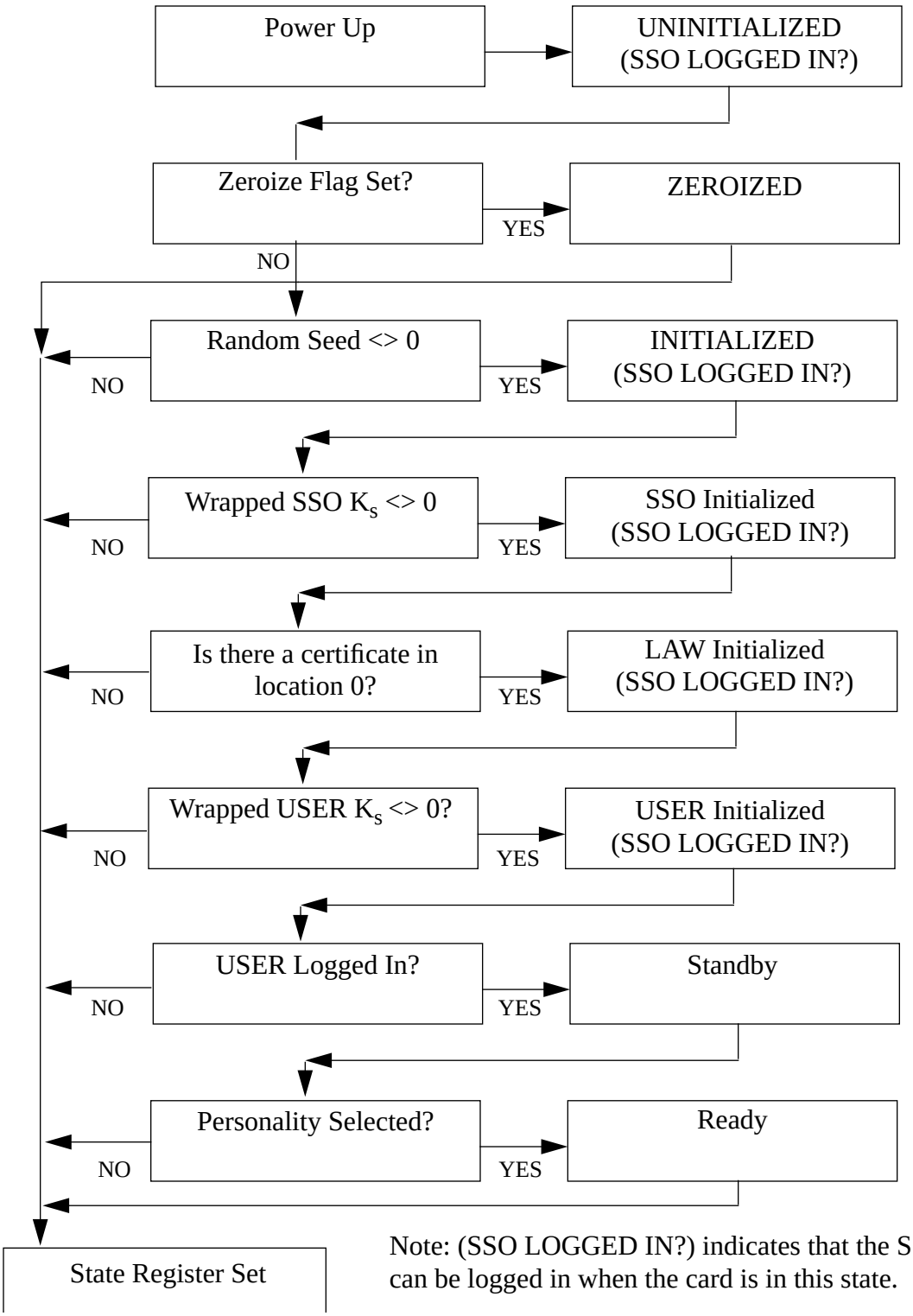
The Card requires a User system logon prior to access of any onboard cryptographic functions. To logon the user must provide the correct User PIN Phrase and select a “User Personality.” The system supports 10 consecutive User logon attempts, storing the logon attempt count in non-volatile memory. After 10 consecutive failed User logon attempts, the Card will delete the stored User PIN Phrase. Certificates and other information stored on the Card in non-volatile memory will remain resident. After every successful User PIN Phrase, the failed User logon count is reset. After a correct PIN Phrase is entered by the user, they can then select one of the stored User “Personality” certificates for operations requiring DSA or KEA exchanges.

After successful user logon, the cryptographic resources of the Card are available. The Card has an on-board public key/private key exchange protocols, user digital signature verification and generation capabilities, and encryption/decryption capability. These resources can be used to facilitate secured E-Mail applications.

Since the Card is targeted for use in cryptographic applications, it has an on board self test. A failure of any of the self tests will cause the Card to go into a fail state. To exit the fail state, a hardware reset must be performed.

Each of these states are defined below:

Card State Determination Flow Chart



Note: (SSO LOGGED IN?) indicates that the SSO can be logged in when the card is in this state.

Power Up to Uninitialized: Power Up state is entered any time power is initially applied to the Card. After the Card runs its power-on self tests, the Card declares itself Uninitialized and proceeds down the state determination flow chart. The resultant state of the card, after power-up self tests, can be read from the State Information Register prior to logging onto the Card.

Uninitialized to Initialized: After the SSO has logged on with the default SSO PIN Phrase, the SSO will define the two initialization values.

Initialized to SSO Initialized: In this state, the SSO should change the SSO default PIN Phrase.

SSO Initialized to LAW Initialized: The Card supports internal storage of User certificates/ personalities and an SSO certificate in non-volatile memory. To cause this state transition, the SSO loads a certificate into location 0.

LAW Initialized to User Initialized: In this state, the SSO should set the User's Pin Phrase. This will transition the card to the User Initialized state. The Card may now be given to the User.

User Initialized to Standby: The User Pin Phrase is validated against the SSO initialized User value. If the User fails to provide the correct Pin Phrase after 10 attempts, the Card will transition to the LAW Initialized state.

Standby to Ready: Since a single User may have multiple personalities, the User selects the personality for the cryptographic functions it will be performing during a session. The User may select a different personality at any time during this session. Prior to selecting a personality, any of the Card resources, e.g. encrypt and decrypt with storage key, that don't require a user's private key to be selected (any KEA or DSA operations) may be performed.

Ready State: In this state, the complete Card resources are available to the application.

Zeroized State: Any state may transition to the Zeroized state by issuing the Zeroize command or after 10 consecutive invalid SSO logon attempts.

Fail State: The card will transition to this state if a card failure is detected during card self test or command execution.

Reserved State: This state code is reserved for future use.

State information can be read out from the Card's Configuration registers discussed further in Chapter 2. The Card assigns the following 4 bit field for state:

Power Up	= X0h	Uninitialized	= X1h	Initialized	= X2h
SSO Initialized	= X3h	LAW Initialized	= X4h	User Initialized	= X5h
Standby	= X6h	Ready	= X7h	Zeroized	= X8h
Reserved	= XEh	Fail	= XFh		

Note: State register values X9h - XDh are undefined.

2.2. Prototype FORTEZZA Crypto Card Hardware Overview

Figure 3.0 is a block diagram of a typical Card. The Card's cryptographic and processor functions are supported by the CAPSTONE chip. Since CAPSTONE has a 32-bit bus, memory is configured 32-bits wide. The interface logic performs the bus arbitration and 32 to 16/8-bit bus access configuration functions required by the PCMCIA 2.1 specification. This logic may also implement the configuration registers that allow the host to access control information while the Card has access to the shared-memory interface. The Ready/Busy line is used to indicate to the adapter/driver that the Card is accessing the shared-memory area and therefore the bus appears tri-stated to the host.

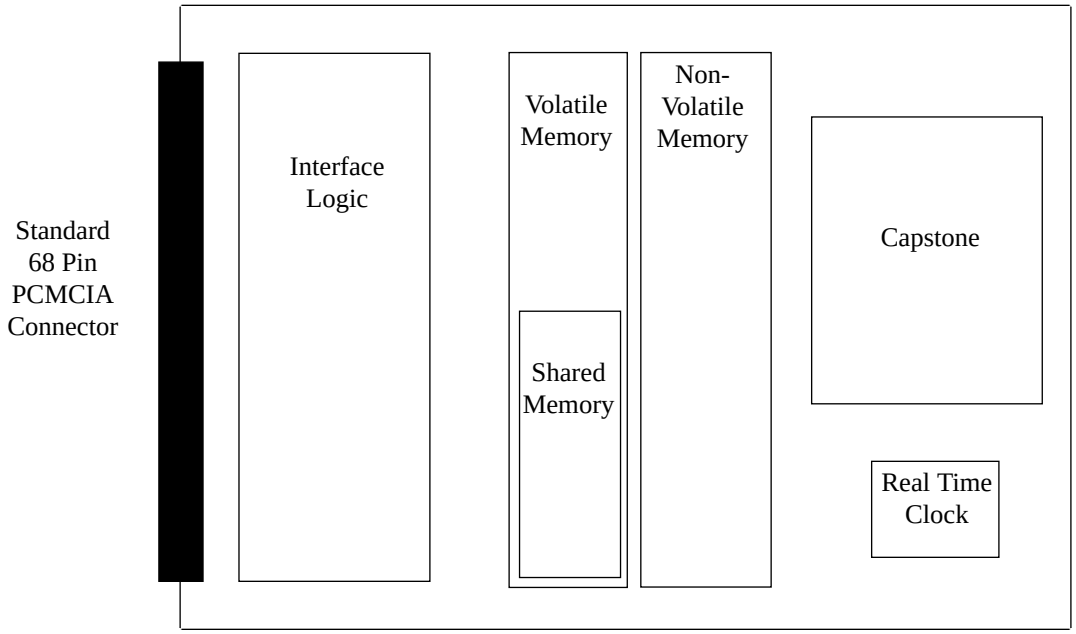


FIGURE 3.0 Hardware Block Diagram

2.2.1. Pin Definitions

All pins are defined to be consistent with the PCMCIA 2.1 specification. Please reference this specification for the pin/polarity configuration.

2.2.2. Power Management

When the Card is not executing commands, most of the card's circuitry is idle. Hence, while the card is waiting for commands, power consumption is minimal.

2.3. Memory Configuration

The Card contains three memory areas. Each of these three areas are shown in Figure 4.0. This figure also shows the type of information stored in each of the defined areas. Through the use of these memory areas, the host has access to all the Card cryptographic functions. This section will briefly describe their function.

The lowest-level area is defined as the Attribute memory area. Part of this memory is read only and is used by the host to obtain information about the Card. The Attribute Memory contains the Card Information Structure (CIS). The CIS provides information about the card manufacturer, user RAM available, ROM, Card Revision, Number of Certificates that can be stored, and the start of the mailbox communications area between the Host and the Card. Within Attribute memory space, a read/write memory area is also defined. This memory area contains the configuration registers. The host uses these registers to obtain state information or request the Card to begin executing a command in the shared memory area. The gate arrays provide the configuration register information to the host.

The second area is defined as the Common memory area. This consists of the RAM space on the Card. The RAM provides the shared memory interface to the host which is used for communications between the host and the processor. This read/write memory is called the Common Shared Memory Area. All communications between the processor and the host use this area, the "Mailbox Area", to communicate through. The CIS, in Attribute memory tells the host the location of this Mailbox Area. Chapter 3 will address the host communications through the mailbox area in shared memory. A portion of the on-board RAM is available for the shared memory interface. The other portion is used for local scratch memory by the Card. Since this is a shared memory interface, the RAM is not accessible to the Host during the time the Card is accessing data from the RAM. The Ready/Busy line is used by the FORTEZZA Crypto Card to arbitrate shared memory access by the Host.

The third and final memory area on the FORTEZZA Crypto card is the non-volatile memory storage area. This area is used to store information for the user and SSO. The SSO initializes this area with the Users Pin phrase and parameters for digital signatures or key exchanges. This area also contains the certificates that have been loaded in. Since this is non-volatile memory, information written to this area will be maintained on board until they are overwritten with new information. This area is not directly accessible by the host system. All communications with the non-volatile storage area must go through the shared memory RAM area. Data is written into

non-volatile storage through the use of the Load Certificate or other appropriate commands. A portion of the non-volatile memory is available for this use. The other portion of non-volatile memory is used as the program storage for the FORTEZZA Crypto Card processor. The CIS in Attribute memory provides the host information about the amount of certificate storage available.

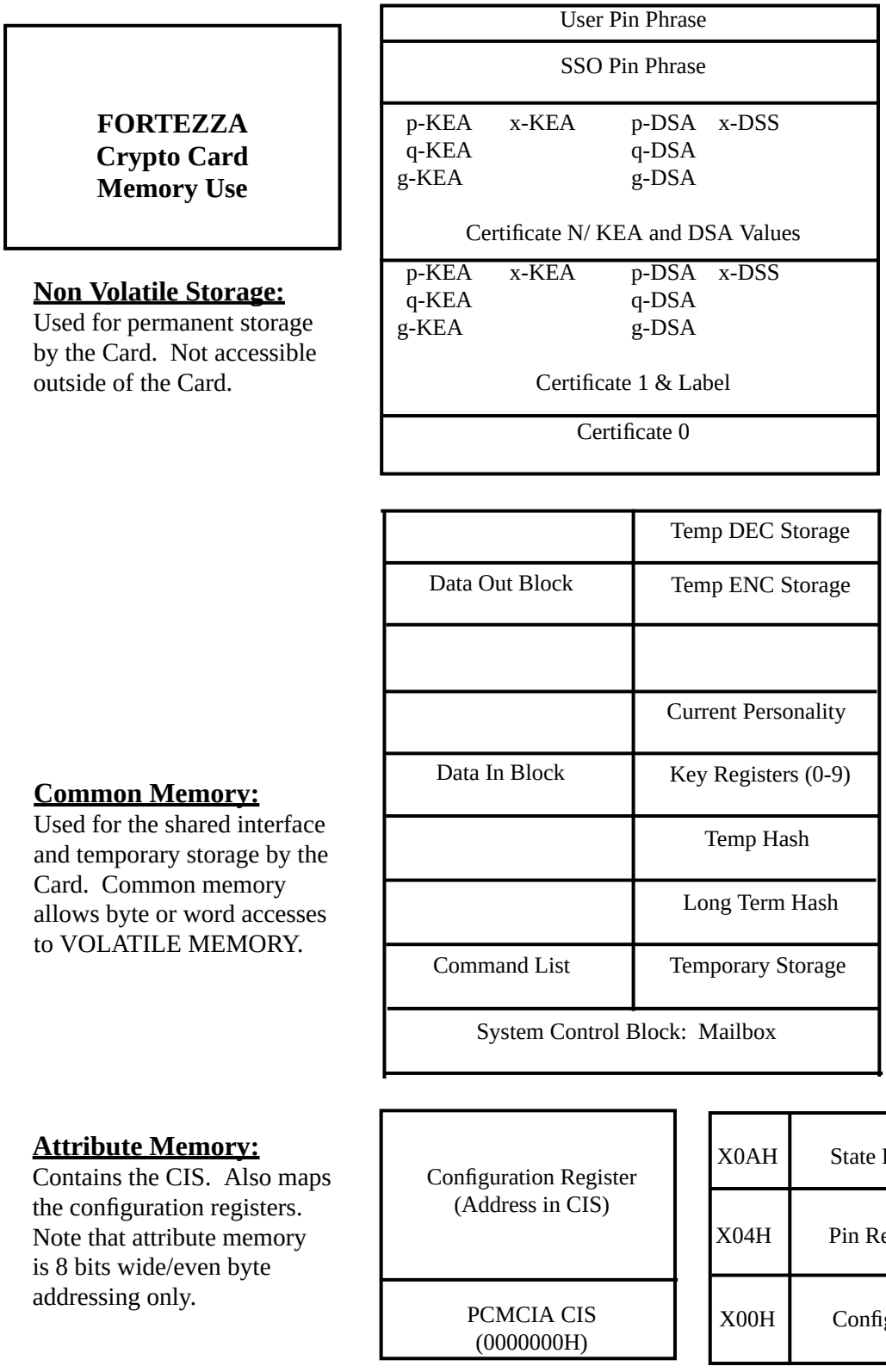


FIGURE 4.0 Memory Use

2.3.1. Attribute Memory Area

The host must access Attribute memory before initiating any communications with the FORTEZZA Crypto Card. By reading the tuples in the CIS, the host can determine the start of the mailbox area used for command execution by the Card's processor. The use and function of Attribute memory is defined in the Card Metaformat section of the PCMCIA 2.1 specification. The FORTEZZA Crypto Card is compliant with this specification and uses Attribute memory to store the hardwired basic compatibility structure. For further information, please reference Section 5 of the PCMCIA 2.1 Specification. The Attribute memory contains the Card Information Structure (CIS) which is read by the host. This memory is at the beginning of the card address space and describes the low-level organization of the data on the card. All information in Attribute memory is written in the tuple format, per the PCMCIA standard.

The Attribute memory structure resides in the even-byte locations starting at 0000h and going to the CIS ending tuple (FNULL). The data included in the CIS consists of tuples. Tuples are a variable-length list of data blocks describing details such as manufacturer's name, the size of shared memory area and the 32-bit absolute address that the Card maps shared memory into. The following section describes the tuple format in the CIS.

2.3.1.1. PCMCIA Card Information Structure

The Card Information Structure (CIS) starts at address zero of the card's Attribute memory. All Attribute memory is accessed on even-byte boundaries. It contains a variable length chain of data blocks (tuples) that conform to a basic format as shown in Table 1. The tuple configuration for the FORTEZZA Crypto Card is PCMCIA 2.1 compliant. For a further definition of the use, reading and configuration of tuples in the CIS, please reference the Card Metaformat section of the PCMCIA 2.1 specification. The following section describes the tuples contained within the FORTEZZA Crypto Card Attribute memory which the host can read to obtain the card's configuration information.

Table 1: Tuple Format

Byte	Data
0	Tuple Code: CISPTL_xx. The tuple code FFH indicates no more tuples on the list.
1	Tuple Link: TPL_LINK. Link to the next tuple in the list. This can be viewed as the number of additional bytes in tuple, excluding this byte. If the link is zero, the tuple body is empty. If the link is FFH, this is the last tuple in the list.
2-n	Bytes specific to this tuple.

2.3.1.1.1. The Device Information Tuple - First Tuple at Location 0000

The first tuple, identified by CISTPL_DEVICE = 01h, contains information about the card's RAM speed and memory size. This is the first information to the host that allows the host to determine how much RAM memory is available and its access speed. For the Card, this information pertains to the Common Memory area (RAM) access time only. This tuple follows the Metaformat definition in the PCMCIA specification. The Device Type field within this tuple

should indicate FUNCTION_SPECIFIC as defined in the PCMCIA specification.

2.3.1.1.2. Level 1 Version/Product Information Tuple - Second Tuple

The next tuple, identified by CISTPL_VER = 15h, contains Level 1 version compliance and card manufacturer information. The Level 1 Version/Production information tuples provide information to the host about the cards production and revision information. This tuple will be used to indicate the base address of the mailbox communications area. This allows hosts to adapt to different manufacturers or revision releases of the card.

Structure defined for the Level 1 Version/Production information are as follows:

TPLLV_MAJOR - Major version number = 04h.

TPLLV_MINOR - Minor version number = 01h (for release 2.1)

TPLLV1_INFO - Each item is an ASCII string terminated with a zero. The values are in 0000 for decimal and 0x0000 for hexadecimal.

Name of Manufacturer	=	“Manufacturer Name”
Name of Product	=	“FORTEZZA Crypto Card” *See Note1 Below
Card Type	=	“Processor = _____”
Number of Certificates	=	“Certificates = _____” *See Note 2 Below
Number of Key Registers	=	“Key Registers = _____”
Number of Concurrent Mailboxes	=	“Multiple Mailboxes = No” *See Note 3 Below
Mailbox Start Address	=	“Mailbox Start Address = ____” *See Note 4 below
Timeout in Milliseconds	=	“Timeout = ____”
ICD Compliance Level	=	“ICD = P1.5”

Note 1: This field MUST contain the “Crypto Card” string to be FORTEZZA Crypto Card compliant.
Note 2: This field contains the number of available User Certificate locations. Certificate location 0, is not a User available location and not displayed as a User available certificate location in this tuple.
Note 3: Cards that support multiple channels will have Yes in this field.
Note 4: For single mailbox cards this address points to the start of the mailbox area. For multiple mailbox cards this address points to a TBD structure map of the mailbox area.

2.3.1.1.3. The Configuration Tuple

This tuple, identified by CISTPL_CONFIG = 1Ah, describes the interface support by the card and locations of the Card configuration registers defined in the next section. This tuple is used to tell the host where the start of configuration registers reside. Please reference the PCMCIA specification for further detail.

TPCC_SZ - Indicates the number of bytes in the configuration registers base address and the configuration register presence mask field.

TPCC_LAST - One byte field containing the configuration Index number of the last entry in the Card Configuration Table. This byte is set to 0 (FORTEZZA is not configurable).

TPCC_RADR - Configuration Registers Base Address in Reg Space.

TPCC_RMSK - Configuration Register Presence Mask Field.

CISTPL_CFTABLE_ENTRY - Card Configuration Table Entry Tuple. (This tuple is defined as 1BH 02H 40H 00H.)

2.3.1.1.3.1. Configuration Registers

The Configuration register's base address is mapped into the Attribute memory area of the Card at the address defined in the TPCC RADR and TPCC_RMSK fields of the configuration tuple. The host reads these locations to determine the location of the first register. The Card Configuration Registers provide information to the host while the Card has control of the shared memory. Control logic in the Gate Array will prohibit host access to the shared memory during command execution but still provide access to the Configuration registers.

The configuration registers are:

Register 0: Configuration Option Register:

REGISTER BIT	NAME
7	SOFT RESET
6	UNUSED
5	UNUSED
4	UNUSED
3	UNUSED
2	UNUSED
1	UNUSED
0	UNUSED

(D7) Software Reset or "Soft Reset".

The Host may command a soft reset by writing to Register 0, bit position 7 (D7) = 1. The Software Reset bit is write only and any reads of this register will return a zero.

(D6:D0) Unused. For any reads of this register all unused bits will return as zeros.

Register 1: Card Configuration and Status Register:

This register is unused. Reads and writes by the host to this register will have no effect. A Host read of this register will return zeros in bit positions (D7:D0).

Register 2: Pin Replacement Register:

REGISTER BIT	NAME
7	UNUSED
6	UNUSED
5	READY/BUSY CHANGE
4	UNUSED
3	UNUSED
2	UNUSED
1	READY/BUSY*
0	UNUSED

(D5) Ready/Busy* Transition Flag

Whenever the Ready/Busy* Flag changes state, this bit is set. The Ready/Busy* Transition Flag can be cleared by writing a zero (0) in position D5 to this register. If this bit is to be used to alert the host that the card has transitioned from Busy to Ready, it must be cleared after the Busy mode has been set.

(D1) Ready/Busy* Flag

--When high (1) the Ready/Busy* flag indicates that the card is in the “ready” mode of operation, allowing host access to the dual-access SRAM memory, attribute memory and attribute registers.

--When low (0) the Ready/Busy* flag indicates that the card is in the “busy” mode of operation, with host access allowed only to the attribute registers.

To execute a command loaded into the mailbox, the host must set the Ready/Busy* Flag to low (Busy). The Ready/Busy* Flag will remain in the Busy mode until the card has finished implementing the commands it was assigned. Upon completion of the assigned task the card will return the Ready/Busy* Flag to the Ready (High) state.

Any reads of this register will return zeros in the positions of all unused bits.

Default States:

(D1) Ready/Busy* will default to Busy after power-up or reset until the card’s self-test is completed. After self-test is completed this bit will change to Ready.

(D5) Ready to Busy Change will default to inactive (0) after power-up or reset.

Register 3: Socket and Copy Register:

This register is unused. Reads and writes by the host to this register will have no effect. A Host read of this register will return zeros in bit positions (D7:D0).

Register 4: Reserved

This register is unused, but this address is reserved for future use. Reads and writes by the host to this register will have no effect. A Host read of this register will return zeros in bit positions (D7:D0).

Register 5: State Register:

REGISTER BIT	NAME
7	RESET DETECT
6	WORST CASE/TYPICAL*
5	RESERVED
4	FAST/SLOW*
3	STATE3
2	STATE2
1	STATE1
0	STATE0

(D3:D0) Represent the Card's State field

The defined states for bits (D3:D0) are:

X0h = Power Up
X1h = Uninitialized
X2h = Initialized
X3h = SSO Initialized
X4h = LAW Initialized
X5h = User Initialized
X6h = Standby
X7h = Ready
X8h = Zeroized
X9h - XDh = Undefined
XEh = Reserved
XFh = Fail

(D4) Fast/Slow* Flag

--When High (1) the processor runs at full clock speed
--When Low (0) the processor runs at a reduced clock speed

(D5) Unused

(D6) Worst-Case/Typical* Flag

--When High (1) the card operates with worst-case memory access timing
--When Low (0) the card operates with typical memory access timing

(D7) Reset Detect

This bit is set when the card has been reset. This flag can be cleared by writing a zero (0) in the position of D7 to this register

Default States:

(D0-D3) The status bits default to zero
(D4) The Fast/Slow* Flag defaults to (1)=Fast
(D5) Unused; defaults to zero
(D6) Worst-Case/Typical* Flag defaults to (1)=Worst-Case
(D7) Reset Detect defaults to zero

Worst-Case/Typical* selection and operation:

The Worst-Case mode of operation is intended to allow the user to continue to use the card even if the timing of its processor are not within the typical timing parameters due to environmental conditions such as low voltage or extreme temperature.

The Typical mode of operation can be used when the processor is running within the typical timing characteristics.

Fast/Slow* selection and operation:

The Fast mode of operation allows the card to drive the processor at full clock speed for normal operation. The Slow mode of operation drives the processor with a reduced clock speed. The slow mode requires less power to run and can be utilized when power consumption is a concern, such as when battery charge is low in the host.

Register 6: Reserved:

This register is unused, but this address is reserved for future use. Reads and writes by the host to this register will have no effect. A Host read of this register will return zeros in bit positions (D7:D0).

Register 7: Reserved:

This register is unused, but this address is reserved for future use. Reads and writes by the host to this register will have no effect. A Host read of this register will return zeros in bit positions (D7:D0).

Register 8 and higher: Unused:

These registers are unused. Some cards may not fully decode attribute memory addresses. Therefore, registers 8 and higher may contain images of registers 0-7.

2.3.1.1.4. Card Information Tuples

This tuple (CISTPL_VERS_2 = 40h) serves to introduce information pertaining to the logical organization of the card's memory and data areas.

TPLLV2_VERS - Structure Version = 00h

TPLLV2_COMPLY - Level of compliance= 00h

TPLLV2_DINDEX - A 16 bit address which specifies the card's first data byte. LSB first.

TPLLV2_RSV6 - Reserved = 00h

TPLLV2_RSV7 - Reserved = 00h

TPLLV2_VSPEC8 - Firmware Version.

TPLLV2_VSPEC9 - Hardware Version.

TPLLV2_NHDR - Number of Copies of CIS on Card.

TPLLV2_OEM - Manufacturer's Name

TPLLV2_INFO - "FORTEZZA Crypto Card".

CISTPL_NO_LINK - This is the last tuple in the CIS. This tuple = (14h).

2.3.2. Common Shared Memory Area - Mailbox Area

Within the RAM, the Card supports both the Card working area RAM as well as the shared memory area for host communications. The Host can obtain the amount of RAM available to the Host by reading the CISTPL_DEVICE, Device Information Tuple.

The common shared memory area of RAM contains the shared memory "mailbox". This area is used by the host to set up commands and data by which the Card executes. Writing a command into this mailbox area, and subsequently writing bit 1=0 in register 2, indicates to the Card that data is available in the shared memory area for command execution. The Card will read the data-in block starting at the location of the mailbox memory area. The base address of the Mailbox Location in the Common Shared Memory Area (as seen by the Card's Processor) is defined within Attribute memory in the TPLLV1_INFO tuple defined above. This tuple tells the Host the location in RAM into which the Card is expecting the Host to write FORTEZZA Crypto Card commands. This tuple contains a 32-bit pointer containing the start address of the mailbox area.

While the Card is accessing this shared memory, the Ready/Busy- line is held Busy. The Host can monitor activity of this RAM by reading bit 1 of register 2. After successful execution of the Host command, the Ready/Busy- line becomes Ready and the Host can read the command response in shared memory. Chapter 3 of this document defines this command interface in more detail.

RAM area dedicated to the Card processing contains temporary storage for Key Registers, save and restore functions and other temporary session-oriented parameters. Keys are stored in Key Registers which the host accesses based on their Key Register Index identifier. The Card contains storage for 10 keys in RAM identified by Key Register Index 0 through 9. Register 0 is used by the FORTEZZA Crypto Card to store the Users K_s value after successful completion of the Check PIN Phrase command.

Non-Volatile memory is not user accessible. This memory contains the User's certificates, PIN Phrase information and other parameters required for operation. The Card contains storage for certificates including one SSO certificate and multiple User certificates. These are stored according to a certificate index. Certificate Index 0 is defined for the SSO certificate. Certificate Indexes 1 through X contain the User Personality certificates.

3.0. Shared Memory Command Interface

An Application and the Card communicate via the shared memory block. The application, via the FORTEZZA Cryptologic Interface Library, will place a Command Block at the mailbox start address of the Card's shared memory. The mailbox start address is defined in the card information structure. Once the Command and Data-In block, if any, is transferred to the Card, the FORTEZZA Crypto Card Library sets bit 1 of register 2 (the Ready/Busy- bit) to 0 (Busy) to begin execution. After this register is written to, the Card disables access to shared memory, sets the Ready/Busy- line of the card to busy and begins to execute the command. Note that the Ready/Busy- Register is also set, to allow the Ready/Busy- line to be polled by host. Cards must support the Ready/Busy- line as defined in the PCMCIA spec., and hosts should respect the status of the line and not try to access the shared memory when the card is busy. When the Card has finished executing the command, it enables access to shared memory and sets the Ready/Busy- line and Ready/Busy- bit to Ready. When the application notices that the Ready/Busy-line is Ready, it then retrieves any status and data from the Card that were a result of executing the command. Figure 5.0 is a high level block diagram that illustrates the interface.

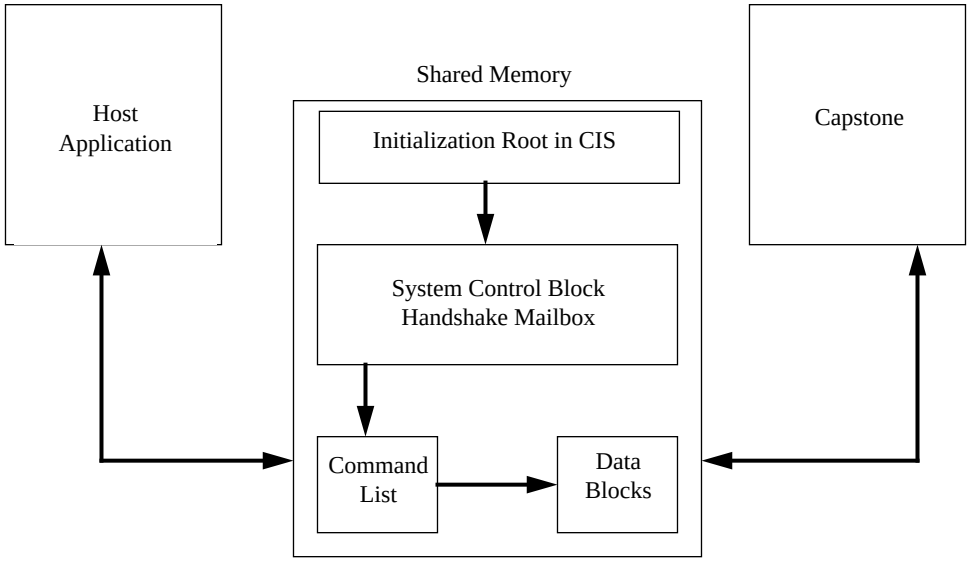


FIGURE 5.0 Shared Memory Interface

3.1. Command Block Structure

The Command Block is the data structure that the host system uses to instruct the Card to perform a command. The Command block is made up of six fields: Command, Pointer to Next Command Block, Pointer to Data-In, Pointer to Data-Out, Response, and Channel Specifier. All fields are 32-bits. The three pointers are 32-bit addresses in the Card's memory space. Note that these addresses are not offsets into the shared memory mailbox area. To calculate the Card memory address from a shared memory mailbox offset, add the mailbox start address from the CIS tuple TPLLV1_INFO to the shared memory offset. The mailbox start address is the address into which the Card has mapped the shared memory mailbox. **Please Note: All Pointers used in the Command Block must be on 32 bit even word boundaries.**

The following will outline the use of each field of a Command Block:

Command Block

byte offset	
00	Command
04	Pointer Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier
32	0

FIGURE 6.0 the Command Block

Offset 0 - Command: The command field contains five separate fields for manipulating the mailbox and also passing the command ID and opcode. This field is defined as follows:

Bits 31-28	27-24	23-20	19-12	11-0
Ownership bits	Reserved	Reserved	Command Set ID	Op Code for Function

Bits 0-11: Function
This is the 12 bit hex opcode for the command to be executed by the card.

Bits 12-19: Command Set ID
This allocates the ID for the FORTEZZA Crypto Card command set. For the card, the Command SET ID is set to all zeros.

Bits 20-23: Reserved
These bits are reserved for future use. For the FORTEZZA Crypto Card, they should be set to all zeros.

Bits 24- 27: Reserved
These bits are reserved for future use. For the FORTEZZA Crypto Card, they should

be set to all zeros.

Bits 28-31: Ownership Bits

There will be four (4) bits allocated to control of the mailbox and its associated data areas. The Ownership bit use is defined as follows:

Bit	Set	Owner	Reason
C- Mailbox Control Bit 31:	0	Set to 0 by the Host	Command has been loaded
	1	Set to 1 by the Card	Card has finished processing command
I- Command/Data-In Mail Bit 30:	0	Set to 0 by the Card or Host	Command and Data-In area available
	1	Set to 1 by the Card or Host	Command and Data-In area in use
O- Data-Out Memory Use Bit 29:	0	Set to 0 by the Card or Host	Command Data-Out area is available
	1	Set to 1 by the Card or Host	Command Data-Out area is in use
E- Command Execution Bit 28:	0	Set to 0 by the Host	Command loaded and ready
	1	Set to 1 by the Card	Command Execution complete

Single mailbox Fortezza cards should check bits 28 and 31 on commands to the card to verify that they are 0, and return an INVALID COMMAND response if they are not. These bits should be returned to the host set to 1. Bits 29 and 30 should not be checked and should be returned to the host as they were received.

Offset 4 - Pointer to Next Command Block: This field contains a pointer to the next command block in a list of commands. The last command will have this field set to NULL (0000 0000h). Note that this is a 32-bit address.

Offset 8 -Pointer to Data-In Block: This field contains a pointer to the Data-In Block. If the command does not require a Data-In Block, then this field must be set to NULL (0000 0000h). Note that this is a 32-bit address.

Offset 12 - Pointer to Data-Out Block: This field contains a pointer to the Data-Out Block. If the command does not require a Data-Out Block then this field must be set to NULL (0000 0000h). Note that this is a 32-bit address.

Offset 16 - Response: This field is used by the card to return a response code to the host at the completion of command execution.

Offset 20 - Channel Specifier: Although not used by the basic single channel card, this field is included to support interoperability with future cards that support multiple channels. For single mailbox cards, this field is set to all zeros (00H 00H 00H 00H).

3.2. The Data-In Block Structure

The Data-In Block is used to provide input data to commands executed on the Card. It is possible to use the same Data-In Block for multiple commands which have identical Data-In Block formats. Therefore, only one copy of the Data-In Block need be on the Card at a time. Each command's Pointer to Data-In Block would point to the same Data-In Block when commands are chained. An example of this is shown in Figure 7.0 where both command blocks point to the same Data-In Block.

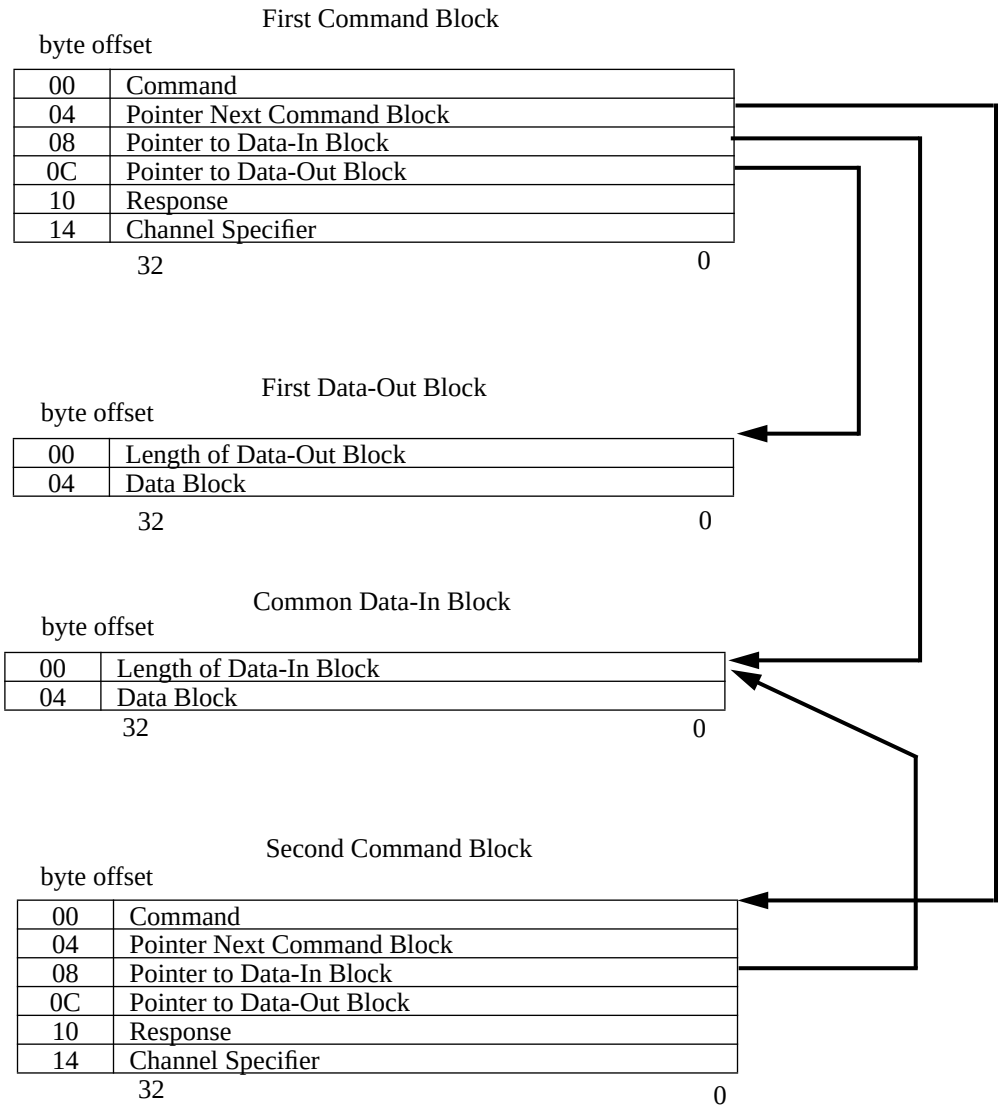


FIGURE 7.0 Chained Commands

In addition to the method described above for using a common Data-In Block, the Encrypt, Decrypt, Get Hash and Hash commands allow the use of a common piece of data, not an entire

Data-In Block. The Data-In Blocks of these commands contain a *pointer* to their primary input data, not the actual data. This allows a common piece of data to be pointed to by a Hash command, and an Encrypt command, for example. An example is shown in Figure 7.1

3.3. The Data-Out Block Structure

See Figure 7.0. The Data-Out block, when necessary, consists of a Length indicator followed by the appropriate data block. When commands are chained together, each command should have its own Data-Out Block. It is up to the host system to ensure that memory is reserved for the entire Data-Out Block.

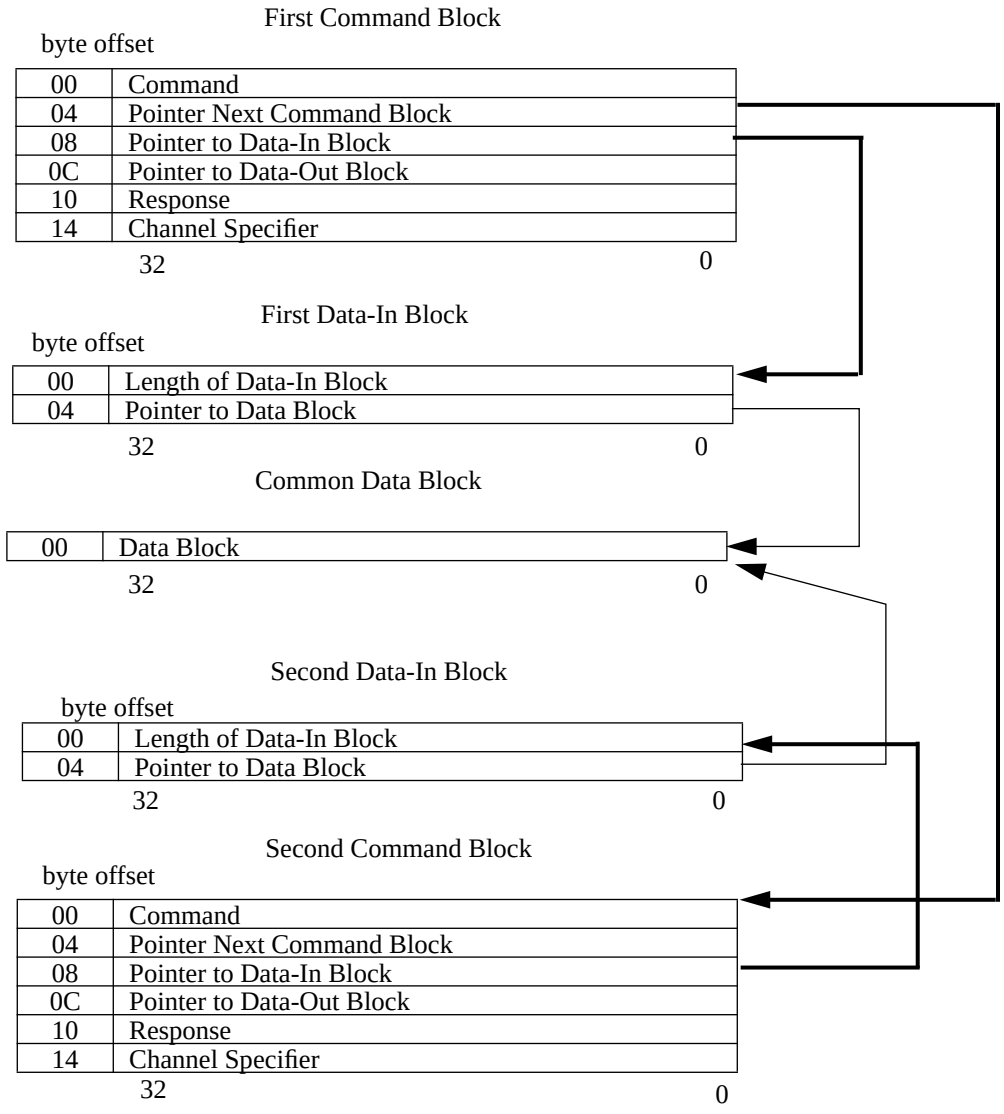


FIGURE 7.1 Common Data Block

4.0. PCMCIA Command Set

The commands for the card are defined in this section. Table 2 is a summary of all of the commands. Figure 8.0 groups commands by their function.

4.1. Commands

The following subsections describe the commands and protocols of the Card Command Interface.

Table 2 provides a list of all commands used by the Card. All values are given in hex. While Table 3 identifies the response codes, section 4.6 provides a list of possible response codes with each command description.

4.2. Security

In order to protect the key material, and thus the user's credentials, stored on the card, a number of security features and rules have been incorporated. These features and rules affect the use of the command set outlined in this document. These include:

(a) After 10 unsuccessful User logon attempts, the User's PIN value is zeroized, requiring the user to return to the SSO.

(b) After 10 unsuccessful SSO logon attempts, the SSO's PIN and all key material is zeroized. After zeroization, the PIN is set to a known ZEROIZED_PIN value.

(c) There is only one SSO and one User per card.

(d) Only an SSO may update card firmware, load the initialization parameters, set the date/time of the real-time clock, set the SSO and User PINs, and Extract an X-value.

(e) The only valid commands that may be performed on a card prior to SSO or User logon are Get_Status, Get_Time, Get_Random_Number, Check_Pin, and Zeroize. Note: no safeguards were placed on the use of the Zeroize command as 10 incorrect SSO logons achieves the same result.

(f) Either a logged on User or SSO may load, generate, or install x-values. This allows the user to receive credentials from sources other than the SSO who issued the FORTEZZA Crypto card and also supports remote rekey. Certificates and key material cannot be loaded, etc., without a successful logon of the User or SSO. Only the SSO may load x-values and/or a certificate into Certificate Index 0.

(g) Only the original SSO may Extract an x-value. The original SSO (the SSO who possesses the SSO PIN) can only extract x-values that the original SSO created/loaded. Thus the SSO cannot Extract x-values that were added to the card by any other source (see e. above).

4.3. Command Code Summary

Table 2. FORTEZZA Crypto Card Command Interface Code Summary

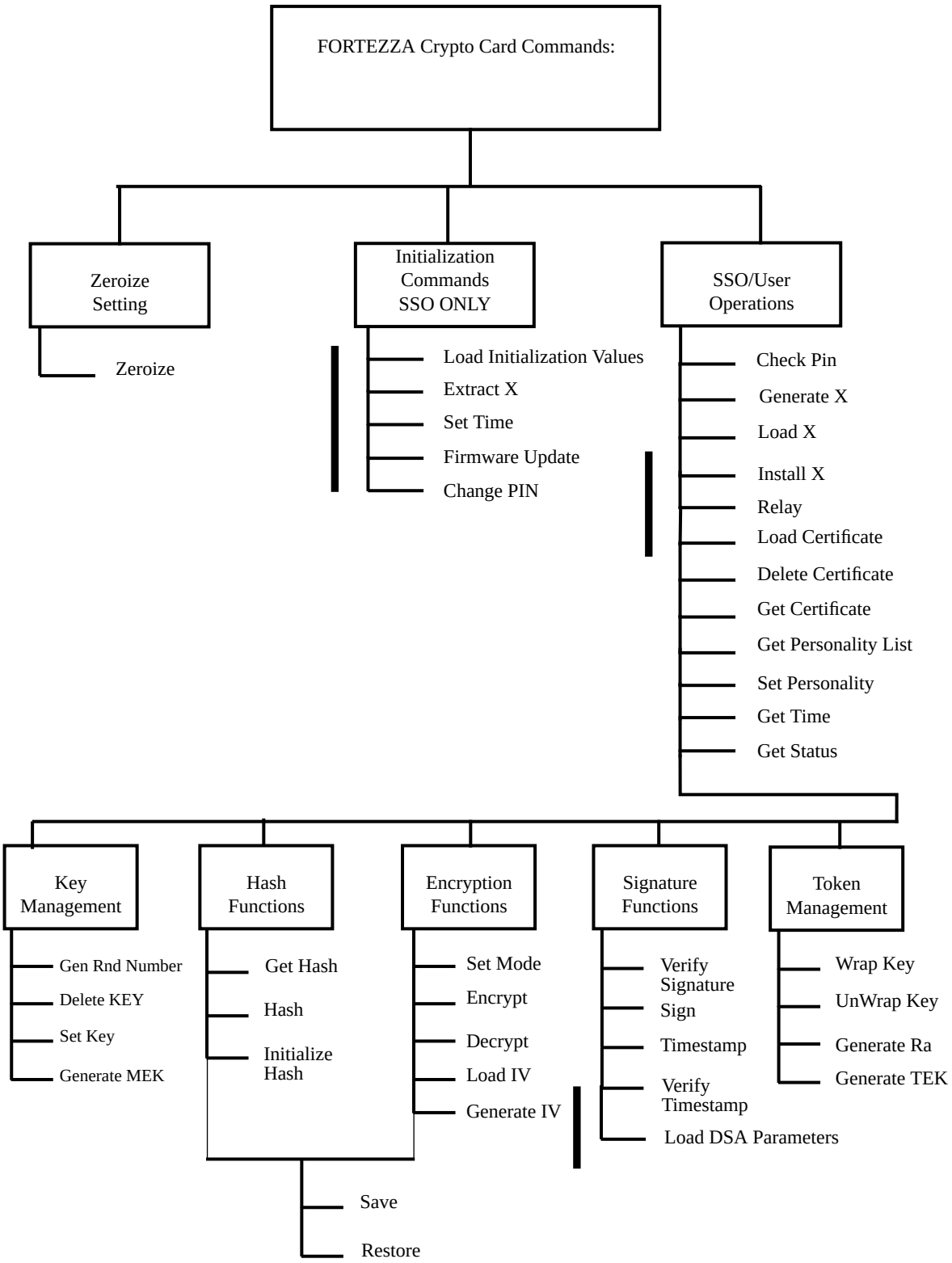
Commands	Hex Value
1) Change Pin Phrase	<X000 006E>
2) Check Pin Phrase	<X000 0004>
3) Decrypt	<X000 0007>
4) Delete Certificate	<X000 0092>
5) Delete Key	<X000 000B>
6) Encrypt	<X000 000D>
7) Extract X	<X000 007C>
8) Firmware Update	<X000 0070>
9) Generate IV	<X000 000E>
10) Generate MEK	<X000 0013>
11) Generate Ra	<X000 0016>
12) Generate Random Number	<X000 0019>
13) Generate TEK	<X000 0083>
14) Generate X	<X000 0085>
15) Get Certificate	<X000 001A>
16) Get Hash	<X000 0020>
17) Get Personality List	<X000 0025>
18) Get Status	<X000 0026>
19) Get Time	<X000 0029>
20) Hash	<X000 002A>
21) Initialize Hash	<X000 002C>
22) Install X	<X000 0086>
23) Load Certificate	<X000 002F>
24) Load DSA Parameters	<X000 0089>
25) Load Initialization Values	<X000 008A>
26) Load IV	<X000 0031>
27) Load X	<X000 008F>
28) Relay	<X000 0091>
29) Restore	<X000 003B>
30) Save	<X000 003E>
31) Set Key	<X000 0051>
32) Set Mode	<X000 0054>
33) Set Personality	<X000 0057>
34) Set Time	<X000 0058>
35) Sign	<X000 005B>
36) Timestamp	<X000 0061>
37) Unwrap Key	<X000 0079>
38) Verify Signature	<X000 0064>
39) Verify Timestamp	<X000 0068>
40) Wrap Key	<X000 006B>
41) Zeroize	<X000 006D>

4.4. Response Codes

Table 3.FORTEZZA Crypto Card Response Code Summary

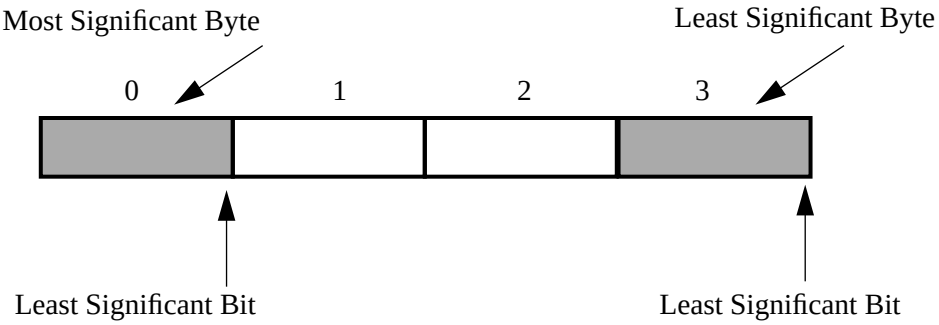
Responses	Hex Value
1) Passed	<0000 0000>
2) Failed	<0000 0001>
3) Checkword Failure	<0000 0002>
4) Invalid Type Value	<0000 0003>
5) Invalid Mode Value	<0000 0004>
6) Invalid Key Index	<0000 0005>
7) Invalid Certificate Index	<0000 0006>
8) Invalid Data Size	<0000 0007>
9) Invalid Header	<0000 0008>
10) Invalid State	<0000 0009>
11) Execution Failure	<0000 000A>
12) No Key Loaded	<0000 000B>
13) No IV Loaded	<0000 000C>
14) No X Value	<0000 000D>
15) (Response Code Deleted)	
16) No Saved Value	<0000 000F>
17) Register In Use	<0000 0010>
18) Invalid Command	<0000 0011>
19) Invalid Pointer	<0000 0012>
20) Bad Clock	<0000 0013>
21) No P,Q,G Loaded	<0000 0014>

Figure 8.0 Command Set Organization

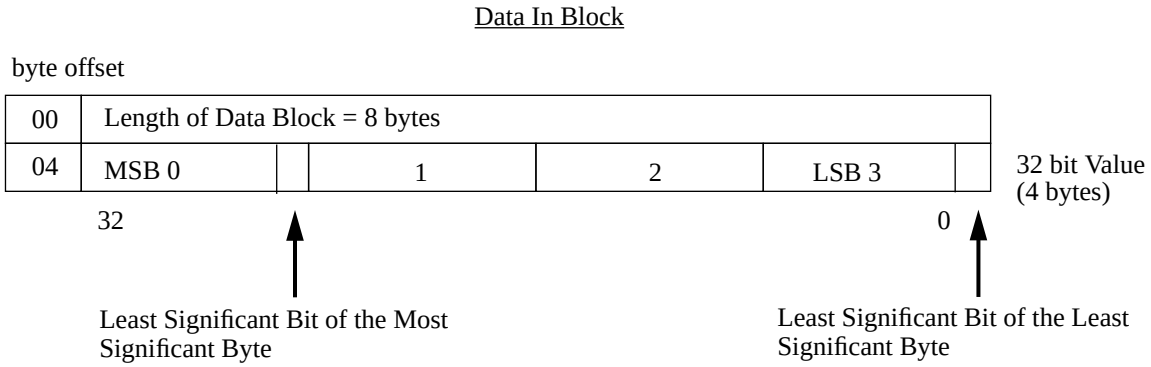


4.5 Data Parameters Bit and Byte Ordering

All commands that include data as a parameter must adhere to the following specification. This includes data to the Card and from the Card. The following figure is an example of the bit and byte ordering of a 32-bit data block.



When putting this value into the Data-In Block, the first (most significant) byte is put into the first memory location offset (lowest memory address) in the Data-In block. The next figure demonstrates this value as a Data-In Block.



When reading data from the Data-Out Block, the same ordering is followed.

4.6. COMMAND SET

4.6.1. CHANGE PIN PHRASE

The CHANGE PIN PHRASE command allows the SSO to change the User or SSO PIN on the card. To change the PIN, the SSO must provide the original PIN and the new PIN phrase, both padded to 12 bytes. The "Type" value 0000 0025 indicates SSO and 0000 002A indicates User. The 12 bytes of PIN phrase are hashed by the Card. The results of this hash will be used to "Unwrap" the previously stored and wrapped K_s value. The card will then compute a check value from the "unwrapped" K_s and compare the calculated check value with a stored check value. If the original PIN was correctly entered, the key material will be rewrapped using the new User or SSO PIN. If there is no User PIN currently on the card, the original PIN input is ignored when changing the User PIN. The "Passed" status returned from the Change PIN Phrase command indicates that the command has been successfully executed and the appropriate PIN has been changed. If the check values do not match, a failed status is returned. Only the SSO may execute this command. If this command fails, or the User PIN is successfully changed, the SSO is logged out.

Change Pin Phrase Command Block

byte offset	
00	Command = X000 006Eh
04	Pointer Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Type: SSO or User (4 bytes)
08	Original PIN Phrase (0Ch bytes)
14	New PIN Phrase (0Ch bytes)

32 0

Entry State	Exit State
Initialized	SSO Initialized when SSO PIN is changed from Default PIN.
SSO Initialized	SSO Initialized
LAW Initialized	User Initialized when User pin is changed. SSO is logged out.
User Initialized	User Initialized

Response Codes

Passed
Failed
Invalid Pointer
Invalid Type
Invalid State
Execution Failure

4.6.2. CHECK PIN PHRASE

The CHECK PIN PHRASE command inputs a PIN phrase (12 bytes) used to authenticate the SSO or User. The host must pad this value to 12 bytes. The "Type" value 0000 0025 indicates SSO and 0000 002A indicates User. The 12 bytes of PIN phrase are hashed by the Card. The results of this hash will be used to "Unwrap" the previously stored and wrapped K_s value. The Card will then compute a check value from the "unwrapped" K_s and compare the calculated check value with a stored check value. If this matches, K_s is moved to register index 0 for future use and the card cryptographic services are available. The "Passed" status returned from Check PIN Phrase command indicates that the command has been successfully executed and a match was received. If the check values do not match, a failed status is returned, and the card is logged out. After ten consecutive User logon failures, the card will zeroize the User PIN and transition to the LAW- Initialized state. After ten consecutive SSO logon failures, the card will zeroize all key material certificates, etc., and transition to the Zeroized state.

The CHECK PIN PHRASE command can also be used to authenticate the presence of a FORTEZZA Crypto Card. To accomplish this, a Random Challenge number is sent to the card, signed with a special signature key programmed on the card and returned in the Data-Out block. This signature may be externally verified.

Check Pin Phrase Command Block

byte offset	
00	Command = X000 0004h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-in Block
04	Type - SSO or User (4 bytes)
08	PIN Phrase (0Ch bytes)
14	Random Challenge (14h bytes)

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	r Value (28h bytes)
2B	s Value (28h bytes)

32 0

Entry State	Exit State
Uninitialized -Type = SSO using SSO Default PIN	Uninitialized Logged out if PIN fails
Zeroized - Type = SSO using Zeroize PIN	Uninitialized and Logged Out
Initialized - Type = SSO using SSO Default PIN	Initialized Logged out if PIN fails
SSO Initialized - Type = SSO	SSO Initialized Logged out if PIN fails
LAW Initialized - Type = SSO	LAW Initialized Logged out if PIN fails
User Initialized - Type = SSO	User Initialized Logged out if PIN fails
User Initialized - Type = User	Standby User Initialized if PIN fails
Standby - Type = SSO	User Initialized Logged out if PIN fails
Standby - Type = User	Standby User Initialized if PIN fails
Ready - Type = SSO	User Initialized Logged out if PIN fails
Ready - Type = User	Ready User Initialized if PIN fails
Any state and fail SSO PIN 10 times	Zeroized
User Initialized, Ready or Standby and fail User PIN 10 times	LAW Initialized

Response Codes

Passed
Failed
Invalid Pointer
Invalid State
Invalid Type
Execution Failure

4.6.3. DECRYPT

The DECRYPT command supports multiple modes of decryption for user data. Prior to executing this command a decryption key must be loaded into the cryptologic. The Load IV command must have been previously executed.

The Capstone chip requires that data to be encrypted or decrypted be preceded by an 8-byte header. This header is prepended to plaintext and ciphertext data areas by the card firmware. The host software, when choosing data areas within the mailbox should be aware of this. The card will check mailbox boundaries to ensure that only valid mailbox memory is addressed and will return an INVALID POINTER response otherwise. The card will not check for overlaps of data areas and it is the responsibility of the host software to manage data area pointers so that data is not unintentionally overwritten.

Decrypt Command Block

byte offset

00	Command = X000 0007h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-in Block in bytes (4 bytes)
04	Length of Cipher Text in Bits
08	Pointer to the Cipher Text

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block in bytes (4 bytes)
04	Length of Plain Text in bits (4 bytes)
08	Plain Text

32 0

4.6.3. DECRYPT (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Size
Invalid Pointer
Invalid State
No Key Loaded
No IV Loaded
Execution Failure

Notes:

64-bit ECB, CBC, OFB, CFB modes of decryption require 64 bit blocks of data (i.e. data length must be divisible by 64 bits). 32-bit CFB, 16-bit CFB and 8-bit CFB require 32 bit blocks of data (i.e. data length must be divisible by 32 bits).

4.6.4. DELETE CERTIFICATE

The DELETE CERTIFICATE command will overwrite the data in the certificate storage location indexed with nulls. This command must pass the certificate index to the Card. After the data is deleted, the appropriate certificate index flag will be cleared.

Note: Only the SSO may delete Certificate Index 0.

Delete Certificate Command Block

byte offset

00	Command = X000 0092h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-in Block (4 bytes)
04	Certificate Index (4 bytes)

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized SSO Initialized if certificate 0 is deleted
User Initialized	User Initialized SSO Initialized if certificate 0 is deleted
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Certificate Index
Invalid Pointer
Invalid State

4.6.5. DELETE KEY

The DELETE KEY command will zeroize the key in the key register indexed. This command must pass the register index to the Card. Note that key register index 0 is not valid for this command.

Delete Key Command Block

byte offset

00	Command = X000 000Bh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Key Register Index (4 bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Key Index
Invalid Pointer
Execution Failure
Invalid State

4.6.6. ENCRYPT

The ENCRYPT command supports multiple modes of encryption of user data. The host must pad the data out with zeros to the appropriate byte length, depending on the mode. Prior to executing this command an encryption key must be loaded into the cryptologic. Additionally, the Generate IV or Load IV command must have been previously executed.

The Capstone chip requires that data to be encrypted or decrypted be preceded by an 8-byte header. This header is prepended to plaintext and ciphertext data areas by the card firmware. The host software, when choosing data areas within the mailbox should be aware of this. The card will check mailbox boundaries to ensure that only valid mailbox memory is addressed and will return an INVALID POINTER response otherwise. The card will not check for overlaps of data areas and it is the responsibility of the host software to manage data area pointers so that data is not unintentionally overwritten.

Encrypt Command Block

byte offset

00	Command = X000 000Dh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-in Block in bytes (4 bytes)
04	Length of Plaintext data in bits (4 bytes)
08	Pointer to Plain Text

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block in bytes (4 bytes)
04	Length of Cipher Text in bits (4 bytes)
08	Cipher Text

32 0

4.6.6. ENCRYPT (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Size
Invalid Pointer
Invalid State
No Key Loaded
No IV Loaded
Execution Failure

Notes: 64 bit ECB, CBC, OFB, CFB modes of encryption require 64 bit blocks of data (i. e. data length must be divisible by 64 bits). 32 bit CFB, 16 CFB and 8 bit CFB require 32 bit blocks of data (i. e. data length must be divisible by 32 bits).

4.6.7. EXTRACT X

The EXTRACT X command is used to safely archive a private x-value, support remote rekey operations, and duplicate personalities between cards as appropriate. A password is supplied to provide positive control of the instantiation of the key material. **Only x-values that were loaded or generated by the SSO may be extracted.** Prior to executing this command, the SSO must use the SET PERSONALITY command to select a storage location containing KEA X-values that will be used by the EXTRACT X command to generate a TEK. The primes, p, q, and g, are those associated with the extracted X-value and may be different than those associated with the KEA X used to generate the TEK.

Internal to the card, this command will perform the following functions:

(a) Load the appropriate target X-value into the cryptologic.

(b) Generate an Ra.

(c) Generate a TEK (Rb=1, use Extractor's KEA X, Installer's KEA Y).

(d) Wrap[0] the X-value with the TEK.

(e) Exclusive-OR the wrapped X-value with the password.

The password must be 24 bytes, with the host performing canonicalization as required.

The Response field in the Command Block from the EXTRACT X command indicates if the command has been successfully completed.

NOTE: In order to prevent unauthorized replication of key material, the following rules will be enforced by the card:

(a) This command can only be executed by the SSO.
(b) Only x-values generated, loaded, or installed by the SSO may be extracted.
(c) The password shall be checked to make sure it is non-zero.

0. This Wrap function is used to encrypt a 160-bit value as described in this document.

4.6.7. EXTRACT X (cont'd)

Extract X Command Block

byte offset

00	Command = X000 007Ch
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Certificate Index Value (4 bytes)
08	Type (DSA=0000 000Ah - KEA=0000 0005h - 4 bytes)
0C	Password (18h bytes)
24	Length of Public Component in hex bytes (4 bytes)
28	Yb (80h Byes)

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	Covered X value (18h bytes)
1C	Ra (80h bytes)
9C	Length of p Parameter in bits (4 bytes)
A0	p Value
	Length of q Parameter in bits (4 bytes)
	q Value
	Length of g Parameter in bits (4 bytes)
	g Value

32 0

4.6.7. EXTRACT X (cont'd)

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized

Response Codes

Passed
Execution Failure
Invalid Pointer
Invalid State
Invalid Type
Invalid Certificate Index
No X Value

4.6.8. FIRMWARE UPDATE

The Firmware Update command is to update the firmware in the Card. At the completion of this command, the Card will run a checksum test on the updated software to verify that it has been loaded correctly. The destructive flag, 0000 00FF, indicates to the Card that all non-volatile memory should be zeroized (including certificates) while the non-destructive flag, 0000 FF00, indicates that certificates should be maintained during this update. In the Length of Current Block, Bit 31 set to 1 will indicate the Last Block of data. Note that the Firmware Update command executes successfully, a PASSED response will not be returned since the newly loaded firmware will be
started.

NOTE: During a firmware update, the following restrictions apply:

A) If more than one block long, data blocks must be downloaded in the correct order as specified by the manufacturer.

B) The host may not issue any other commands until the FIRMWARE UPDATE sequence is complete.

Firmware Update Command Block

byte offset

00	Command = X000 0070h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Destructive or Non-Destructive flag (4 bytes)
08	Length of the checksum in bytes (4 bytes)
0C	Checksum for the software (4 bytes)
10	Length of the Current Block in bytes & Last Block (4 bytes)
14	Firmware Update

32 0

4.6.8. FIRMWARE UPDATE (cont'd)

Entry State	Exit State
Uninitialized	Uninitialized - Card is reset
Initialized	Uninitialized - Card is reset
SSO Initialized	SSO Initialized - Card is reset Uninitialized if Destructive
LAW Initialized	LAW Initialized - Card is reset Uninitialized if Destructive
User Initialized	User Initialized - Card is reset Uninitialized if Destructive

Response Codes

Invalid Pointer
Invalid Data Size
Checkword Failure
Invalid State
Failed

4.6.9. GENERATE IV

The GENERATE IV command is used prior to an Encrypt command. This command will generate, check and then write the 192-bit IV value into the cryptologic. The output from the Generate IV command is also written into the Data-Out area. A Set Key command must be executed prior to this command.

Generate IV Command Block

byte offset	
00	Command = X000 000Eh
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	IV Data Results (18h bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Passed
No Key Loaded
Invalid Pointer
Invalid State

4.6.10. GENERATE MEK

The GENERATE MESSAGE ENCRYPTION KEY (MEK) command will use the internal random number generator to produce a key for message encryption. The results of this command will then be stored in the key-register index indicated for future use. The key is not available in plain text form outside of the Card. Although the MEK is available for use immediately after generation, a Set Key command should be issued to insure that the correct MEK is used.

This command can be used prior to executing the Encrypt or Wrap command.

Generate MEK Command Block

byte offset

00	Command = X000 0013h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Key Register Index (4 Bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Key Index
Checkword Failure
Passed
Execution Failure
Register In Use
Invalid State

4.6.11. GENERATE R_a

The GENERATE R_a command is used with the Generate TEK command to support the public/ private key exchange. This command generates a 1024-bit random value that is written to the Data-Out Block.

Generate Ra Command Block

byte offset	
00	Command = X000 0016h
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	Ra (80h bytes)

32 0

Entry State	Exit State
Ready	Ready

Response Codes

Invalid Pointer
Checkword Failure
Passed
Execution Failure
Invalid State
No X Value

4.6.12. GENERATE RANDOM NUMBER

GENERATE RANDOM NUMBER will generate a 160-bit random number and return the value in the Data Out-Block.

Generate Random Number Command Block

byte offset

00	Command = X000 0019h
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	Random Number (14h Bytes)

32 0

Entry State	Exit State
Uninitialized	Uninitialized
Initialized	Initialized
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Note: This command may be used before logging onto the card.

Response Codes

Invalid Pointer
Passed

4.6.13. GENERATE TEK

The GENERATE TOKEN ENCRYPTION KEY (TEK) command is used to support public/ private key exchanges. This command is used by both initiator and recipient after the key exchange. This command generates an 80-bit key based on the parameters passed in with the Data-In Block. This key is stored in the key register index indicated. The Generate R_a command is executed prior to this command with the results passed as an input parameter to the Generate TEK command. For an initiator, the Recipient's Public Component (KEA-Y_b) and R_b values are also input parameters to this command. For a recipient, The parameters are described as follows:

1) Key Register index to store the resultant TEK (1-9)
2) R_a-1024-bit Initiators random number
   Results of the Generate R_a command (if initiator),
   or received from initiator (if recipient)
3) R_b-Recipient's 1024-bit random number
   Received from recipient (if initiator), Results of the Generate R_a command (if recipient), or set to 0000.....0001h (initiator and recipient) for FORTEZZA E-Mail.
4) Length of the Public component in hex bytes
5) Y_b (if initiator) or Y_a (if recipient) - Other party's 1024-bit KEA Public component
6) Initiator or Recipient of the key exchange-Used to indicate whether this is the negotiator (Party A) of the key exchange or if this command is being executed as the recipient (Party B) in the exchange. For the initiator, this value is 0000 0000. For the recipient, this value is 0000 0001.

This command generates a TEK used by the Encrypt, Decrypt or Wrap command. Prior to executing this command, the host must be sure that the KEA parameters are loaded. If this TEK is to be used for token generation, the Set Personality command will load the proper KEA parameters.

Generate TEK Command Block

byte offset	32 0
00	Command = X000 0083h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

4.6.13. GENERATE TEK (cont'd)

Data-In Block

byte offset

00	Length of Data-in Block (4 bytes)
04	Register Index (4 Bytes)
08	Ra (80h bytes)
88	Rb (80h bytes)
108	Initiator or Recipient of the key exchange (4 bytes)
10C	Length of Public Component in hex bytes (4 bytes)
110	Yb-Public Component

32 0

Notes: For FORTEZZA E-Mail, the R_b is preset to all zeros with the last (LSB) bit set to 1.

Entry State	Exit State
Ready	Ready

Response Codes

Invalid Pointer
Invalid Key Index
Register In Use
Checkword Failure
Passed
Execution Failure
Invalid State
No X Value

4.6.14. GENERATE X

The GENERATE X command allows the SSO or User to generate the unique X & Y components prior to loading its Certificate. The certificate index and the component type to be generated (KEA, DSA or both) are parameters passed within the Data-In Block. The specific type value for KEA is 0000 0005h, DSA is 0000 000Ah or both is 0000 000Fh. Additionally, the p, q and g parameters are also inputs to this command. Each p, q and g input value is preceded by a length, in bits, of the following p, q and g parameters value. The values for p, q and g must follow the following:

0) Type-KEA (value of 0000 0005h), DSA (0000 000Ah) or both (0000 000Fh)
1) p size and g size-denotes the size of the prime modulus in bits
2) q size-denotes the size of the prime divisor in bits
3) p - the prime modulus: 512-1024 bits (64-128 bytes)
4) q - the prime divisor: 160 bits (20 bytes)
5) g - a value: 512-1024 bits (64-128 bytes) Equal to the size of p

This command will generate the specified certificate's X and Y values. The resultant Y for DSA or KEA is written to the Data-Out Block. The secret X value is wrapped with K_s and stored in non-volatile memory in their associated certificate index location. Please note that Index 0 is not a valid index for this command.

Generate X Command Block

byte offset	32 0
00	Command = X000 0085H
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

4.6.14. GENERATE X (cont'd)

Data-In Block

byte offset

00	Length of Data-in Block (4 bytes)
04	Certificate Index (4 Bytes))
08	Type: DSA or KEA or Both (4 bytes)
0C	Length of p Parameter in bits(4 bytes)
10	p Parameter
	Length of q Parameter in bits(4 bytes)
	q Parameter
	Length of g Parameter in bits(4 bytes)
	g Parameter

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	Length of Public Component in hex bytes (4 bytes)
08	Resultant Yb (KEA=80h bytes and DSA=80h bytes)

32 0

Entry State	Exit State
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Certificate Index
Invalid Type Value
Passed
Execution Failure
Invalid State
Invalid Data Size

4.6.15. GET CERTIFICATE

The GET CERTIFICATE command returns the entire Certificate indicated by the Certificate index value provided in the Data-In Block.

Get Certificate Command Block

byte offset	
00	Command=X000 0001Ah
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Certificate Index (4 bytes)

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	Certificate (800h bytes)

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Invalid Certificate Index
Passed
Invalid Pointer
Invalid State

4.6.16. GET HASH

The GET HASH command returns the current hash value for a block(s) of data that has been hashed. This command expects the last or only block of data to be hashed as input to this command. If the block hash is on a 512-bit boundary, a last block of zero length is valid. The card will maintain the entire length of the hashed message.

The Capstone chip requires that data to be hashed be preceded by a 12-byte header. This header is prepended to hash data areas by the card firmware. The host software, when choosing data areas within the mailbox should be aware of this. The card will check mailbox boundaries to ensure that only valid mailbox memory is addressed and will return an INVALID POINTER response otherwise. The card will not check for overlaps of data areas and it is the responsibility of the host software to manage data area pointers so that data is not unintentionally overwritten.

Get Certificate Command Block

byte offset	
00	Command=X000 00020h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Length of Last Block Hash Data in bits (4 bytes)
08	Pointer to Last Block Hash Data (4 bytes)

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	Hash Value (14h bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Passed
Invalid State

4.6.17. GET PERSONALITY LIST

The GET PERSONALITY LIST command provides the Host system with the list of the previously stored personalities available to the user. The Card will return an entry for all certificate locations including index 0. All the certificate names stored in non-volatile memory will be sent to the host for the User to select. Locations which have not been loaded with a valid certificate will return a NULL (0000 0000h) value for all 32 bytes. All certificate names have a length of 32 bytes.

Get Personality List Command Block

byte offset

00	Command=X000 00025h
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	String 0 (20h bytes)
24	String 1 (20h bytes)
44	String 2 (20h bytes)
N	String N (20h bytes)-Certificate Maximum

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Pointer
Invalid State

4.6.18. GET STATUS

The GET STATUS command allows the SSO or User to obtain information as to the current status of the card. Status information returned includes current state, Capstone serial number, mode, personality, which key registers are in use, and which certificate slots are in use.

Key Register Status will be indicated as the most significant 10 bits in the 4-byte return code. Key Register 0 will be represented in Bit 7, Key Register 1 in Bit 6, etc. A set bit will indicate that the key register is in use, a cleared bit will indicate that the key register is available. Key Register 0 bit will always be set.

Certificate Status will be indicated as the most significant X bits in the 16-byte return code (where X= Maximum number of certificates on the card). Certificate 0 will be represented in Bit 7 of the MSB, Certificate 1 in Bit 6, etc. A set bit will indicate that a Load Certificate command has loaded data into the certificate location. A cleared bit will indicate that a Delete Certificate command has deleted the data in the certificate location and new data may be written there. Currently, a maximum of 128 certificate flag bits are available. Note that hosts need to keep track of which certificates have X values associated with them since the Get Status command does not indicate this.

Key Register Count and Certificate Count are the maximum number of Key Registers and Certificate storage locations on the card.

For the Current Mode Indicator, the lower 2 bytes will be used to indicate the Decrypt Mode and the upper 2 bytes will be used to indicate the Encrypt Mode. The 2 byte representation of the modes will be the same as listed in the SET MODE command.

The CAPSTONE Serial Number is padded to 8 bytes for consistency with future cards. The format is 4 bytes of 00H followed by the 32-bit serial number in hex.

4.6.18. GET STATUS (cont'd)

Get Status Command Block

00	Command=X000 0026h
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

00	Length of Data-Out Block (4 bytes)
04	CAPSTONE Serial Number (8 bytes)
0C	Current State (4 bytes)
10	Current Mode (4 bytes)
14	Current Personality (4 bytes) or 0000 if in Standby
18	Key Register Count (4 bytes)
1C	Key Register Flags (4 bytes)
20	Certificate Count (4 bytes)
24	Certificate Flags (16 bytes)

32 0

Entry State	Exit State
Uninitialized	Uninitialized
Zeroized	Zeroized
Initialized	Initialized
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Note: This command may be used before logging onto the card.

Response Codes

Invalid Pointer
Passed

4.6.19. GET TIME

The GET TIME command allows the user to retrieve the current time from the onboard real-time clock.

In the command format, XX indicates unused digits and will be zero-filled (XX=00H).

Get Time Command Block

byte offset

00	Command=X000 0029h
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-Out Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Date/Time (YYYYMMDDHHMMSSXX-16 bytes)

32 0

Entry State	Exit State
Uninitialized	Uninitialized
Initialized	Initialized
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Note: This command may be used before logging onto the card.

Response Codes

Invalid Pointer
Bad Clock
Passed

4.6.20. HASH

The HASH command hashes the data provided with the command. It will continue to hash using the current hash state as a starting point. The hash value may be reset by the INITIALIZE HASH command or restored by the RESTORE command. This command will not output the Hash value. The GET HASH command must be used to obtain the hash value.

The hash command requires the host to input the length of the data to be hashed, in bits, as a Data-In Block parameter. The Current Block Size defines the length of the data, in bits, that is included with this Data-In Block. The number of bits to hash must be a multiple of 512 bits (64 bytes) greater than 0.

The Capstone chip requires that data to be hashed be preceded by a 12-byte header. This header is prepended to hash data areas by the card firmware. The host software, when choosing data areas within the mailbox should be aware of this. The card will check mailbox boundaries to ensure that only valid mailbox memory is addressed and will return an INVALID POINTER response otherwise. The card will not check for overlaps of data areas and it is the responsibility of the host software to manage data area pointers so that data is not unintentionally overwritten.

Hash Command Block

byte offset

00	Command=X000 002Ah
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block in bytes (4 bytes)
04	Current Block size in bits (4 bytes)
08	Pointer to the Data to be Hashed

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Data Size
Invalid Pointer
Passed
Invalid State

4.6.21. INITIALIZE HASH

The INITIALIZE HASH command will initialize the onboard Hash function according to the NIST SHA-1 standard.

Initialize Hash Command Block

byte offset

00	Command=X000 002Ch
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid State

4.6.22. INSTALL X

The INSTALL X command is used by the SSO or the user to restore an archived private DSA or KEA X-value, support remote rekey operations, and duplicate personalities between cards as appropriate A password is required to verify positive control of the instantiation of the key material.

Prior to issuing the INSTALL X command, the host must use the SET PERSONALITY command to access the installer's KEA X that will be used to uncover the TEK. (The KEA X does not have to be from the same personality that is receiving the installed X.)

Internal to the FORTEZZA card, this command will perform the following functions:

(a) Load the password, wrapped X-value, R_a,Y_a, and primes (from the Extract X command).
(b) Exclusive-OR the password and the password protected X-value.
(c) Generate the TEK (Rb=1, Installer's KEA X[2], and Extractor's KEA Y).
(d) Unwrap the X-value.
(e) Cover the X-value with the K_s.
(f) Place the X-value in the appropriate memory location.

The password must be 24 bytes, with the host performing canonicalization as required.

Certificate index 0 cannot be specified by the user.

The Response field in the Command Block from the INSTALL X command indicates if the command has been successfully completed.

[2] This KEA X is selected by the SET PERSONALITY command that must be executed prior to the

INSTALL X command.

4.6.22. INSTALL X (cont'd)

Install X Command Block

byte offset

00	Command=X000 086h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Certificate Index (4 bytes)
08	Type (DSA or KEA-4 bytes) DSA = 0000 000Ah KEA = 0000 0005h
0C	Password (18h bytes)
18	Length of Yb (4 bytes)
1C	Yb (80h bytes)
9C	Covered X value (18h bytes)
B4	Ra (80h bytes)
134	Length of p Parameter in bits (4 bytes)
	p Value
	Length of q Parameter in bits (4 bytes)
	q Value
	Length of g Parameter in bits (4 bytes)
	g Value

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Passed
Invalid Pointer
Invalid State
Invalid Certificate Index
Invalid Data Size
Invalid Type Value

4.6.23. LOAD CERTIFICATE

The LOAD CERTIFICATE command allows the SSO or user to load User's certificates for storage into the Card's non-volatile certificate memory area. The certificates loaded at this time define the "Personalities" available to the User of the card. The Card Information Structure (CIS) tuples can be read to determine the maximum number of certificates that can be stored on the card. The User's Certificate is loaded after a Generate X or Load X command has generated the certificate's unique X and Y values. The certificate is loaded into the index location containing the stored secret X value. This command is repeated for each certificate to be stored. Each certificate is associated with a "Personality" User name. The certificate label is included with the certificate as a 32-byte character string.

NOTE: Certificate Index 0 is reserved for use by the SSO. Only the SSO may load a certificate at index 0.

Load Certificate Command Block

byte offset	
00	Command=X000 02Fh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Certificate Index Identifier (4 bytes)
08	Certificate Label (20h bytes)
28	Length of Certificate in bytes (4 bytes)
2C	Certificate Data (800h bytes)

32 0

4.6.23. LOAD CERTIFICATE (cont'd')

Entry State	Exit State
SSO Initialized	LAW Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Certificate Index
Passed
Execution Failure
Invalid State

Notes: For FORTEZZA, User certificates are generated and signed by the LAW and the Load Certificate command is used to load the resultant certificate into the User's Card. Before the certificate can be generated and signed, the SSO or User will use the Card with the Generate X or Load X command to obtain the public component Y used with the certificate.

4.6.24. LOAD DSA PARAMETERS

This command allows the user to load externally supplied p, q and g parameters to use for signature verification outside of the domain of the currently selected personality. These are the parameters associated with the Digital Signature Algorithm (DSA).

These parameters are defined as follows:

1) psize and gsize-denotes the size of the prime modulus in bits
2) qsize-denotes the size of the prime divisor in bits
3) p-the prime modulus: 512-1024 bits (64-128 bytes)
4) q-the prime divisor: 160 bits (20 bytes)
5) g-a value: 512-1024 bits (64-128 bytes) Equal to the size of p

The psize and qsize must be set to increments of 32-bits. Note that p and q may be padded with zeros to meet this incremental size requirement. If parameters other than the default (SSO loaded) set are used, it is the user's responsibility to insure that the proper set is loaded prior to any cryptographic operations.

NOTE: The card will perform all signature verification operations subsequent to the issuance of this command with the loaded DSA primes. After this command is issued, any previously selected personality will no longer be selected, and the card will transition from the READY state to the STANDBY state. Signature generation will no longer be possible until the SET PERSONALITY command is re-issued. It is the Host's responsibility to coordinate these commands.

Load DSA Parameters Command Block

byte offset

00	Command=X000 089h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

4.6.24. LOAD DSA PARAMETERS (cont'd)

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Length of p Parameter in bits (4 bytes)
08	p Value
	Length of q Parameter in bits (4 bytes)
	q Value
	Length of g Parameter in bits (4 bytes)
	g Value

32 0

Entry State	Exit State
Standby	Standby
Ready	Standby

Response Codes

Invalid Pointer
Invalid Data Size
Passed
Invalid State

4.6.25. LOAD INITIALIZATION VALUES

This command allows the SSO to load the Card's initialization parameters. These parameters include:

1) Random Seed Value (64 bits)
2) User Storage Key Variable K_s-A plaintext value (80 bits)

The random seed value is loaded into the card and used to seed the internal random number generator. The User Storage Key, K_s, is used in combination with the PIN authentication algorithm. K_s is wrapped by the hash of the User's PIN Phrase value, after completion of the Change (User) Pin Phrase command by the SSO. The K_s value is loaded in as a plain text value by this command.

Load Initialization Values Command Block

byte offset

00	Command=X000 008Ah
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Random Seed (8 bytes)
0C	User K_s (0Ah bytes)

32 0

Entry State	Exit State
Uninitialized	Initialized

Response Codes

Invalid Pointer
Passed
Invalid State

4.6.26. LOAD IV

The LOAD INITIALIZATION VECTOR (IV) command is used prior to an Encrypt or Decrypt command. This command writes the 192-bit value into the decrypt cryptologic. A Set Key command must be executed prior to this command.

Load IV Command Block

byte offset

00	Command=X000 0031h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	IV Value (18h bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
No Key Loaded
Execution Failure
Passed
Invalid State

4.6.27. LOAD X

The LOAD X command allows the SSO or user to load the secret X value into the card during initialization. This command must specify for which certificate this is being loaded and whether the value being loaded is for the DSA or KEA mode. The input parameters to this command are the type of x being loaded (DSA or KEA or both), the x value and it's p, q and g parameters. These parameters are defined as follows:

0) Type-KEA (value of 0000 0005h), DSA (0000 000Ah) or both (0000 000Fh)
1) Length of the x Value-denotes the size of the x in bits
2) x value-secret value
3) psize-denotes the size of the prime modulus in bits
4) p-the prime modulus: 512-1024 bits (64-128 bytes)
5) qsize-denotes the size of the prime divisor in bits
6) q-the prime divisor: 160 bits (20 bytes)
7) g size-denotes the size of the prime modulus in bits
8) g-a value: 512-1024 bits (64-128 bytes-the size of p)

After the X value and its associated p, q and g parameters are received, the Card will generate the public key (Y) corresponding to the X value passed. The 1024 bits of public key will be returned to the host in the Data-Out Block. The X value will be stored at the location specified by the certificate index. Note that certificate index 0 is not a valid input for this command. The X value is wrapped with K_s prior to storage on the Card. After the Certificate is generated, it is loaded into the Card using the same certificate index value.

Load X Command Block

byte offset	
00	Command=X000 008Fh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

4.6.27. LOAD X (cont'd)

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Certificate Index (4 Bytes)
08	Type: DSA or KEA, or Both (4 bytes)
0C	Length of x value (4 bytes)
10	x value
	Length of p Parameter in bits (4 bytes)
	p Value
	Length of q Parameter in bits (4 bytes)
	q Value
	Length of g Parameter in bits (4 bytes)
	g Value

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	Length of Public Component in hex bytes (4 bytes)
08	Yb Resultant (KEA=80h bytes or DSA or both=80h bytes)

32 0

Entry State	Exit State
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Certificate Index
Invalid Data Size
Invalid Type Value
Passed
Execution Failure
Invalid State

4.6.28. RELAY

The RELAY command may used in the process of restoring an archived private x-value, to support remote rekey operations, and to duplicate personalities between cards as appropriate. A password is required to verify positive control of the instantiation of the key material.

Prior to executing the RELAY command, the host must use the SET PERSONALITY command to indicate which KEA X will be used to uncover the TEK. This KEA X must correspond to the Installer's KEA Y used in the EXTRACT X operation.

Internal to the card, this command will perform the following functions:

(a) Load the password, wrapped X-value, Ra, and extractor's Ya, and installer's Yb
(b) Exclusive-OR the original password and the password protected Wrapped X-value
(c) Generate the TEK (use Rb=1, extractor's Ya, Relayer's KEA X, Ra)
(d) Unwrap the Wrapped X-value
(e) Generate a new Ra, now Ra'
(f) Generate a new TEK (use Rb=1, Relayer's KEA X, Installer's KEA Yb, Ra')
(g) Wrap the X-value with the new TEK
(h) Exclusive-OR the new password and the Wrapped X-value

The password must be 24 bytes, with the host performing canonicalization as required.

The Response from the RELAY command indicates if the command has been successfully completed.

4.6.28. RELAY (cont'd)

Relay Command Block

byte offset

00	Command=X000 0091h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Original Password (18h bytes)
1C	Length of Y (4 bytes)
20	Y (80h bytes)
A0	Ra (80h bytes)
120	Password protected Wrapped x-value (18h bytes)
138	Password for Relayed x-value (18h bytes)
150	Length of Yb (4 bytes)
154	Yb (80h bytes)

32 0

Data-Out Block

byte offset

00	Length of Data-In Block (4 bytes)
04	New Ra (80h bytes)
84	Password protected Wrapped x-value (18h bytes)

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Ready	Ready

Response Codes

Invalid Pointer
Passed
Execution Failure
Invalid State

4.6.29. RESTORE

The RESTORE command will restore the state of the cryptologic operation as specified by the Crypto Type input parameter. Additionally, if the Data-In block contains data, the card will use this value as the input value. If the length of the restore data is zero, the card will use the internally stored data (stored as a result of a Save command).

If the Crypto Type is Hash, 0000 0002h, and the length of the restore data is zero, then the hash value that had been previously saved in the Long Term Hash Storage Register into the Temporary Hash Storage Register. If Hash is specified, and the length of the restore data is not zero, the card will load the supplied intermediate hash value and number of bits (previously) hashed.

If the Crypto Type is Encrypt or Decrypt, 0000 0000h and 0000 0001h respectively, and the length of the restore data is nonzero, the following operation is performed.

A) Exclusive-OR the saved XOR value with the 80 Least Significant Bits of the appropriate wrapped-MEK or TEK[4].
B) Set the Mode and Enc/Dec settings in accordance with the saved parameters.
C) Load the encrypt or decrypt state.

Restore Command Block

byte offset

00	Command=X000 003Bh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Crypto Type (4 Bytes: Hash, Encrypt or Decrypt
08	Intermediate Hash (20 bytes) or XOR value (12 bytes)
	Number of bits hashed or Zero (00H) Pad (8 bytes)

32 0

[4]This wrapped-MEK o r TEK is selected with the SET KEY command that must be executed prior to the RESTORE command.

4.6.29. RESTORE (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
No Saved Value
Invalid Mode Value
No Key Loaded
Execution Failure
Passed
Invalid State
Invalid Type Value

4.6.30. SAVE

The SAVE command saves the state of the cryptologic operation specified by the Crypto Type input parameter. If the Crypto Type indicates Hash, 0000 0002h, the card transfers the current hash value from the Temporary Hash Value Storage Register into the Long Term Hash Storage Register and also outputs this data. If the Crypto Type indicates Encrypt or Decrypt, 0000 0000h or 0000 0001h respectively, the card performs the operation outlined below which outputs the encrypt or decrypt state, and also stores the encrypt or decrypt state on the card. This command is later followed by a RESTORE command to continue the cryptologic operation.

If hash is specified, the card will return the intermediate hash value and the number of bits hashed.

If encrypt or decrypt is specified, the following operation is performed:

A) The encrypt or decrypt state is extracted from the cryptologic.
B) Construct
   (State [64 bits] + 000H [12 bits] + Mode [3 bits] + Enc/Dec [1 bit])
   where:
   Mode corresponds to the lower 3 bits of the cryptographic mode as set by the SET MODE command, and Enc/Dec is set to one (1) for Decrypt and cleared (0) for Encrypt.
C) Exclusive-OR this value with the 80 Least Significant Bits of the wrapped MEK or TEK selected by a previous Set Key command
D) Return the value.

Save Command Block

byte offset	
00	Command=X000 003Eh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Crypto Type (4 Bytes: Hash, Encrypt or Decrypt)

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	Intermediate Hash [20 bytes] or XOR value [12 bytes]
	Number of bits hashed [8 bytes] or Zero (00H) Pad [8 bytes]

32 0

4.6.30. SAVE (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Type Value
No Key Loaded
Passed
Invalid State

4.6.31. SET KEY

The SET KEY command will load the key value from the specified register index into the cryptologic for encryption/decryption data processing. Execution of this command does not delete the key from the register indexed. This command must be performed prior to a LOAD IV or GENERATE IV command.

Set Key Command Block

byte offset

00	Command=X000 0051h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Key Register Index (4 bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Key Index
Checkword Failure
Execution Failure
No Key Loaded
Passed
Invalid State

4.6.32. SET MODE

The SET MODE command sets the Encrypt or Decrypt command modes used by the cryptologic. The command passes a mode parameter to specify if the mode being set is for the encrypt or decrypt logic. Encrypt mode is 0000 0000h and Decrypt mode is 0000 0001h. The next parameter defines the cipher mode type. If the mode is not set, the Card will default to 64-bit Cipher Block Chaining (CBC) (Mode Type 1).

The following defines the Algorithm Type values:

0000 0000h=64-bit ECB	0000 0003h=64-bit CFB
0000 0001h=64-bit CBC	0000 0004h=32-bit CFB
0000 0002h=64-bit OFB	0000 0005h=16-bit CFB
	0000 0006h=8-bit CFB

Set Mode Command Block

byte offset

00	Command=X000 0054h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Mode (4 bytes)
08	Algorithm Type (4h bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Type Value
Passed
Invalid State
Invalid Mode Value

4.6.33. SET PERSONALITY

The SET PERSONALITY command allows the User to select the personality to use. The Get Personality List command is used prior to this command, to get the list of certificate labels. This command is then used to select one of the personalities on the list. Upon proper execution of this command, the Card will output a response and use this Personality for the remainder of the session or until a Set Personality or a Load DSA Parameters command is executed. The Set Personality command can be executed at any time during the user's session. The card uses the personality to select which x-values and DSA/KEA parameters to use for signature and key exchange functions. There must be an X value, either DSA or KEA, associated with the Certificate Index value specified in the Data-In block, or a NO X VALUE error response will be returned.

Set Personality Command Block

byte offset

00	Command=X000 0057h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Certificate Index Value (4 bytes)

32 0

Entry State	Exit State
LAW Initialized	LAW Initialized
User Initialized	User Initialized
Standby	Ready
Ready	Ready

Response Codes

Invalid Pointer
Passed
Invalid Certificate Index
No X Value
Invalid State

4.6.34. SET TIME

The SET TIME command allows the SSO to set the current date and time for the onboard real-time clock. This command shall only be necessary when the card is initialized or possibly after battery replacement. The user will never be allowed to change the time. Time should be set in accordance with local policy, i. e., Greenwich Mean Time, local time, etc. The card will insure that the new date and time setting is later than the last stored value, thus lessening the chance of a signature being backdated.

NOTE: To accommodate forward clock drift, the clock may be stopped in one of two methods:
(1) (For cards with removable batteries) Remove the battery or
(2) (For cards with permanent batteries) Issue the SET TIME command with an all-zero value.

Set Time Command Block

byte offset

00	Command=X000 0058h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Date/Time (YYYYMMDDHHMMSS00-16 bytes)

32 0

Entry State	Exit State
Uninitialized	Uninitialized
Initialized	Initialized
SSO Initialized	SSO Initialized
LAW Initialized	LAW Initialized
User Initialized	User Initialized

Response Codes

Invalid Pointer
Execution Failure
Invalid State
Bad Clock
Passed

4.6.35. SIGN

The SIGN command computes the digital signature values, r and s, over the host provided data. The data is signed with the private key associated with the certificate selected by the Set Personality command.

This command works with the output from the HASH commands defined earlier. The 20-byte value from the hash provides the input for the Sign command. After the hash has been computed, the Sign command is executed to return a digital signature computed on the hash of a message using the Personality private key. The Card writes the resultant r and s values into 320 bit fields. For FORTEZZA, only the top 160 bits of the r and s values are used as the signature. The lower 160 bits of each value are discarded.

Sign Command Block

byte offset	
00	Command=X000_005Bh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Hash Value (14h Bytes)

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	r Value (28h bytes)
2C	s Value (28h bytes)

32 0

Entry State	Exit State
Ready	Ready

Response Codes

Invalid Pointer
Execution Failure
Passed
Invalid State
No X Value

4.6.36. TIMESTAMP

The TIMESTAMP command computes the digital signature values, r and s, over the host provided hash value and the date/time obtained from the onboard clock. To prevent spoofing, the date/time value is cryptographically processed prior to being included into the digital signature. The following steps will be performed on the card:

(a) Concatenate the 160-bit message hash value with the 96-bit date/time value of the onboard real-time clock.
(b) Internally hash the result of (a)
(c) Digitally sign the hash from (b) using a special signature key common to all CAPSTONE chips.

This command works with the output from the HASH commands defined earlier. The 20-byte value from the message hash provides the input for the TIMESTAMP command. After the message hash has been computed, the TIMESTAMP command is executed to return a digital signature computed on the hash of a message and the value of the real-time clock. The Card writes the resultant r and s values into 320-bit fields. For FORTEZZA, only the top 160 bits of the r and s values are used as the signature. The lower 160 bits of each value are discarded.

Timestamp Command Block

byte offset	
00	Command=X000 0061h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Hash Value (14h bytes)

32 0

Data-Out Block

byte offset	
00	Length of Data-Out Block (4 bytes)
04	r value (28 bytes)
2C	s value (28 bytes)
54	Signed Date/Time (16 bytes)

32 0

4.6.36. TIMESTAMP (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Bad Clock
Invalid Pointer
Execution Failure
Passed
Invalid State

4.6.37. UNWRAP KEY

The UNWRAP KEY command will unwrap the key data provided using the key indicated by the register index. After the key is unwrapped, the card will perform a checkword test and compare the generated value to the unwrapped value. If they compare, the key will then be loaded into the indicated register. Note that K_s, key index 0, can be used as the unwrapping key, but not as the unwrapped key destination.

1) Key Register Index-Register containing Key used for the unwrap function (Must be a TEK)
2) Key Register Index-Register in which to place the Unwrapped key in
3) Key-(Wrapped-96-bit value) (an MEK wrapped with a TEK)

Unwrap Key Command Block

byte offset

00	Command=X000 0079h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block(Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Key Register Index for unwrapping TEK (4 bytes)
08	Key Register Index for unwrapped MEK (4 bytes)
0C	TEK Wrapped MEK (0C bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Invalid Key Index
Register In Use
No Key Loaded
Execution Failure
Passed
Invalid State
Checkword Failure

4.6.38. VERIFY SIGNATURE

The VERIFY SIGNATURE command validates a received digital signature value. The host provides the newly generated hash value, the public component DSA-Y of the sender and the r and s values received in the message. The Card Data In Block defines a fixed 320-bit field for r and s. For FORTEZZA, write the 160 bit r and s values into the top bits of the defined 320-bit input field and fill the remaining 160 bits with zeros. The DSA parameters must be loaded prior to executing this command via the Set Personality or the Load DSA Parameters command. Subsequent Verify Signature commands will continue to use these parameters until another Set Personality or Load DSA Parameters command is issued.

Verify Signature Command Block

byte offset	
00	Command=X000 0064h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Calculated Hash Value (14h bytes)
18	r Value (28h bytes)
40	s Value (28h bytes)
68	Length of Public Component in hex bytes (4 bytes)
6C	Originator Public Component Y (80h bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Execution Failure
Failed
Passed
No P,Q,G Loaded
Invalid State

4.6.39. VERIFY TIMESTAMP

The VERIFY TIMESTAMP command validates a received digitally signed date/time value. The host provides the newly generated hash value and the r, s, and date/time values received in the message. The Card Data In Block defines a fixed 320-bit field for r and s. For FORTEZZA, write the 160 bit r and s value into the top bits of the defined 320-bit input field and fill the remaining 160 bits with zeros. This command is the reverse of the TIMESTAMP command and returns a pass or fail in the status block. Internally, the card performs the following steps:

(a) Concatenate the 160-bit message hash value with the 96-bit date/time value received from the message originator.
(b) Internally hash the result of (a).
(c) Digitally verify the TIMESTAMP value from the message originator based on (b) and a public key value common to all CAPSTONE.

Verify Timestamp Command Block

byte offset	
00	Command=X000 0068h
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset	
00	Length of Data-In Block (4 bytes)
04	Calculated Hash Value (14h bytes)
18	r Value (28h bytes)
40	s Value (28h bytes)
68	Date/Time value (16 bytes)

32 0

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Execution Failure
Failed
Passed
Invalid State

4.6.40. WRAP KEY

The WRAP KEY command will use the key indicated by the first register index to wrap the key indicated by the second register index. The result of this wrap is then returned in the data-out block as a 96-bit value. The key to be wrapped cannot be a TEK. Please note that a 16-bit check word will be generated over the key to be wrapped. Both the key and the generated checkword value will be wrapped. Register 0, K_s, can be used as the key to use for the wrapping function. PLEASE NOTE that Register 0, which contains K_s, cannot be used as the Register B index. The K_s value can not leave the card.

1) Register A -Key [Ka] used for the wrapping function (Must be a TEK)
2) Register B -Key [Kb] to be wrapped (encrypted) (Must be an MEK for Capstone)

Wrap Key Command Block

byte offset

00	Command=X000 006Bh
04	Pointer to Next Command Block
08	Pointer to Data-In Block
0C	Pointer to Data-Out Block
10	Response
14	Channel Specifier

32 0

Data-In Block

byte offset

00	Length of Data-In Block (4 bytes)
04	Register Index for Ka TEK (4 bytes)
08	Register Index for Kb MEK(4 bytes)

32 0

Data-Out Block

byte offset

00	Length of Data-Out Block (4 bytes)
04	Wrapped Key Value (0Ch bytes)

32 0

4.6.40. WRAP KEY (cont'd)

Entry State	Exit State
Standby	Standby
Ready	Ready

Response Codes

Invalid Pointer
Execution Failure
Invalid Key Index
No Key Loaded
Passed
Invalid State

4.6.41. ZEROIZE

The ZEROIZE command will sanitize the Card data buffers, internal buffers and key management information. After execution of this command the Card will enter the Zeroized state. The Card will need to be re-initialized by the SSO before operation. Note that the SSO will need to use the Zeroize default PIN to authenticate to the card.

Zeroize Command Block

byte offset	
00	Command=X000 006Dh
04	Pointer to Next Command Block
08	Pointer to Data-In Block (Null)
0C	Pointer to Data-Out Block (Null)
10	Response
14	Channel Specifier

32 0

Entry State	Exit State
Uninitialized	Zeroized
Initialized	Zeroized
SSO Initialized	Zeroized
LAW Initialized	Zeroized
User Initialized	Zeroized
Standby	Zeroized
Ready	Zeroized

Note: This command may be used before logging onto the card.

Response Codes

Passed